COMUNICAZIONI DIGITALI





COMUNICAZIONI DIGITALI Teoria dell'Informazione, Comunicazioni via Radio e su Cavo

Marco Luise Università di Pisa, 6 dicembre 2023



Una pubblicazione dell'UNIVERSITÁ DI PISA



PREFAZIONE



"Ma sopra tutte le invenzioni stupende, qual eminenza fu quella di colui che s'immaginò di trovar modo di comunicare i suoi più reconditi pensieri a qualsivoglia altra persona, benché distante per lunghissimo intervallo di luogo e di tempo?"

— Galileo Galilei, Dialogo sopra i due massimi sistemi del mondo, Giornata Prima

Firenze, 1632

Ringrazio (in ordine sparso): Silvia, Umberto Mengali, Emerson Lake & Palmer, Riccardo De Gaudenzi, Ezio Biglieri, Franco Russo, Luca Sanguinetti, Giorgio Vitetta, Stefano&Roberta, Nino D'Amico, i miei studenti.



Indice

Prefazione

1	Com	unicare	Digitale	1	
	1.1	Interne	et e "The Black Cloud"	2	
	1.2	Sulle S	Spalle dei Giganti	3	
	1.3	Essere	Digitali	3	
	1.4	Questi	Appunti	5	
2	ĽA-I	B-C di S	egnali Sistemi e Comunicazioni	7	
	2.1	Analis	i di Fourier dei segnali analogici	8	
		2.1.1	Segnali Periodici e Serie di Fourier	8	
		2.1.2	Segnali non-periodici e la Trasformata di Fourier	11	
		2.1.3	Segnali a Banda limitata	13	
		2.1.4	La funzione Delta di Dirac	14	
	2.2	Filtrag	gio Lineare	16	
		2.2.1	Sistemi e Segnali	16	
		2.2.2	Caratterizzazione degli SLS nel Tempo e nella Frequenza	18	
	2.3	Filtrag	gio di Segnali aleatori	20	
		2.3.1	Fondamenti di segnali aleatori	20	
		2.3.2	Media, funzione di Autocorrelazione, e Densità Spettrale di		
			Potenza	21	
				vii	

viii INDICE

		2.3.3	Filtraggio dei segnali aleatori	24	
	2.4	Segnali	e Sistemi in Banda Passante	27	
		2.4.1	Equivalenti in Banda-Base di segnali Passa-Banda	27	
		2.4.2	Equivalente in banda base del Rumore Gaussiano Bianco	28	
		2.4.3	Il modulatore I-Q	29	
		2.4.4	Il demodulatore I-Q	30	
	2.5	Analisi	di Fourier dei segnali Digitali	31	
		2.5.1	Segnali Analogici e Segnali Digitali	31	
		2.5.2	TF di Segnali Digitali	31	
		2.5.3	Filtraggio e Interpolazione di una segnale digitale	35	
		2.5.4	Il teorema del campionamento	37	
	2.6	Trasmis	ssione e Ricezione di Dati Digitali	39	
		2.6.1	Segnali Dati in Banda Base	39	
		2.6.2	Spettro dei Segnali Dati	40	
		2.6.3	Ricezione di un segnale dati su canale AWGN	42	
	2.7	Modula	zione/Demodulazione a Radiofrequenza e Architettura dei		
		Modem		47	
		2.7.1	Modulazione lineare digitale I/Q	47	
		2.7.2	Le costellazioni I/Q	48	
		2.7.3	Demodulazione I/Q di segnali radio	50	
		2.7.4	Architettura dei Modem basati su Digital Signal Processing		
			(DSP)	52	
	2.8	Multipl	azione e Accesso Multiplo	54	
		2.8.1	Multiplazione = Accesso Multiplo ?	55	
		2.8.2	Multiplazione con Segnali Ortogonali	55	
		2.8.3	FDMA, TDMA, CDMA, ALOHA	62	
		2.8.4	Cenno all'Accesso Multiplo con Criteri Statistici	68	
		2.8.5	Efficienza del Multiplexing/Multiple Access Deterministico	73	
3	Rido	ndanza	ed Efficienza - CompinaciDai	77	
	3.1	Che cos	s'è la "Codifica di Sorgente" ?	78	
		3.1.1	Probabilità sbilanciate	79	
		3.1.2	Codifica con perdita di qualità	79	
		3.1.3	Correlazione tra valori della sorgente	80	
	3.2	Codific	a senza perdita di una sorgente senza memoria	83	
	3.3	Shanno	n Information	88	
	3.4	Quantit	à di informazione di una sorgente senza memoria	89	
	3.5	Codici	(ottimi) per sorgente senza memoria	94	
		3.5.1	Codifica della sorgente estesa	98	
		3.5.2	Codifica Aritmetica	99	
		3.5.3	Codifica Aritmetica Adattativa	103	
	3.6	Sorgent	i di informazione con memoria	103	

		INI	
		3.6.1 Sorgenti Markoviane	106
	3.7	Codifica Universale (Adattativa) per sorgenti con memoria	110
	3.8	Da "Source Compression" a "Data Compression": La trasformata	di
		Burrows-Wheeler	112
	3.9	E adesso? Codifica di sorgente con perdita	114
4	Alla	Ricerca dell'Affidabilità Perduta	117
	4.1	Comunicazioni Digitali su di un "canale rumoroso"	118
	4.2	Il canale di comunicazione nella Teoria dell'Informazione	119
		4.2.1 Modelli di Canale	119
		4.2.2 Equivocazione, Informazione Mutua, Irrilevanza	122
		4.2.3 Canali notevoli	124
	4.3	Capacità di Shannon di un canale di comunicazione con disturbo	126
		4.3.1 Capacità dei canali notevoli	127
	4.4	Il secondo teorema della codifica (o teorema della codifica con	
		disturbo) di Shannon	131
		4.4.1 Generalità sulla Codifica di canale	131
		4.4.2 Codifica di Canale per Negati	132
		4.4.3 Il Secondo Teorema di Shannon	136
	4.5	Il canale AWGN	138
		4.5.1 Canali "hard" e "soft"	138
		4.5.2 Il canale Gaussiano a tempo discreto	144
		4.5.3 Il canale Gaussiano con ingresso binario (BIAWGN)	147
		4.5.4 Il canale AWGN complesso e la capacità delle costellazio	ni I/Q150
		4.5.5 Il canale Gaussiano a tempo continuo	151
	4.6	Efficienza dei metodi di modulazione e codifica	152
		4.6.1 Separazione Sorgente/Canale	155
5	Dist(0rc3r3 ma n0n tr0pp0 - Codifica di sorgente con perdita	157
	5.1	Perché codificare con perdita?	158
	5.2	Codifica di Sorgente con una Misura di Distorsione	158
		5.2.1 Curva Tasso-Distorsione	160
		5.2.2 Curve Tasso-Distorsione Notevoli	162
		5.2.3 Dalla teoria alla pratica	165
	5.3	Codifica delle immagini digitali in formato JPEG	165
	5.4	Codifica dei segnali video digitali in formato MPEG	173
	5.5	Codifica dei segnali audio digitali in formato MP3	179
6	Codi	fica di Vanale Codifica yi Canale Codifica di Canale	183
	6.1	Capacità di Shannon e Fondamenti della Codifica di Canale	184
		6.1.1 Codici a Blocco e Decodifica Ottima	184
		6.1.2 Codici a Blocco Algebrici	188
		e	

х INDICE

6.2	Codici	Convoluzionali e Decodifica di Viterbi	198		
	6.2.1	Codifica convoluzionale non sistematica	198		
	6.2.2	Codifica convoluzionale sistematica	200		
	6.2.3	Decodifica di un codice convoluzionale	201		
	6.2.4	BER Performance	208		
	6.2.5	Decodifica Probabilistica ad uscita soft: l'algoritmo BCJR			
		per codici convoluzionali	211		
6.3	Decodi	fica Iterativa (Turbo) di Codici Convoluzionali Concatenati	213		
	6.3.1	Il Codificatore Turbo PCCC	213		
	6.3.2	Decodifica iterativa del Turbo Code	216		
	6.3.3	BER performance	218		
6.4	Codici	Low-Density Parity-Check e Decodifica Iterativa a Scambio di			
	Messag	ıgi	220		
	6.4.1	Che cos'è un codice LDPC	220		
	6.4.2	Decodifica iterativa dei codici LDPC - I: Hard-Input	223		
	6.4.3	Decodifica iterativa dei codici LDPC - II: Soft-Input	224		
6.5	Codici	Polari con Decodifica a Cancellazioni Successive	228		
	6.5.1	C'è davvero necessità dell'(n+1)-esimo codice?	228		
	6.5.2	Teorema di polarizzazione e codici polari	231		
	6.5.3	Costruzione e decodifica dei codici polari	232		
	6.5.4	Decodifica dei codici polari con esempi	236		
6.6	Codific	a Link-Layer con codici "Rateless"	238		
	6.6.1	Rateless/Fountain Codes	239		
	6.6.2	Il codice a fontana lineare casuale	239		
	6.6.3	FC basati sulla trasformata di Luby	241		
	6.6.4	Codici "Raptor"	243		
Anyw	/here, A	nytime - Elementi di Propagazione Radio	245		
71	Il canal	e wireless per comunicazioni digitali	246		
/.1	7 1 1	Il canale radio a cammini multinli (multinath)	240 247		
	7.1.1	Canali radio selettivi nella frequenza e/o nel tempo	251		
	7.1.2	Canali Radio con Fading (Evanescenza)	251		
72	Modell	istica ed emulazione del canale radio	255		
1.2	wieden		201		
Un A	Arcobaleno di Dati - Le Tecnologie Multiportante		269		
8.1	Le Con	nunicazioni Multiportante	270		
	8.1.1	Cammini Multipli e Distorsione	270		
8.2	Dal Ge	nerico Formato Multicarrier all'OFDM	273		
	8.2.1	Il modulatore multiportante	273		
	8.2.2	Demodulatore Multiportante e OFDM	276		
	8.2.3	Efficienza spettrale dell' OFDM	277		
8.3	Implem	nentazione DSP del Modem OFDM	279		
	 6.2 6.3 6.4 6.5 6.6 Anyw 7.1 7.2 Un A 8.1 8.2 8.3 	6.2 Codici 6.2.1 6.2.2 6.2.3 6.2.3 6.2.4 6.2.5 6.3 Decodi 6.3.1 6.3.2 6.3.3 6.4 Codici Messag 6.4.1 6.4.2 6.4.3 6.5 Codici 6.5.1 6.5.2 6.5.3 6.5.4 6.6.1 6.6.2 6.6.3 6.6.4 Anywhere, A 7.1 Il canal 7.1.1 7.2 Modell Un Arcobale 8.1 Le Con 8.1.1 8.2 Dal Ge 8.2.1 8.2.2 8.2.3 8.3 Implem	 6.2 Codici Convoluzionali e Decodifica di Viterbi 6.2.1 Codifica convoluzionale non sistematica 6.2.2 Codifica convoluzionale sistematica 6.2.3 Decodifica i un codice convoluzionale 6.2.4 BER Performance 6.2.5 Decodifica Probabilistica ad uscita soft: l'algoritmo BCJR per codici convoluzionali 6.3 Decodifica Iterativa (Turbo) di Codici Convoluzionali Concatenati 6.3.1 II Codificatore Turbo PCCC 6.3.2 Decodifica iterativa del Turbo Code 6.3.3 BER performance 6.4 Codici Low-Density Parity-Check e Decodifica Iterativa a Scambio di Messaggi 6.4.1 Che cos'è un codice LDPC 6.4.2 Decodifica iterativa dei codici LDPC - I: Hard-Input 6.4.3 Decodifica iterativa dei codici LDPC - I: Soft-Input 6.4.3 Decodifica dei codici polari con cessità dell'(n+1)-esimo codice? 6.5.2 Teorema di polarizzazione e codici polari 6.5.3 Costruzione e decodifica dei codici polari 6.5.4 Decodifica dei codici polari con esempi 6.6 Codifica Link-Layer con codici "Rateless" 6.6.1 Rateless/Fountain Codes 6.6.2 II codice a fontana lineare casuale 6.6.3 FC basati sulla trasformata di Luby 6.6.4 Codici "Raptor" Anywhere, Anytime - Elementi di Propagazione Radio 7.1 II canale radio a cammini multipli (multipath) 7.1.2 Canali radio selettivi nella frequenza e/o nel tempo 7.1.3 Canali Radio con Fading (Evanescenza) 7.2 Modellistica ed emulazione del canale radio 8.1 Le Comunicazioni Multiportante 8.1.1 Cammini Multipli e Distorsione 8.2 Demodulatore multiportante 8.2.1 II modulatore multiportante 8.2.1 II modulatore multiportante 8.2.2 Demodulatore Multiportante 8.2.3 Efficienza spettrale dell' OFDM 8.2.3 Efficienza spettrale dell' OFDM	6.2 Codici Convoluzionali e Decodifica di Viterbi 198 6.2.1 Codifica convoluzionale non sistematica 198 6.2.2 Codifica convoluzionale sistematica 200 6.2.3 Decodifica di un codice convoluzionale 201 6.2.4 BER Performance 208 6.2.5 Decodifica Irobabilistica ad uscita soft: l'algoritmo BCJR per codici convoluzionali 211 6.3 Decodifica iterativa (Turbo) di Codici Convoluzionali Concatenati 213 6.3.1 Il Codificatore Turbo PCCC 213 6.3.2 Decodifica iterativa del Turbo Code 216 6.3.3 BER performance 218 6.4 Codici Low-Density Parity-Check e Decodifica Iterativa a Scambio di Messaggi 220 6.4.1 Che cos'è un codice LDPC 221 6.4.2 Decodifica iterativa dei codici LDPC - I: Hard-Input 223 6.5.3 Codici Polari con Decodifica a Cancellazioni Successive 228 6.5.1 Ci davero necessità dell'(n+1)-esimo codice? 228 6.5.2 Teorema di polarizzazione e codici polari 232 6.5.4 Decodifica dei codici ipolari con esempi 236 6.6.2	 6.2 Codici Convoluzionali e Decodifica di Viterbi 198 6.2.1 Codifica convoluzionale non sistematica 198 6.2.2 Codifica convoluzionale non sistematica 200 6.2.3 Decodifica di un codice convoluzionale 201 6.2.4 BER Performance 201 6.2.5 Decodifica Probabilistica ad uscita soft: l'algoritmo BCJR per colici convoluzionali 211 6.3 Decodifica Iterativa (Turbo) di Codici Convoluzionali Concatenati 213 6.3.1 Il Codificatore Turbo PCCC 213 6.3.2 Decodifica iterativa del Turbo Code 216 6.3.3 BER performance 218 Codici Low-Density Parity-Check e Decodifica Iterativa a Scambio di Messaggi 6.4 Codici Low-Density Parity-Check e Decodifica Iterativa a Scambio di Messaggi 6.4.1 Che cos'è un codice LDPC 220 6.4.1 Che cos'è un codice LDPC - I: Hard-Input 223 6.4.2 Decodifica iterativa dei codici LDPC - I: Hard-Input 224 6.5 Codicio Polari con Decodifica da Cancellazioni Successive 228 6.5.1 C'è davvero necessità dell'(n+1)-esimo codice? 228 6.5.2 Teorema di polarizzazione e codici polari 231 6.5.3 Costruzione decodifica dei codici polari 231 6.5.4 Decodifica dei codici polari 233 6.5.4 Decodifica dei codici polari 239 6.6.2 Il codice a fontana lineare casuale 239 6.6.3 FC basati sulla trasformata di Luby 241 6.6.4 Codici "Raptor" 238 6.6.2 Codici ca fontana lineare casuale 239 6.6.3 FC basati sulla trasformata di Luby 241 7.1.1 Il canale radio a cammini multipli (multipath) 247 7.1.3 Canali Radio con Fading (Evanescenza) 255 7.2 Modellistica ed enuluzione del canale radio 246 Un Ar-coberto of Dati - Le Tecnologie Multiportante 270 8.1.1 Cammini Multipit Distorsione 270 8.2.1 Il modulatore multiportante e OFDM 275

	INDICE	xi
		070
	8.3.1 Modulatore OFDM Digitale	279
	8.3.2 Demodulatore OFDM Digitale	281
0.4	8.3.3 Sottoportanti Virtuali	281
8.4	Equalizzazione di OFDM nel Dominio della Frequenza	283
	8.4.1 Tempo di guardia e prefisso ciclico	283
	8.4.2 Stima e compensazione della risposta in frequenza del canale	289
	8.4.3 BER del ricevitore OFDM con equalizzazione ZF	293
	8.4.4 Modellistica ed emulazione del canale wireless nel dominio	
	della frequenza	296
8.5	Vantaggi, svantaggi, e varianti della tecnologia OFDM	298
	8.5.1 Multiplazione e accesso multiplo flessibili	298
	8.5.2 Svantaggi delle tecnologie OFDM	299
8.6	Multicarrier meets Information Theory: Le tecnologie DSL e la	
	capacità del canale Gaussiano colorato	305
	8.6.1 Architettura di un sistema xDSL per l'ultimo miglio	305
	8.6.2 Duplexing e Modellistica di Canale per ADSL	306
8.7	Capacità del canale Gaussiano "colorato"	309
	8.7.1 Massimizzazione della capacità del canale Gaussiano	
	"colorato"	311
	8.7.2 Criterio di water-filling: interpretazione della massimizzazione	
	della capacità del canale Gaussiano "colorato"	313
8.8	La modulazione DMT (Discrete MultiTone)	314
Qua	ndo il Disturbo è più forte del Segnale - Comunicazioni Spread	-
Spe	ctrum	319
9.1	Comunicare con un segnale a spettro espanso	320
	9.1.1 Banda stretta e a banda larga: Direct-Sequence Spread	
	Spectrum	320
	9.1.2 Ricezione del segnale DS/SS su canale AWGN	325
9.2	Frequency-Hopping Spread Spectrum e Altri Formati	329
	9.2.1 Frequency-Hopping Spread Spectrum	329
	9.2.2 DS/SS a Banda Limitata	334
	9.2.3 Codici di Spreading	336
	9.2.4 Spread-Spectrum e Capacità di Shannon	340
9.3	Alice, Bob, Eve & Mallory	341
	9.3.1 Jamming	341
	9.3.2 Eavesdropping/Low Probability of Intercept	343
	9.3.3 Spoofing	346
9.4	DS/SS in ambiente Multipath	346
	9.4.1 Cancellazione del Multipath	346
	9.4.2 Ricevitore "Rake"	347
9.5	CDMA Riveduto e Corretto	351

xii	INDICE	

9.5.2 Capacità di Shannon 353 10 L'Unione fa la Forza - Comunicazioni MIMO 357 10.1 L'A-B-C delle Comunicazioni MIMO (Multiple-Input, Multiple-Output 358 10.2 La Ricezione in Diversità (tradizionale) diventa Trasmissione in Diversità 358 10.2.1 Strategie di combinazione 360 10.2.2 Diversità in trasmissione 365 10.3 Modellistica di canale MIMO e utilizzo intuitivo 367 10.3.1 Modellistica di canale MIMO e utilizzo intuitivo 367 10.3.2 Capacità di Shannon del canale MIMO 372 10.3.3 Cadacità con allocazione della potenza al trasmettitore 377 10.3.2 Capacità di Shannon fel canale MIMO 382 10.3.3 Cadacita e pazio relutori servizio sul canale di Rayleigh 383 10.4 MIMO Multiutente, Beamforming, Space-Division Multiple Access 385 10.4.1 Diagramma d'irradiazione d' un'antenna 392 10.4.2 Antenna ad array lineare e fattore d'array 388 10.4.2 Antenna da array lineare e fattore d'array 388 10.4.3 Beamforming e vettore di steering 392 10.4.4 Space-Division Multiple Access (SDMA) and Multiuser 410 11.1 Comunicazioni Radio e Comunicazioni Ottiche <t< th=""><th></th><th></th><th>9.5.1</th><th>Accesso Multiplo e Multiple Access Interference (MAI)</th><th>351</th></t<>			9.5.1	Accesso Multiplo e Multiple Access Interference (MAI)	351
10 L'Unione fa la Forza - Comunicazioni MIMO (Multiple-Input, Multiple-Output 358 10.1 L'A-B-C delle Comunicazioni MIMO (Multiple-Input, Multiple-Output 358 10.2 La Ricezione in Diversità (tradizionale) diventa Trasmissione in Diversità 358 10.2.1 Strategie di combinazione 360 10.2.2 Diversità ontro il Fading di Rayleigh 361 10.3.0 Diversità in trasmissione 365 10.3 Modellistica e Capacità di Shannon di un Collegamento MIMO 377 10.3.1 Modellistica di canale MIMO e utilizzo intutivo 367 10.3.2 Capacità di Shannon del canale MIMO 372 10.3.3 Capacità con allocazione della potenza al trasmettiore 377 10.3.4 Codifica Spazio-Tempo 388 10.4 MIMO Multiutente, Beamforming, Space-Division Multiple Access 385 10.4.1 Diagramma d'irradiazione di urantenna 388 10.4.3 Beamforming e vettore di steering 392 10.4.4 Space-Division Multiple Access (SDMA) and Multiuser 401 11.1 Icomu capello, grandi come Il mondo: le Fibre Ottiche 409 10.5 La Nuova Frontiera: Massive MIMO 399 <t< th=""><th></th><th></th><th>9.5.2</th><th>Capacità di Shannon</th><th>353</th></t<>			9.5.2	Capacità di Shannon	353
10 L'Unione fa la Forza - Comunicazioni MIMO 357 10.1 L'A-B-C delle Comunicazioni MIMO (Multiple-Input, Multiple-Output 358 10.2 10.4 La Ricezione in Diversità (tradizionale) diventa Trasmissione in Diversità in trasmissione 366 10.2.1 Strategie di combinazione 360 10.2.2 Diversità in trasmissione 365 10.3 Modellistica e Capacità di Shannon di un Collegamento MIMO 367 10.3.1 Modellistica di canale MIMO e utilizzo intuitivo 367 10.3.2 Capacità con allocazione della potenza al trasmettitore 372 10.3.3 Capacità con allocazione della potenza al trasmettitore 377 10.3.4 Codifica Spazio-Tempo 380 10.3.5 MIMO OFDM 382 10.3.6 Capacità ergofica e fuori servizio sul canale di Rayleigh 383 10.4 MIMO Multiutente, Beamforming, Space-Division Multiple Access 385 10.4.1 Diagramma d'irradiazione d'array 388 10.4.2 Antenna ad array lineare e fattore d'array 388 10.4.3 Beamforming, vectore di steering 392 10.4.4 Space-Division Multiple Access 396					
10.1 L'A-B-C delle Comunicazioni MIMO (Multiple-Input, Multiple-Output 358 10.2 La Ricezione in Diversità (tradizionale) diventa Trasmissione in Diversità contro il Fading di Rayleigh 361 10.2.1 Strategie di combinazione 365 10.3 Modellistica e Capacità di Shannon di un Collegamento MIMO 367 10.3.1 Modellistica di canale MIMO e utilizzo intuitivo 367 10.3.2 Capacità di Shannon del canale MIMO 372 10.3.4 Codifica Spazio-Tempo 370 10.3.5 MIMO OFDM 382 10.3.6 Capacità con allocazione della potenza al trasmettitore 377 10.3.6 Capacità ergodica e fuori servizio sul canale di Rayleigh 383 10.4 MIMO OFDM 382 10.3.5 MIMO OFDM 382 10.4.1 Diagramma d'irradiazione di un' antenna 385 10.4.2 Antenna ad array lineare e fattore d'array 388 10.4.3 Beamforming e vettore di steering 392 10.4.4 Space-Division Multiple Access (SDMA) and Multiuser MIMO 396 10.5 La Nuova Frontiera: Massive MIMO 396 10.5 La Nuova Frontiera: Massive MIMO 399 10.6 Modellistica ed emulazione del canale MIMO-OFDM 401 11 Icoma-base di un link su fibra ottica 410 1	10	ĽUni	one fa la	a Forza - Comunicazioni MIMO	357
10.2 La Ricezione in Diversità (tradizionale) diventa Trasmissione in Diversità 358 10.2.1 Strategie di combinazione 360 10.2.2 Diversità contro il Fading di Rayleigh 361 10.2.3 Diversità in trasmissione 365 10.3 Modellistica e Capacità di Shannon di un Collegamento MIMO 367 10.3.1 Modellistica di canale MIMO e utilizzo intuitivo 367 10.3.2 Capacità di Shannon del canale MIMO 372 10.3.3 Capacità di Shannon del canale MIMO 372 10.3.4 Codifica Spazio-Tempo 380 10.3.5 MIMO OFDM 382 10.3.6 Capacità ergodica e fuori servizio sul canale di Rayleigh 383 10.4 MIMO Multiutente, Beamforming, Space-Division Multiple Access 385 10.4.1 Diagramma d'irradiazione di un' antenna 385 10.4.2 Antenna ad array lineare e fattore d'aray 388 10.4.3 Beamforming e vettore di steering 392 10.4.4 Space-Division Multiple Access (SDMA) and Multiuser MIMO MIMO 396 396 396 10.5 La N		10.1	L'A-B-0	C delle Comunicazioni MIMO (Multiple-Input, Multiple-Outpu	ıt 358
Diversità 358 10.2.1 Strategie di combinazione 360 10.2.2 Diversità contro il Fading di Rayleigh 361 10.2.3 Diversità in trasmissione 365 10.3 Modellistica e Capacità di Shannon di un Collegamento MIMO 367 10.3.1 Modellistica di canale MIMO e utilizzo intuitivo 367 10.3.1 Modellistica di canale MIMO e utilizzo intuitivo 367 10.3.2 Capacità di Shannon del canale MIMO 372 10.3.3 Capacità con allocazione della potenza al trasmettitore 377 10.3.4 Codifica Spazio-Tempo 380 10.3.5 MIMO OFDM 382 10.3.6 Capacità ergodica e fuori servizio sul canale di Rayleigh 383 10.4 MIMO Multiutente, Beamforming, Space-Division Multiple Access 385 10.4.1 Diagramma d'irradiazione di un'antenna 385 10.4.2 Antenna ad array lincare e fattore d'array 388 10.4.3 Beamforming e vettore di steering 392 10.4.4 Space-Division Multiple Access (SDMA) and Multiuser MIMO 396 10.5 La Nuova Frontiera: Massive MIMO 399 10.6 Modellistica ed emulazione del canale MIMO-OFDM 401 11 Sottili come un capello, grandi come il mondo: le Fibre Ottiche 410 11.1.1 Comunicazioni Radio e Comunicazioni Ottiche 410 11.1.2 Schema-base di un link su fibra ottica 410 11.1.2 Schema-base di un link su fibra ottica 410 11.1.1 Comunicazioni Radio e Comunicazioni Ottiche 410 11.1.2 Schema-base di un link su fibra ottica 410 11.1.1 Comunicazioni Radio e Comunicazioni Ottiche 410 11.1.2 Schema-base di un link su fibra ottica 410 11.1.1 Fibre Multi-Mode Step-Index e Graded-Index 416 11.1.2 Schema-base di un link su fibra ottica 410 11.1.1 Limiti di capacità del link: Dispersione Intramodale e Attennazione della fibra SM 426 11.2 Modem per Fibra Ottica 1: On-Off Keying (OOK) e Rivelazione Celtromagnetica 413 11.2.1 Non ci stanchiamo della fisica: Interazioni materia/radiazione el eltromagnetica (DD, Direct Detection) 434 11.2.1 Non ci stanchiamo della fisica: Interazioni materia/radiazione eltromagnetica (DD D, Direct Detection) 434		10.2	La Rice	zione in Diversità (tradizionale) diventa Trasmissione in	
10.2.1Strategic di combinazione36010.2.2Diversità contro il Fading di Rayleigh36110.2.3Diversità in trasmissione36510.3Modellistica e Capacità di Shannon di un Collegamento MIMO36710.3.1Modellistica di canale MIMO e utilizzo intuitivo36710.3.2Capacità di Shannon del canale MIMO37210.3.3Cadifica Spazio-Tempo38010.3.4Codifica Spazio-Tempo38010.3.5MIMO OFDM38210.3.6Capacità cegodica e fuori servizio sul canale di Rayleigh38310.4MIMO Multiutente, Beamforming, Space-Division Multiple Access38510.4.1Diagramma d'irradiazione di un'antenna38510.4.2Antenna ad array lineare e fattore d'array38810.4.3Beamforming e vettore di steering39210.4.4Space-Division Multiple Access (SDMA) and Multiuser MIMO39610.5La Nuova Frontiera: Massive MIMO39910.6Modellistica ed emulazione del canale MIMO-OFDM40111Sottill come un capello, grandi come il mondo: le Fibre Ottiche40911.1Il come ei il perché dei Collegamenti in Fibra Ottica41011.1.2Schema-base di un link su fibra ottica41211.1.3Che cos'é una fibra ottica?41411.1.4Fibre Multi-Mode Step-Index e Graded-Index41611.1.5Le fibre Single-Mode (SM)42411.1.6Le fibre Single-Mode (SM)42411.1.7Limiti di capacità del li			Diversit	à	358
10.2.2 Diversità contro il Fading di Rayleigh 361 10.2.3 Diversità in trasmissione 365 10.3 Modellistica e Capacità di Shannon di un Collegamento MIMO 367 10.3.1 Modellistica di canale MIMO e utilizzo intuitivo 367 10.3.2 Capacità di Shannon del canale MIMO 372 10.3.3 Capacità con allocazione della potenza al trasmettitore 377 10.3.4 Codifica Spazio-Tempo 380 10.3.5 MIMO OFDM 382 10.3.6 Capacità ergodica e fuori servizio sul canale di Rayleigh 383 10.4 MIMO Multiutente, Beamforming, Space-Division Multiple Access 385 10.4.1 Diagramma d'irradiazione di un'antenna 385 10.4.2 Antenna ad array lineare e fattore d'array 388 10.4.3 Space-Division Multiple Access (SDMA) and Multiuser 392 10.4.4 Space-Division Multiple Access (SDMA) and Multiuser 409 11.1 Il comunicazioni Radio e Comunicazioni Ottiche 410 11.1.2 Schema-base di un link su fibra ottica 412 11.1.1 Comunicazioni Radio e Comunicazioni Ottiche 416 11.1.2			10.2.1	Strategie di combinazione	360
10.2.3 Diversità in trasmissione 365 10.3 Modellistica e Capacità di Shannon di un Collegamento MIMO 367 10.3.1 Modellistica di canale MIMO e utilizzo intuitivo 367 10.3.2 Capacità di Shannon del canale MIMO 372 10.3.3 Capacità con allocazione della potenza al trasmettiore 377 10.3.4 Codifica Spazio-Tempo 380 10.3.5 MIMO OFDM 382 10.3.6 Capacità ergodica e fuori servizio sul canale di Rayleigh 383 10.4 MIMO Multiutente, Beamforming, Space-Division Multiple Access 385 10.4.1 Diagramma d'irradiazione di un'antenna 385 10.4.2 Antenna ad array lineare e fattore d'array 388 10.4.3 Beamforming e vettore di steering 392 10.4.4 Space-Division Multiple Access (SDMA) and Multiuser 392 10.5 La Nuova Frontiera: Massive MIMO 399 10.6 Modellistica ed emulazione del canale MIMO-OFDM 401 11 Sottill come un capello, grandi come Il mondo: le Fibre Ottiche 409 11.1 Il comunicazioni Radio e Comunicazioni Ottiche 410 11.1.2			10.2.2	Diversità contro il Fading di Rayleigh	361
10.3 Modellistica e Capacità di Shannon di un Collegamento MIMO 367 10.3.1 Modellistica di canale MIMO e utilizzo intuitivo 367 10.3.2 Capacità di Shannon del canale MIMO 372 10.3.3 Capacità di Shannon del canale MIMO 372 10.3.4 Codifica Spazio-Tempo 380 10.3.5 MIMO OFDM 382 10.3.6 Capacità ergodica e fuori servizio sul canale di Rayleigh 383 10.4 MIMO Multiutente, Beamforming, Space-Division Multiple Access 385 10.4.1 Diagramma d'irradiazione di un'antenna 385 10.4.2 Antenna ad array lineare e fattore d'array 388 10.4.3 Beamforming e vettore di steering 392 10.4.4 Space-Division Multiple Access (SDMA) and Multiuser MIMO MIMO 396 399 396 10.5 La Nuova Frontiera: Massive MIMO 396 10.5 La Nuova Frontiera: Massive MIMO oFDM 401 11 Sottili come un capello, grandi come il mondo: le Fibre Ottiche 409 11.1 Il comunicazioni Radio e Comunicazioni Ottiche 410 11.1.1 Comunicazioni Radio e Comuni			10.2.3	Diversità in trasmissione	365
10.3.1 Modellistica di canale MIMO e utilizzo intuitivo 367 10.3.2 Capacità di Shannon del canale MIMO 372 10.3.3 Capacità con allocazione della potenza al trasmettitore 377 10.3.4 Codifica Spazio-Tempo 380 10.3.5 MIMO OFDM 382 10.3.6 Capacità ergodica e fuori servizio sul canale di Rayleigh 383 10.4 MIMO Multitutente, Beamforming, Space-Division Multiple Access 385 10.4.1 Diagramma d'irradiazione di un'antenna 385 10.4.2 Antenna ad array lineare e fattore d'array 388 10.4.3 Beamforming e vettore di steering 392 10.4.4 Space-Division Multiple Access (SDMA) and Multiuser MIMO MIMO 396 399 10.6 Modellistica ed emulazione del canale MIMO-OFDM 401 11 Sottili come un capello, grandi come il mondo: le Fibre Ottiche 409 11.1 Il comunicazioni Radio e Comunicazioni Ottiche 410 11.1.2 Schema-base di un link su fibra ottica 410 11.1.3 Che cos'é una fibra ottica? 414 11.1.4 Fibre Multi-Mode Step-Index e Graded-		10.3	Modelli	stica e Capacità di Shannon di un Collegamento MIMO	367
10.3.2 Capacità di Shannon del canale MIMO 372 10.3.3 Capacità con allocazione della potenza al trasmettitore 377 10.3.4 Codifica Spazio-Tempo 380 10.3.5 MIMO OFDM 382 10.3.6 Capacità ergodica e fuori servizio sul canale di Rayleigh 383 10.4 MIMO Multiutente, Beamforming, Space-Division Multiple Access 385 10.4.1 Diagramma d'irradiazione di un'antenna 385 10.4.2 Antenna ad array lineare e fattore d'array 388 10.4.3 Beamforming e vettore di steering 392 10.4.4 Space-Division Multiple Access (SDMA) and Multiuser MIMO MIMO 396 399 10.6 Modellistica ed emulazione del canale MIMO-OFDM 401 11 Sottili come un capello, grandi come il mondo: le Fibre Ottiche 409 11.1 Il comucicazioni Radio e Comunicazioni Ottiche 410 11.1.2 Schema-base di un link su fibra ottica 412 11.1.3 Che cos' é una fibra ottica? 414 11.1.4 Fibre Multi-Mode Step-Index e Graded-Index 416 11.1.5 Le fibre a indice graduato (graded-index) </th <th></th> <th></th> <th>10.3.1</th> <th>Modellistica di canale MIMO e utilizzo intuitivo</th> <th>367</th>			10.3.1	Modellistica di canale MIMO e utilizzo intuitivo	367
10.3.3 Capacità con allocazione della potenza al trasmettitore 377 10.3.4 Codifica Spazio-Tempo 380 10.3.5 MIMO OFDM 382 10.3.6 Capacità ergodica e fuori servizio sul canale di Rayleigh 383 10.4 MIMO Multiutente, Beamforming, Space-Division Multiple Access 385 10.4.1 Diagramma d'irradiazione di un'antenna 385 10.4.2 Antenna ad array lineare e fattore d'array 388 10.4.3 Beamforming e vettore di steering 392 10.4.4 Space-Division Multiple Access (SDMA) and Multiuser 396 10.5 La Nuova Frontiera: Massive MIMO 396 10.6 Modellistica ed emulazione del canale MIMO-OFDM 401 11 Sottili come un capello, grandi come il mondo: le Fibre Ottiche 409 11.1 I comunicazioni Radio e Comunicazioni Ottiche 410 11.1.2 Schema-base di un link su fibra ottica 412 11.1.3 Che cos'é una fibra ottica? 414 11.1.4 Fibre Multi-Mode Step-Index e Graded-Index 416 11.1.5 Le fibre Single-Mode (SM) 422 11.1.6 Le fibre Single-Mode (SM) </th <th></th> <th></th> <th>10.3.2</th> <th>Capacità di Shannon del canale MIMO</th> <th>372</th>			10.3.2	Capacità di Shannon del canale MIMO	372
10.3.4 Codifica Spazio-Tempo 380 10.3.5 MIMO OFDM 382 10.3.6 Capacità ergodica e fuori servizio sul canale di Rayleigh 383 10.4 MIMO Multiutente, Beamforming, Space-Division Multiple Access 385 10.4.1 Diagramma d'irradiazione di un'antenna 385 10.4.2 Antenna ad array lineare e fattore d'array 388 10.4.3 Beamforming e vettore di steering 392 10.4.4 Space-Division Multiple Access (SDMA) and Multiuser MIMO MIMO MIMO 396 10.5 La Nuova Frontiera: Massive MIMO 399 10.6 Modellistica ed emulazione del canale MIMO-OFDM 401 11 Sottili come un capello, grandi come il mondo: le Fibre Ottiche 409 11.1 II comunicazioni Radio e Comunicazioni Ottiche 410 11.1.2 Schema-base di un link su fibra ottica 412 11.1.3 Che cos'é una fibra ottica? 414 11.1.4 Fibre Multi-Mode Step-Index e Graded-Index 416 11.1.5 Le fibre aindice graduato (graded-index) 422 11.6 Le fibre Single-Mode (SM) 424			10.3.3	Capacità con allocazione della potenza al trasmettitore	377
10.3.5 MIMO OFDM 382 10.3.6 Capacità ergodica e fuori servizio sul canale di Rayleigh 383 10.4 MIMO Multitutente, Beamforming, Space-Division Multiple Access 385 10.4.1 Diagramma d'irradiazione di un'antenna 385 10.4.2 Antenna ad array lineare e fattore d'array 388 10.4.3 Beamforming e vettore di steering 392 10.4.4 Space-Division Multiple Access (SDMA) and Multiuser 396 10.5 La Nuova Frontiera: Massive MIMO 399 10.6 Modellistica ed emulazione del canale MIMO-OFDM 401 11 Sottili come un capello, grandi come il mondo: le Fibre Ottiche 409 11.1 Il come e il perché dei Collegamenti in Fibra Ottica 410 11.1.2 Schema-base di un link su fibra ottica 412 11.1.3 Che cos'é una fibra ottica? 414 11.1.4 Fibre Multi-Mode Step-Index e Graded-Index 416 11.1.5 Le fibre Single-Mode (SM) 424 11.1.6 Le fibre Single-Mode (SM) 424 11.1.7 Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM 426 11.2.1 <th></th> <th></th> <th>10.3.4</th> <th>Codifica Spazio-Tempo</th> <th>380</th>			10.3.4	Codifica Spazio-Tempo	380
10.3.6 Capacità ergodica e fuori servizio sul canale di Rayleigh 383 10.4 MIMO Multiutente, Beamforming, Space-Division Multiple Access 385 10.4.1 Diagramma d'irradiazione di un'antenna 385 10.4.2 Antenna ad array lineare e fattore d'array 388 10.4.3 Beamforming e vettore di steering 392 10.4.4 Space-Division Multiple Access (SDMA) and Multiuser 396 10.5 La Nuova Frontiera: Massive MIMO 399 10.6 Modellistica ed emulazione del canale MIMO-OFDM 401 11 Sottili come un capello, grandi come il mondo: le Fibre Ottiche 409 11.1 II come e il perché dei Collegamenti in Fibra Ottica 410 11.1.2 Schema-base di un link su fibra ottica 412 11.1.3 Che cos'é una fibra ottica? 414 11.1.4 Fibre Multi-Mode Step-Index e Graded-Index 416 11.1.5 Le fibre a indice graduato (graded-index) 422 11.1.6 Le fibre Single-Mode (SM) 424 11.1.7 Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM 426 11.2 Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione			10.3.5	MIMO OFDM	382
10.4 MIMO Multiutente, Beamforming, Space-Division Multiple Access 385 10.4.1 Diagramma d'irradiazione di un'antenna 385 10.4.2 Antenna ad array lineare e fattore d'array 388 10.4.3 Beamforming e vettore di steering 392 10.4.4 Space-Division Multiple Access (SDMA) and Multiuser 396 10.5 La Nuova Frontiera: Massive MIMO 399 10.6 Modellistica ed emulazione del canale MIMO-OFDM 401 11 Sottili come un capello, grandi come il mondo: le Fibre Ottiche 409 11.1 Il come e il perché dei Collegamenti in Fibra Ottica 410 11.2.2 Schema-base di un link su fibra ottica 412 11.1.3 Che cos'é una fibra ottica? 414 11.1.4 Fibre Multi-Mode Step-Index e Graded-Index 416 11.1.5 Le fibre single-Mode (SM) 422 11.1.6 Le fibre Single-Mode (SM) 426 11.2 Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione 434 11.2.1 Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica 434 11.2.1 Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagneti			10.3.6	Capacità ergodica e fuori servizio sul canale di Rayleigh	383
10.4.1 Diagramma d'irradiazione di un'antenna 385 10.4.2 Antenna ad array lineare e fattore d'array 388 10.4.3 Beamforming e vettore di steering 392 10.4.4 Space-Division Multiple Access (SDMA) and Multiuser 396 10.5 La Nuova Frontiera: Massive MIMO 399 10.6 Modellistica ed emulazione del canale MIMO-OFDM 401 11 Sottili come un capello, grandi come il mondo: le Fibre Ottiche 409 11.1 Il comunicazioni Radio e Comunicazioni Ottiche 410 11.2 Schema-base di un link su fibra ottica 412 11.3 Che cos'é una fibra ottica? 414 11.1.4 Fibre Multi-Mode Step-Index e Graded-Index 416 11.1.5 Le fibre a indice graduato (graded-index) 422 11.1.6 Le fibre Single-Mode (SM) 424 11.1.7 Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM 426 11.2 Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione Diretta (DD, Direct Detection) 434 11.2.1 Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica 434 11.2.1 Non ci stanchiamo		10.4	MIMO	Multiutente, Beamforming, Space-Division Multiple Access	385
10.4.2 Antenna ad array lineare e fattore d'array 388 10.4.3 Beamforming e vettore di steering 392 10.4.4 Space-Division Multiple Access (SDMA) and Multiuser 396 10.5 La Nuova Frontiera: Massive MIMO 399 10.6 Modellistica ed emulazione del canale MIMO-OFDM 401 11 Sottili come un capello, grandi come il mondo: le Fibre Ottiche 409 11.1 Il comunicazioni Radio e Comunicazioni Ottiche 410 11.1.2 Schema-base di un link su fibra ottica 410 11.1.3 Che cos'é una fibra ottica? 414 11.1.4 Fibre Multi-Mode Step-Index e Graded-Index 416 11.1.5 Le fibre a indice graduato (graded-index) 422 11.1.6 Le fibre Single-Mode (SM) 424 11.1.7 Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM 426 11.2 Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione 434 11.2.1 Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica 434 11.2.1 Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica 437 11.2.3 La Modu			10.4.1	Diagramma d'irradiazione di un'antenna	385
10.4.3 Beamforming e vettore di steering 392 10.4.4 Space-Division Multiple Access (SDMA) and Multiuser 396 10.5 La Nuova Frontiera: Massive MIMO 399 10.6 Modellistica ed emulazione del canale MIMO-OFDM 401 11 Sottili come un capello, grandi come il mondo: le Fibre Ottiche 409 11.1 Il comunicazioni Radio e Comunicazioni Ottiche 410 11.1.2 Schema-base di un link su fibra ottica 412 11.1.3 Che cos'é una fibra ottica? 414 11.1.4 Fibre Multi-Mode Step-Index e Graded-Index 416 11.1.5 Le fibre a indice graduato (graded-index) 422 11.1.6 Le fibre Single-Mode (SM) 424 11.1.7 Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM 426 11.2 Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione Diretta (DD, Direct Detection) 434 11.2.1 Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica 434 11.2.1 Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica 434 12.2.1 LASER per Telecomunicazioni 437 <			10.4.2	Antenna ad array lineare e fattore d'array	388
10.4.4 Space-Division Multiple Access (SDMA) and Multiuser MIMO 396 10.5 La Nuova Frontiera: Massive MIMO 399 10.6 Modellistica ed emulazione del canale MIMO-OFDM 401 11 Sottili come un capello, grandi come il mondo: le Fibre Ottiche 409 11.1 Il come e il perché dei Collegamenti in Fibra Ottica 410 11.1.1 Comunicazioni Radio e Comunicazioni Ottiche 410 11.1.2 Schema-base di un link su fibra ottica 412 11.1.3 Che cos'é una fibra ottica? 414 11.1.4 Fibre Multi-Mode Step-Index e Graded-Index 416 11.1.5 Le fibre a indice graduato (graded-index) 422 11.1.6 Le fibre Single-Mode (SM) 424 11.1.7 Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM 426 11.2 Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione Diretta (DD, Direct Detection) 434 11.2.1 Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica 437 11.2.2 I LASER per Telecomunicazioni 437 11.2.3 La Modulazione d'Intensità (IM) 441			10.4.3	Beamforming e vettore di steering	392
MIMO 396 10.5 La Nuova Frontiera: Massive MIMO 399 10.6 Modellistica ed emulazione del canale MIMO-OFDM 401 11 Sottili come un capello, grandi come il mondo: le Fibre Ottiche 409 11.1 Il come e il perché dei Collegamenti in Fibra Ottica 410 11.1.2 Schema-base di un link su fibra ottica 412 11.1.3 Che cos'é una fibra ottica? 414 11.1.4 Fibre Multi-Mode Step-Index e Graded-Index 416 11.1.5 Le fibre a indice graduato (graded-index) 422 11.1.6 Le fibre Single-Mode (SM) 424 11.1.7 Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM 426 11.2 Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione Diretta (DD, Direct Detection) 434 11.2.1 Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica 434 11.2.2 ILASER per Telecomunicazioni 437 11.2.3 La Modulazione d'Intensità (IM) 441			10.4.4	Space-Division Multiple Access (SDMA) and Multiuser	
10.5 La Nuova Frontiera: Massive MIMO 399 10.6 Modellistica ed emulazione del canale MIMO-OFDM 401 11 Sottili come un capello, grandi come il mondo: le Fibre Ottiche 409 11.1 Il come e il perché dei Collegamenti in Fibra Ottica 410 11.1.2 Schema-base di un link su fibra ottica 412 11.1.3 Che cos'é una fibra ottica? 414 11.1.4 Fibre Multi-Mode Step-Index e Graded-Index 416 11.1.5 Le fibre a indice graduato (graded-index) 422 11.1.6 Le fibre Single-Mode (SM) 424 11.1.7 Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM 426 11.2 Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione Diretta (DD, Direct Detection) 434 11.2.1 Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica 434 11.2.1 LASER per Telecomunicazioni 437 11.2.3 La Modulazione d'Intensità (IM) 441				МІМО	396
10.6Modellistica ed emulazione del canale MIMO-OFDM40111Sottili come un capello, grandi come il mondo: le Fibre Ottiche40911.1II come e il perché dei Collegamenti in Fibra Ottica41011.1.1Comunicazioni Radio e Comunicazioni Ottiche41011.1.2Schema-base di un link su fibra ottica41211.1.3Che cos'é una fibra ottica?41411.1.4Fibre Multi-Mode Step-Index e Graded-Index41611.1.5Le fibre a indice graduato (graded-index)42211.1.6Le fibre Single-Mode (SM)42411.1.7Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM42611.2Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione Diretta (DD, Direct Detection)43411.2.1Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica43411.2.2ILASER per Telecomunicazioni43711.2.3La Modulazione d'Intensità (IM)441		10.5	La Nuo	va Frontiera: Massive MIMO	399
11Sottili come un capello, grandi come il mondo: le Fibre Ottiche40911.1II come e il perché dei Collegamenti in Fibra Ottica41011.1.1Comunicazioni Radio e Comunicazioni Ottiche41011.1.2Schema-base di un link su fibra ottica41211.1.3Che cos'é una fibra ottica?41411.1.4Fibre Multi-Mode Step-Index e Graded-Index41611.1.5Le fibre a indice graduato (graded-index)42211.1.6Le fibre Single-Mode (SM)42411.1.7Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM42611.2Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione Diretta (DD, Direct Detection)43411.2.1Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica43411.2.2I LASER per Telecomunicazioni43711.2.3La Modulazione d'Intensità (IM)441		10.6	Modelli	stica ed emulazione del canale MIMO-OFDM	401
11.1Il come e il perché dei Collegamenti in Fibra Ottica41011.1.1Comunicazioni Radio e Comunicazioni Ottiche41011.1.2Schema-base di un link su fibra ottica41211.1.3Che cos'é una fibra ottica?41411.1.4Fibre Multi-Mode Step-Index e Graded-Index41611.1.5Le fibre a indice graduato (graded-index)42211.1.6Le fibre Single-Mode (SM)42411.1.7Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM42611.2Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione Diretta (DD, Direct Detection)43411.2.1Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica43411.2.2I LASER per Telecomunicazioni43711.2.3La Modulazione d'Intensità (IM)441	11	Sottil	i come	un capello, grandi come il mondo: le Fibre Ottiche	409
11.1.1Comunicazioni Radio e Comunicazioni Ottiche41011.1.2Schema-base di un link su fibra ottica41211.1.3Che cos'é una fibra ottica?41411.1.4Fibre Multi-Mode Step-Index e Graded-Index41611.1.5Le fibre a indice graduato (graded-index)42211.1.6Le fibre Single-Mode (SM)42411.1.7Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM42611.2Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione Diretta (DD, Direct Detection)43411.2.1Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica43411.2.2I LASER per Telecomunicazioni43711.2.3La Modulazione d'Intensità (IM)441		11.1	Il come	e il perché dei Collegamenti in Fibra Ottica	410
11.1.2Schema-base di un link su fibra ottica41211.1.3Che cos'é una fibra ottica?41411.1.4Fibre Multi-Mode Step-Index e Graded-Index41611.1.5Le fibre a indice graduato (graded-index)42211.1.6Le fibre Single-Mode (SM)42411.1.7Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM42611.2Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione Diretta (DD, Direct Detection)43411.2.1Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica43411.2.2I LASER per Telecomunicazioni43711.2.3La Modulazione d'Intensità (IM)441			11.1.1	Comunicazioni Radio e Comunicazioni Ottiche	410
11.1.3Che cos'é una fibra ottica?41411.1.4Fibre Multi-Mode Step-Index e Graded-Index41611.1.5Le fibre Multi-Mode Step-Index e Graded-index)42211.1.6Le fibre Single-Mode (SM)42411.1.7Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM42611.2Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione Diretta (DD, Direct Detection)43411.2.1Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica43411.2.2I LASER per Telecomunicazioni43711.2.3La Modulazione d'Intensità (IM)441			11.1.2	Schema-base di un link su fibra ottica	412
11.1.4Fibre Multi-Mode Step-Index e Graded-Index41611.1.5Le fibre a indice graduato (graded-index)42211.1.6Le fibre Single-Mode (SM)42411.1.7Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM42611.2Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione Diretta (DD, Direct Detection)43411.2.1Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica43411.2.2I LASER per Telecomunicazioni43711.2.3La Modulazione d'Intensità (IM)441			11.1.3	Che cos'é una fibra ottica?	414
11.1.5Le fibre a indice graduato (graded-index)42211.1.6Le fibre Single-Mode (SM)42411.1.7Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM42611.2Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione Diretta (DD, Direct Detection)43411.2.1Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica43411.2.2I LASER per Telecomunicazioni43711.2.3La Modulazione d'Intensità (IM)441			11.1.4	Fibre Multi-Mode Step-Index e Graded-Index	416
11.1.6Le fibre Single-Mode (SM)42411.1.7Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM42611.2Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione Diretta (DD, Direct Detection)43411.2.1Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica43411.2.2I LASER per Telecomunicazioni43711.2.3La Modulazione d'Intensità (IM)441			11.1.5	Le fibre a indice graduato (graded-index)	422
11.1.7 Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM 426 11.2 Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione Diretta (DD, Direct Detection) 434 11.2.1 Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica 434 11.2.2 I LASER per Telecomunicazioni 437 11.2.3 La Modulazione d'Intensità (IM) 441			11.1.6	Le fibre Single-Mode (SM)	424
Attenuazione della fibra SM 426 11.2 Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione Diretta (DD, Direct Detection) 434 11.2.1 Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica 434 11.2.2 I LASER per Telecomunicazioni 437 11.2.3 La Modulazione d'Intensità (IM) 441			11.1.7	Limiti di capacità del link: Dispersione Intramodale e	
11.2 Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione Diretta (DD, Direct Detection) 434 11.2.1 Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica 434 11.2.2 I LASER per Telecomunicazioni 437 11.2.3 La Modulazione d'Intensità (IM) 441				Attenuazione della fibra SM	426
Diretta (DD, Direct Detection)43411.2.1Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica43411.2.2I LASER per Telecomunicazioni43711.2.3La Modulazione d'Intensità (IM)441		11.2	Modem	per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione	
11.2.1Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica43411.2.2I LASER per Telecomunicazioni43711.2.3La Modulazione d'Intensità (IM)441			Diretta	(DD. Direct Detection)	434
elettromagnetica 434 11.2.2 I LASER per Telecomunicazioni 437 11.2.3 La Modulazione d'Intensità (IM) 441			11.2.1	Non ci stanchiamo della fisica: Interazioni materia/radiazione	
11.2.2I LASER per Telecomunicazioni43711.2.3La Modulazione d'Intensità (IM)441				elettromagnetica	434
11.2.3 La Modulazione d'Intensità (IM) 441			11.2.2	I LASER per Telecomunicazioni	437
			11.2.3	La Modulazione d'Intensità (IM)	441

				INDIC	e xiii	
			11.2.4	Il diodo Fotorivelatore pin	442	
			11.2.5	Prestazioni del Link OOK/DD su Fibra	445	
		11.3	Variazi	oni sul Tema: WDM e Amplificatori Ottici	451	
			11.3.1	Aumentare il bit-rate in fibra: Wavelength Division		
				Multiplexing	451	
			11.3.2	Gli Amplificatori Ottici per aumentare la Sensibilità del		
				ricevitore OOK	452	
		11.4	Moden	n per Fibra Ottica II: Modulazione di Fase/Ampiezza e		
			Rivelaz	zione Coerente	456	
			11.4.1	Il Ricevitore Coerente con LASER Locale	456	
			11.4.2	Prestazioni dei Link Coerenti su Fibra	461	
		11.5	Rete di	Trasporto e Rete di Accesso: dalla Backbone all'FTTx	464	
			11.5.1	Le Tecnologie FTTx	464	
			11.5.2	Le PON per la Rete di Accesso	465	
			11.5.3	Un esempio di standard: la Gigabit PON (G-PON) ITU e	le	
				sue evoluzioni future	468	
4	4	Ottim	alità de	I codice di Huffman	469	
E	3	Teore	ma dell	a Codifica di Canale	471	
-	0	L'Algo	oritmo E	3CJR (Modificato)	477	
٦	D	L'Algo	oritmo "	Message-Passing" per Decodificare i Codici LDPC	485	
E		Deco	difica a	Cancellazioni Successive dei Codici Polari	493	
F		Rispo	sta in F	requenza di una Fibra Ottica Dispersiva	497	
C	G	Crite	rio di De	ecisione Ottimo e Funzione di Log-verosimiglianza	del	
		Rice	vitore IN	//DD	499	
E	Bibl	iografia			503	



CAPITOLO 1

COMUNICARE DIGITALE



"Captain Kirk to Enterprise, Captain Kirk to Enterprise."

-Star Trek TV movies series, 1966-1969

Il Capitano Kirk ha nelle sue mani un oggetto che negli anni '60 era piena fantascienza, e che oggi invece, dopo soli 60 anni, è il simbolo delle tecnologie digitali del XXI secolo: nel suo gergo, un "communicator" o, nel nostro, uno "smartphone"

Comunicazioni Digitali, I Edizione. di M. Luise Copyright © 2023 University of Pisa

2 COMUNICARE DIGITALE



Figura 1.1 The Black Cloud: rappresentazione dei principali collegamenti Internet nel mondo

1.1 Internet e "The Black Cloud"

Fred Hoyle (1915-2001) è stato un grande scienziato britannico, l'inventore del termine "Big Bang" per indicare l'origine dell'Universo, curiosamente coniato in senso ironico per controbattere l'idea della "nascita" di una cosa che Hoyle stesso riteneva invece immanente e non mutevole.

Hoyle è stato anche un grande scrittore di fantascienza, e il suo capolavoro è senz'altro *The Black Cloud*, apparso nel 1957, nel quale si narra la vicenda immaginaria di un'enorme nube di gas interstellare, una "nuvola" che si avvicina alla terra e la circonda schermando i raggi solari e mettendo a repentaglio la vita sul pianeta. Un gruppo di scienziati riesce a capire che la nuvola è in realtà un organismo vivente, e riesce ad entrare in comunicazione con questo essere dotato di una forma avanzatissima di intelligenza (stupita a sua volta che possa esistere vita intelligente su di un pianeta). La Nuvola comprende il danno che sta facendo e abbandona la Terra dopo aver cercato, senza riuscirvi, di infondere tramite messaggi sensoriali ultra-veloci un livello di conoscenza superumano nello scienziato-capo Chris Kingsley - Kingsley non sopravvive allo shock.

Durante uno dei colloqui, la Black Cloud spiega lucidamente che la sua propria grande intelligenza deriva da una moltitudine di menti elementari che amplificano i loro singoli intelletti attraverso dei link di comunicazione ramificati e pressoché istantanei (oggi diremmo, con il linguaggio di Internet, una *mesh network*) che creano di fatto un'unica mente innervata da una fittissima rete: *intelligence out of communications*.

Ho letto questo libro da ragazzino e ne sono rimasto affascinato - ogni tanto estraggo la *Nuvola Nera* (ho sia l'edizione originale in lingua inglese, sia la traduzione italiana Garzanti, penso introvabile) dalla libreria del mio studio; lo sfoglio, me lo rigiro tra le mani come oggetto di culto. Cosa meglio della nuvola nera puo rappresentare l'idea che abbiamo oggi di Internet? L'onniscienza, quasi la suprema intelligenza del motore di ricerca? Dopotutto, ogni volta che vogliamo rappresentare Internet su di un foglio di carta, in un diagramma, disegniamo una Nuvola...

SULLE SPALLE DEI GIGANTI 3



Figura 1.2 Claude E. Shannon (1916-2001)

E che cosa sarebbe Internet senza le tecnologie di comunicazione che la innervano e la costituiscono?

1.2 Sulle Spalle dei Giganti

Tutto nacque negli anni '40 con una serie di contributi scientifici di Claude E. Shannon, al tempo supersegreti e successivamente resi pubblici in un articolo fondamentale, del quale vediamo la prima pagina in Fig. 1.3. Il campo di studio, che annoverava anche i grandi Norbert Wiener a John von Neumann, era genericamente chiamato "cibernetica", ma Shannon se ne distaccò presto perchè ciò che aveva inventato costituì di fatto il fondamento di quella che oggi chiamiamo *civiltà dell'informazione*. Quando oggi attraverso WhatsApp riceviamo un videoclip, sappiamo che è di qualità scarsa rispetto all'originale perché è stato *compresso* per non occupare troppo spazio sullo smartphone. Siamo anche preoccupati che questo video, magari di carattere personale, non venga da qualcuno postato su Internet perché il nostro telefono è stato *hackerato*. Se invece abbiamo già salvato il video su YouTube e lo rivediamo sullo smartphone, magari si blocca ("freeza") o si sgrana perchè la *qualità* del collegamento wireless non è ottimale.

Ho elencato tutti i tre principali aspetti toccati in modo assolutamente visionario e in anticipo di mezzo secolo dal paper di Shannon che ho appena citato: l'*efficienza* nella rappresentazione dell'informazione, la *sicurezza* dell'informazione, la *qualità* nella trasmissione o memorizzazione dell'informazione. Aspetti sui quali la Teoria dell'Informazione dice cose fondamentali, entrate dopo decenni nella nostra vita quotidiana - un segno dell'Ingegneria con la I maiuscola.

1.3 Essere Digitali

Nel 1995 Nicholas Negroponte pubblica *Being Digital*, che scuote il grande pubblico, e che conferma ciò che noi addetti ai lavori conoscevamo già: il futuro è digitale, chi non sarà

4 COMUNICARE DIGITALE

Reprinted with corrections from *The Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, July, October, 1948.

A Mathematical Theory of Communication

By C. E. SHANNON

INTRODUCTION

THE recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist¹ and Hartley² on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have meaning; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one *selected from a set* of possible messages. The system must be designed to operate for each possible selection, not just the one which will actually be chosen since this is unknown at the time of design.

If the number of messages in the set is finite then this number or any monotonic function of this number can be regarded as a measure of the information produced when one message is chosen from the set, all choices being equally likely. As was pointed out by Hartley the most natural choice is the logarithmic function. Although this definition must be generalized considerably when we consider the influence of the statistics of the message and when we have a continuous range of messages, we will in all cases use an sentially logarithmic measure.

The logarithmic measure is more convenient for various reasons:

- 1. It is practically more useful. Parameters of engineering importance such as time, bandwidth, number of relays, etc., tend to vary linearly with the logarithm of the number of possibilities. For example, adding one relay to a group doubles the number of possible states of the relays. It adds 1 to the base 2 logarithm of this number. Doubling the time roughly squares the number of possible messages, or doubles the logarithm, etc.
- 2. It is nearer to our intuitive feeling as to the proper measure. This is closely related to (1) since we intuitively measures entities by linear comparison with common standards. One feels, for example, that two punched cards should have twice the capacity of one for information storage, and two identical channels twice the capacity of one for transmitting information.
- 3. It is mathematically more suitable. Many of the limiting operations are simple in terms of the logarithm but would require clumsy restatement in terms of the number of possibilities

The choice of a logarithmic base corresponds to the choice of a unit for measuring information. If the base 2 is used the resulting units may be called binary digits, or more briefly bits, a word suggested by J. W. Tukey. A device with two stable positions, such as a relay or a flip-flop circuit, can store one bit of information. N such devices can store N bits, since the total number of possible states is 2^N and $\log_2 2^N = N$. If the base 10 is used the units may be called decimal digits. Since

$$\log_2 M = \log_{10} M / \log_{10} 2$$
$$= 3.32 \log_{10} M,$$

¹Nyquist, H., "Certain Factors Affecting Telegraph Speed," Bell System Technical Journal, April 1924, p. 324; "Certain Topics in Telegraph Transmission Theory," A.J.E.E. Trans., v. 47, April 1928, p. 617.
²Hartley, R. V. L., "Transmission of Information," Bell System Technical Journal, July 1928, p. 535.

1





digitale rischierà di essere travolto dalla marea.

Sono passati quasi trent'anni nel momento in cui scrivo queste note, e il messaggio di Negroponte (un altro visionario...) è ancora più valido, e fortunatamente è stato recepito da molti strati della società civile. In Italia, uno scrittore di origine e tradizione umanistica che più umanistica non si può (Alessandro Baricco) si è sentito in dovere assai recentemente di dire qualcosa sul mondo digitale (il libro *The Game*, 2018) che conferma e addirittura estremizza la visione di Negroponte della civiltà digitale con i suoi benefici e i suoi pericoli.

La Teoria dell'Informazione appartiene alla "buona" ingegneria e alla "buona" informatica - è quindi un insieme di "technicalities"? No. Non solo. La Teoria dell'Informazione ha introdotto un gran numero di idee e concetti, come ho già accennato, relativi al linguaggio, alla privacy, all'efficienza, che sono ormai cultura generale e lezione quotidiana per ogni cittadino/a della moderna era delle *Information and Communication Technologies* (ICT). Ergo, chiunque voglia professionalmente occuparsi del mondo digitale *deve* a maggior ragione conoscere questi argomenti.

E il "braccio armato" della Teoria dell'Informazione sono le Comunicazioni Digitali, con le sue molte tecniche di utilizzo efficiente e sostenibile delle risorse fisiche (bande radio, cavi in fibra, cavi telefonici) per dare quella *connettività* su lunghissime come cortissime distanze, senza la quale Internet come oggi lo conosciamo semplicemente non esisterebbe.

1.4 Questi Appunti...

...sono il risultato di più di un decennio di docenza di Comunicazioni Ottiche, Comunicazioni Digitali e Teoria dell'Informazione all'Università di Pisa e sono scritti in lingua italiana per un motivo ben preciso: se da una parte è vero che il mondo dell'ICT usa l'inglese come una specie di Esperanto, è anche vero che ad oggi non esiste un testo o una raccolta che spieghi efficacemente i concetti di questi appunti in lingua italiana - una lacuna grave che a mio giudizio deve essere colmata, a maggior ragione perchè la comunità scientifica, accademica e industriale italiana in questo settore è *fortissima* nello scenario internazionale. Sospettiamo che nell'affernazione "Ma tanto questi argomenti si capiscono meglio se si legge un libro in inglese..." che talvolta si sente usata a giustificazione della penuria della letteratura in lingua italiana in questo campo nasconda un po' di pigrizia da parte degli autori...

Le pagine che seguono coprono argomenti classici e moderni nell'ambito delle tecnologie di comunicazione ed elaborazione dati all'interno di Internet, con una commistione tra Information Theory e Digital Communications, con una spruzzata di Signals and Systems, senza trascurare un po' di tecnologia. Questo mix rappresenta anche la mia personale esperienza e conoscenza, che ritengo utile mettere a disposizione dei lettori - non sarebbe leale scrivere di cose che non so. La scelta degli argomenti e del livello di profondità è quindi del tutto soggettiva e ovviamente non esaustiva. Penso però che rappresenti una bella rassegna dei fondamenti scientifici e tecnologici che ancora oggi, in alcuni casi a distanza di diversi decenni dall'invenzione, sono pienamente attuali e nient'affatto contingenti, patrimonio culturale perenne della civiltà digitale.

Ogni osservazione o suggerimento all'indirizzo di email marco.luise@unipi.it sarà benvenuta. Buona lettura.



CAPITOLO 2

L'A-B-C DI SEGNALI SISTEMI E COMUNICAZIONI



"Le système Chappe: Il primo telegrafo senza fili su lunga distanza sviluppato ai tempi di Napoleone Bonaparte. Trasportava *messaggi digitali* attraverso la Francia del XIX secolo tramite *segnali ottici* ricevuti e ritrasmessi da una rete di ripetitori che si estendeva da Lione a Venezia"

-Ripetitore del telegrafo ottico nei pressi del Castello di Rohan in Alsazia, Francia, restaurato nel 1998

Comunicazioni Digitali, I Edizione. di M. Luise Copyright © 2023 University of Pisa



Figura 2.1 Esempio di segnale periodico

Il motto di questo Capitolo è: torniamo ai principi di base. Infatti il suo scopo è quello di rivedere i concetti principali dei segnali analogici e digitali e della loro elaborazione. Il lettore che ha familiarità con tali argomenti può passare al Capitolo successivo.

2.1 Analisi di Fourier dei segnali analogici

Se è vero che questo testo si occupa di comunicazioni digitali, è anche vero che i dati digitali (audio, video, testo) che transitano su di una connessione Internet sono trasportati nella comunicazione da un qualche *segnale* di carattere fisico, cioè *analogico*, che a sua volta transita su di un mezzo fisico (fibra, radio, rame). Dunque il "Coltellino Svizzero" di ogni ingegnere delle telecomunicazioni che si occupa di comunicazioni digitali è lo studio dei segnali, e in particolare l' *analisi di Fourier*. In questa sezione non vogliamo/possiamo scendere nei dettagli di un argomento così vasto seppur centrale - richiameremo piuttosto i risultati principali e fisseremo una notazione che useremo nei prossimi capitoli per esporre molti concetti che stanno appunto alla base dell'analisi dei segnali a *tempo continuo* e a *tempo discreto*.

2.1.1 Segnali Periodici e Serie di Fourier

Sin dalle scuole elementari, sappiamo che il colore bianco è il risultato della combinazione di tutti i colori (ricordate il disco di Newton?) - un esempio opportunamente elementare di analisi di Fourier. Per entrare un po' più nel merito, l'analisi di Fourier ci dice che ogni segnale *periodico* $x_p(t)$ come in Fig. 2.1 (cioè tale che $x_p(t) = x_p(t + T_0)$ per un certo $T_0 > 0$ che viene definito *periodo*) può essere espresso come la somma di segnali periodici più semplici, cioè di *sinusoidi*:

$$x_p(t) = A_0 + A_1 \cos(2\pi f_0 t + \theta_1) + A_2 \cos(2\pi 2f_0 t + \theta_2) + \dots + A_k \cos(2\pi k f_0 t + \theta_k) + \dots$$
(2.1)

Notiamo che le sinusoidi oscillano alle frequenze kf_0 , chiamate frequenze armoniche, che sono multipli interi della frequenza fondamentale o frequenza di ripetizione $f_0 = 1/T_0$ (il valore costante A_0 si può intendere oscillante alla frequenza kf_0 con k = 0, cioè frequenza nulla).

Vediamo anche che la k-esima componente nello sviluppo (2.1) ha una certa ampiezza A_k , e una fase θ_k . Mentre il valore della frequenza di oscillazione kf_0 è il medesimo per qualunque segnale di periodo $T_0 = 1/f_0$, il valore di A_k e θ_k dipende dalla forma dello

ANALISI DI FOURIER DEI SEGNALI ANALOGICI 9



Figura 2.2 Keith Emerson davanti al Moog negli anni '70

specifico segnale $x_p(t)$ in analisi. La sequenza dei coefficienti A_k è chiamata spettro di ampiezza di $x_p(t)$, mentre la sequenza delle fasi θ_k è lo spettro di fase. Per rappresentare un segnale basta conoscerne gli spettri di ampiezza e di fase: dalla (2.1) possiamo infatti sintetizzare il segnale stesso nel dominio del tempo a partire da tali spettri. Per questo motivo la (2.1) è chiamata equazione di sintesi. Ed è ancora per questo motivo che il famoso strumento musicale elettronico Moog degli anni '70 fu chiamato sintetizzatore (vedi Fig. 2.2): esso implementava la (2.1), attraverso un geniale hardware analogico, per generare le (varie) forme d'onda periodiche usate nelle composizioni musicali prog-rock al posto dei suoni generati dagli strumenti tradizionali.

L'equazione (2.1) è la forma più semplice della *serie di Fourier* che rappresenta un segnale periodico. La difficoltà di questa rappresentazione risiede nel fatto che la sintesi *esatta* del segnale periodico richiede teoricamente un numero *infinito* di componenti. Tuttavia, tale rappresentazione può essere usata nella pratica *troncando* (cioè limitando) la serie alla somma di un numero (ristretto) di componenti significative, così come nel caso del MiniMoog. Nella figura 2.3 si può vedere come un treno di impulsi rettangolari si possa ottenere dalla sovrapposizione di un numero *finito* N di componenti sinusoidali elementari secondo la (2.1).

Naturalmente, quando si vuole ottenere una determinata forma d'onda, il problema è: quali sono i valori corretti di $A_k \in \theta_k$ da usare nell'equazione di sintesi? Per rispondere a questa domanda bisogna *analizzare* il segnale $x_p(t)$ mediante una specifica relazione, appunto, di *analisi* che dobbiamo ricavare. Per fare ciò, si ricorre ad una rappresentazione in forma *complessa* delle serie di Fourier. Ricordando la formula di Eulero per la funzione



Figura 2.3 Sintesi di un treno di impulsi rettangolari con un numero finito N=15 di componenti sinusoidali

coseno otteniamo:

$$A_k \cos(2\pi k f_0 t + \theta_k) = \frac{1}{2} \left[A_k \exp(j2\pi k f_0 t) \cdot \exp(j\theta) + A_k \exp(-j2\pi k f_0 t) \cdot \exp(-j\theta) \right]$$
(2.2)

La funzione oscillante reale viene espressa come la somma di due *vettori rotanti* nel piano complesso. Il primo, $\exp(j2\pi k f_0 t)$ ruota in senso orario a una frequenza f_0 giri/s (Hz) mentre il secondo, $\exp(-j2\pi k f_0 t)$, ruota (in senso antiorario) a una frequenza negativa $-f_0$ Hz. La somma dei due vettori rotanti complessi corrisponde all'oscillazione cosinusoidale a valori reali dalla quale siamo partiti.

A questo punto possiamo conglobare gli spettri di ampiezza e di fase in un unico coefficiente a valori complessi $X_k = A_k \exp[j\theta_k]$ (con k positivo) che è chiamato coefficiente di Fourier del segnale $x_p(t)$. Combinando la (2.1) con (2.2), otteniamo la seguente espressione delle serie di Fourier a coefficienti complessi X_k :

$$x_p(t) = \sum_{k=-\infty}^{\infty} X_k \exp(j2\pi k f_0 t)$$
(2.3)

che è esattamente equivalente alla forma a valori reali (2.1). Il lettore può ricavare autonomamente l'espressione di X_k per k negativi.

Ricavare l'equazione di analisi che stavamo cercando è assai più semplice per la serie espressa in questa forma. Dopo alcuni passaggi, si trova infatti

$$X_k = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} x_p(t) \exp(-j2\pi k f_0 t) dt$$
(2.4)

che ci permette di valutare la fase e l'ampiezza di ogni componente armonica nello sviluppo.

Esempio 2.1





Analizziamo esattamente il *treno di impulsi* $x_p(t)$ che è stato parzialmente sintetizzato nella Fig. 2.3. Il suo coefficiente di Fourier è dato da

$$X_{k} = \frac{1}{T_{0}} \int_{-T_{0}/2}^{T_{0}/2} x_{p}(t) \exp(-j2\pi k f_{0}t) dt = \frac{1}{T_{0}} \int_{-T_{0}/2}^{T_{0}/2} x_{p}(t) \cos(2\pi k f_{0}t) dt \quad (2.5)$$

dove abbiamo usato la proprietà di simmetria dovuta alla parità della forma d'onda. Adesso, considerando che $x_p(t)$ è a tratti costante, possiamo scrivere

$$X_{k} = \frac{1}{T_{0}} \int_{-T_{0}/4}^{T_{0}/4} \cos(2\pi k f_{0}t) dt = \frac{1}{T_{0}} \frac{\sin(2\pi k f_{0}t)|_{-T_{0}/4}^{10/4}}{2\pi k f_{0}} = \begin{cases} 1/2 & k = 0\\ 0 & k = 2m, m \neq 0\\ \frac{(-1)^{m}}{\pi k} & k = 2m + 1 \end{cases}$$
(2.6)

I coefficienti di Fourier X_k sono *reali* (grazie alla simmetria pari di $x_p(t)$); lo spettro di ampiezza di $x_p(t)$ è rappresentato nella Fig. 2.4 ed è un tipico *spettro a righe*.

2.1.2 Segnali non-periodici e la Trasformata di Fourier

Quanto detto finora relativamente all'analisi/sintesi di Fourier è applicabile soltanto a segnali *periodici*. Cosa succede se un segnale x(t) non ha alcuna ripetitività? La Figura 2.5 mostra



Figura 2.5 Impulso rettangolare

un esempio di impulso rettangolare definito come segue:

$$\operatorname{rect}(t/T) \stackrel{\triangle}{=} \begin{cases} 1 & |t| \le T/2 \\ 1/2 & |t| = T/2 \\ 0 & \text{elsewhere} \end{cases}$$

(2.7)

che evidentemente non presenta alcun periodo di ripetizione. Possiamo trattare questo tipo di segnali con l'analisi/sintesi di Fourier? La risposta è (ovviamente) positiva, purché si consideri un segnale non-periodico come un segnale con periodo di *lunghezza infinita*. La frequenza fondamentale diventa quindi infinitamente piccola, e di conseguenza due componenti frequenziali dello spettro del segnale, le cui frequenze differivano l'una dall'altra di $\Delta f = (k + 1)f_0 - kf_0 = f_0$, adesso diventano infinitesimamente vicine. Lo spettro del segnale in frequenza, che prima era *discreto* (si vedano le righe dello spettro in Fig. 2.4 separate da un "quanto" di frequenza f_0 , adesso diventa *continuo*, e l'equazione di analisi diventa

$$X(f) = \int_{t=-\infty}^{\infty} x(t) \exp(-j2\pi f t) dt$$
(2.8)

dove la frequenza f, che è l'argomento della quantità complessa X(f), assume tutti i valori *reali* da $-\infty$ a $+\infty$ con continuità.

Questa nuova versione X(f) del coefficiente di Fourier è chiamata *Trasformata Fourier* (TF) di x(t) e ha lo stesso significato di X_k . Indicheremo con |X(f)| l'ampiezza di X(f) e con $\angle X(f)$ la sua fase, in modo tale da avere ancora, rispettivamente, spettri di ampiezza e di fase (continui). Poiché adesso lo spettro è continuo, l'equazione di sintesi non è più una serie ma, al contrario, viene espressa nella forma di un *Integrale di Fourier*:

$$x(t) = \int_{f=-\infty}^{\infty} X(f) \exp(j2\pi ft) df$$
(2.9)

Questa relazione è anche chiamata Antitrasformata o Trasformata Inversa di Fourier di X(f). Fisicamente, le due relazioni (2.8)-(2.9) hanno lo stesso significato della coppia

coefficiente di Fourier-serie di Fourier: l'antitrasformata è un'equazione di *sintesi* che descrive come costruire il nostro segnale partendo da componenti più semplici (sinusoidi complesse), mentre la trasformata è l'equazione di *analisi* che descrive i valori specifici dell'ampiezza e della fase di ogni sinusoide che devono essere usati nel processo di sintesi per costruire il dato segnale. Si può facilmente dimostrare che la TF X(f) di un segnale reale x(t) (che denoteremo con $\mathcal{F}[x(t)]$) ha un tipo particolare di simmetria chiamata Hermitiana: $X(-f) = X^*(f)$, cioè, |X(-f)| = |X(f)|, e $\angle X(-f) = -\angle X(f)$.

Non ci soffermeremo a descrivere tutte le caratteristiche della TF nel progetto e nell'analisi dei segnali dell'ingegneria delle telecomunicazioni. Ci limiteremo a richiamare alcuni risultati elementari che useremo nei capitoli seguenti. Per esempio, lasciamo al lettore come semplice esercizio la dimostrazione che la TF della versione traslata nel tempo $x(t - t_0)$ di x(t) è

$$\mathcal{F}[x(t-t_0)] = X(f) \exp(-\jmath 2\pi f t_0)$$
(2.10)

in questo modo lo spettro di ampiezza del segnale non cambia , mentre la variazione dello spettro di fase è proporzionale alla frequenza di ogni componente.

Allo stesso modo, si dimostra facilmente cosa succede se effettuiamo una modulazione a radio-frequenza del segnale x(t) come segue:

$$x_{RF}(t) = x(t)\cos(2\pi f_0 t) = \frac{x(t)\exp(j2\pi f_0 t) + x(t)\exp(-j2\pi f_0 t)}{2}$$
(2.11)

dove f_0 è la frequenza portante. La variazione corrispondente della TF è

$$X_{RF}(f) = \frac{X(f+f_0) + X(f-f_0)}{2}$$
(2.12)

che corrisponde a una traslazione in frequenza delle componenti del segnale modulante.

Esempio 2.2

Ricaviamo la TF della funzione rect in (2.7):

$$X(f) = \int_{-\infty}^{+\infty} \operatorname{rect}(t) \exp(-j2\pi ft) dt = \int_{-T/2}^{+T/2} \exp(-j2\pi ft) dt$$
$$= \frac{\sin(\pi fT)}{\pi f} = T \frac{\sin(\pi fT)}{\pi fT} = T \operatorname{sinc}(fT)$$
(2.13)

dove abbiamo introdotto una particolare forma d'onda che useremo spesso in seguito: la cosiddetta funzione sinc (seno cardinale) rappresentata nella Fig.2.6 e definita come sinc(α) = sin($\pi \alpha$)/($\pi \alpha$).

2.1.3 Segnali a Banda limitata

Un segnale è *a banda limitata* se ha uno spettro di ampiezza |X(f)| diverso da zero in un intervallo limitato di frequenze [-B, B]. Questa definizione si può intendere anche in senso approssimato se si riscontra che fuori dall'intervallo [-B, B] le componenti frequenziali di un dato segnale, sebbene non esattamente nulle, sono così piccole da poter essere trascurabili.





Un esempio di segnale a banda rigorosamente limitata è il ben noto Frequency Raised Cosine (FRC) (o impulso di Nyquist) dato da

$$g_N(t) = \operatorname{sinc}(t/T) \frac{\cos(\beta \pi t/T)}{1 - (2\beta t/T)^2}$$

$$G_N(f) = \begin{cases} T & |f| < (1-\beta)/2T \\ \frac{T}{2} \left\{ 1 + \cos\left[\frac{\pi T}{\beta} \left(|f| - \frac{1-\beta}{2T}\right)\right] \right\} & (1-\beta)/2T \le |f| \le (1+\beta)/2T \\ 0 & \text{altrove} \end{cases}$$

$$(2.14)$$

il cui spettro/forma d'onda sono rappresentati nella Fig. 2.7. In questo caso, la banda è

 $B = (1 + \beta)/2T$, dove β , $0 \le \beta \le 1$, è un parametro che regola appunto l'occupazione spettrale del segnale ed è chiamato fattore di roll-off. Il valore 1/2T è la cosiddetta frequenza *di Nyquist* e corrisponde alla banda minima dell'impulso quando $\beta = 0$.

2.1.4 La funzione Delta di Dirac

La *delta* di Dirac $\delta(t)$ è un segnale particolare che useremo spesso nei prossimi capitoli. In realtà $\delta(t)$ non è un vero è proprio segnale, nel senso di una ben definita funzione del tempo, ma è un'entità matematica più complicata che viene ben definita nell'ambito della teoria delle distribuzioni o delle funzioni generalizzate. Essa viene definita infatti mediante una proprietà integrale detta del campionamento:

$$\delta(t) \quad : \quad \int_{-\infty}^{-\infty} x(t)\delta(t)dt = x(0) \tag{2.15}$$

dove x(t) è un qualsiasi segnale che non ha discontinuità in t = 0. Dalla (2.15) ricaviamo come caso particolare la proprietà

$$\int_{-\infty}^{-\infty} \delta(t)dt = 1 \tag{2.16}$$



Figura 2.7 Forma d'onda (a) e Trasformata di Fourier (b) dell'impulso FRC

16 L'A-B-C DI SEGNALI SISTEMI E COMUNICAZIONI

cioè si può affermare che la $\delta(t)$ ha un'"area unitaria".

Si capisce facilmente che non esiste alcuna funzione ordinaria avente la proprietà (2.15), che però può essere compresa e giustificata considerando una *successione* di funzioni. Consideriamo infatti una funzione rettangolare $\delta_{\varepsilon}(t)$ di durata $T = 2\varepsilon$ e ampiezza $A = 1/2\varepsilon$, come in Fig. 2.8(a). L''area" di questo segnale è pari a 1, indipendentemente da ε . Se adesso immaginiamo di diminuire la durata temporale di $\delta_{\varepsilon}(t)$, vedremo l'impulso aumentare di ampiezza mantenendo area unitaria come si vede in Fig.2.8(a). Il *limite* di tale impulso quando $\varepsilon \to 0$ è una rappresentazione euristica della $\delta(t)$: un "segnale" con una durata nel tempo pari a 0 (e quindi quasi ovunque nullo salvo t = 0) ma con ampiezza infinita per t = 0, in modo da avere area unitaria. La rappresentazione grafica della "funzione" $\delta(t)$ è mostrata nella Fig. 2.8(b).

Dunque se vogliamo dare una definizione della $\delta(t)$ che segua la nostra rappresentazione euristica, possiamo scrivere che

$$\delta(t) \stackrel{\triangle}{=} \lim_{\varepsilon \to 0} \delta_{\varepsilon}(t) = \lim_{\varepsilon \to 0} \frac{1}{2\varepsilon} \operatorname{rect}\left(\frac{t}{2\varepsilon}\right)$$
(2.17)

Questa relazione non solo dà un'idea dell'"aspetto" della delta, ma può essere usata nella pratica, purché i) $\delta(t)$ compaia in una relazione integrale simile alla (2.15), e ii) il limite nella (2.17) venga portato *fuori* dall'integrale, cioè venga calcolato *dopo* aver valutato l'integrale. In questo modo la definizione di $\delta(t)$ risulta ben posta, operativa, e porta risultati esatti e utili nell'analisi dei segnali e dei sistemi. Il lettore può usare tale definizione per verificare la proprietà di campionamento (2.16).

E' anche facile dimostrare che l'integrale definito della $\delta(t)$ su di un intervallo temporale finito del tipo $\int_b^a \delta(t) dt$ risulta essere uguale a 1 quando l'istante t = 0 cade entro l'intervallo (a, b), altrimenti il risultato è 0.

La delta di Dirac è particolare anche per quanto riguarda la sua Trasformata di Fourier $\Delta(f)$. Per prima cosa, il problema di calcolare la trasformata della $\delta(t)$ è ben posto dato che la trasformata di Fourier (2.8) è per definzione una trasformazione integrale; inoltre, il suo calcolo è banale, secondo la (2.15):

$$\Delta(f) = \int_{t=-\infty}^{\infty} \delta(t) \exp(-j2\pi ft) dt = \exp(-j2\pi ft)|_{t=0} = 1$$
 (2.18)

La TF dell'impulso unitario è quindi *costante* su tutte le frequenze, senz'alcuna limitazione di banda.

2.2 Filtraggio Lineare

Esaminiamo adesso alcune funzioni utili per elaborare i segnali trattati da un modem per comunicazioni digitali.

2.2.1 Sistemi e Segnali

L'operazione più semplice ma fondamentale è il *filtraggio*. Filtrare significa progettare un dispositivo, un circuito elettronico, una procedura software o, in una parola, un *sistema* che trasforma un segnale di *input/ingresso* x(t) in un segnale di *output/uscita* y(t) seguendo opportuni criteri. Secondo la nostra notazione:

$$y(t) = \mathcal{T}[x(\alpha); t] \tag{2.19}$$



Figura 2.8 Definizione della funzione di Dirac (a), e sua rappresentazione simbolica (b)

dove \mathcal{T} è un *operatore* che rappresenta la funzione di trasformazione del segnale implementata nel sistema, e dove l'elaborazione del segnale dipende in generale dall'intera forma d'onda in ingresso $x(\alpha)$. ma anche dal tempo. La famiglia di sistemi più semplice è quella dei *filtri lineari* che soddisfano il *principio di sovrapposizione degli effetti*. Supponiamo di conoscere

$$y_1(t) = \mathcal{T}[x_1(\alpha); t] , \quad y_2(t) = \mathcal{T}[x_2(\alpha); t]$$
 (2.20)

e costruiamo un segnale x(t) come un *sovrapposizione* (combinazione lineare) di x_1 e x_2 , cioè $x(t) = \alpha_1 x_1(t) + \alpha_2 x_2(t)$. Il sistema è un filtro *lineare* se e soltanto se

$$y(t) = \mathcal{T}[x(\alpha); t] = \mathcal{T}[\alpha_1 x_1(\alpha) + \alpha_2 x_2(\alpha); t] = \alpha_1 y_1(t) + \alpha_2 y_2(t)$$
(2.21)

Questo significa che l'uscita y(t) si può ottenere come una combinazione lineare delle "singole" uscite, applicando gli stessi coefficienti α_1 e α_2 dei singoli ingressi x_1 e x_2 .

In aggiunta alla proprietà di linearità, molti filtri che usiamo comunemente, sono anche *tempo-invarianti* o *stazionari*, cioè la trasformazione che viene applicata non cambia nel tempo. In particolare, sapendo che $y(t) = \mathcal{T}[x(\alpha); t]$, e introducendo "successivamente" nel sistema una versione dello stesso segnale traslata nel tempo, $x_{TS}(t) \stackrel{\triangle}{=} x(t - t_0)$, ci aspettiamo che l'output del filtro $y_{TS}(t)$ sia la stessa forma d'onda che avevamo inizialmente, traslata nel tempo della stessa quantità presente all'ingresso:

$$y_{TS}(t) = \mathcal{T}[x_{TS}(\alpha); t] = \mathcal{T}[x(\alpha - t_0); t] = y(t - t_0)$$
(2.22)

I Sistemi Lineari e Stazionari (SLS) sono particolarmente semplici da progettare e analizzare, e sono utili in molte applicazioni di elaborazione dei segnali, soprattutto nella rivelazione (ricezione) di segnali noti disturbati da rumore.



Figura 2.9 Rappresentazione grafica della trasformazione indotta da un sistema LTI

2.2.2 Caratterizzazione degli SLS nel Tempo e nella Frequenza

Le proprietà di ogni SLS sono completamente determinate dalla conoscenza di un particolare segnale che è associato al sistema, cioè la sua *risposta impulsiva*. La risposta impulsiva di un sistema LTI non è altro che l'uscita del sistema quando in ingresso si ha $\delta(t)$:

$$h(t) \stackrel{\Delta}{=} \mathcal{T}\left[\delta(\alpha); t\right] \tag{2.23}$$

In particolare, si può dimostrare che la risposta di un SLS ad un generico input x(t) è data da

$$y(t) = \int_{-\infty}^{\infty} x(\alpha)h(t-\alpha)d\alpha = x(t) \otimes h(t)$$
(2.24)

Questo modo di "combinare" i due segnali $x \in h$ per ottenere y merita un nome e una notazione specifici: l'operazione 2.24 viene chiamata *convoluzione aperiodica* tra due segnali, e si indica con il simbolo \otimes come indicato nella (2.24). Questo *operatore* è simmetrico, associativo e distributivo, come si può facilmente verificare.

La trasformazione di un segnale x(t) in ingresso ad un SLS che fornisce in uscita il segnale y(t) è spesso rappresentata in forma grafica come nella Fig. 2.9, dove la risposta impulsiva del sistema è indicata esplicitamente.

L'analisi di Fourier si rivela uno strumento importante anche per determinare le proprietà del SLS. Ritornando alla relazione "costitutiva" (2.24) del sistema, e calcolando la TF di entrambi i membri, troviamo:

$$Y(f) = X(f) \cdot H(f) \tag{2.25}$$

dove H(f) è la trasformata della risposta impulsiva h(t). Si vede che l'operazione (complicata) della convoluzione nel dominio del tempo diventa un'operazione (semplice) di prodotto tra due trasformate nel dominio della frequenza. Da qui si può comprendere che molto spesso l'analisi e il design di un SLS è più semplice nel dominio della frequenza piuttosto che nel dominio del tempo poiché implicitamente il sistema tratta la trasformata del segnale in ingresso con operazioni più semplici.

La quantità H(f) è la *risposta in frequenza* del filtro, ed è un modo alternativo per avere una completa caratterizzazione del SLS. In particolare, si può verificare che la risposta ad un input puramente sinusoidale $x(t) = \cos(2\pi f_0 t)$ è un altro segnale sinusoidale alla stessa frequenza nella forma

$$x(t) = \cos(2\pi f_0 t) \implies y(t) = |H(f_0)| \cdot \cos(2\pi f_0 t + \angle H(f_0))$$
(2.26)

L'effetto prodotto dal filtro sul segnale sinusoidale è una variazione di ampiezza di un fattore pari a $|H(f_0)|$ (la *risposta in ampiezza* del sistema alla frequenza f_0), e una rotazione di fase di $\angle H(f_0)$ (la *risposta in fase*). La versione in forma complessa della (2.26) è

$$x(t) = \exp(j2\pi f_0 t) \quad \Rightarrow \quad y(t) = H(f_0) \cdot \exp(j2\pi f_0 t) \tag{2.27}$$

FILTRAGGIO LINEARE 19



Figura 2.10 Risposta in ampiezza di un filtro passa-basso

Se cambiamo la frequenza del segnale sinusoidale, cambieranno l'ampiezza e la fase dell'uscita, quindi il sistema risponde in modo diverso a componenti a frequenze diverse. Per questo motivo H(f) è chiamata la risposta in frequenza, e indica anche che il sistema in generale presenta un comportamento *selettivo* al variare della frequenza: componenti diverse nello spettro del segnale vengono trattate in modo diverso.

Un esempio di comportamento *intenzionalmente* selettivo è dato dal filtro *passa-basso* ideale, la cui risposta in frequenza $H_{LP}(f)$ è rappresentata nella figura 2.10 (linea continua). Le componenti a "bassa" frequenza passano inalterate (vengono moltiplicate per $H_{LP} = 1$), mentre le componenti ad alta frequenza, cioè quelle al di fuori della banda passante B sono bloccate. Un comportamento simile si riscontra nella riposta H(f) ancora in Fig. 2.10 ma con linea tratteggiata, che non rappresenta un filtro "ideale", ma che ne è comunque una buona approssimazione, magari (più) facilmente realizzabile.

Esempio 2.3

Ricaviamo la risposta impulsiva e in frequenza di un *integratore a finestra mobile*. Questo SLS restituisce un segnale in uscita il cui valore a un determinato istante di tempo corrisponde al valor medio integrale del segnale in ingresso calcolato su di un intervallo temporale di durata T precedente all'istante considerato (ampiezza della finestra), come in Fig. 2.11:

$$h(t) = \frac{1}{T} \int_{t-T}^{t} x(\alpha) d\alpha \qquad (2.28)$$

Il lettore può verificare che la (2.28) definisce un SLS. Di conseguenza, la risposta impulsiva è per definizione l'uscita del sistema quando l'input è $\delta(t)$ ed è data da

$$y(t) = \frac{1}{T} \int_{t-T}^{t} \delta(\alpha) d\alpha$$
(2.29)

Come sappiamo dalle proprietà della delta di Dirac, quest'ultimo integrale risulta uguale a 1 se l'istante $\alpha = 0$ (dove si applica $\delta(\alpha)$) è compreso nell'intervallo di integrazione, altrimenti è uguale a 0. Questo significa che il segnale in uscita sarà uguale a 1 finchè 20 L'A-B-C DI SEGNALI SISTEMI E COMUNICAZIONI



Figura 2.11 Integratore a finestra mobile

 $t - T \le 0 < t$, e cioè se $0 < t \le T$, mentre sarà pari a 0 fuori da tale intervallo. La risposta impulsiva che stiamo cercando è un impulso rettangolare *causale* (cioè diverso da 0 solo per $t \ge 0$) che si può scrivere nella forma

$$h(t) = \frac{1}{T} \operatorname{rect}\left(\frac{t - T/2}{T}\right)$$
(2.30)

La risposta in frequenza è banalmente la trasformata di Fourier della h(t), cioè

$$H(f) = \operatorname{sinc}(fT) \exp(-\jmath \pi fT)$$
(2.31)

2.3 Filtraggio di Segnali aleatori

I segnali aleatori, chiamati anche *processi aleatori* nel gergo della teoria della probabilità, hanno un ruolo fondamentale nell'ingegneria delle telecomunicazioni. Un processo aleatorio è la rappresentazione matematica per tutti i tipi di *disturbi* derivanti da cause esterne non controllabili (interferenza, rumore termico ecc.) che possono degradare il segnale digitale da elaborare/ricevere.

D'altro canto, un processo aleatorio è anche la rappresentazione dell'*informazione digitale* trasportata dal segnale stesso: se i dati digitali fossero noti al ricevitore in anticipo, non servirebbe che il trasmettitore li inviasse! Come vedremo approfonditamente a partire dal prossimo Capitolo, la casualità di un segnale del genere è legata alla quantità di informazione che esso stesso contiene: più il segnale sembrerà aleatorio al ricevitore *prima* di essere ricevuto, più elevata sarà la quantità di informazione che porterà *dopo* la ricezione.

2.3.1 Fondamenti di segnali aleatori

Un segnale (processo) aleatorio è una funzione del tempo che ha una forma non (completamente) nota in anticipo. Il valore all'istante di tempo t di un segnale deterministico x(t), come quelli visti finora, è perfettamente noto per ogni possibile t perché se ne conosce la formula matematica, oppure perché l'abbiamo registrato e salvato in un file sul nostro computer. Al contrario, il valore di un segnale aleatorio ad un certo istante di tempo t è una variabile aleatoria. Un processo aleatorio è quindi un insieme di variabili aleatorie dipendenti dal tempo t, che indicheremo con la stessa notazione di un comune segnale deterministico, w(t). Se è vero che non è possibile conoscere a priori il valore di un segnale aleatorio per ogni t (che è un insieme di variabili aleatorie), è anche vero che di queste variabili si possono conoscere le *proprietà statistiche*. Quando osserviamo un segnale aleatorio, otteniamo una *realizzazione* delle variabili aleatorie ad ogni istante di tempo e quindi, dopo la nostra osservazione, quel che resta è un segnale deterministico (la realizzazione o *funzione-campione* del processo) che però si può prevedere in anticipo.

Per caratterizzare completamente le proprietà del segnale potrebbe sembrare sufficiente avere a disposizione, come descrizione statistica, la funzione densità di probabilità (pdf) $f_w(a;t)$ della variabile aleatoria w(t) all'istante t - in questo modo potremmo calcolare qualunque probabilità legata a w(t), ad esempio la probabilità che $w(t) \leq 0$. In realtà la conoscenza di $f_w(a;t)$ non è sufficiente. Per convincersene, basta considerare un caso in cui serva conoscere la probabilità che $w(t) \leq 0$ come sopra, ma che anche dopo un certo tempo τ , si abbia $w(t+\tau) \geq 0$. Questa è una probabilità congiunta che non si può calcolare semplicemente da $f_w(a;t)$ - serve è la pdf congiunta di secondo ordine $f_w(a_1,a_2;t,t+\tau)$. Generalizzando, si capisce che per caratterizzare completamente il processo aleatorio serve conoscere la classe delle pdf congiunte di ordine K

$$f_w(a_1, a_2, \dots, a_K; t, t + \tau_1, \dots, t + \tau_{K-1})$$
(2.32)

per *qualunque* K (arbitrariamente grande). Ricavare (e gestire) questa descrizione è piuttosto difficile, a meno di casi particolari. Uno di questi è il caso dei *processi Gaussiani* per cui le pdf (2.32) sono (congiuntamente) Gaussiane per ogni K.

2.3.2 Media, funzione di Autocorrelazione, e Densità Spettrale di Potenza

In pratica, è spesso sufficiente avere una conoscenza *parziale* delle proprietà statistiche di un processo aleatorio per risolvere gran parte dei problemi di ingegneria delle telecomunicazioni.

La proprietà più semplice è la funzione valor medio:

$$\eta_w(t) \stackrel{\triangle}{=} \int_{-\infty}^{-\infty} a f_w(a; t) da \tag{2.33}$$

che rappresenta, istante per istante, il valore del processo che ci si attende di osservare. La relazione (2.33) viene spesso sintetizzata con la notazione semplificata:

$$\eta_w(t) = \mathbf{E}\{w(t)\}$$
(2.34)

dove abbiamo introdotto l'*operatore lineare* $E\{\cdot\}$ chiamato Valore Atteso (Expectation) o (più frequentemente ma impropriamente) media cui definizione è evidente. In molti esempi di segnali aleatori che riguardano i sistemi di comunicazione (sia che trasportino informazione o che siano soltanto rumore), vedremo che $\eta_w(t)$ è una costante, e molto spesso $\eta_w(t) \equiv \eta = 0$.

Esempio 2.4

È dato il processo aleatorio

$$w(t) = A\cos(2\pi f_0 t + \Theta) \tag{2.35}$$

dove A e f_0 sono noti, mentre Θ è una variabile aleatoria uniformemente distribuita nell'intervallo $[-\pi/2, \pi/2)$. Ricaviamo per prima cosa il valore atteso $\eta_w(t)$. Secondo il

22 L'A-B-C DI SEGNALI SISTEMI E COMUNICAZIONI

teorema della media, invece di usare la definizione (2.33) che necessita della conoscenza di $f_w(a;t)$, possiamo calcolare "indirettamente" $\eta_w(t)$ usando le proprietà statistiche del parametro Θ da cui il processo dipende:

$$\eta_w(t) = \mathbb{E}\left\{w(t)\right\} = E\left\{A\cos(2\pi f_0 t + \Theta)\right\}$$
$$= \int_{-\infty}^{-\infty} A\cos(2\pi f_0 t + \theta)f_{\Theta}(\theta)d\theta = A\int_{-\pi/2}^{\pi/2}\cos(2\pi f_0 t + \theta)\frac{1}{\pi}d\theta$$
$$= -\frac{2A}{\pi}\sin(2\pi f_0 t)$$
(2.36)

Il valore atteso $\eta_w(t)$ di w(t) è una quantità statistica del *primo ordine*. Esso si calcola attraverso la pdf del primo ordine, e descrive il comportamento statistico del nostro segnale aleatorio quando è osservato ad un *singolo* istante di tempo. Un altro esempio di statistica del primo ordine è la *potenza media istantanea* del processo:

$$P_w(t) \stackrel{\triangle}{=} \mathbb{E}\{w^2(t)\} = \int_{-\infty}^{-\infty} a^2 f_w(a;t) da$$
(2.37)

L'esempio principale di statistica del *secondo* ordine è la *funzione di autocorrelazione* o semplicemente *autocorrelazione* del processo:

$$R_w(t,\tau) \stackrel{\triangle}{=} E\{w(t)w(t-\tau)\}$$

=
$$\int_{a_1=\infty}^{-\infty} \int_{a_2=\infty}^{-\infty} a_1 \cdot a_2 f_w(a_1,a_2;t,t-\tau) da_1 da_2 \qquad (2.38)$$

La funzione di autocorrelazione si calcola come *correlazione statistica* tra le due variabili aleatorie $w(t) e w(t - \tau)$ estratte dal processo ai due istanti di tempo rispettivamente t e $t - \tau$, e gioca un ruolo fondamentale nell'analisi spettrale di un segnale aleatorio (come del resto per un segnale determinato).

Per i segnali determinati, sappiamo che l'autocorrelazione è una funzione del solo ritardo τ , mentre, secondo la (2.38), $R(t, \tau)$ è in generale una funzione sia di t che di τ . Introduciamo allora la nozione di processo *stazionario in senso lato* (SSL). Supponiamo che

$$R_w(t,\tau) = R_w(\tau) \quad , \ \eta_w(t) \equiv \eta_w \tag{2.39}$$

Questo significa che le proprietà di valore atteso e autocorrelazione di un processo aleatorio di questo genere sono sempre le stesse indipendentemente dall'istante che scegliamo come origine dei tempi: la media non dipende dal tempo, e l'autocorrelazione dipende soltanto dal *ritardo* tra i due istanti $t_1 = t$ e $t_2 = t - \tau$ che consideriamo sul nostro segnale. (Anche) la potenza istantanea di un processo SSL è costante nel tempo, dato che

$$P_w(t) = R_w(t,0) \equiv R_w(0) = P_w$$
(2.40)

Per i processi SSL, possiamo definire la funzione *densità spettrale di potenza* (psd) come la trasformata di Fourier della funzione di autocorrelazione:

$$S_w(f) \stackrel{\Delta}{=} \int_{-\infty}^{+\infty} R_w(\tau) \exp(-j2\pi f\tau) d\tau = 2 \int_0^{+\infty} R_w(\tau) \cos(2\pi f\tau) d\tau \qquad (2.41)$$
(2.43)

dove la seconda uguaglianza deriva dal fatto (facile da dimostrare) che $R_w(\tau) = R_w(-\tau)$. La psd indica come è distribuita la potenza di un certo segnale (processo) sulle diverse componenti spettrali (frequenziali), e la potenza totale del segnale si può ricavare come

$$P_w = \int_{-\infty}^{+\infty} S_w(f) df \tag{2.42}$$

Come per ogni variabile aleatoria, la *varianza* di un processo SSL è data da $\sigma_w^2 = P_w - \eta_w^2$.

Esempio 2.5

Ritorniamo al processo dell'Esempio 4, ma supponiamo adesso che Θ sia uniforme in $[-\pi, \pi)$. Si trova facilmente che il processo ha valor medio *nullo* indipendentemente dal tempo. Ci si può chiedere allora se il processo sia SSL. Per verificarlo, calcoliamo la funzione di autocorrelazione $R_w(t, \tau) = E\{w(t)w(t - \tau)\}$ attraverso il teorema della media:

$$R_w(t,\tau) = \mathbb{E}\{\cos(2\pi f_0 t + \Theta)\cos(2\pi f_0(t-\tau) + \Theta)\}$$
$$\int_{-\infty}^{-\infty} A\cos(2\pi f_0 t + \theta)A\cos(2\pi f_0(t-\tau) + \theta)f_\Theta(\theta)d\theta$$
$$= \frac{A^2}{2\pi}\int_{-\pi}^{\pi}\cos(2\pi f_0 t + \theta)\cos(2\pi f_0(t-\tau) + \theta)d\theta$$

Usando le ben note formule di Werner, è facile dimostrare che:

$$R_w(t,\tau) = \frac{A^2}{2}\cos(2\pi f_0\tau) = R_w(\tau)$$
(2.44)

che *non* dipende da t. Come si può vedere, la nostra intuizione sulla stazionarietà in senso lato del processo era ben fondata!

Un particolare esempio di segnale aleatorio SSL è il processo di *rumore bianco*. La luce bianca è l'unica costituita da tutti i colori, come abbiamo già anticipato. Il colore è associato alla lunghezza d'onda, e la lunghezza d'onda è associata alla frequenza. Un rumore bianco è un processo a media nulla, e la sua psd è *costante in tutto l'intervallo di frequenze* - contiene cioè tutti possibili "colori":

$$S_w(f) = N_0/2 \quad \forall f \tag{2.45}$$

La corrispondente funzione di autocorrelazione si ottiene antitrasformando la densità costante:

$$R_w(\tau) = N_0/2 \cdot \delta(\tau) \tag{2.46}$$

Il rumore bianco è chiaramente una idealizzazione, poiché un segnale del genere non esiste: la sua potenza sarebbe infinita (dalla (2.42)). È però un buon modello per un segnale aleatorio che ha una banda molto più ampia rispetto a un altro segnale di "riferimento" o del sistema in esame. Questo è il caso del rumore presente in ogni dispositivo elettronico attivo o passivo (rumore termico, rumore di Johnson), o del rumore dell"ambiente" elettromagnetico circostante captato da un'antenna, che viene usualmente modellato come rumore bianco. In questi esempi, il processo di rumore è generato dalla sovrapposizione di una molteplicità di componenti elementari indipendenti (elettroni con moto casuale ecc.). Di conseguenza, per il ben noto teorema-limite centrale della teoria della probabilità (uno dei molti teoremi-limite,

come la legge debole/forte dei grandi numeri) qualsiasi set di variabili aleatorie $w(t_1)$, $w(t_2),...,w(t_N)$ ottenuto dall'osservazione del segnale agli istanti di tempo $t_1, t_2,...,t_N$ ha una pdf Gaussiana, e quindi il processo di rumore bianco è esso stesso Gaussiano.

2.3.3 Filtraggio dei segnali aleatori

2.3.3.1 Il problema fondamentale Il filtraggio è un'operazione fondamentale anche per i segnali aleatori, vediamo dunque di capire che cosa accade, in particolare, quando un processo w(t) viene filtrato con un SLS per ottenere il processo aleatorio di uscita n(t). Si può facilmente comprendere come si ricava la funzione valore atteso: dalla (2.33) vediamo infatti che il calcolo di questa funzione è ottenuto attraverso operazioni *lineari*: l'operatore "Valore Atteso" è lineare. Possiamo richiamare la proprietà "commutativa" degli operatori lineari ed affermare che:

$$\eta_n(t) = \mathbb{E}\{n(t)\} = \mathbb{E}\{\mathcal{T}[w(\alpha); t]\} = \mathcal{T}[\mathbb{E}\{w(\alpha)\}; t]$$
$$= \mathcal{T}[\eta_w(\alpha); t] = \eta_w(t) \otimes h(t)$$
(2.47)

In particolare, se w(t) è a media nulla, n(t) sarà ancora a media nulla, indipendentemente dal tipo di filtro che usiamo.

Esempio 2.6

Ritorniamo al processo parametrico dell'Esempio 4

$$w(t) = A\cos(2\pi f_0 t + \Theta) \tag{2.48}$$

dove Θ è una variabile aleatoria uniformemente distribuita nell'intervallo $[-\pi/2, \pi/2)$. I suo valor medio risulta essere

$$\eta_w(t) = -\frac{2A}{\pi}\sin(2\pi f_0 t)$$
(2.49)

Che succede se w(t) viene filtrato dall'integratore a finestra mobile dell'Esempio 3? Chiamiamo n(t) il segnale filtrato. Dalla (2.47) sappiamo che $\eta_n(t) = \eta_w(t) \otimes h(t)$. Ma $\eta_w(t)$ è *sinusoidale* nel tempo, quindi il risultato del filtraggio può essere calcolato facilmente attraverso la nozione della risposta del filtro in frequenza, come nella (2.26):

$$\eta_n(t) = -\frac{2A}{\pi} |H(f_0)| \sin(2\pi f_0 t + \angle H(f_0))$$

= $-\frac{2A \operatorname{sinc}(f_0 T)}{\pi} \sin(2\pi f_0 t + \pi f_0 T)$ (2.50)

Poiché già sappiamo dalla (2.47) come calcolare il valore atteso di un processo filtrato a partire da quello d'ingresso, possiamo risolvere completamente il problema del filtraggio di un processo SSL (che, si dimostra facilmente, resta SSL anche dopo filtraggio) se riusciamo a trovare l'espressione della funzione di autocorrelazione o della densità spettrale di potenza del processo di uscita. Il calcolo è un po' più complesso rispetto alla (2.47), ma il risultato è semplice:

$$S_n(f) = |H(f)|^2 \cdot S_w(f)$$
(2.51)

dove $|H(f)|^2$ è la risposta in potenza del SLS.

FILTRAGGIO DI SEGNALI ALEATORI 25



Figura 2.12 Segnale a durata limitata s(t) originario (a) e disturbato da rumore r(t) (b)

2.3.3.2 *Filtraggio Adattato* Immaginiamo che il segnale s(t) rappresentato nella Fig. 2.12 (a) venga inviato su di un lungo doppino telefonico. All'altro lato del collegamento si riceve tale segnale disturbato da Rumore Bianco Gaussiano Additivo(AWGN, Additive White Gaussian Noise) w(t) con psd $N_0/2$ - in Fig. 2.12 (b) vediamo un possibile segnale di questo tipo. Il segnale ricevuto r(t) = s(t) + w(t) viene filtrato con un SLS avente risposta impulsiva h(t) per ottenere il segnale $y(t) = r(t) \otimes h(t) = x(t) + n(t)$, dove x e n sono rispettivamente il segnale originale filtrato e il rumore filtrato. Supponiamo che s(t) sia limitato nell'intervallo temporale [0, T) come in Fig. 2.12 (a), e che la risposta all'impulso del filtro di ricezione sia

$$h(t) = \frac{1}{E_s} s(T - t)$$
 (2.52)

cioè lo stesso impulso trasmesso ma i) ribaltato nel tempo, ii) ritardato in modo da renderlo di nuovo causale, e iii) scalato di un fattore $E_s = \int s^2(t) dt$ che rappresenta per definizione l'*energia* del segnale s.

L'uscita del filtro in ricezione all'istante t = T è:

$$y(T) = x(T) + n(T) = \frac{1}{E_s} \int_{-\infty}^{+\infty} s(\alpha)s((T - (T - \alpha)))d\alpha + N = 1 + N \quad (2.53)$$

dove N è una variabile Gaussiana (perché ottenuta tramite una trasformazione lineare da un processo Gaussiano) a media nulla (perché il rumore bianco ha media nulla) e con varianza

$$\sigma_N^2 = \int_{-\infty}^{+\infty} S_n(f) df = \int_{-\infty}^{+\infty} \frac{N_0}{2} |H(f)|^2 df$$
$$= \frac{N_0}{2} \frac{1}{E_s^2} \int_{-\infty}^{+\infty} |S(f)|^2 df = \frac{1}{2E_s/N_0}$$
(2.54)

Adesso possiamo calcolare il *rapporto segnale rumore* (SNR, Signal to Noise Ratio) all'uscita del filtro di ricezione come il rapporto tra il quadrato del campione di segnale all'istante t = T (potenza istantanea del segnale) e la varianza del rumore filtrato allo stesso istante (potenza del rumore):

$$\operatorname{SNR} \stackrel{\triangle}{=} \frac{x^2(T)}{\sigma_N^2} = \frac{1}{(2E_s/N_0)^{-1}} = \frac{2E_s}{N_0}$$
(2.55)

Si può dimostrare che il valore (2.55) rappresenta il *miglior* SNR (cioè il massimo) che si possa ottenere filtrando r(t) attraverso *qualsiasi* SLS. La risposta impulsiva h(t) come nella (2.52) è quella che "meglio si adatta" al segnale ricevuto, nel senso di esaltare il segnale e abbattere il rumore, per cui questo particolare SLS viene chiamato *filtro adattato* (matched filter, MF).

Esempio 2.7

Vediamo che forma particolare assume il filtro adattato quando l'impulso lanciato su di un canale AWGN è un impulso rettangolare di durata T, che nelle comunicazioni digitali rappresenta, come vedremo successivamente, il cosiddetto formato NRZ (No Return to Zero):

$$s(t) = \operatorname{rect}\left(\frac{t - T/2}{T}\right)$$

L'energia dell'impulso è

$$E_{s} = \int s^{2}(t) dt = \int \operatorname{rect}^{2} \left(\frac{t - T/2}{T} \right) dt = \int_{0}^{T} dt = T$$
 (2.56)

e quindi il matched filter (2.52) è

$$h(t) = \frac{1}{T} \operatorname{rect}\left(\frac{T/2 - t}{T}\right)$$
(2.57)

Per rivelare l'impulso in AWGN dovremo filtrare un segnale ricevuto $r(t) \operatorname{con} h(t) \operatorname{come}$ sopra, e poi valutare l'uscita all'istante t = T:

$$y(T) = r(t) \otimes h(t)|_{t=T} = \int r(\alpha)h(t-\alpha) d\alpha \Big|_{t=T} = \int r(\alpha)h(T-\alpha) d\alpha$$
$$= \frac{1}{T} \int r(\alpha)s(\alpha) d\alpha = \frac{1}{T} \int_0^T r(t) dt$$
(2.58)

La variabile ottima di decisione per rivelare l'impulso NRZ in AWGN si ottiene quindi da un semplice filtro integratore che effettua la "media" del segnale ricevuto r(t) sul periodo dell'impulso (si indica talvolta con il nome di "Integrate & Dump"). La catena "trasmittente e ricevente" di quest'esempio è raffigurata nella Fig. 2.13.



Figura 2.13 Rivelazione di un impulso NRZ con filtro adattato "I & D"

2.4 Segnali e Sistemi in Banda Passante

2.4.1 Equivalenti in Banda-Base di segnali Passa-Banda

I segnali coinvolti negli apparati per le telecomunicazioni su bande radio o su fibra ottica sono di tipo passa-banda, cioè occupano una banda di frequenza molto piccola rispetto al valore della frequenza centrale della banda stessa (la frequenza portante f_0).

La forma generale di un segnale passa-banda sinusoidale alla frequenza f_0 senz'alcuna modulazione digitale (cioè un'oscillazione non modulata alla frequenza portante) è

$$x_{BP}(t) = A\cos(2\pi f_0 t + \vartheta)$$

dove A è l'ampiezza del segnale e ϑ è la fase. Una forma alternativa della (2.59) è

$$\begin{aligned} x_{BP}(t) &= A\cos(\vartheta)\cos(2\pi f_0 t) - A\sin(\vartheta)\sin(2\pi f_0 t) \\ &= x_I\cos(2\pi f_0 t) - x_Q\sin(2\pi f_0 t) \\ &= \Re\{(x_I + \jmath x_Q)\exp(\jmath 2\pi f_0 t)\} \\ &= \Re\{x\exp(\jmath 2\pi f_0 t)\} \end{aligned}$$
(2.60)

dove sono presenti nuove quantità oltre all'ampiezza e alla fase della sinusoide, che useremo in seguito. Per prima cosa, abbiamo introdotto le componenti In fase e in Quadratura $(I/Q) x_I = A\cos(\vartheta) e x_Q = A\sin(\vartheta)$ come le "proiezioni" della sinusoide lungo le due componenti ortogonali della portante alla frequenza f_0 , rispettivamente $\cos(2\pi f_0 t)$ e $-\sin(2\pi f_0 t)$. Inoltre, abbiamo introdotto la notazione in forma complessa dell'equivalente *in banda base* $x \stackrel{\triangle}{=} x_I + j x_Q$ del segnale sinusoidale $x_{BP}(t)$. Il modulo di x è l'ampiezza del segnale sinusoidale, la fase di x ne è la fase iniziale, $x = A \exp(j\vartheta)$. Il tutto è rappresentato in forma mnemonica in Fig. 2.14

Sebbene le relazioni tra le varie quantità siano chiare, il segnale oscillante alla frequenza f_0 che abbiamo appena considerato non porta alcun messaggio degno di essere inviato. Supponiamo però che l'ampiezza e/o la fase di $x_{BP}(t)$ vengano fatte variare lentamente nel tempo attraverso una qualche legge di modulazione digitale per associare al segnale un certo messaggio che debba essere trasmesso. Avremo allora:

$$x_{BP}(t) = A(t)\cos(2\pi f_0 t + \vartheta(t))$$

(2.59)



Figura 2.14 Rappresentazione grafica delle componenti I/Q e dell'ampiezza/fase di un segnale sinusoidale



Figura 2.15 Spettri di campioni di un segnale passa-banda e dell'equivalente in banda-base

$$= A(t)\cos(\vartheta(t))\cos(2\pi f_0 t) - A(t)\sin(\vartheta(t))\sin(2\pi f_0 t)$$

$$= x_I(t)\cos(2\pi f_0 t) - x_Q(t)\sin(2\pi f_0 t)$$

$$= \Re\{(x_I(t) + \jmath x_Q(t))\exp(\jmath 2\pi f_0 t)\}$$

$$= \Re\{x_{BB}(t)\exp(\jmath 2\pi f_0 t)\}$$
(2.61)

Le quantità (componenti I/Q, ampiezza/fase, equivalente in banda base) che abbiamo prima menzionato non sono più delle costanti, bensì cambiano (lentamente) nel tempo, dove "lentamente" è da intendersi "su una scala temporale molto maggiore di $1/f_0$ ". Lo spettro di $x_{BP}(t)$ come in (2.61) non è più monocromatico, ma occupa una banda stretta attorno alla frequenza portante f_0 . La banda passante di questo spettro è $B \ll f_0$, e quindi il segnale è quasi-monocromatico o passa-banda. Al contrario, x(t) ha uno spettro che è confinato alla cosiddetta banda base (cioè per frequenze vicine allo 0), con una banda B_{BB} molto minore di f_0 perché la legge di modulazione fa variare lentamente le componenti in banda base. Poiché il segnale è complesso, lo spettro di x(t) non avrà simmetria Hermitiana attorno allo 0. La Figura 2.15 mostra degli esempi fittizi di spettri in banda passante (modulati, quasi-monocromatici), e l'equivalente in banda-base (senza simmetria Hermitiana).

2.4.2 Equivalente in banda base del Rumore Gaussiano Bianco

Anche i processi aleatori stazionari caratterizzati da una densità spettrale di potenza di tipo passa-banda possono essere rappresentati attraverso gli equivalenti in banda base complessi. Senza entrare in troppi dettagli, un processo w(t) Gaussiano bianco con d.s.p. $S_w(f) = N_0/2$ su di una certa banda passante B (Fig. 2.16) ammette come equivalente in



Figura 2.16 Spettri del rumore bianco in banda passante e base

banda base un processo ancora Gaussiano bianco a valori complessi

$$w_{BB}(t) = w_I(t) + \jmath w_Q(t) \tag{2.62}$$

dove le due componenti in banda base sono processi indipendenti Gaussiani, entrambi a media nulla e con densità spettrale di potenza $S_{w_I}(f) = S_{w_Q}(f) = N_0$ costante sulla banda B/2. (Fig. 2.16). La varianza (potenza) σ_w^2 del rumore w(t) si trova facilmente dalla Fig. 2.16:

$$\sigma_w^2 = \int S_w(f) \, df = 2 \int_{f_0 - B/2}^{f_0 + B/2} \frac{N_0}{2} \, df = N_0 B \tag{2.63}$$

La varianza di entrambe le componenti in banda base del rumore è anche

$$\sigma_I^2 = \sigma_Q^2 = \int_{-B/2}^{B/2} N_0 \, df = N_0 B \tag{2.64}$$

La relazione tra la potenza effettiva del processo in banda passante e di quello in banda base è infatti

$$\sigma_w^2 \stackrel{\triangle}{=} \mathbb{E}\{w^2(t)\} = \frac{1}{2}\mathbb{E}\{|w_{BB}|^2(t)\} = \frac{1}{2}\left[\mathbb{E}\{w_I^2(t) + w_Q^2(t)\}\right]$$
$$= \frac{1}{2}\left[\mathbb{E}\{w_I^2(t)\} + \mathbb{E}\{w_Q^2(t)\}\right] = \frac{1}{2}\left[\sigma_w^2 + \sigma_w^2\right] = \sigma_w^2 \tag{2.65}$$

Talvolta, anziché utilizzare separatamente le d.s.p. $S_{w_I}(f)$ e $S_{w_Q}(f)$ delle componenti del processo, si utilizza la d.s.p. *totale* $S_{w_{BB}}(f)$, definita attraverso la funzione di autocorrelazione $R_{w_{BB}}(\tau)$ del processo $w_{BB}(t)$ a valori complessi:

$$R_{w_{BB}}(\tau) \stackrel{\triangle}{=} \mathbb{E}\{w(t)w^*(t-\tau)\} \quad , \quad S_{w_{BB}}(f) \stackrel{\triangle}{=} \int R_{w_{BB}}(\tau)e^{-\jmath 2\pi f\tau} d\tau \qquad (2.66)$$

Per il nostro rumore bianco, la d.s.p. totale è pari alla somma delle d.s.p. delle due componenti indipendenti: $S_{w_{BB}}(f) = 2N_0 \operatorname{rect}(f/B)$.

2.4.3 II modulatore I-Q

Una volta nota la frequenza portante, il segnale passa-banda è totalmente rappresentato dal suo equivalente in banda-base (chiamato anche *inviluppo complesso*) x(t). L'Equazione (2.61) mostra che $x_{BP}(t) = x_I(t) \cos(2\pi f_0 t) - x_Q(t) \sin(2\pi f_0 t)$. Questa non è soltanto



Figura 2.17 Architettura generale di un modulatore I-Q

una rappresentazione matematica, ma corrisponde anche all'architettura del cosiddetto *modulatore I-Q* che è usato nella pratica per generare un segnale qualsiasi modulato in banda passante a partire dalle sue componenti I-Q in banda base, come in Fig. 2.17. Nei capitoli seguenti, adotteremo sistematicamente la notazione in forma complessa per i segnali equivalenti in banda-base, implicitamente considerando che tutti i segnali sono inviluppi complessi, a meno che non sia specificato altrimenti.

2.4.4 II demodulatore I-Q

Come vedremo successivamente, le varie tecniche di modulazione digitale associano secondo differenti criteri l'informazione digitale alle componenti $x_I(t) e x_Q(t)$, e il segnale viene poi inviato sul mezzo fisico (banda radio, cavo in fibra o in rame) dopo modulazione I/Q passa-banda. Il ricevitore deve in generale ricostruire le componenti in banda base $x_I(t)$ e $x_Q(t)$ per recuperare (*demodulare*) i dati digitali. Il modo più semplice per fare ciò parte dall'espressione generale del segnale passa-banda:

$$x_{BP}(t) = \Re\{x_{(t)} \exp(j2\pi f_{0}t)\}$$

=
$$\frac{x(t) \exp(j2\pi f_{0}t) + x^{*}(t) \exp(-j2\pi f_{0}t)}{2}$$
(2.67)

Se immaginiamo di effettuare una conversione di frequenza con una oscillazione complessa alla frequenza $-f_0$ e ampiezza 2 otteniamo

$$x_{BP}(t) \cdot 2\exp(-\jmath 2\pi f_0 t) = x(t) + x^*(t)\exp(-\jmath 2\pi \cdot 2f_0 t)$$
(2.68)

Questo segnale complesso (quindi con due componenti reale/immaginaria) convertito in frequenza contiene due termini: il primo è proprio quello che ci interessa, mentre il secondo, che non ci è utile, è centrato alla frequenza $-2f_0$. Per liberarci di quest'ultimo e mantenere solo il primo termine utile, processiamo il segnale (entrambe le componenti I/Q) con un filtro *passa-basso* con una banda pari a quella di x(t) per rimuovere le componenti alla frequenza doppia $-2f_0$. Il risultato di questo procedimento è semplice:

$$\{x_{BP}(t) \cdot 2\exp(-j2\pi f_0 t)\} \otimes h_{LP}(t) = x(t) = x_I(t) + jx_O(t)$$
(2.69)

dove $h_{LP}(t)$ è la risposta impulsiva del filtro passa-basso.

Questa procedura di demodulazione per recuperare $x_I(t \in x_Q(t) \text{ non è puramente})$ matematica, bensì descrive (anche) lo schema del *demodulatore I-Q* rappresentato in Fig. 2.18, così come viene implementato nella maggior parte dei moderni ricevitori radio. Si può



Figura 2.18 Architettura generale di un demodulatore I-Q

notare che il prodotto tra il segnale (reale) ricevuto in banda passante $x_{BP}(t)$ e l'oscillazione (complessa) $2 \exp(-\jmath 2\pi f_0 t)$ è implementato come una *coppia* di prodotti reali tra il primo e le due componenti reale e immaginaria della seconda; il filtro passa-basso viene poi applicato a entrambe le componenti.

2.5 Analisi di Fourier dei segnali Digitali

2.5.1 Segnali Analogici e Segnali Digitali

Nella sezione precedente, abbiamo rivisto i concetti e i principali risultati dell'analisi di Fourier dei segnali a tempo continuo, cioè *analogici*. Gli stessi risultati si possono estendere ai segnali *digitali*, o, più correttamente, a *tempo discreto*. Il modo più immediato per generare un segnale digitale è quello di usare un convertitore analogico-digitale (ADC, Analog to Digital Converter), ovvero, nel gergo dell'analisi dei segnali, fare un'operazione di *campionamento*. Campionare un segnale analogico x(t) ad una certa frequenza di campionamento f_s campioni/sec (o Hz) (ovvero con un periodo di campionamento $T_s = 1/f_s$, significa estrarre da x(t) una sequenza di campioni x[n] tale che

$$x[n] = x(nT_s) \tag{2.70}$$

Le parentesi quadre indicano che l'indice di tempo n è discreto, contrariamente al tempo continuo t che è invece tra parentesi tonde. Il valore all'istante n di x[n] è un numero reale, e quindi per rappresentarlo servirebbe teoricamente una sequenza infinita di cifre. Nella pratica, la conversione A/D rappresenta ogni valore come un intero su un numero fisso(limitato) di cifre binare. Di conseguenza si avrà un (piccolo) errore di rappresentazione: l'uscita del convertitore ADC sarà $x_q[n] = x[n] + q[n]$, dove x_q è la versione quantizzata di x[n], e q[n] è il cosiddetto rumore di quantizzazione. Quando il numero di bit nella forma rappresentazione digitale di $x_q[n]$ è grande (≥ 16), il rumore di quantizzazione può tranquillamente essere trascurato (come faremo in seguito sistematicamente).

2.5.2 TF di Segnali Digitali

Il campionamento e la conversione analogico-digitale sono alla base dell'Elaborazione Numerica dei Segnali. Spesso quest'ultima consiste nell'applicazione di tecniche che si basano sull'analisi di Fourier, pertanto è utile richiamare i concetti fondamentali della trasformate di Fourier (anche) per i segnali a tempo-discreto.

32 L'A-B-C DI SEGNALI SISTEMI E COMUNICAZIONI

Generalizzando le nozioni già viste per i segnali analogici, si può facilmente dimostrare che una sequenza non periodica x[n] si può scomporre come

$$x[n] = \frac{1}{f_s} \int_{-f_s/2}^{+f_s/2} \overline{X}(f) \exp(j2\pi nf/f_s) df$$
(2.71)

dove $\overline{X}(f)$ è la TF della sequenza x[n]. L'Equazione (2.71) è chiaramente un'equazione di *sintesi* molto simile alla (2.8) per segnali analogici. La corrispondente equazione di *analisi* è

$$\overline{X}(f) = \sum_{n=-\infty}^{+\infty} x[n] \exp(-j2\pi nf/f_s)$$
(2.72)

Tuttavia, esiste una differenza sostanziale tra le TF di un segnale analogico e di un segnale discreto. Dalla (2.71) si può notare che il segnale digitale si può sintetizzare partendo da un intervallo *finito* di componenti continue in frequenza, mentre il segnale analogico richiede una componente *per ogni* frequenza sull'intero asse reale. La spiegazione di quanto appena detto si trova nel fatto che la TF $\overline{X}(f)$ di una sequenza è una funzione *periodica* della frequenza, con periodo pari alla frequenza di campionamento f_s (in quanto tutte le oscillazioni $\exp(-j2\pi nf/f_s)$ nell'equazione di analisi sono esse stesse periodiche dello stesso periodo). Quindi le uniche componenti indipendenti (significative) di x[n] si trovano entro un intervallo di frequenze di ampiezza pari a f_s , e per la sintesi del segnale non ne servono ulteriori.

Per capire meglio la questione della periodicità della TF di una sequenza e/o della limitatezza dell'intervallo frequenziale delle componenti, esaminiamo le proprietà delle sinusoidi campionate. La sequenza estratta dal campionamento di un segnale sinusoidale analogico $x_1(t)$ alla frequenza f_1 è

$$x_1[n] = x_1(n/f_s) = \cos(2\pi n f_1/f_s)$$
(2.73)

Il rapporto f_1/f_s viene talvolta indicato come F_1 ed è chiamato frequenza *normalizzata*. Supponiamo adesso di avere un *secondo* segnale sinusoidale $x_2(t)$ alla frequenza $f_2 = f_1 + f_s \neq f_1$, che risulta ovviamente (molto) diverso da $x_1(t)$. Campionando (anche) $x_2(t)$ otteniamo

$$x_2[n] = \cos(2\pi n f_2/f_s) = \cos(2\pi n f_1/f_s + 2\pi n) = \cos(2\pi n f_1/f_s) = x_1[n] \quad (2.74)$$

Dopo il campionamento cioè, i due segnali sinusoidali, assai diversi nel dominio delle frequenze analogiche, diventano identici! Questo vuol dire che, per sintetizzare un segnale nel dominio digitale, non ha senso pensare a componenti di frequenza al di fuori dell'intervallo "base" di ampiezza f_s , che sono repliche identiche di componenti già presenti e già considerate – non sono per niente necessarie.

Esempio 2.8

Dato un segnale esponenziale a tempo continuo causale $x(t) = \exp(-t/\alpha)u(t)$, effettuiamo un campionamento con un intervallo pari a T_s . Otteniamo la sequenza

$$x[n] = x(nT_s) = \exp(-nT/\alpha) = a^n \mathfrak{u}[n]$$
(2.75)

dove u(t) e u[n] sono le funzioni gradino rispettivamente a tempo continuo e a tempo discreto, uguali a 1 quando il loro argomento è non negativo e uguali a 0 altrimenti. La

costante reale $a \stackrel{\triangle}{=} \exp(-T/\alpha) < 1$ dipende dalla costante di tempo del segnale analogico e dalla frequenza di campionamento. La TF della sequenza ottenuta è

$$\overline{X}(f) = \sum x[n] \exp(-j2\pi n f T_s) = \sum_{n=0}^{\infty} a^n \exp(-j2\pi n f T_s)$$
$$= \sum_{n=0}^{\infty} [a \exp(-j2\pi f T_s)]^n = \frac{1}{1 - a \exp(-j2\pi f T_s)}$$
(2.76)

dove la certezza sulla convergenza della serie scaturisce dalla condizione $|a \exp(-\jmath 2\pi f T_s)| =$ |a| < 1. Lo spettro di fase e di ampiezza di x[n] sono mostrati nella Fig. 2.19. Gli spettri sono mostrati soltanto su di un intervallo pari a $\pm f_s/2$ poiché le due funzioni sono come sappiamo periodiche di tale periodo.

Molte delle proprietà già viste per le TF dei segnali analogici continuano ad essere ancora valide per i segnali digitali, con qualche piccola modifica. Ci riferiamo in particolare ai teoremi della simmetria Hermitiana, del ritardo, della modulazione, e così via.

Esempio 2.9

Esiste un analogo a tempo discreto della funzione Delta di Dirac (Sect. 2.1.4)? La risposta è più semplice di quanto ci si aspetti. Mentre la $\delta(t)$ è un segnale del tutto particolare, la sua versione a tempo discreto è la sequenza, particolare ma del tutto convenzionale,

$$\delta[n] \stackrel{\triangle}{=} \begin{cases} 1 & n = 0\\ 0 & \text{elsewhere} \end{cases}$$

La sua TF sarà chiaramente

$$\overline{\Delta}(f) = \sum_{n=-\infty}^{+\infty} \delta[n] \exp(-j2\pi nf/f_s) = 1$$
(2.78)

come la TF di $\delta(t)$.

Il problema fondamentale del campionamento in ambito frequenziale è ovviamente il seguente: se la nostra sequenza x[n] è il risultato del campionamento di un segnale analogico x(t), avente TF X(f), qual è l'espressione della TF $\overline{X}(f)$ della sequenza x[n]? La risposta risiede nella seguente formula di Poisson:

$$\overline{X}(f) = f_s \sum_{k=-\infty}^{+\infty} X(f - kf_s)$$
(2.79)

Questa relazione ci dice che la TF di x[n] è la sovrapposizione di una serie infinita di ripetizioni della TF originale del segnale analogico, con un periodo di ripetizione nel dominio frequenziale uguale alla frequenza di campionamento f_s . Ovviamente, il risultato sarà una funzione periodica $\overline{X}(f)$ di periodo f_s , come già sappiamo.

33

(2.5



Figura 2.19 Spettri di ampiezza (a) e fase (b) di una sequenza esponenziale

Nella periodicizzazione implicata dalla formula di Poisson (2.79) può nascere una certa "interferenza" tra ripetizioni adiacenti dello spettro analogico originale, che creano una discrepanza tra lo spettro del segnale campionato e quello originario. Tale discrepanza è chiamata (errore di) *aliasing*. Un esempio di aliasing dovuto al campionamento è mostrato in Fig. 2.20 (b) che rappresenta la TF $\overline{X}(f)$ della sequenza x[n] ottenuta campionando il segnale analogico x(t) la cui TF X(f) è mostrata nella Fig. 2.20 (a). La sovrapposizione di spettri adiacenti *non* si verifica soltanto se il segnale originale x(t) ha una banda *B* limitata, e se la frequenza di campionamento è maggiore di 2*B*. La relazione $f_s > 2B$ è fondamentale ed è chiamata *condizione di Nyquist*: quando è verificata, non ci sarà aliasing nello spettro del segnale campionato. Si confronti a tal proposito la Fig. 2.20 (b), che rappresenta la $\overline{X}(f)$ ricavata dal segnale campionato in Fig. 2.20 (a), ma questa volta con una frequenza di campionamento che soddisfa la condizione di Nyquist. Si nota che la copia "principale" con k = 0 nella formula di Poisson (2.79) (cioè quella che ricade nell'intervallo principale $[-f_s/2, f_s/2)$ della TF) è una copia fedele dello spettro originale X(f) (a meno di un trascurabile fattore di scala f_s).

2.5.3 Filtraggio e Interpolazione di una segnale digitale

Anche i concetti generali di filtro lineare e di SLS si trasportano pressoché invariati nel dominio dei segnali digitali. La caratterizzazione completa di un SLS a tempo discreto viene dalla conoscenza della relativa risposta impulsiva

$$h[n] \stackrel{\Delta}{=} \mathcal{T}[\delta[n]; n] \tag{2.80}$$

e la relazione ingresso-uscita del sistema è (ancora) la convoluzione aperiodica (a tempo discreto) tra il il segnale di ingresso x[n] e h[n]:

$$y[n] = x[n] \otimes h[n] \stackrel{\Delta}{=} \sum_{m=-\infty}^{+\infty} x[m] \cdot h[n-m] = \sum_{m=-\infty}^{+\infty} h[m] \cdot x[n-m]$$
(2.81)

Il sistema sarà ugualmente caratterizzato (anche) dalla propria risposta in frequenza

$$\overline{H}(f) \stackrel{\triangle}{=} \sum_{n=-\infty}^{+\infty} h[n] \exp(-j2\pi n f/f_s)$$
(2.82)

in modo che la relazione ingresso-uscita nel dominio della frequenza è (ancora)

$$\overline{Y}(f) = \overline{X}(f)\overline{H}(f) \tag{2.83}$$

Se la risposta all'impulso h[n] del SLS ha un numero pari ad N, cioè *finito*, di campioni non nulli, il filtro è detto FIR (Finite Impulse Response), altrimenti è del tipo IIR (Infinite Impulse Response). Le operazioni necessarie per implementare un filtro FIR sono rappresentate in Fig. 2.21, dove i blocchi etichettati con z^{-1} introducono un ritardo di un campione sulla sequenza al loro ingresso (elemento di ritardo unitario), e dove abbiamo ipotizzato che il filtro sia casuale, cioè che h[n] = 0 quando n < 0.

Dopo aver trasformato un segnale analogico in una sequenza digitale attraverso il campionamento, e dopo aver elaborato il (segnale digitale), oppure dopo aver riletto un segnale digitale precedentemente memorizzato su di un supporto fisico (un hard-disk, una pen drive ecc.), può essere necessario ricostruire un segnale analogico a partire dalla sequenza



Figura 2.20 Spettro di un segnale analogico (a), del segnale digitale campionato con aliasing (b), e del segnale digitale campionato senza aliasing (c)

ANALISI DI FOURIER DEI SEGNALI DIGITALI 37



Figura 2.22 Impulsi interpolanti: (a) interpolatore a mantenimento; (b) interpolatore lineare

elaborata o riletta. Questa operazione inversa al campionamento viene chiamata *interpolazione* e in pratica viene relizzata attraverso un *Convertitore Digitale-Analogico*(DAC, Digital-to-Analog Converter).

La forma generale di un segnale interpolato $\hat{x}(t)$ è

$$\hat{x}(t) = \sum_{n = -\infty}^{+\infty} x[n] \cdot p(t - nT_s)$$
(2.84)

dove x[n] è la sequenza da interpolare, e p(t) è la forma dell'impulso di interpolazione che caratterizza lo specifico interpolatore utlizzato. Se p(t) è l'impulso rettangolare di Fig. 2.22 (a), avremo un interpolatore *a mantenimento*, perchè il valore del segnale analogico nell'intervallo temporale $[nT_s, (n+1)T_s)$ è pari a quello del campione della sequenza x[n]che viene *mantenuto* costante per tutto l'intervallo. Se invece p(t) è l'impulso triangolare come in Fig. 2.22 (b), l'andamento del segnale interpolato $\hat{x}(t)$ nell'intervallo $[nT_s, (n + 1)T_s)$ si ottiene, istante per istante, dalla retta di interpolazione lineare che unisce i due valori della sequenza x[n] all'istante nT_s e x[n + 1] all'istante $(n + 1)T_s$.

La versione nel dominio della frequenza della (2.84) è molto semplice:

$$X(f) = P(f)X(f)$$
(2.85)

dove sia $\hat{X}(f)$ che P(f) sono TF di segnali analogici, mentre $\overline{X}(f)$ è la TF della sequenza x[n] da interpolare.

2.5.4 Il teorema del campionamento

Dallo studio del problema del campionamento e dell'interpolazione, con le formule di Poisson (2.79) e di interpolazione (2.85), possiamo risolvere un problema fondamentale:



Figura 2.23 Funzionamento dell'interpolatore cardinale nel dominio della frequenza

una volta che un segnale analogico viene campionato, e i suoi campioni vengono salvati, è possibile recuperare *fedelmente* il segnale originale senza alcuna perdita? A primo impatto si direbbe di NO, poiché quando si converte un segnale a tempo continuo in una sequenza a tempo discreto, tutto ciò che "sta tra un campione e l'altro" sembra esser perso per sempre. MA... dando un'occhiata nel dominio della frequenza si scopre che non è così. Se il segnale ha banda *B* limitata e soddisfiamo la condizione di Nyquist $f_s \ge 2B$, sappiamo già che non avremo aliasing, e vediamo una "copia" *non distorta* (cioè fedele) dello spettro del segnale analogico nello spettro della nostra sequenza (la copia con k = 0 nella formula di Poisson (2.79)). Resta però il problema di recuperare questa copia e ritornare nel dominio analogico. La risposta è relativamente semplice: è necessario usare un interpolatore appropriato che preserva la copia con k = 0 mentre cancella tutte le altre. Riferendoci allo spettro del segnale analogico in Fig. 2.20 (a), si capisce immediatamente che serve un impulso di interpolazione p(t) la cui TF P(f) sia *piatta* entro l'intervallo di frequenze $[-f_s/2, f_s/2)$ che contiene la copia principale con k = 0, e che sia *nulla* al di fuori di quest'intervallo - si dovrà poi anche compensare il fattore di scala f_s nella relazione di Poisson.

In breve, per ricostruire un segnale a banda limitata campionato secondo la condizione di Nyquist, dobbiamo utilizzare un interpolatore

$$P(f) = \frac{1}{f_s} \operatorname{rect}\left(\frac{f}{f_s}\right) = T_s \operatorname{rect}(fT_s)$$
(2.86)

Usando questo interpolatore, è evidente che $\hat{X}(f) = X(f)$, cosicché possiamo dire che il problema di ricostruire il segnale campionato è risolto. L'impulso interpolante che corrisponde alla P(f) è banalmente $p(t) = \operatorname{sinc}(t/T_s)$, e quindi la formula di interpolazione è

$$\hat{x}(t) = \sum_{n=-\infty}^{+\infty} x[n] \cdot \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right)$$
(2.87)

chiamata *interpolazione cardinale*. Poiché sappiamo che $\hat{X}(f) = X(f)$, è anche chiaro che $\hat{x}(t) = x(t)$. Questo risultato fondamentale prende il nome di *teorema del campionamento* di Shannon.

2.6 Trasmissione e Ricezione di Dati Digitali

2.6.1 Segnali Dati in Banda Base

Il formato più semplice di un segnale che "trasporta" informazione digitale è quella di un segnale digitale con formato Non Ritorno a Zero (NRZ)

$$x(t) = A \sum_{n=-\infty}^{+\infty} a[n]g(t-nT)$$
 (2.88)

dove A è l'ampiezza del segnale, g(t) è un impulso rettangolare di ampiezza unitaria e durata nel tempo pari a T:

$$g_{NRZ}(t) = \operatorname{rect}\left(\frac{t - T/2}{T}\right)$$
(2.89)

 $g(t) = \operatorname{rect}[(t - T/2)/T)]$, a[n] è la sequenza dei dati digitali (anche chiamati *simboli*), e T è l'intervallo di segnalazione o *intervallo di simbolo*. Il segnale non contiene alcuna oscillazione portante, quindi non è di tipo passa banda, e il suo spettro, cone vedremo in seguito, occupa la cosiddetta *banda base*. La frequenza di presentazione dei dati (simboli) $R \triangleq 1/T$ è chiamata *velocità di segnalazione, frequenza di simbolo* (symbol rate) o *frequenza di clock*, e si misura in *simboli/sec* o *Baud*. Segnali di questo tipo sono utilizzati all'interno di ogni apparato digitale per rappresentare flussi informativi, e non necessariamente per effettuare una comunicazione a distanza. Tipicamente, i dati a[n] sono *binari* sui livelli 0 e 1 e la costante A dipende dal formato fisico del segnale. Se la durata temporale dell'impulso rettangolare g(t) è più piccola di T, il formato del segnale dati è chiamato RZ (Ritorno a Zero) poiché l'impulso associato ad ogni bit torna a 0 prima che venga trasmesso il bit successivo.

Un altro formato molto utilizzato prevede che i livelli siano *antipodali* binari, cioè a[n]è uguale a -1 o +1 a seconda della specifica sequenza (*pattern*) di dati da trasmettere. I valori +1 e -1 sono associati ai valori logici rispettivamente 1 e 0 dei bit della sequenza dati. Un formato di questo tipo viene utilizzato nella trasmissione su reti Ethernet su cavo in rame, congiuntamente ad un particolare impulso di simbolo g(t) di durata T chiamato Manchester:

$$g_{Manch}(t) = \operatorname{rect}\left(\frac{t - T/4}{T/2}\right) - \operatorname{rect}\left(\frac{t - 3T/4}{T/2}\right)$$
(2.90)

Il segnale dati nella forma appena descritta si può vedere come il risultato di un'interpolazione (2.84) della sequenza digitale a[n] caratterizzata da una frequenza di campionamento R = 1/T, con un impulso di interpolazione p(t) = g(t). Questi formati di segnale vengono tradizionalmente indicati con il nome "storico" di *Pulse-Amplitude Modulation* (PAM).

Il ricevitore non conosce in anticipo il valore specifico dell'*n*-esimo simbolo a[n] all'istante di tempo *n*. Pertanto, il segnale dati viene modellato come un *processo aleatorio* e come tale viene trattato con gli strumenti e metodi che abbiamo affrontato nel paragrafo 2.3. L'ipotesi più semplice, che chiariremo nel Capitolo 3, è che a[n] sia una sequenza di variabili aleatorie i.i.d. (indipendenti e identicamente distribuite) a valori equiprobabili.

Abbiamo chiamato i dati a[n] "simboli" e non semplicemente "bit" perché non necessariamente sono a due (soli) livelli. Ad esempio, i simboli dei segnali dati che erano usati nelle storiche (anni '90...) comunicazioni digitali ISDN in banda base sul doppino telefonico erano *quaternari*, cioè appartenenti all'*alfabeto* {-3, -1, +1, +3}. Quale di questi livelli viene effettivamente inviato ad un certo istante di clock dipende dalla regola (chiamata *mappatura*) che associa ognuno di essi ad una specifica coppia di bit di informazione, ad esempio $00 \rightarrow -3$, $01 \rightarrow -1$, $11 \rightarrow 1$, $10 \rightarrow 3$. In questo esempio, ogni singolo simbolo trasporta l'informazione di *due* bit. Pertanto, la *frequenza di bit* (o *bit-rate*) R_b della trasmissione (misurata in bit/s) è in generale *diversa* della frequenza di simbolo: nel nostro caso, $R_b = 2R$.

In generale, se il simbolo a[n] può assumere uno tra gli M possibili valori di un alfabeto M-ario, la frequenza di bit sarà $R_b = \log_2(M) \cdot R$. L'alfabeto più semplice per i segnali M-PAM è

$$\mathcal{A} \equiv \{-(M-1), -(M-3), \dots, -1, +1, \dots, (M-3), (M-1)\}$$
(2.91)

Dobbiamo ancora chiarire il motivo per cui si utilizzano questi formati multilivello (piuttosto che il semplice formato binario "nativo"), che nel campo delle comunicazioni digitali sono ormai la regola piuttosto che l'eccezione. Il motivo di ciò è da ricondursi alla *banda* dei segnali in questione.

2.6.2 Spettro dei Segnali Dati

Abbiamo già visto l'"aspetto" di un segnale dati nel dominio del tempo. Quel che resta da scoprire è l'andamento del suo spettro e l'occupazione di banda. Per prima cosa occorre trovare la funzione di autocorrelazione del segnale dati in banda base (2.88):

$$R_{x}(t,\tau) \stackrel{\simeq}{=} E\left\{x(t)x(t-\tau)\right\}$$

$$= A^{2}E\left\{\sum_{n=-\infty}^{+\infty} a[n]g(t-nT)\sum_{m=-\infty}^{+\infty} a[m]g(t-\tau-mT)\right\}$$

$$= A^{2}\sum_{n=-\infty}^{+\infty}\sum_{m=-\infty}^{+\infty} E\left\{a[n]a[m]\right\}g(t-nT)g(t-\tau-mT)$$

$$= A^{2} \cdot A_{2}\sum_{n=-\infty}^{+\infty}g(t-nT)g(t-\tau-nT)$$
(2.92)

dove A_2 è la potenza media di a[n], $A_2 = E\{a^2[n]\}$, e dove si tiene in considerazione che a[n] e a[m] sono a media nulla e sono indipendenti quando $n \neq m$. Si può vedere che il segnale dati *non* è un processo SSL in quanto la funzione di autocorrelazione dipende anche da t e non soltanto da τ . Vediamo anche che $R_x(t,\tau)$ è una funzione *periodica* di periodo T rispetto al tempo t per qualunque valore di τ . Quando ciò si verifica, il processo si dice *ciclostazionario*.

Per i processi ciclostazionari, si definisce una densità spettrale di potenza *media* (quella che si può fisicamente misurare con un *analizzatore di spettro*) basata sulla media temporale della funzione periodica di autocorrelazione, dove l'operazione di media viene eseguita su un intervallo di tempo pari al periodo:

$$\rho_x(\tau) \stackrel{\triangle}{=} \frac{1}{T} \int_{t=-T/2}^{T/2} R_x(t,\tau) dt$$
(2.93)

La psd media è definita adesso come l'usuale TF della $\rho_x(\tau)$.

Se riprendiamo la (2.92), otteniamo

$$\rho_{x}(\tau) \stackrel{\triangle}{=} \frac{A^{2}}{T} \int_{t=-T/2}^{T/2} A_{2} \sum_{n=-\infty}^{+\infty} g(t-nT)g(t-\tau-nT)dt
= \frac{A^{2} \cdot A_{2}}{T} \sum_{n=-\infty}^{+\infty} \int_{t=-T/2}^{T/2} g(t-nT)g(t-\tau-nT)dt
= \frac{A^{2} \cdot A_{2}}{T} \int_{-\infty}^{+\infty} g(t)g(t-\tau)dt$$
(2.94)

Come si può notare, la funzione (media) di autocorrelazione del segnale dati aleatorio x(t) è proporzionale alla funzione di autocorrelazione del segnale deterministico g(t). Calcolando la TF della $\rho_x(\tau)$ si trova

$$S_x(f) = \frac{A^2 \cdot A_2}{T} |G(f)|^2$$
(2.95)

Alla fine di questi calcoli, utili seppur lunghi, si trova che la funzione di psd di un segnale dati è determinata dal (quadrato del) modulo della TF dell'impulso dei dati. Di conseguenza, l'occupazione di banda di un segnale digitale è uguale alla banda del suo impulso elementare.

Per il formato NRZ, la densità spettrale di potenza è quindi

$$S_x(f) = A^2 \cdot A_2 T \operatorname{sinc}^2(fT) \tag{2.96}$$

mostrata in Fig. 2.24, la cui banda è teoricamente *infinita*. Una misura pratica dell'occupazione di banda di uno spettro di questo tipo è la cosiddetta *banda al primo nullo*, cioè l'intervallo di frequenze tra la frequenza 0 e il primo nullo dello spettro, nel nostro caso pari a 1/T, pari cioè alla velocità di simbolo R. Da qui si intuisce l'importanza dei formati multilivello: assegnata la banda utilizzabile sul mezzo fisico (cavo in rame, fibra ottica, canale radio), con un formato a M livelli è possibile aumentare la velocità di bit R_b di un fattore $\log_2(M)$ mantenendo la banda invariata - il limite di questa tecnologia verrà analizzato nel Cap. 6.

Se lo spettro del segnale dati deve essere strettamente limitato in una banda *B*, piuttosto che filtrare un formato NRZ, causando distorsione e una cattiva ricezione dei dati, è più opportuno ricorrere a un segnale dati *intrinsecamente limitato in banda*, come l'impulso FRC di Nyquist $g_N(t)(2.14)$, la cui banda è $B = R(1 + \beta)/2$. La forma più comune di impulso a banda limitata usato nella trasmissione dati è il cosiddetto SRFRC o Square-Root Frequency Raised Cosine la cui TF è proporzionale alla radice quadrata della TF dell'impulso di Nyquist FRC $g_N(t)$:

$$G(f) = T\sqrt{G_N(f)/T} = \sqrt{TG_N(f)}$$
(2.97)

la cui forma d'onda è

$$g(t) = \frac{\sin\left(\pi(1-\beta)\frac{t}{T}\right) + 4\beta\frac{t}{T}\cos\left(\pi(1+\beta)\frac{t}{T}\right)}{\pi\frac{t}{T}\left[1 - 16\beta^2\left(\frac{t}{T}\right)^2\right]}$$
(2.98)

e la cui banda è ancora pari a $R(1 + \beta)/2$.

La potenza totale del segnale dati si trova mediante l'integrale della (2.95)

$$P_x = \frac{A^2 \cdot A_2}{T} \int_{-\infty}^{+\infty} |G(f)|^2 df = \frac{A^2 \cdot A_2}{T} \int_{-\infty}^{+\infty} |g(t)|^2 dt = A^2 \cdot A_2 E_g/T \quad (2.99)$$



- /

Figura 2.24 Spettro di potenza di un segnale dati NRZ. Scala lineare (a) e scala logaritmica (dB) (b)

dove E_g è l'energia dell'impulso dati g(t). In molti casi,come ad esempio per l'impulso NRZ, o l'impulso SRFRC, l'energia dell'impulso risulta essere $E_g = T$, cosicché $P_x = A^2 \cdot A_2$. Per un segnale PAM multilivello con alfabeto a M simboli $\{-(M-1), -(M-3), \ldots, -1, +1, \ldots, (M-3), (M-1)\}$ si ottiene $A_2 = (M^2 - 1)/3$.

2.6.3 Ricezione di un segnale dati su canale AWGN

Uno dei problemi fondamentali nelle comunicazioni digitali è la ricezione affidabile dei dati quando il segnale fisico è disturbato da rumore aleatorio. Lo scenario più comune è la trasmissione del segnale su di un canale AWGN, come avviene nelle comunicazioni radio

tra due punti senza ostacoli intermedi (ad esempio la ricezione della televisione digitale via satellite), e come già brevemente delineato nell' Esempio 2.3.3.2. Considerando il caso semplice di un segnale binario con un impulso g(t) avente energia pari a T, il segnale ricevuto è

$$r(t) = \sqrt{P_r} \sum_{n=-\infty}^{+\infty} a[n]g(t - nT) + w(t)$$
(2.100)

dove w(t) è AWGN con spettro di potenza pari a $N_0/2$ e dove la potenza di segnale ricevuta è P_r . Di conseguenza, l'energia ricevuta per ogni simbolo dati è pari a $E_s = P_r T$.

Il problema della ricezione dati si può facilmente sintetizzare come segue: il ricevitore, sulla base del segnale ricevuto r(t), deve recuperare fedelmente (o *rigenerare*) il flusso dei dati trasmessi, nonostante la presenza del disturbo. Quando la *realizzazione* del processo di rumore è particolarmente sfavorevole, può accadere che il dato rigenerato $\hat{a}[n]$ è diverso da a[n] per un qualche n: viene a crearsi un *errore di simbolo*. Ovviamente, il particolare valore di n per cui si produce un errore non è predicibile, proprio per la natura aleatoria dei dati e del rumore. Ciò che possiamo valutare sulla base delle proprietà statistiche del rumore (e anche dei dati) è la *probabilità* che si verifichi un errore di simbolo (e/o un errore sui bit associati al simbolo). Quindi il parametro più importante che caratterizza l'affidabilità di un canale di comunicaizone è il *Symbol Error Rate* (SER) o il *Bit Error Rate* (BER), dove *rate*, in italiano frequenza, è l'interpretazione pratica di *probabilità*. Ricevere in modo affidabile il segnale dati vuol dire quindi: *determinare le operazioni per elaborare il segnale ricevuto* r(t) *in modo da rigenerare il flusso di simboli dei dati originali a*[n] *con il minimo SER*.

Tale problema non è semplice da risolvere, dovremo quindi procedere gradualmente. Supponiamo *in primis* di dover estrarre un *singolo* dato isolato a[0] con impulso g(t), cioè supponiamo di ricevere $r(t) = \sqrt{P_r}a[n]g(t) + w(t)$. Questo problema è stato risolto negli anni 50' ricorrendo a un'elegante interpretazione geometrica basata sull'espansione di r(t) in uno spazio Hilbertiano di funzioni ortonormali (ortogonali). La soluzione complessivamente equivale allo schema di elaborazione mostrato in Fig. 2.25 (a) usato per i segnali PAM con simboli i.i.d. ed equiprobabili - il ricevitore a *filtro adattato*. Come abbiamo anticipato nell'Esempio 2.52, il segnale viene filtrato da un SLS adattato a g(t), cioè avente risposta impulsiva $h_{MF}(t) = g^*(t_0 - t)/E_g h_{MF}(t) = g(t_0 - t)/E_g$ con un t_0 opportuno (dipendente dal formato dell'impulso). Il segnale risultante è

$$y(t) = \sqrt{P_r a[0]g_r(t) + n(t)}$$
(2.101)

dove $g_r(t) = g(t) \otimes h_{MF}(t)$ è l' impulso dati filtrato, e n(t) è rumore Gaussiano a media nulla filtrato, avente densità spettrale di potenza $S_n(f) = N_0 |H_{MF}(f)|^2/2$.

Il segnale filtrato viene valutato (campionato) all'istante di tempo t_0 per ottenere la cosiddetta *variabile di decisione*

$$y(0) = \sqrt{P_r a[0]g_r(t_0) + n(0)}$$
(2.102)

Per un impulso NRZ, $t_0 = T$, mentre per un impulso SRFRC simmetrico $t_0 = 0$. In entrambi i casi, $g_r(t_0) = 1$. Inoltre, n(0) è una variabile aleatoria Gaussiana (poiché estratta da un processo Gaussiano) con media nulla e varianza

$$\sigma_n^2 = \frac{N_0}{2} \int_{-\infty}^{+\infty} |H_{MF}(f)|^2 df = \frac{N_0}{2E_g^2} \int_{-\infty}^{+\infty} |G(f)|^2 df = \frac{N_0}{2E_g} = \frac{N_0}{2T}$$
(2.103)

L'uscita "soft" (cioè a valori reali) y(0) viene infine elaborata con un rivelatore "hard" che applica la non linearità *a soglia* mostrata in Fig. 2.25 (b) per tornare nel "mondo digitale". La soglia produce infatti il simbolo rigenerato a valori discreti $\hat{a}[0]$.

(a)



(b)



La probabilità di errore di simbolo (che, nel nostro caso di segnale binario, è pari alla probabilità di errore di bit) è

$$SER = BER = \frac{1}{2} \Pr\{y(0) > 0 \mid a[0] = -1\} + \frac{1}{2} \Pr\{y(0) \le 0 \mid a[0] = +1\}$$
$$= \frac{1}{2} \Pr\{-\sqrt{P_r} + n(0) \le 0\} + \frac{1}{2} \Pr\{\sqrt{P_r} + n(0) \le 0\}$$
$$= \Pr\{n(0) > \sqrt{P_r}\} = Q\left(\frac{\sqrt{P_r}}{\sigma_n}\right) = Q\left(\sqrt{\frac{2E_s}{N_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$
(2.104)

dove $Q(\cdot)$ è la ben nota funzione integrale Gaussiana

$$Q(\alpha) \stackrel{\triangle}{=} \frac{1}{\sqrt{2\pi}} \int_{-\alpha}^{+\infty} \exp(-\beta^2/2) d\beta$$
(2.105)

Mediante calcoli un po' più elaborati, si trova anche il SER di una *M*-PAM generica con alfabeto (2.91):

$$SER = 2\left(1 - \frac{1}{M}\right)Q\left(\sqrt{\frac{3}{M^2 - 1}}\sqrt{\frac{2E_s}{N_0}}\right) \quad (M-PAM) \tag{2.106}$$

Al posto del rapporto E_s/N_0 , si preferisce spesso utilizzare il rapporto E_b/N_0 tra l'energia ricevuta *per bit* E_b e la psd del rumore N_0 . Per una *M*-PAM, $E_s = \log_2(M) \cdot E_b$ e quindi

$$SER = 2\left(1 - \frac{1}{M}\right)Q\left(\sqrt{\frac{3\log_2(M)}{M^2 - 1}}\sqrt{\frac{2E_b}{N_0}}\right)$$
 (M-PAM) (2.107)

Una variante del ricevitore a filtro adattato che risulta essere ugualmente ottima per i canali AWGN è mostrata nella Fig. 2.26. Per capirne l'origine, riprendiamo in considerazione la (2.101) che descrive l'uscita del filtro adattato per una trasmissione a singolo simbolo:

$$y(t) = r(t) \otimes h_{MF}(t) + n(t) = \int_{-\infty}^{\infty} r(\alpha)h_{MF}(t-\alpha)d\alpha$$



Figura 2.26 Ricevitore a correlazione per segnali in banda-base

$$=\frac{1}{E_g}\int_{-\infty}^{\infty}r(\alpha)g(t_0+\alpha-t)d\alpha$$
(2.108)

Osservando il segnale filtrato all'istante $t = T_0$, si ottiene la variabile di decisione ($E_g = T$)

$$y(t_0) = \frac{1}{T} \int_{-\infty}^{\infty} r(\alpha)g(\alpha)d\alpha = \frac{1}{T} \int_{-\infty}^{\infty} r(t)g(t)dt$$
(2.109)

il cui "processing" è appunto rappresentato nella Fig. 2.26, ed equivale al calcolo della *correlazione* tra il segnale ricevuto r(t) e una forma d'onda locale che corrisponde alla trasmissione di un simbolo (bit) uguale a 1. Il *ricevitore a correlazione* che abbiamo ottenuto è un'alternativa a quello con filtro adattato, e quindi l'andamento del BER è il medesimo.

Si ottengono gli stessi risultati di BER/SER del filtro adattato/ricevitore a correlazione se normalizziamo l'equazione (2.102). Dividendo sia il segnale che il rumore per la deviazione standard $\sigma_n = \sqrt{N_0/2T}$ di n(0) (e lasciando quindi l'SNR invariato), otteniamo

$$y(0) = \sqrt{\frac{2E_s}{N_0}} a[0]g_r(t_0) + n_1(0)$$
(2.110)

dove $n_1(0)$ è una variabile aleatoria Gaussiana a media nulla con varianza *unitaria*. In alternativa, possiamo normalizzare rispetto a $\sqrt{P_r}$ ottenendo

$$y(0) = a[0]g_r(t_0) + N(0)$$
(2.111)

dove adesso N(0) è una variabile aleatoria Gaussiana a media nulla con varianza $\sigma_N^2 = N_0/(2E_s)$. Menzioniamo esplicitamente queste forme alternative poiché in seguito useremo indifferentemente le tre espressioni appena ricavate: la forma non normalizzata (2.102), la forma a varianza unitaria (2.110), la forma ad ampiezza unitaria (2.111).

L'Espressione (2.104) del BER/SER è chiamata *limite del filtro adattato* (Matched Filter Bound), e la sua dipendenza dal rapporto E_s/N_0 (che fondamentalmente è una misura dell'SNR sul simbolo dati) è mostrata in Fig. 2.27. Perché questo valore viene chiamato *limite*? Perché è ciò che si otterrebbe in caso di trasmissione di un *singolo* simbolo di dati. Riprendendo in considerazione (anche) i simboli adiacenti, ciò che in effetti osserviamo all'uscita del filtro adattato è

$$y(t) = \sqrt{P_r} \sum_{n=-\infty}^{+\infty} a[n]g_r(t - nT) + n(t)$$
 (2.112)

che all'istante t_0 vale

$$y(t_0) = \sqrt{P_r} \sum_{n=-\infty}^{+\infty} a[n]g_r(t_0 - nT) + n(t_0)$$



Figura 2.27 Prestazioni di BER del ricevitore a filtro adattato per una M-PAM

$$= \sqrt{P_r} a[0] + \sum_{n \neq 0} a[n]g_r(t_0 - nT) + n(t_0)$$

(2.113)

Oltre al simbolo dati a[0] da rigenerare e al rumore, notiamo la presenza di un ulteriore termine dovuto alla trasmissione di tutti gli altri simboli del flusso dati, che si manifestano come una componente di interferenza sul simbolo a[0]: l'*Interferenza InterSimbolica* (ISI). La presenza di ISI comporta un peggioramento del BER rispetto al valore (2.104) dato dal limite del filtro adattato.

Fortunatamente, l'ISI può essere controllata, e in particolare *annullata*. Se l'impulso $g_r(t)$ è tale per cui

$$g_r(t_0 - nT) \equiv 0 , \ n \neq 0$$
 (2.114)

l'ISI sarà *nulla*, e il BER sarà uguale al limite del filtro adattato. La condizione (2.114) è soddisfatta, per esempio, da un impulso NRZ (simmetrico) filtrato con il corrispondente filtro adattato. La $g_r(t)$ risultante è un impulso triangolare come in Fig. 2.22 (b), e si può notare che i campioni ai multipli del tempo di simbolo sono zero tranne quello all'istante $t = t_0 = 0$.

Anche l'impulso SRFRC non dà luogo a ISI, sebbene a primo impatto possa sembrare diversamente. Per comprendere meglio, osserviamo che la risposta impulsiva del filtro adattato a un SRFRC è a sua volta un impulso SRFRC, poiché g(t) è pari e quindi simmetrico. Riportando la definizione di filtro adattato (2.52) nel dominio della frequenza otteniamo

$$G_{r}(f) = G(f) \cdot H_{MF}(f) = G(f) \cdot G^{*}(f) / E_{g}$$

= $\sqrt{TG_{N}(f)} \cdot (1/T) \sqrt{TG_{N}(f)} / T = G_{N}(f)$ (2.115)

e quindi $g_r(t) = g_N(t)$. Dalla (2.14) e dalla Fig. 2.7 si può facilmente vedere che tutti i campioni dell'impulso di Nyquist $g_N(t)$ all'istante $t_n = nT$ sono *nulli*, tranne l'unico con

n = 0 che vale 1. Alla stessa conclusione si arriva ragionando nel dominio della frequenza. Se consideraimo la (2.114) con $t_0 = 0$, notiamo che la sequenza dei campioni di $g_r(t)$ alla frequenza di simbolo è tale per cui $g_r(nT) = \delta[n]$. Quindi, calcolando la TF di entrambe le sequenze e considerando la relazione di Poisson (2.79)), troviamo la relazione

$$R\sum_{k=-\infty}^{+\infty} G_r \left(f - kR \right) = 1$$
 (2.116)

che è chiamata il *criterio di Nyquist* per l'assenza di ISI. Se sostituiamo G_r con G_N come sopra, si vede facilmente che la (2.116) è soddisfatta grazie alla particolare simmetria rispetto al punto f = R/2 nella regione di roll-off di $G_N(f)$ in Fig. 2.7, per qualsiasi valore di β .

Adesso possiamo comprendere facilmente il motivo per cui viene usato l'impulso SRFRC in molti formati di comunicazione a banda limitata: il ricevitore a filtro adattato per questo tipo di impulso combina la robustezza contro il rumore all'assenza di ISI sul canale a banda limitata.

2.7 Modulazione/Demodulazione a Radiofrequenza e Architettura dei Modem

2.7.1 Modulazione lineare digitale I/Q

I segnali in banda base come quelli visti nel paragrafo precedente, non sono adatti per le comunicazioni via radio (wireless). Infatti i segnali radio possono essere generati e rivelati efficientemente soltanto quando le dimensioni dell'antenna trasmittente/ricevente sono dello stesso ordine di grandezza della *lunghezza d'onda* $\lambda_0 = c/f_0$ (*c* velocità della luce) dell'onda radio. Per fare un esempio, la lunghezza d'onda $\lambda_0 = c/f_0$ (*c* velocità della alla frequenza $f_0=100$ kHz è $\lambda_0 = c/f_0=3$ km, quindi la trasmissione di un segnale a tale frequenza (o comunque aventi componenti a frequenze vicine a tale f_0) risulta essere alquanto problematica. La soluzione si trova usando una frequenza portante (assai più) alta (per esempio, nelle wireless LAN secondo standard Wi-Fi, la frequenza portante è $f_0=2.4$ GHz) e concentrando lo spettro del segnale dati attorno a tale frequenza. Questo significa che dobbiamo *modulare* (cioè variare) l'ampiezza e/o la fase dell'oscillazione portante attraverso il segnale dati in banda base, come abbiamo anticipato nella Sez.2.4 a proposito dei segnali passa-banda e del modulatore I/Q rappresentato in Fig. 2.17.

Quando si vuole realizzare una modulazione digitale, bisogna dunque specificare quale sia la relazione tra le componenti I/Q in banda base del segnale passa-banda e i dati da trasmettere. I vari formati di modulazione digitali sono in ultima analisi delle "appendici" da aggiungere al generico modulatore radio I/Q di Fig. 2.17. Da questo punto di vista, il formato più semplice è la BPSK (Binary Phase Shift-Keying) dove la componente Q $x_Q(t)$ è *nulla*, e la componente I è un segnale binario NRZ antipodale come nella (2.88). L'equivalente in banda base del segnale modulato BPSK risulta quindi

$$x(t) = A \sum_{n = -\infty}^{+\infty} a[n]g(t - nT) + j0$$
 (BPSK) (2.117)

Nel generico intervallo di simbolo $nT \le t < (n+1)T$, la componente I della portante $\cos(2\pi f_0 t)$ viene moltiplicata o per un numero positivo (fase 0), o per un numero negativo



Figura 2.28 Rappresentatione dei segnali BPSK (a) e QPSK (b) nel piano complesso (A = 1)

(fase π), a seconda della polarità dei dati da trasmettere. Quindi, il segnale radio (passabanda) si ottiene *spostando* la *fase* della portante (phase-shift) tra *due* valori (binary) a seconda del dato che vogliamo *codificare* (keying).

2.7.2 Le costellazioni I/Q

La generalizzazione più semplice della BPSK si ottiene usando anche la componente Q che nella BPSK è assente. Per far ciò, si ricavano *due* segnali NRZ antipodali a partire dal flusso di dati binari di partenza, usando nel primo soltanto i bit nelle posizioni pari a[2m], e nel secondo soltanto i bit nelle posizioni dispari a[2m + 1]. Identificando questi due segnali rispettivamente come x_I e x_Q , otteniamo un segnale *Quaternary Phase-Shift Keying* (QPSK):

$$x(t) = A \sum_{m=-\infty}^{+\infty} a[2m]g(t-mT) + jA \sum_{m=-\infty}^{+\infty} a[2m+1]g(t-mT)$$
$$+\infty$$

$$= A \sum_{m=-\infty}^{+\infty} \left(a[2m] + ja[2m+1] \right) g(t-mT) \quad (\text{QPSK})$$
(2.11)

8)

In ogni intervallo di simbolo $mT \le t < (m+1)T$, il segnale x(t) assume stavolta uno dei quattro possibili valori $A(\pm 1 \pm j)$ che giacciono su di una circonferenza nel piano complesso con raggio pari a $\sqrt{2}A$, come mostrato in Fig. 2.28 (b). I quattro punti hanno modulo identico ma *fase* diversa; per tale ragione il segnale viene detto QPSK: i due bit di informazione identificano di fatto una tra le possibili 4 fasi del segnale. Un segnale BPSK al contrario è caratterizzato da due soli punti, entrambi sull'asse reale come in Fig. 2.28 (a).

E' evidente che a parità di velocità di simbolo R = 1/T (e quindi a parità di banda), i due segnali supportano diverse velocità di bit. La BPSK ha bit-rate $R_b = R$, mentre la QPSK ha $R_b = 2R$, come per qualunque segnale a quattro livelli.

È possibile generalizzare ulteriormente e ottenere formati ancora più efficienti della QPSK utilizzando il modulatore generalizzato di Fig. 2.29, nel quale il flusso di dati binario a[n] a frequenza di bit R_b è segmentato in *parole* di lunghezza pari a N_b ciascuna, con frequenza di presentazione di tali parole $R_w = R_b/N_b$. Ogni "parola" identifica un *simbolo* specifico sul piano complesso scelto in un insieme (*alfabeto*) di $M = 2^{N_b}$ elementi - la cosiddetta *costellazione* della modulazione digitale. La frequenza di presentazione delle





 $S_Q(t)$



s_Q[m]

parole R_w diventa dunque la frequenza di simbolo $R = 1/T = R_b/N_b$. Per la QPSK abbiamo $N_b = 2$ e M = 4, ma sono utilizzate anche costellazioni M-PSK (M uguale a 8,16 o 32) caratterizzate da M punti uniformemente distribuiti su una circonferenza sul piano complesso - la Fig. 2.30 (a) rappresenta una costellazione 8-PSK, dove per ogni simbolo viene indicato il particolare valore della parola a 3 bit che ne che comporta la trasmissione.

La corrispondenza tra le parole di bit e i simboli della costellazione indicata nella Fig. 2.30 (a) per la 8-PSK è la *mappatura* che viene implementata nel corrispondente modulo di Fig. 2.29. Il mappatore genera dunque all'istante di tempo mT il simbolo complesso $s[m] = s_I[m] + \jmath s_Q[m]$ selezionato tra gli M della costellazione sulla base del particolare valore della parola di N_b bit che deve essere inviata.

Anche i segnali modulati devono essere limitati in banda (centrata sulla frequenza portante f_0) per rispettare certi limiti stabiliti dagli enti regolatori (ad esempio lo *European Telecommunications Standards Institute* www.etsi.org a livello europeo o la *International Telecommunication Union* www.itu.int a livello mondiale). Questa caratteristica si ottiene scegliendo un particolare impulso di simbolo g(t) limitato in banda da utilizzare nella costruzione delle componenti I/Q del segnale equivalente in banda base prima della modulazione radio. Per questo motivo, nella Fig. 2.29 è stato inserito un filtro *sagomatore* (interpolatore) con impulso g(t) a banda limitata, solitamente del tipo SRFRC.

La forma generale del segnale all'uscita del modulatore è infine

$$s(t) = A \sum_{m=-\infty}^{+\infty} s[m]g(t - mT)$$
 (2.119)



Figura 2.31 Demodulatore dati digitale I/Q

dove s[m] è l'*m*-esimo simbolo complesso appartenente ad una costellazione specificata, e g(t) è l'impulso di simbolo. Come esempio di costellazione, si veda la 64-QAM di Fig. 2.30 (b), che è generata tramite la composizione di due segnali I/Q ciascuno su 8 livelli con alfabeto $\{-7, -5, -3, -1, +1, +3, +5, +7\}$ su ogni componente, e che "trasporta" $N_b=6$ bit/simbolo.

2.7.3 Demodulazione I/Q di segnali radio

Il criterio con cui effettuare la demodulazione dati di un segnale radio con modulazione I/Q secondo una certa costellazione assegnata è immediata immaginando di combinare le operazioni di demodulazione radio I/Q come in Fig. 2.18 e di filtraggio adattato con campionamento e decisione come in Fig. 2.25, seguiti da un'appendice che inverte la mappatura di Fig. 2.29. Quel che si ottiene è il demodulatore dati digitale di Fig. 2.31, nel quale i filtri passa-basso del demodulatore di Fig. 2.18 non compare più perché tale funzione viene (già) implicitamente eseguita dai filtri adattati. All'uscita dei campionatori di segnale agli istanti di simbolo, otteniamo due valori "soft" che rappresentano una versione disturbata da rumore delle componenti I/Q dei simboli trasmessi s[m]. Se raccogliamo alcuni campioni e li riportiamo come punti sul piano complesso, otteniamo il *diagramma I/Q* di Fig. 2.32, riferita alla costellazione 16-QAM.

Il "decisore complesso" in Fig. 2.18 è l'analogo per una costellazione I/Q del decisore a soglia per i segnali in banda base. Il piano complesso viene suddiviso in diverse regioni, chiamate *zone di decisione* che di fatto implementano una mappa inversa (costruita *una tantum* secondo un criterio ottimale di decisione) per ricostruire i simboli di costellazione ricevuti: quando $y(mT) = y_I(mT) + jy_Q(mT)$ ricade in una certa regione, il decisore indica in uscita il simbolo di costellazione corrispondente alla zona. I simboli rigenerati $\hat{s}[m]$ sono poi de-mappati per tornare alla corrispondente parola originale di N_b bit (nell'esempio $N_b = 4$). La funzione risultante dalla cascata del decisore complesso e del demappatore è rappresentata nella Fig. 2.33, dove in ogni zona compare direttamente la parola di N_b bit rigenerata: le etichette delle zone di decisione sono già i bit demappati. Quando la componente di rumore è elevata, y(mT) può oltrepassare il confine della regione di decisione a cui appartiene s[m], per cui viene decodificato un simbolo sbagliato: $\hat{s}[m] \neq s[m]$.

Esempio 2.10

Calcoliamo la SER/BER del ricevitore a filtro adattato per un segnale digitale QPSK con sagomatura SRFRC in AWGN. L'equivalente complesso in banda base (I/Q) del segnale





ricevuto è

$$r(t) = \sqrt{\frac{2P_r}{A_2}} \sum_{n=-\infty}^{+\infty} a[n]g(t-nT) + w(t)$$

dove g(t) è appunto l'impulso SRFRC normalizzato in modo da avere energia $E_g = T$, a[n] è l'*n*-esimo simbolo di costellazione, $a[n] = a_I[n] + ja_q[n]$ è il simbolo QPSK che quindi prende uno tra i 4 possibili valori $\{\pm 1 \pm j\}$ (Fig. 2.28), $A_2 = E\{|a[n]^2|\} = 2$ è la potenza dei simboli di costellazione, P_r è la potenza totale di segnale ricevuta, e w(t) è l'equivalente in banda base del rumore AWGN, $w(t) = w_I(t) + jw_Q(t)$, $S_I(f) = S_Q(f) = N_0$ come nel paragrafo 2.4.2.

Dopo filtraggio adattato (con risposta in frequenza $H(f) = G^*(f)/T$) e campionamento agli istanti ideali nT, e considerando che l'impulso $g_r(t)$ all'uscita del filtro adattato è privo di ISI otteniamo la variabile di decisione complessa

$$z[n] = \sqrt{\frac{2P_r}{A_2}}a[n] + w[n]$$
(2.121)

(2.120)

dove $w[n] = w_I[n] + jw_Q[n]$ è una variabile aleatoria complessa Gaussiana le cui componenti I/Q sono indipendenti, a media nulla ed entrambe con varianza

$$\sigma_w^2 = \int N_0 |H(f)|^2 df = N_0 \int |G^*(f)/T|^2 df = \frac{N_0}{T} \int G_N(f) df = \frac{N_0}{T} \quad (2.122)$$

Da questa relazione, è semplice ricavare la BER del collegamento: entrambe le componenti I/Q del segnale mappano un bit di sorgente diverso. Di fatto, la relazione complessa (2.121) è equivalente a due canali reali indipendenti (perchè sia i simboli che le componenti di rumore sui due rami sono indipendenti):

$$\begin{cases} z_{I}[n] = \sqrt{\frac{2P_{r}}{A_{2}}} a_{I}[n] + w_{I}[n] \\ z_{Q}[n] = \sqrt{\frac{2P_{r}}{A_{2}}} a_{Q}[n] + w_{Q}[n] \end{cases}$$
(2.123)





La BER di entrambi i canali è la stessa, ed è quindi uguale alla BER complessiva:

$$BER = BER_I = BER_Q = Q\left(\sqrt{\frac{2P_r/A_2}{N_0/T}}\right) = Q\left(\sqrt{\frac{E_s}{N_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$
(2.124)

avendo tenuto conto che $E_s = P_sT$, $A_2 = 2$ ed $E_s = 2E_b$ perché ogni simbolo QPSK mappa 2 bit. Abbiamo ritrovato il familiare limite del filtro adattato (2.104) che vale quindi anche per la QPSK. Se immaginiamo di "degradare" la QPSK a BPSK ponendo $a_Q[n] \equiv 0$, si trova facilmente il medesimo risultato.

Il calcolo del SER per una costellazione complessa generica è semplice ma lungo, pertanto sorvoliamo su questa questione. Per una M-QAM, un limite superiore piuttosto accurato, specialmente quando il SER è basso, è il seguente:

$$SER \le 4\left(1 - \frac{1}{\sqrt{M}}\right)Q\left(\sqrt{\frac{3}{2(M-1)}}\sqrt{\frac{2E_s}{N_0}}\right)$$
 (M-QAM) (2.125)

2.7.4 Architettura dei Modem basati su Digital Signal Processing (DSP)

Come vengono implementate le tecniche di modulazione e demodulazione viste finora? Il modem (MOdulatore-DEModulatore) radio di un apparato di comunicazione, sia che si tratti di uno smartphone economico o di una complessa Stazione Radio Base che serve una intera cella di una rete cellulare, comprende una sezione di banda base nella quale i segnali vengono il più possibile elaborati nel dominio digitale (modulazione, demodulazione, filtraggio, ecc.), lasciando i componenti analogici soltanto nella parte radio.

Prendendo in considerazione la parte ricevente del modem (la più complicata da implementare) si hanno diverse alternative per quanto riguarda la collocazione del modulo ADC che stabilisce il confine tra la sezione analogica e quella digitale. Il primo approccio, mostrato in Fig. 2.34, è principalmente adottato nei terminali mobili a basso



Figura 2.34 Demodulazione di un segnale I/Q con campionamento in banda-base

costo/dimensione/consumo di potenza. Il segnale viene convertito in banda-base usando un demodulatore I/Q analogico convenzionale seguito da una coppia di convertitori I/Q ADC in banda base con una frequenza di campionamento f_s , e la demodulazione dati viene implementata nel dominio digitale in banda base. Nelle implementazioni più economiche e a minimo consumo, manca il primo stadio di conversione alla *frequenza intermedia* (IF, intermediate frequency), e il segnale RF viene convertito direttamente in banda base (I/Q) (architettura *zero-IF*).

I punti critici di questa architettura sono i possibili sbilanciamenti di ampiezza dei due rami analogici I e Q, come anche l'imprecisione di quadratura tra le oscillaizoni I/Q analogiche usate per la conversione da IF in banda base che possono non risultare esattamente sfasate di $\pi/2$. Una diversa architettura che supera i problemi di sbilanciamento della precedente ma che comporta minori vincoli di consumi o di costi è quella che prevede il campionamento IF mostrata in Fig. 2.35 (a), dove il modulo ADC è ulteriormente spostato verso l'antenna.

Quest'architettura è quella caratteristica delle cosiddette *software-defined radio* (SDR) dove tutta l'elaborazione dei segnali, a meno della conversione iniziale da RF a IF che viene eseguita nel dominio analogico, viene svolta su segnali digitalizzati, e può anche essere svolta da un processore general-purpose con un opportuno software o da un componente riprogrammabile FPGA con opportuno firmware di elaborazione del segnale. Con la SDR, cambiare il formato di comunicazione supportato dal modem equivale semplicemente a cambiare il SW/FW dedicato senz'alcuna modifica all'HW della piattaforma utilizzata, evitando quindi di dover rimpiazzare una card, o l'intero modem come negli apparati radio convenzionali.

Tornando alla Fig. 2.35 (a), il convertitore ADC opera direttamente sul segnale IF, e l'uscita digitale è ancora in banda passante su di una frequenza intermedia *digitale* $f_{DIF} = f_{IF} \pm k \cdot f_{sa}$ (con k intero). La parte iniziale analogica viene semplificata, e il compito della conversione in banda base viene svolto nella sezione digitale (senza problemi di sbilanciamenti di ampiezza o imprecisioni di quadratura). Il problema principale dell'approccio del campionamento IF riguarda i requisiti (più) stringenti per l'ADC che adesso deve trattare segnali IF (invece dei segnali in banda base come in 2.34) e quindi a banda più larga. In particolare, i tempi di salita e discesa del convertitore, cioè il tempo per "aprire" e "chiudere" la porta logica del campionatore, devono essere proporzionati alla frequenza IF analogica, e risultano essere più brevi di quelli dei convertitori che operano in banda base. Di conseguenza, il campionatore IF ADC è più costoso e consuma più energia dei due convertitori in banda base di Fig. 2.34.

54 L'A-B-C DI SEGNALI SISTEMI E COMUNICAZIONI



Figura 2.35 Demodulazione di un segnale I/Q con campionamento IF (a) o RF (b)

La versione "ultimativa" dell'approccio SDR è quella mostrata in Fig. 2.35 (b), ed è chiamata ricevitore a *conversione RF diretta*. Il modulo ADC opera direttamente alla frequenza radio f_0 , e le componenti analogiche sono ridotte al minimo - in particolare non serve alcuna conversione IF. Ovviamente, i problemi legati ai costi e ai consumi di potenza dell'ADC, già menzionati a proposito dell'architettura a campionamento IF, in questo caso sono ancora più marcati. Questo tipo di architettura è usata per applicazioni speciali o militari che richiedono particolari caratteristiche come acquisizione e rivelazione rapida del segnale e totale ri-configurabilità dei terminali.

La nozione di SDR che abbiamo presentato relativamente alla sezione di *ricezione* del modem, vale naturalmente anche per la sezione di *trasmissione*, per realizzare la quale si devono "invertire" gli schemi generali dei trasmettitori appena presentati, e alla quale si applicano le medesime considerazioni riguardo i pro e i contro. Ad esempio, la Fig. 2.36 rappresenta lo schema della sezione di trasmissione con interpolazione (digital-to-analog conversion) I/Q in banda base "gemello" del ricevitore della Fig. 2.34.

2.8 Multiplazione e Accesso Multiplo

Nel resto di questi Appunti incontreremo le varie tecniche che sono utilizzate nelle comunicazioni digitali per far coesistere all'interno della rete diversi flussi (segnali) digitali provenienti da o destinati a diversi utenti. In questo paragrafo esamineremo brevemente le tecniche base, e svilupperemo un quadro generale che le comprenda tutte quante.

Immaginiamo dunque di avere a che fare con N flussi dati indipendenti $s_i[n]$. Questi rappresentano il collegamento Internet di diversi utenti o possono anche rappresentare diversi



Figura 2.36 Trasmettitore I/Q con DAC in bada base

flussi indipendenti (ad esempio, audio, video e dati) generati da uno stesso utente. Ogni segnale identifica un canale i cui dati devono essere consegnati al rispettivo destinatario indipendentemente dalla presenza degli altri canali concomitanti - i vari segnali di cui sopra vengono talvolta indicati con il nome di tributari. Dobbiamo dunque esaminare le tecnologie mediante le quali è possibile condividere un unico mezzo fisico (ad esempio, una certa banda radio o un'unica fibra ottica) senza che i canali perdano la loro individualità o vengano disturbati da mutua interferenza.

2.8.1 Multiplazione = Accesso Multiplo ?

Il caso più semplice è rappresentato in Fig. 2.37: i segnali tributari vengono generati dallo stesso apparato (ad esempio, i flussi audio/video/dati di uno smartphone), oppure vengono instradati in forma digitale attraverso la rete terrestre verso un unico apparato che "costruisce" il segnale fisico. L'apparato è il MUX (multiplexer, ad esempio la stazione radio base di una rete cellulare) che costruisce l'*unico* segnale inviato sul mezzo fisico $s_{MUX}(t)$ come aggregato dei flussi digitali $s_i[n]$ di tutti i tributari. Questa operazione è appunto il multiplexing o multiplazione ed è tipico del forward link (dalla rete ai terminali utente) delle reti cellulari.

Prendiamo adesso in considerazione il return link: in questo caso abbiamo un insieme di terminali utente che necessariamente producono N segnali fisici diversi $s_i(t)$ ciascuno contentone il rispettivo flusso individuale $s_i[n]$. producono i flussi individuali , the situation is different. I vari segnali, costruiti dai vari terminali utente in generale in modo non coordinato (e quindi anche non sincronizzato), vengono inviati sul mezzo fisico (tipicamente una certa banda radio) da siti differenti e vengono raccolti dall'(unica) antenna ricevente della stazione radio base per fornire il segnale ricevuto complessivo $s_{MA}(t)$. Il mezzo fisico è acceduto separatamente e direttamente da ciascun terminale utente in siti diversi, come già detto, e in maniera non coordinata: abbiamo il cosiddetto accesso multiplo come raffigurato in in Fig. 2.38.

Queste due modalità (multiplazione e accesso multiplo) spesso vengono confuse e/o non correttamente identificate. Non si tratto solo di una questione nominalista o accademica: la differenza tra i due modi può richiedere anche differenti soluzioni implementative.

2.8.2 Multiplazione con Segnali Ortogonali

2.8.2.1 Time-Division Multiplexing Cominciamo ad esaminare il caso di N segnali tributari in formato NRZ binario, tutti caratterizzati dallo stesso tempo di simbolo T.



Figura 2.37 Multiplexing di flussi tributari





Poiché tutti i tributari "convergono" in un unico apparato, possiamo supporre che tutti i flussi dati condividono lo stesso riferimento di clock (sono sincronizzati), cioè i fronti dell'impulso NRZ di tutti i flussi dati sono perfettamente allineati, come rappresentato in Fig. 2.39 (a). Vogliamo trovare una maniera di creare un singolo *multiplex*, cioè segnale multiplato $s_{MUX}(t)$ contenente tutti i flussi dati e in modo che ogni tributario possa essere successivamente demultiplato senz'alcuna interferenza da parte degli altri.

È chiaro che non possiamo *sic et simpliciter* sommare i vari segnali nella loro forma originale - dobbiamo al contrario assegnare a ciascun canale una *firma (signature)* o in generale una qualche proprietà distintiva, che possa caratterizzarlo e renderlo separabile (demultiplabile) dagli altri prima di effettuare la somma. La Figura 2.39 (b) rappresenta un esempio di *firme* $c_i(t)$, i = 0, ..., N - 1. Come si vede, il tempo di segnalazione T viene ripartito in tanti *time slots* quanti canali dobbiamo multiplare, e il codice di multiplazione $c_i(t)$ è semplicemente un impulso (di durata T/N) che "marca" lo slot n. *i*. Il segnale multiplato è dunque

$$s_{MUX}(t) = \sum_{i=0}^{N-1} s_i(t) \cdot c_i(t) = \sum_{i=1}^{N-1} s_i[n]c_i(t)$$
(2.126)

Formalmente, $s_{MUX}(t)$ è ancora un segnale NRZ binario, ma con una velocità di segnalazione N/T pari a N volte la velocità originaria R = 1/T dei segnali tributari. In particolare, nello slot #1, il multiplex ha il valore del dato del tributario 1, nello slot #2 ha il valore del adato del tributario 2, e così via. Il tempo di un singolo simbolo originario è stato *suddiviso* e allocato a turno (*round-robin*) ai vari tributari: abbiamo creato un Time-Division Multiplex (TDM). Questa tecnica è stata introdotta negli anni '60 con l'avvento dele reti telefoniche digitali.



Figura 2.39 Segnali tributari sincronizzati, firme per la multiplazione, segnale TDM

Come possiamo demultiplare i vari canali a partire dal segnale multiplato $s_{MUX}(t)$? Aggiungiamo anche il consueto disturbo di ricezione, e supponiamo di ricevere $r(t) = s_{MUX}(t) + w(t)$, dove w(t) isè AWGN, e cerchiamo di recuperare il canale # m. Se non vi fossero gli altri tributari, ma solo rumore di ricezione, il ricevitore ottimo sarebbe un filtro adattato o il ricevitore a correlazione di Fig. 2.26, che fornisce la variabile di decisione

$$y_m[0] = \frac{1}{T} \int_0^T s_{MUX}(t) c_m(t) + n(T)$$
(2.127)

relativa al simbolo utente $s_m[0]$, dove n(T) è AWGN filtrato campionato al tempo T. Che cosa accade considerando anche la presenza degli altri canali? Dalla (2.127) e dalla (2.126) abbiamo

$$y_m[0] = \frac{1}{T} \int_0^T \sum_{i=0}^{N-1} s_i[0]c_i(t) \cdot c_m(t) + n(T) = \sum_{i=0}^{N-1} s_i[0] \frac{1}{T} \int_0^T c_i(t) \cdot c_m(t) + n(T)$$
(2.128)

D'altra parte, si vede facilmente dalla Fig. 2.39 (b) che le "firme" sono tali per cui

$$\frac{1}{T} \int_0^T c_i(t) \cdot c_m(t) dt = \begin{cases} 1/N & i = m \\ 0 & i \neq m \end{cases} = \frac{1}{N} \delta[i - m]$$
(2.129)

In altre parole, i segnali firma $c_i(t)$ usati per la multiplazione costituiscono un insieme di segnali *ortogonali*. È proprio quest proprietà di *ortogonlità* (2.129) che garantisce l'assenza di interferenza tra i canali multiplati. Il fattore di scala 1/N nella(2.129) non ha effetto

58 L'A-B-C DI SEGNALI SISTEMI E COMUNICAZIONI

perchè può essere compensato semplicemente modificando in proporzione il ricevitore a correlazione.

In pratica, l'implementazione del TDM è in generale leggermente diversa. Il capostipite di tutti i TDM è il multiplexing digitale che è a tutt'oggi implementato nella telefonia digitale T1 (negli USA e Giappone) ed E1 (in Europa). Il multiplexing non viene eseguito bit per bit come nella nostra descrizione, ma *byte per byte*: nella gerarchia E1, viene preso un byte di dati binari (8 bit) da ciascun canale tributario a 64 kbit/s per un tempo totale di 125 μ s e viene ri-clockato e compresso nel tempo in uno slot da 3.90625 μ s, cioè di un fattore 32. Il multiplex E1 da 2.048 Mbit/s comprende infatti 30 canali vocali più 2 canali di servizio per un totale di *N*=32 canali.

La multiplazione ha un costo in termini di *banda*. La banda di ciascun tributario con symbol rate $1/T \in B \approx 1/T$. Il segnale TDM (2.126) è fondamentalmente un segnale digitale binario con formato NRZ il cui impulso.base ha una durata $T_{TDM} = T/N$, e quindi con una velocità di segnalazione $R_{TDM} = 1/T_{TDM} = N/T$ - la banda occupata sarà dunque $B_{TDM} \approx N/T$ cioè N volte maggiore di quella di ciascuno dei segnali tributari. Se è vero che la multiplazione avviene in ambito temporale (il tempo necessario per inviare N bit di N canali diversi è uguale a quello necessario per inviare singolarmente un solo bit dei tributari), é anche vero che lo "spazio" necessario ad allocare gli N segnali viene di fatto reperito nel dominio della frequenza.

2.8.2.2 Frequency-Division Multiplexing Una volta stabilito il principio di ortogonalità (2.129), possiamo cercare di individuare *altri* insiemi di funzioni ortogonali per realizzare forme di multiplexing diverse dal TDM. Ad esempio, possiamo usare insiemi di oscillazioni sinusoidali ortogonali per realizzare il cosiddetto *Frequency Division Multiplexing* (FDM):

$$c_i(t) = \exp(j \, 2\pi i \, t/T)$$
 (2.130)

Verifichiamone l'ortogonalità, tenendo conto della corretta definizione per segnali complessi (equivalenti in banda base) che comprende il coniugio di uno dei due termini:

$$\frac{1}{T} \int_0^T c_i(t) \cdot c_m^*(t) \, dt = \frac{1}{T} \int_0^T \exp(j \, 2\pi t/T \, (i-m)) \, dt$$
$$= \frac{\exp(j \, 2\pi t/T \, (i-m)|_0^T}{j \, 2\pi \, (i-m)} = \begin{cases} 1 & i=m\\ 0 & i\neq m \end{cases} = \delta[i-m]$$
(2.131)

cioè l'insieme è (anche) *ortonormale*. Scrivendo esplicitamente l'espressione del segnale multiplex capiamo facilmente la ragione del nome FDM:

$$s_{MUX}(t) = \sum_{i=0}^{N-1} s_i(t) \cdot c_i(t) = \sum_{i=0}^{N-1} s_i[n]c_i(t) = \sum_{i=0}^{N-1} s_i[n] \exp(j \, 2\pi i \, t/T) \quad , \quad nT \le t < (n+1)T$$
(2.132)

Vediamo che il generico simbolo dell'utente *i*-esimo $s_i[n]$ viene convertito (modulato) attorno alla frequenza $f_i = i/T$. Ogni canale è dunque caratterizzato da una particolare *banda frequenziale* centrata attorno alla propria frequenza portante, anziché da uno slot temporale come nel TDM: ecco perchè questa tecnica è chiamata FDM. I segnali tributari sono totalmente sovrapposti nel tempo ma "occupano" bande frequenziali diverse - e per questo risultano ortogonali.

Il ricevitore/demultiplatore per questo set di segnali segue di nuovo il principio gà visto per il TDM: usiamo un ricevitore a correlazione per il canale #m generico, che è ottimale
rispetto all'AWGN, e nello stesso tempo, data l'ortogonalità, non è soggetto a problemi di interferenza:

$$y_m[0] = \frac{1}{T} \int_0^T s_{MUX}(t) c_m^*(t) dt$$
 (2.133)

Così come l'ortogonalità dei tributari TDM è garantita dalla sincronizzazione nel *tempo* delle firme, per il nostro FDM, l'ortogonalità è garantita dalla particolare *frequenza* delle varie portanti (2.130), che sono tutte distanziate sull'asse delle frequenze di una certa *spaziatura* Δf pari al symbol rate 1/T dei vari tributari. Viceversa, nell'FDM non occorre il perfetto allineamento in *fase* delle portanti, come dimostriamo con il seguente

Esempio 2.11

Supponiamo di voler fare FDM con un *pettine* di frequenze portanti (2.130) *non coerenti*, cioè non sincrone in fase:

$$c_i(t) = \exp[j \left(2\pi i t/T + \theta_i\right)] \tag{2.134}$$

Usando queste portanti per realizzare un FDM otteniamo

$$s_{MUX}(t) = \sum_{i=0}^{N-1} s_i[n] \exp[j(2\pi i t/T + \theta_i)]$$
(2.135)

Per estrarre il segnale #*m* dobbiamo usare in ogni caso una replica *coerente* di $c_m(t)$ (che cioè conosce θ_m) nel ricevitore a correlazione:

$$y_m[0] = \frac{1}{T} \int_0^T r(t) \cdot c_m^*(t) dt = \frac{1}{T} \int_0^T \sum_{i=0}^{N-1} s_i[0] c_i(t) \cdot c_m^*(t) dt + n(T)$$
$$= \sum_{i=0}^{N-1} s_i[0] \frac{1}{T} \int_0^T e^{j \left[(2\pi i \ t/T + \theta_i)\right]} e^{-j \left[(2\pi m \ t/T + \theta_m)\right]} dt + n(T)$$
$$= \sum_{i=0}^{N-1} s_i[0] e^{j \ (\theta_i - \theta_m)} \frac{1}{T} \int_0^T e^{j \left[2\pi (i-m) \ t/T\right]} dt + n(T)$$
(2.136)

L'integrale nella (2.136), includendo anche il coefficiente 1/T, è pari a $\delta[i - m]$, e quindi l'ortogonalità non è vanificata dall'incoerenza di fase.

Nuovamente, la multiplazione necessita di banda. La Figura 2.40 rappresenta la tecnica FDM ortogonale nel dominio della frequenza. Vediamo in particolare che su ogni frequenza portante $f_i = i/T$ è centrato lo spettro con andamento sinc² dei vari segnali tributari NRZ (statisticamente indipendenti) dopo multiplazione (modulazione sulla portante). La densità spettrale di potenza totale del segnale multiplato è pari alla somma delle singole densità di ogni tributario (torneremo più in dettaglio su questa questione nel paragrafo 8.2.3), ed è mostrata in Fig. 2.40 a tratto spesso. Se ci riferiamo, come già fatto per la (2.96), alla banda al primo nullo, vediamo che la banda totale dell'FDM è $B_{FDM} = 1/T + (N-1)(1/T) + 1/T \simeq N/T$ se N è grande - di nuovo proporzionale al numero di tributari.

2.8.2.3 Code-Division Multiplexing La terza modalità di multiplazione che presentiamo è ancora una volta caratterizzata da un diverso set di segnali ortogonali di firma $c_i(t)$, la cui



separabilità non sussiste per divisione (differenza) di tempo come nel TDM o di frequenza come nell'FDM. Le firme ortogonali sono formalmente forme d'onda binarie NRZ note al trasmettitore e al ricevitore chiamate *codici* - la tecnica di multiplazione risultante prende il nome di Code-Division Multiplexing ovvero CDM. Un esempio molto utilizzato di codici per CDM è basato sull'insieme di sequenza binarie di *Walsh-Hadamard* (WH). L'insieme di sequenze WH ha cardinalità $N = 2^W$, W intero, e viene usato per multiplare N segnali: il tempo di simbolo T viene partizionato in N slot di durata $T_s = T/N$ chip di durata $T_s = T/N$, e il codice WH *#i* assume il valore -1 or +1 in ogni slot a seconda del valore della *i*-esima riga della matrice di Hadamard di ordine N, definita ricorsivamente come segue:

$$\mathbf{H}_{2} \stackrel{\triangle}{=} \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix} , \quad \mathbf{H}_{N} = \mathbf{H}_{2} \otimes \mathbf{H}_{N/2} = \begin{bmatrix} \mathbf{H}_{N/2} & \mathbf{H}_{N/2} \\ \mathbf{H}_{N/2} & -\mathbf{H}_{N/2} \end{bmatrix}$$
(2.137)

dove in questo contesto il simbolo \otimes indica l'operazione di *prodotto di Kronecker* tra matrici (vedi anche (10.134)). Ad esempio,

e l'insieme delle forma d'onda (codici) di firma è mostrato nella Fig. 2.41.

L'idea di suddividere il tempo di simbolo in N time slot assomiglia a quella che abbiamo utilizzato per definire il TDM, ma le firme di multiplazione sono nei due casi radicalmente



Figura 2.41 Codici di firma di Walsh-Hadamard con N=8

diverse. Le forme d'onda di Fig. 2.39 sono diverse da 0 solo in un time-slot e hanno la sola funzione di suddividere il tempo (una specie di "funzione-porta"). Le firme del CDM invece sono sempre diverse da 0 (non c'è alcuna suddivisione temporale), ma alternano tra +1 and -1 in modo apparentemente casuale, speci se N è grande. Nel gergo del CDM, i valori binari dei codici si chiamano chip (perchè si utlizzano per "affettare" l'intero simbolo in N chip) e la durata di uno slot si chiama *chip time* e normalmente si indica con il simbolo $T_c = T/N$.

Dalla (2.137), si intuisce che la correlazione (2.129) tra due qualunque codici WH $c_i(t)$ e $c_m(t)$ coincide con il *prodotto scalare* tra la *i*-esima e la *m*-esima riga di **H**_N. E si può facilmente dimostrare dalla (2.137) che due qualunque righe distinte sono (per costruzione...) ortogonali. Per la terza volta, il segnale multiplato assume la forma

$$s_{MUX}(t) = \sum_{i=0}^{N-1} s_i[n] \cdot c_i(t) , \quad nT \le t < (n+1)T$$

Se raccogliamo gli N simboli $s_i[n]$, i = 1, ..., N dei tributari all'istante n in un vettore riga N-dimensionale s[n], il vettore riga $s_{MUX}[n]$ contenente gli N valori (costanti ma non binari) che il segnale CDM $s_{MUX}(t)$ assume negli N time-slots contenuti nel tempo di simbolo T si ottiene dal prodotto riga-matrice

$$\mathbf{s}_{MUX}[n] = \mathbf{s}[n] \mathbf{H}_N \tag{2.139}$$

Questa è per definizione la *trasformata di Walsh-Hadamard* $\mathbf{s}[n]$ e si può implementare efficientemente con un algoritmo di complessita $O(N \log N)$ analoga a quella della Fast Fourier



Figura 2.42 Rappresentazione del CDM

Transform (FFT) e derivante dalla definizione ricorsiva (2.137) di H_N . La trasformata inversa è elementare, dato che le righe della matrice di WH sono ortogonali:

$$\mathbf{H}_{N}^{-1} = \frac{1}{N} \mathbf{H}_{N} \tag{2.140}$$

Nel CDM con i codici di WH, i segnali tributari risultano sovrapposti nel tempo e nella banda di frequenze, ma possono essere separati con il consueto ricevitore a correlazione grazie all'ortogonalità dei codici. Questa proprietà è suggerita dalla Fig. 2.42 (che può essere riformulata anche per TDM ed FDM) ove i tre assi rappresentano le tre "dimensioni" rispetto alle quali è possibile fare multiplazione. La relazione (2.140) di antitrasformata WH può essere immediatamente implementata in un ricevitore per realizzare in modo digitale la demultiplazione equivalente a un banco di ricevitori a correlazione, uno per ciascuno codice.

Relativamente alla banda occupata dal segnale CDM, se consideriamo la (2.139), vediamo che in ogni intervallo di chip (time-slot) il segnale CDM assume un valore casuale che è dato da una combinazione lineare dei simboli tributari $s_i[n]$, i = 0, ..., N - 1. Tali valori non sono binari, ma si dimostra facilmente che sono incorrelati a causa della proprietà (2.140) di \mathbf{H}_N . Quindi, il multiplex CDM è fondamentalmente un segnale NRZ con simboli incorrelati e con un tempo di simbolo che è pari al tempo di chip $T_c = T/N$ dei codici WH. La larghezza di banda è quindi (di nuovo) N volte quella del singolo segnale tributario, $B_{MUX} = N/T$.

La conclusione di questo paragrafo è che le varie tecniche di multiplexing ortogonale che abbiamo preso in considerazione sono equivalenti in termini di occupazione di banda, e non danno luogo ad alcuna interferenza tra i canali quando ogni tributario viene demultiplato con un ricevitore a correlazione. Vi sono differenze implementative che possono consigliare l'utilizzo dell'una piuttosto che dell'altra, ma tutte quante vengono a tutt'oggi ampiamente utilizzate in pratica.

2.8.3 FDMA, TDMA, CDMA, ALOHA

Già nel paragrafo precedente abbiamo delineato la differenza tra multiplazione e accesso multiplo. Dobbiamo in questo paragrafo illustrare la ragione per cui le tecniche utilizzate nel multiplexing non possono essere direttamente utilizzate nell'accesso multiplo, e suggerire le modifiche necessarie.

2.8.3.1 Frequency-Division Multiple Access Storicamente, la prima tecnica di accesso multiplo utilizzata nelle radiotrasmissioni era basata sulla divisione di frequenza (Frequency Division Multiple Access, FDMA) introdotta per la trasmissione di diversi programmi radiofonici modulati in ampiezza su differenti frequenze portanti. Lo FDMA è facile da implementare e può essere applicato a qualsiasi segnale, indipendentemente dalla sua natura (cioé con modulazione analogica o digitale), purché *limitato in banda*. Lo FDMA non richiede alcun particolare requisito di sincronicità nel tempo o coerenza di fase tra i canali: l'unico vincolo è che le frequenze portanti di centro-canale assegnate a ciascun canale siano generate da ogni apparato con precisione e stabilità (cioè precisione mantenuta nel tempo). Supponiamo infatti che $s_i(t)$ sia un segnale QAM con impulsi SRFRC g(t) (2.98), la cui larghezza di banda (a radiofrequenza) è uguale a $(1 + \beta)/T$ (con β il fattore di roll-off). L'equivalente in banda base del segnale FDMA si ottiene dopo che il generico tributario # i viene modulato su di una frequenza portante $f_i = i \cdot (1 + \beta)/T$, per cui

$$s_{FDMA}(t) = \sum_{i=0}^{N-1} s_i(t) e^{j \left[2\pi i (1+\beta)t/T\right]}$$
(2.141)

Il trasmettitore # i, prima di accedere alla banda radio, deve ovviamente essere informato che la frequenza (il canale) che intende utilizzare non è già assegnato ad un altro trasmettitore - una qualche forma di coordinamento per assegnare i canali è comunque richiesta.

In ambito frequenziale, la rappresentazione dell'FDMA è ugualmente semplice richiamando il teorema della modulazione (2.11) della Trasformata di Fourier:

$$S_{FDMA}(f) = \sum_{i=0}^{N-1} S_i \left(f - i \cdot \frac{(1+\beta)}{T} \right)$$
(2.142)

I canali possono essere separati dato che non sono sovrapposti nel dominio della frequenza la spaziatura tra le frequenza portanti è stata proprio scelta in modo che gli spettri dei vari canali non interferiscano. Per essere certi che l'interferenza non si verifichi mai, neanche in caso di instabilità della frequenza portante, implementazione non accurata del filtro di sagomatura dell'impulso SRFRC ecc., la spaziatura tra le portanti viene incrementata rispetto al minimo teorico utilizzato nella (2.142):

$$S_{FDMA}(f) = \sum_{i=0}^{N-1} S_i \left(f - i \left[\frac{(1+\beta)}{T} + B_g \right] \right)$$
(2.143)

dove B_g è la *banda di guardia*. Lo spettro comprensivo di questo fattore è mostrato nella Fig. 2.43, e può essere confrontato con quello della FDM ortogonale della Fig. 2.40.

Possiamo dire che (anche) nell'FDMA (2.143) i vari tributari sono ortogonali? Dati due segnali $x(t) \in y(t)$, il *Teorema di Parseval* dice che

$$\int_{-\infty}^{\infty} x(t) \cdot y^*(t) dt = \int_{-\infty}^{\infty} X(f) \cdot Y^*(f) df$$
(2.144)

Considerando la Fig. 2.43, e identificando con x ed y due qualunque segnali (canali), vediamo che il secondo membro della (2.144) è sicuramente 0 perchè gli spettri dei tributari non si sovrappongono. DI conseguenza, anche il primo membro è uguale a 0:

$$\int_{-\infty}^{\infty} x(t) \cdot y^{*}(t) dt = 0$$
 (2.145)



Figura 2.43 Densità spettrale di potenza dell'FDMA con impulso SRFRC - $\beta = 0.2$, $B_q = 0.1/T$

Questa relazione rappresenta un'estensione del concetto di ortogonalità a tutto l'asse dei tempi, visto che la forma d'onda SRFRC si estende da $-\infty$ a $+\infty$; la conclusione è che i canali FDMA sono ortogonali e possono essere separati con il consueto ricevitore a correlazione. Immaginando di voler estrarre il canale # m, isoliamo per semplicità l'impulso relativo al simbolo con n = 0 e assumiamo anche $B_g = 0$; il segnale ricevuto è

$$r_0(t) = s_m [0] g(t) e^{j [2\pi i (1+\beta)t/T]}$$
(2.146)

dove abbiamo trascurato gli altri canali, sapendo che sono ortogonali e che daranno interferenza *nulla* all'uscita del ricevitore, e dove non abbiamo considerato l'AWGN, sapendo che il ricevitore a correlazione è ottimo nei suoi confronti. Per rivelare s_m [0], il segnale ricevuto deve essere dunque correlato con la forma d'onda $g(t) \exp\{j [2\pi i(1+\beta)t/T]\}$, e la correlazione può essere implementata in diverse maniere, come indicato in Fig. 2.44: prodotto con $\exp\{-j [2\pi i(1+\beta)t/T]\}$, poi con g(t), infine integrazione sullasse dei tempi (a). Le ultime due operazioni equivalgono di fatto alla correlazione con l'impulso g(t); sappiamo già dalle (2.108)-(2.109) che il calcolo della correlazione con l'impulso g(t) è equivalente al filtraggio adattato a g(t) e campionamento all'istante t_0 (per impulsi SRFRC, $t_0=0$). Dunque la parte finale del ricevitore a correlazione può essere anche implementata come in Fig. 2.44 (b). 'architgettura risultante è (più) intuitiva: abbiamo unio stadio di *demodulazione* che riporta il canale #m dalla portante $m(1+\beta)/T$ in banda base, dopodiché il simbolo s_m [0] viene rivelato con filtro adattato.

2.8.3.2 Time-Division Multiple Access L'accesso multiplo a divisione di tempo (Time-Division Multiple Access, TDMA) non è di altrettanto semplice implementazione. Si capisce infatti che la questione della *sincronizzazione temporale* tra utenti (canali) risulta cruciale e non può essere trascurata come per l'FDMA: è necessario una procedura piuttosto complicata di *sincronizzazione di rete* per mantenere i segnali sincronizzati e quindi non sovrapposti (ortogonali) nel tempo. In particolare, la procedura risulta semplificata se il tempo viene suddiviso tra i vari utenti non bit per bit ma *burst* per *burst*, dove il burst dati è di fatto un pacchetto di più bit di durata fissa T_d contribuito da ciascun utente. Senza

MULTIPLAZIONE E ACCESSO MULTIPLO 65



(a)

(b)

Figura 2.44 Ricevitore ottimo per FDMA: Correlazione (a) e demodulatore-filtro adattato (b)



la sincronizzazione di rete, i burst di dati prodotti da diversi utenti arrivano al ricevitore centralizzato (ad esempio, la stazione radio base di una rete cellulare) in tempi diversi, perché i ritardi di propagazione possono essere in generale notevolmente diversi, a seconda delle diverse distanze tra i trasmettitori e il ricevitore. In questa situazione, il rischio è che burst appartenenti a utenti diversi si sovrappongano nel tempo al ricevitore creando interferenza temporale (collisione) e impedendo il demultiplexing.

Nella procedura di sincronizzazione di rete, il ricevitore centralizzato osserva il tempo di arrivo dei burst di dati di ogni terminale, e re-invia comandi opportuni ai trasmettitori in modo che essi possano gradualmente anticipare o ritardare il tempo di trasmissione del burst successivo in modo da altrettanto gradualmente allinearsi allo slot temporale assegnato *in ricezione* all'interno della *trama* (frame) TDMA. A regime, questa procedura di feedback arriva a uno stato stabile e la trama TDMA al ricevitore è sincronizzata. Così come nell'FDMA adottiamo una banda di guardia per assicurarci che gli spettri adiacenti non siano sovrapposti, nel TDMA è pratica comune usare un *tempo di guardia* per assicurarsi che i burst di dati adiacenti non si sovrappongano nel tempo quando arrivano al ricevitore centralizzato con eventuali (piccole) imprecisioni residue di allineamento. In altre parole, il time-slot allocato ad ogni utente è più lungo della durata totale del burst di dati T_d di un tempo di guardia T_g come illustrato nella Fig. 2.45. Il requisito di precisione della procedura di sincronizzazione di rete viene quindi leggermente allentato, ma questo diminuisce l'efficienza complessiva della rete perché una frazione del tempo totale di frame non viene utilizzato per la comunicazione dei dati.

Un esempio storico di accesso multiplo *misto* FDMA/TDMA è stato introdotto negli anno '80 con la rete cellulare 2G nota come GSM, nella quale la banda radio allocata al

66 L'A-B-C DI SEGNALI SISTEMI E COMUNICAZIONI

servizio (ad esempio 890-915 MHz nell'uplink) era partizionata in un certo numero di canali (portanti) FDMA con spaziatura 200 kHz. Ogni canale FDMA portava poi 8 canali TDMA con burst dati di lunghezza 148 simboli e tempo di guardia 8.25 tempi di simbolo.

2.8.3.3 Code-Division Multiple Access L'analogo del CDM in versione accesso multiplo è ovviamente il Code-Division Multiple Access (CDMA). La differenza principale tra multiplexing e multiple access è che in ques'ultimo caso non è realistico assumere che i vari codici utente possano essere perfettamente sincronizzati e resi ortogonali nel momento in cui vengono generati da apparati diversi e inviati al ricevitore centralizzato da punti geograficamente diversi. Per capire il perché, facciamo il confronto tra TDMA e CDMA.

Nel TDMA, la sincronizzazione di rete tollera errori confrontabili con il tempo di guardia, cioè pari in genere a qualche tempo di simbolo. Riprendendo l'esempio del GSM, $T_s \simeq 3.69$ μ s e $T_G \simeq 30.5 \ \mu$ s. Il massimo errore tollerabile può essere fissato a circa metà tempo di guardia, cioè 15 μ s che è appunto la tolleranza richiesta alla sincronizzazione di rete - un dato ragionevole. Cerchiamo ora di capire se e come è possibile sincronizzare *anche* il CDMA per mantenere l'ortogonalità anche nell'accesso multiplo.

Nel CDMA, il segnale ricevuto dal ricevitore centralizzato è (trascurando per semplicità eventuali costanti di ampiezza e fase per concentrarci sul problema di sincronizzazione)

$$s_{CDMA}(t) = \sum_{i=0}^{N-1} d_i (t - \tau_i) \cdot c_i (t - \tau_i)$$
(2.147)

dove chiaramente $d_i(t) \in c_i(t)$ sono rispettivamente il segnale dati e il codice dell'utente *i*-esimo, e τ_i è il ritardo di propagazione dell'*i*-esimo utente. È chiaro che i segnali, in presenza di ritardi distinti, *non* possono essere ortogonali. Semplifichiamo il problema limitandoci a N = 2 utenti e concentrandoci sul simbolo dati per n = 0 di tutti i canali, in modo che la forma d'onda ricevuta sia

$$s_{CDMA}(t) = s_1[0]c_1(t-\tau_1) + s_2[0]c_2(t-\tau_2)$$

Immaginiamo poi di voler estrarre il canale # 1 con un ricevitore a correlazione sincronizzato sul codice $c_1(t - \tau_1)$. All'uscita del ricevitore, oltre al termine utile $s_1[0]$ abbiamo un'interferenza diversa da zero a causa della presenza del canale # 2:

$$r_{2,1}[0] = s_2[0] \frac{1}{T} \int_{\tau_1}^{\tau_1 + T} c_1(t - \tau_1) \cdot c_2^*(t - \tau_2) dt = s_2[0] \frac{1}{T} \int_0^T c_1(t) \cdot c_2^*(t - \Delta \tau) dt$$
$$= s_2[0] R_{c_1 c_2}(\Delta \tau) , \quad \Delta \tau \stackrel{\triangle}{=} \tau_2 - \tau_1 \qquad (2.148)$$

Il termine di interferenza è tanto maggiore quanto maggiore è il termine di *correlazione* incrociata $R_{c_1c_2}(\Delta \tau)$ tra i due codici, e che dipende fortemente dal ritardo relativo $\Delta \tau$. Supponiamo quindi di usare i codici ortogonali di WH, e di voler utilizzare una procedura di sincronizzazione di rete per mantenere i codici sincronizzati, e quindi ortogonali, al *ricevitore*. Che precisione occorre in questa procedura? Usando codici ortogonali, la cross-correlazione nella (2.148) sarebbe trascurabile se il ritardo relativo fosse mantenuto molto più piccolo del tempo di chip $T_c = T/N$, diciamo $\Delta \tau < 0.1 Tc$, in modo da avere la quasi-sincronizzazione e quindi la quasi-ortogonalità dei codici e assicurarsi che $r_{2,1}[0] \simeq 0$.

Quanto è stringente questo vincolo? Per fare il confronto con il TDMA, supponiamo che i tempi di simbolo dati T siano gli stessi nei due casi; questo significa che il tempo di chip CDMA T_c è uguale al tempo di segnalazione (multiutente) T/N del TDMA. In

quest'ultimo, il vincolo di sincronia è come già accennato pari a metà tempo di guardia, cioè qualche *simbolo*. Nel caso del CDMA, il vincolo è un decimo di chip, cioè *due-tre ordini di grandezza più stringente* che nel TDMA. Nelle reti cellulari 3G (UMTS) degli anni 2000, il tempo di chip era pari a 0.26 μs e quindi l'accuratezza della sincronizzazione di rete che sarebbe stata necessaria è pari a 26 *ns*: troppo stringente per potersi ragionevolmente implementare. Nella letteratura tecnica, talvolta il CDMA è anche indicato come A-CDMA (Asynchronous), mentre il CDM viene indicato (impropriamente) come S-CDMA (Synchronous).

Dunque non ha senso utilizzare codici ortogonali per fare accesso multiplo: l'ortogonalità richiede la sincronia, e la sincronia non può essere garantita. I segnali CDMA (asincroni e non ortogonali) usano allora codici con sequenze di chip *pseudo-casuali* o *pseudo-noise* (PN) con lunghezza di ripetizione maggiore di un simbolo dati (tipica del CDM). Le sequenze PN sono codici binari che "sembrano" generati casualmente, ma che in realtà sono prodotti da un algoritmo ben preciso noto al trasmettitore e ricevitore, e hanno alcune proprietà in comune con il rumore, in particolare densità spettrale di potenza pressoché *costante.* Riprendiamo adesso l'espressione del segnale CDMA con N utenti, considerando l'utente # 1 come quello desiderato:

$$s_{MUX}(t) = d_1(t)c_1(t) + \sum_{i=2}^{M} d_i(t - \Delta\tau_i) \cdot c_i(t - \Delta\tau_i) = d_1(t)c_1(t) + I(t) \quad (2.149)$$

dove abbiamo posto per semplicità $\tau_1 = 0$, (e quindi di fatto $\Delta \tau_i \stackrel{\triangle}{=} \tau_i - \tau_1 = \tau_i$, i > 1), e dove M è il numero di utenti CDMA attivi (non necessariamente M = N). La componente I(t) nella (2.149) è la cosiddetta *Multiple Access Interference* (MAI) che si aggiunge in ricezione al consueto AWGN e che, stante la non ortogonalità dei segnali, non può essere ridotta a zero dal ricevitore. La MAI è costituita dalla sovrapposizione di M - 1 segnali statisticamente indipendenti (perchè provenienti da utenti diversi con flussi dati indipendenti), tutti con la stessa potenza ricevuta P_r^{-1} , uguale alla potenza ricevuta dell'utente desiderato. Se il numero di utenti attivi M è abbastanza grande, diciamo maggiore di 10, le statistiche della MAI I(t) possono essere considerate con buona approssimazione Gaussiane in virtù del teorema-limite centrale, ed è possibile quantificare facilmente l'effetto dell'accesso multiplo sulle prestazioni del ricevitore.

Al ricevitore avremo il segnale

$$r(t) = d_1(t)c_1(t) + I(t) + W(t) = d_1(t)c_1(t) + N(t)$$
(2.150)

dove W(t) è il consueto AWGN con densità spettrale di potenza N_0 per componente, e dove N(t) è il disturbo *complessivo* ancora Gaussiano (perchè somma di due processi Gaussiani indipendenti) e a banda larga, cioè praticamente *bianco sulla banda del ricevitore*. In quest'ipotesi, vedremo nel paragrafo 9.5 che la BER all'uscita del ricevitore a correlazione per il canale utile con costellazione BPSK/QPSK è pari a

$$BER = Q\left(\sqrt{\frac{2E_b}{N_0 + I_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0 + (M-1)\frac{P_r}{N/T}}}\right)$$

¹L'ipotesi di uguale potenza ricevuta da ogni utente è realistica perché in generale le reti CDMA implementano un sistema in feedback di *controllo della potenza* (simile nel principio alla sincronizzazione di rete nel TDMA) in cui i terminali utente vengono comandati dalla stazione radio base di alzare o abbassare la propria potenza trasmessa, finché le potenze *ricevute* non risultino bilanciate. In questo modo, la MAI è a sua volta ben bilanciata e non c'è rischio che un utente molto vicino alla stazione base tenda a "prevaricare" un utente (più) lontano (*near-far effect*).

$$=Q\left(\sqrt{\frac{2E_b}{N_0}}\cdot\sqrt{\frac{1}{1+\frac{M-1}{N}\frac{E_b}{N_0}}}\right)$$
(2.151)

dove la quantità I_0 è il livello della *densità spettrale della MAI*, che come detto risulta costante (bianca) all'interno della banda del ricevitore:

$$S_I(f) = (M-1)\frac{P_r}{N/T} \stackrel{\triangle}{=} I_0 \ , \ -1/T \le f \le 1/T$$
 (2.152)

e che ovviamente risulta proporzionale al numero di interferenti M - 1. Il peggioramento delle prestazioni del ricevitore dovuto alla MAI è molto chiaro, ed è tanto maggiore quanto più "affollata" è la rete (M grande). La Fig. 2.46 mostra le prestazioni di BER nel caso di N = 128 al variare del numero di utenti attivi M (costellazione BPSK), (a) in funzione del rapporto E_b/N_0 del singolo utente desiderato, oppure (b) in funzione del numero di utenti attivi M per alcuni valori di E_b/N_0 (b). Tale effetto di degradazione non è presente nel FDMA o TDMA che in qualche modo preservano l'ortogonalità. D'altra parte, osserviamo che il CDMA è di fatto la solo tecnica di accesso multiplo tra quelle esaminate che consente un metodo di accesso accesso puramente *casuale* senz'alcuna necessità di coordinamento tra gli utenti; ciascun terminale può avviare la propria trasmissione utilizzando un codice pseudocasuale senza nessun'assegnazione di time-slot o di canale frequenziale da parte di un'entità centrale. Da questo punto di vista, il CDMA assomiglia molto alle tecniche *statistiche* ad accesso casuale per trasmissioni a pacchetto come ALOHA/Slotted ALOHA.

2.8.4 Cenno all'Accesso Multiplo con Criteri Statistici

Le tecniche di accesso multiplo che abbiamo appena esaminato risultano particolarmente adatte al caso in cui gli utenti sono attivi (cioè producono effettivamente dati digitali) per tempi di connessione molto più lunghi dei tempi di simbolo, burst, o anche frame - si chiamano infatti *orientati alla connessione*. In molti casi però si ha a che fare con trasmissioni intermittenti, nelle quali cioè la sorgente digitale è per sua natura più o meno discontinua o addirittura sporadica (ad esempio nelle applicazioni *Internet of Things*). I dati si presentano sotto forma di burst relativamente corti, chiamati *pacchetti*, intervallati da (lunghi) periodi di inattività, e si desidera inoltrare il pacchetto dati non appena si produce, con accesso al mezzo *casuale*, cioè *non coordinato*, senz'alcuna suddivisione preordinata di tempo/frequenza/codice - è possibile allora usare criteri di accesso multiplo *orientati al pacchetto*.

Immaginiamo dunque di avere la consueta popolazione di N utenti attivi che generano pacchetti dati di durata (per semplicità) fissa T_p che vengono generati ad istanti casuali di tempo $t_0, t_1, ..., t_k, ...$, con una cadenza *media* fissa e pari per ogni utente a μ pacchetti/s (Fig. 2.47). Se il pacchetto è costituito da N_p bit, in modo che $T_p = N_pT$, il bit rate medio dell'utente è $R_b = \mu N_p$. In molti esempi di sorgente digitale a pacchetto, il *tempo di interarrivo* tra un pacchetto e l'altro, cioè la quantità $\Delta_k = t_k - t_{k-1}$ è modellabile come una variabile aleatoria distribuita *esponenzialmente*:

$$f_{\Delta_k}(\delta) = \mu \, \exp(-\mu\delta) \quad \delta \ge 0 \tag{2.153}$$

con $E{\Delta_k} = 1/\mu$ in modo che il numero medio di pacchetti per secondo risulti effettivamente pari a μ . Con questa ipotesi, gli istanti di nascita dei pacchetti dati $t_k = t_{k-1} + \Delta_k$ hanno statistiche *di Poisson*. Dal punto di vista del ricevitore centralizzato, questo aggregato di traffico prodotto dagli N utenti è equivalente a un traffico *collettivo* avente una cadenza



Figura 2.46 BER del ricevitore a correlazione con CDMA - (a) in funzione di E_b/N_0 ; (b) in funzione del numero di utenti M.



Figura 2.47 Traffico dati di una sorgente a pacchetto

media pacchetti al secondo pari a $N\mu$, e quindi con un tempo di interarrivo globale Δ_g a sua volta distribuito esponenzialmente, $E{\Delta_g} = 1/(N\mu)$.

Nel protocollo ALOHA (il capostipite di tutte le multiplazioni statistiche), ogni utente accede il mezzo e invia il proprio pacchetto sul "canale comune", cioè il mezzo acceduto da tutti gli utenti, non appena questo viene generato. In questo modo si possono avere interferenze o collisioni, cioè è possibile che al ricevitore centralizzato uno o più pacchetti si sovrappongano e non possano essere ricevuti correttamente. Questi eventi di collisione possono essere gestiti secondo due diverse modalit à: se è disponibile un canale di ritorno, il ricevitore puó segnalare la corretta ricezione di ogni pacchetto al trasmettitore con un messaggio di acknowledgement (ACK). In caso di collisione, il trasmettitore evidentemente non riceve il messaggio di ACK, e quinjdi, dopo aver atteso un tempo casuale (con una legge che deve essere specificata quando si implementa il protocollo, ma che qui non interessa) e ritrasmette il medesimo pacchetto. E' chiaro che tale meccanismo di ripetizione del pacchetto impegna le risorse del canale di comunicazione e quindi ne diminuisce il bit-rate utile, ma evita perdite di pacchetto. L'alternativa è quella di non implementare alcun meccanismo di richiesta di ripetizione (vuoi perché non è disponibile il canale di ritorno, vuoi perché il ritardo connesso alla ritrasmissione non è accettabile, come in applicazioni real-time) e quindi accettare il fatto che alcuni pacchetti inviati dal trasmettitore vengono perduti nella comunicazione e non giungono al ricevitore, causando cancellazione (erasure) di parte del messaggio originario.

Evidentemente, se il parametro μ è troppo alto, le collisioni diventano (molto) frequenti e il protocollo perde di efficienza. Dobbiamo dunque valutare l'*efficienza* (throughput) ρ_{ALOHA} del protocollo, intesa come rapporto tra il numero di pacchetti complessivamente ricevuti con ACK dal ricevitore centralizzato, e il numero di pacchetti complessivamente generati dagli utenti - ovviamente l'efficienza sarà inferiore a 1. Per valutare questo parametro dobbiamo calcolare la probabilità P_{NC} che un pacchetto *non* subisca collisione. Dato un certo istante t_k di trasmissione di un pacchetto di un dato utente, questo non subisce collisione se nell'intervallo $[t_k - T_p; t_k + T_p]$ non si verifica alcuna altra trasmissione da parte degli altri utenti (Fig. 2.48 (a)). Questa probabilità *non* dipende da t_k ma solo dall'ampiezza dell'intervallo libero da pacchetti $2T_p$, ed è pari alla probabilità che il tempo di interarrivo tra due pacchetti nel traffico globale sia maggiore di $2T_p$:

$$P_{NC} = \Pr\{\Delta_g \ge 2T_p\} = \int_{2T_p}^{\infty} (N\mu) \, \exp(-(N\mu)\delta) \, d\delta = \exp(-2N\mu T_p) \qquad (2.154)$$

Per trovare l'efficienza di rete ρ_{ALOHA} , dobbiamo considerare il numero di pacchetti complessivamente generati nella rete in un intervallo di tempo di riferimento T_p , e che è pari a $G \stackrel{\triangle}{=} N \mu T_p$ e moltiplicarlo per la probabilità (2.154) che il pacchetto vada a buon fine, ottenendo la relazione

$$\rho_{ALOHA} = G \exp(-2G) \tag{2.155}$$

MULTIPLAZIONE E ACCESSO MULTIPLO 71



Figura 2.48 Collisioni di pacchetti nei protocolli ALOHA (a) e Slotted-ALOHA (b)



che è rappresentata nella Fig. 2.49 al variare di G. Quando il carico della rete G è piccolo, non vi sono quasi collisioni e l'efficienza è sostanzialmente pari a G stesso. Quando però il traffico aumenta, aumentano anche le collisioni finché per alto G pressoché ogni pacchetto in rete è colpito da collisioni e l'efficienza cala a 0. Dalla figura si vede che l'efficienza massima vale $1/(2e) \simeq 0.18$ e si ha quando G = 0.5 (valori che si ottengono facilmente uguagliando a 0 la derivata di (2.155) rispetto a G).

Una modifica dell'ALOHA si ottiene imponendo una *sincronizzazione di rete* analoga a quella del TDMA per cui i pacchetti, quando vengono spediti dai vari utenti, sono allineati ad istanti di inizio pacchetto periodici mT_p , m = 0, 1, ... condivisi da tutti terminali della rete - i pacchetti vengono cioè confinati in *slot* temporali fissi e condivisi, e il protocollo si chiama *slotted* ALOHA (S-ALOHA). Per meglio dire, se un pacchetto viene generato dalla sorgente digitale a un certo istante t_k , viene inviato sul canale non immediatamente, ma al primo istante slottizzato successivo $m_kT_p \ge t_k$ per m_k intero, cioè nello slot # $m_k = \lceil t_k/T_p \rceil$. L'efficienza dello S-ALOHA è migliore rispetto all'ALOHA puro, perché le collisioni tra pacchetti sono stavolta o "complete" o assenti, come mostrato nella Fig. 2.48 (b) (cioè non vi sono collisioni parziali come nell'ALOHA puro), e quindi il numero di pacchetti con collisioni cala sostanzialmente alla metà.

Per calcolare l'efficienza dello S-ALOHA, dobbiamo di nuovo ricavare la probabilità di non avere collisioni che, tenendo conto della slottizzazione, è stavolta la probabilità che il tempo di interarrivo del flusso totale sia maggiore della durata del pacchetto:

$$P_{NC} = \Pr\{\Delta_g \ge T_p\} = \exp(-N\mu T_p) \tag{2.156}$$

per cui l'efficienza è

$$\rho_{S-ALOHA} = G \exp(-G) \tag{2.157}$$

con un valore massimo pari a $1/e \simeq 0.37$ quando G = 1 (Fig. 2.49). G = 1 significa che la rete è complessivamente "piena" di pacchetti in quanto si ha in media esattamente un pacchetto nel tempo di un pacchetto. Dei pacchetti nella rete però, solo il 37% non sono colpiti da collisione.

2.8.5 Efficienza del Multiplexing/Multiple Access Deterministico

Adesso che con il concetto di accesso casuale abbiamo introdotto la misura dell'efficienza di rete, possiamo tornare sui protocolli di accesso multiplo cosiddetti *deterministici* (F/T/CDMA) e valutarne parimenti l'efficienza per confrontarli tra di loro e con gli ALOHA. Stavolta valuteremo l'efficienza in senso spettrale, cioè come rapporto tra il bit-rate totale smaltito dalla rete e la banda occupata con segnalazione binaria.

Se ci concentriamo sul multiplexing, è chiaro che l'efficienza di F/T/CDMA è in ogni caso *unitaria* e le tre tecniche sono equivalentiPer esempio, nel TDM il bit-rate totale è $R_{MUX} = NR_b$ dove R_b è quello di ogni tributario. La banda occupata totè poi $B_{MUX} = NB$, dove B è quella di ogni tributario. Dunque, $\rho_{TDM} = R_b/B$, e, nel caso di segnalazione binaria, $\rho_{TDM} = 1$. Analogamente, si trova che

$$\rho_{TDM} = \rho_{FDM} = \rho_{CDM} = 1 \tag{2.158}$$

Riassumendo: la multiplazione ortogonale ha efficienza unitaria.

L'analisi dell'accesso multiplo è relativamente semplice per FDMA e TDMA che sono ancora pseudo-ortogonali; meno per CDMA che non è ortogonale e che sta al confine, com vedremo, tra accesso casuale e deterministico. Per quanto riguarda FDMM, rispetto FDM c'è soltanto una piccola perdita di efficienza dovuta alla *banda di guardia*:

$$\rho_{FDMA} = \frac{NR_b}{N[(1+\beta)R_b + B_G]} = \frac{1}{1+\beta + B_G/R_b}$$
(2.159)

La perdita è normalmente meno del 10% e rappresenta il costo, come già abbiamo discusso, dell'accesso distribuito - siamo comunque molto vicini all'unità. Stesso ragionamento per il TDMA che paga la presenza del *tempo di guardia*:

$$\rho_{TDMA} = \frac{T_d}{T_d + T_G} = \frac{1}{1 + T_G / (M_B T_b)}$$
(2.160)

Per burst sufficientemente lunghi la perdita di efficienza è minimale.

La questione del CDMA è più complicata a causa della MAI (2.149). D'altro canto, tra i tre protocolli cosiddetti deterministici, CDMA è l'unico totalmente privo di coordinamento e può quindi essere assimilato a un accesso casuale, come gli ALOHA del paragrafo precedente. La sua efficienza piuttosto che con il (quasi) 1 dei protocolli deterministici deve essere confrontata con lo 0.18 di un ALOHA puro, tra l'altro ottenibile in condizioni ottimali che non è detto vengano presentate dalla rete. Anche lo Slotted-ALOHA non è totalmente non coordinato perchè richiede la sincronizzazione di rete come il TDMA.

Il calcolo dell'efficienza del CDMA è un po' più complicato, perché il numero di utenti M che possono essere allocati in una data larghezza di banda $1/T_c = N/T$ non è determinato in modo univoco, come possiamo facilmente vedere considerando la Fig. 2.46 (b). Il numero di canali M è infatti determinato solo specificando un requisito di BER o, in altre parole, specificando una certa *Quality of Service* (QoS). Prendiamo l'esempio N = 128 di Fig. 2.46 con $E_b/N_0 = 11.3$ dB che porta a una $BER = 10^{-7}$ (quindi piuttosto buona) in assenza di MAI. Se fissiamo una QoS nella forma di una certa BER = 10^{-2} , troviamo che il numero massimo di canali attivi è M = 38. L'efficienza è quindi $\rho_{CDMA_1} = 38/128 \simeq 0.3$. Già questo valore supera ALOHA, ma una QoS di questo tipo, sicuramente adatta alla trasmissione di voce digitalizzata (che tollera un BER alto), non è adeguata ad un collegamento video o dati. Per un collegamento video, dobbiamo richiedere una QoS più elevata, con un BER almeno uguale a 10^{-5} . Il nuovo requisito porta a M = 5 per un deludente $\rho_{CDMA_2} = 5/128 \simeq 0.04$.



Figura 2.50 Codifica di Canale e CDMA

Questi due esempi mostrano chiaramente una caratteristica specifica di CDMA, e cioè la possibilita, insita nel protocollo, di variare QoS e capacità privilegiando l'una piuttosto che l'altra. Resta la (forte) limitazione di efficienza dovuta alla MAI, che però può essere migliorata attraverso l'uso della *codifica di canale*, che esamineremo in grande dettaglio nel Cap. 6. Il risultato, tutt'altro che soddisfacente di $rho_{CDMA_2} simeq0, 04$ deriva da un requisito di QoS elevato, cioè una BER (molto) bassa, che deve necessariamente derivare da un altrettanto basso livello di MAI \Rightarrow scarsa capacità. Nel Capitolo 6 vedremo che con un uso appropriato di un codice di canale, possiamo migliorare la BER per un certo livello di disturbo (nel nostro caso, AWGN + MAI), cioè ottenere una buona QoS (anche) con un livello di MAI relativamente alto \Rightarrow maggiore capacità. Vedremo anche nel paragrafo 6.1 che in generale il costo dell'applicazione di un codice di canale è l'aumento della banda occupata dal segnale codificato, quantificato nel fattore 1/r, r < 1 tasso di codifica - questa condizione si verifica però con segnali digitali convenzionali privi di codice di firma per il CDMA. È proprio vero che la codifica fa aumentare la larghezza di banda del segnale CDMA?

Supponiamo di voler applicare CDMA con N = 128, con un un aumento della larghezza di banda di 128 volte rispetto a un singolo canale. In assenza di codifica di canale, sappiamo che il numero massimo di utenti con BER = 10^{-5} è pari M = 5 (Fig. 2.46). Immaginiamo però di utilizzare un codice LDPC con r = 1/2 ad alte prestazioni, con le performance di Fig. 6.28. Senza CDMA, r = 1/2 significa raddoppio della banda occupata; ma, con CDMA possiamo adottare lo schema di Fig. 2.50: partiamo dai bit di dati informativi con bit time T, applichiamo la codifica r = 1/2 per fornire simboli binari di codice con il tempo di simbolo rT = T/2, infine applichiamo i codici di firma con lo stesso tempo di chip $T_c = T/128$ rispetto al caso senza codifica. Evidentemente, anzichè avere N = 128 chip per ciascun bit come in assenza di codici, avremo $N/2=64 \ {\rm chip}$ per ogni simbolo codificato prodotto dal codice LDPC, ma la banda occupata sarà comunque pari a $1/T_c = 128/T$. La MAI ha le stesse caratteristiche che in assenza di codice, e quindi possiamo ricalcolare, stavolta per simulazione, la BER sul generico canale desiderato considerando in presenza di MAI e con la codifica di canale, ottenendo le curve di Fig. 2.51 che sono le controparti di quelle di Fig. 2.46. Abbiamo intenzionalmente mantenuto la stessa scalatura su entrambi gli assi per visualizzare il grande cambiamento nelle curve di capacità. Vediamo che con lo stesso requisito di BER = 10^{-5} con E_b/N_0 = 11,3 dB, adesso abbiamo M = 117 e $\rho_{CDMA-cod} = 0.91$, praticamente pari a TDMA e FDMA, se si considera il tempo di guardia / banda di guardi. Addirittura, se allentiamo il nostro requisiti di QoS, vediamo che possiamo paradossalmente anche avere più di M = 128 canali attivi (non mostrato in Fig. 2.51)! Questo è chiamato sistema soprassaturo, e ancora una volta è un effetto peculiare del

MULTIPLAZIONE E ACCESSO MULTIPLO 75



Figura 2.51 BER del CDMA con codice LDPC a tasso 1/2 in funzione del numero di utenti attivi *M*.

CDMA: efficienza maggiore di uno.

Torneremo più in dettaglio su questo argomento nel paragrafo 9.5 forti di una maggiore conoscenza delle proprietà dei segnali spread-spectrum e CDMA, e della conoscenza approfondita della capacità di Shannon di un link digitale (Capitolo 4).



CAPITOLO 3

RIDONDANZA ED EFFICIENZA -COMPRIMEREIDATI

	U W X X Z		
J K L H H H H H H H H H H H H H H H H H H			

" • • • – – – • • • "

-Tabella del celebre codice Morse per la telegrafia, inventato da Samuel Morse (1791-1862) nel 1837

Il codice Morse è senz'alcun dubbio il primo esempio largamente diffuso di codifica di sorgente digitale, e con lunghezza variabile delle parole di codice in modo da minimizzare la lunghezza media del messaggio.

Comunicazioni Digitali, I Edizione. di M. Luise Copyright © 2023 University of Pisa

78 RIDONDANZA ED EFFICIENZA - COMPRIMEREIDAII

3.1 Che cos'è la "Codifica di Sorgente" ?

Nell'ambito della rete Internet, tutti i segnali che devono essere trasmessi vengono rappresentati in maniera digitale. Attraverso opportune operazioni di codifica, i segnali audio o video, voce o immagini vengono "tradotti" in un flusso di simboli binari (bit) che devono essere trasferiti su di un link di comunicazione, oppure memorizzati e ri-utilizzati successivamente. Naturalmente, questa operazione di codifica deve essere effettuata in modo ottimale, cosicché il flusso dati generato in termini di bit/s non sia eccessivamente gravoso per la rete che deve supportare la comunicazione e/o la memorizzazione. La velocità di trasmissione R_b in bit/s rappresenta infatti la *capacità* (talvolta erroneamente chiamata banda) che la rete deve mettere a disposizione della sorgente. Maggiore è il numero di bit prodotto dall'operazione di codifica, maggiore sarà la capacità richiesta (la dimensione del *file* risultante) e dunque il costo del servizio.

Esempio 3.12

Un segnale digitale video HDTV è costituito dalla successione di 25 (in Europa) quadri (cioè immagini) al secondo. Ogni immagine è composta da 1920 (sull'asse orizzontale H) \times 1080 (sull'asse verticale V) elementi (cioè pixel), ciascuno dei quali è caratterizzato da 256 livelli (8 bit) dei tre colori fondamentali Red, Green, Blue. Il bit rate totale del flusso video secondo questo formato è pari a

 $R_{b,HDTV} = 25 \times 1920 \times 1080 \times 3 \times 8 = 1.24416 \text{ Gbit/s}$ (3.1)

Per il formato UHD (Ultra-High-Definition) o 4k, si deve ulteriormente moltiplicare per 4 questo valore perché le risoluzioni H/V sono entrambe aumentate di un fattore 2, e volendo includere anche la modalità ad alto contrasto (HDR) nella quale i colori vengono rappresentati su 10 bit anziché 8, con un ulteriore aumento di un fattore 10/8, arriviamo a

 $R_{b,UHD-HDR} = 25 \times 3840 \times 2160 \times 3 \times 10 = 6.2208 \text{ Gbit/s}$ (3.2)

cioè un valore del tutto ragguardevole e al di là della ragionevole capacità di qualunque (ragionevole) collegamento digitale odierno (2020).

Riuscire a ridurre il numero di bit prodotti dall'operazione di codifica, cioè effettuare l'operazione di compressione della sorgente, significa quindi aumento della capacità totale di smistamento segnali nella rete e/o diminuzione del costo di trasmissione di un singolo segnale. Oppure, se l'informazione prodotta attraverso la codifica deve essere memorizzata (su hard disk, su di un Blue-Ray DVD ecc.), ridurre la velocità di informazione si riflette nell'aumento di capacità del mezzo di memorizzazione. Questa è dunque la ragion d'essere delle tecniche di codifica digitale e compressione di segnali audio (MP3) o vocali per trasmissioni telefoniche (vocoder), ovvero di segnali video (MPEG) o di immagini fisse (GIF, JPEG), nonché dei banali compressori/espansori di file generici come 7-zip ecc.

Prima di esaminare nei prossimi paragrafi i fondamenti teorici di queste tecniche, vediamo alcuni esempi elementari di compressione della sorgente che riusciremo in seguito a classificare in un contesto più ampio, ma che servono come punto di inizio della discussione

CHE COS'È LA "CODIFICA DI SORGENTE" ? 79



Figura 3.1 Scansione B/N di un documento di testo

3.1.1 Probabilità sbilanciate

Qualche volta i simboli binari "1" e "0" prodotti da una sorgente si presentano con frequenza molto diversa. Si pensi alla scansione di un documento di testo per il salvataggio in formato .pdf (Fig. 3.1): la maggior parte dei pixel nell'immagine sono bianchi e vengono codificati con "0", e solo occasionalmente si incontrano dei pixel neri codificati con "1". Uno spezzone del flusso binario prodotto dallo scanner:

Si nota che i simboli "0" tendono ad "addensarsi" in lunghe ripetizioni intervallate da pochi simboli "1". Prima di trasmettere questo flusso adottiamo allora una procedura di codifica particolare: ogni sequenza di "0" racchiusa tra due "1" non consecutivi viene sostituita dal solo "0" iniziale, seguito da un certo numero di bit, diciamo 3, che, in codifica binaria naturale, rappresentano la lunghezza totale di quella particolare ripetizione di "0". Se la ripetizione è più lunga di 7 (max. intero rappresentabile su 3 bit), si ripete 0 e si inseriscono altri 3 bit con la lunghezza della ripetizione restante e così via. La stringa data diventa come segue:

che risulta considerevolmente più breve della stringa originale: la sorgente binaria è stata compressa con un algoritmo di compressione senza perdita (lossless). Infatti dalla stringa codificata è possibile ricostruire esattamente, senza perdita di fedeltà e quindi di informazione, quella originaria. La codifica della lunghezza di ripetizione (con modalità leggermente diverse da quelle qui presentate, ma con lo stesso principio di funzionamento) è prevista nello standard di trasmissione delle diffusissime macchine fax gruppo 3 ITU V.29 a 9600 bit/s.

3.1.2 Codifica con perdita di qualità

Abbiamo visto che la codifica a standard CD di segnali audio richiede circa 700 kbit/s. Questa capacità è decisamente eccessiva per un rete di telecomunicazioni quando, piuttosto che un segnale ad alta qualità, si intenda trasmettere un segnale vocale con il solo intento della intellegibilita della voce o, in condizioni ottimali, del riconoscimento del parlatore.



Figura 3.2 Campionamento di un segnale

La velocità di trasmissione, e quindi la capacità richiesta, diminuiscono sensibilmente se si accetta dunque una perdita di fedeltà, cioè una distorsione, del segnale, La sorgente può essere sensibilmente compressa usando una strategia di codifica a perdita (lossy) che non preserva interamente il contenuto informativo del segnale, ma ne salvaguarda solo una parte ritenuta essenziale. Una strategia di codifica efficace sotto questo punto di vista è quella del PCM telefonico standard in cui il segnale analogico viene limitato in banda a 4 kHz (prima causa di perdita di informazione), e campionato a 8 kcampioni/s per ottenere la sequenza i cui campioni vengono rappresentati (previa operazione di compressione dei livelli secondo la legge A o μ , qui irrilevante) su 8 bit (quantizzazione rozza, seconda fonte di perdita di fedeltà). Il segnale digitale risultante, come è noto, ha una velocità di 64 kbit/s, meno di 1/10 della velocità originale del PCM a qualità CD. Accettando dunque gradi più o meno alti di perdita di informazione si possono realizzare rapporti di compressione della sorgente (intesi come rapporto tra la velocità del flusso originale e di quello compresso) molto alti.

3.1.3 Correlazione tra valori della sorgente

Compiamo adesso un salto quantistico. Dimentichiamo cioè completamente il bel segnale di qualità CD a 700 kbit/s, e concentriamoci su di un segnale vocale digitalizzato, ad esempio la cosiddetta codifica "PCM telefonico" a 64 kbit/s. È possibile ridurre ulteriormente questa velocità senza un'apprezzabile ulteriore perdita di fedeltà ? Da questo punto di vista, non è consigliabile scendere sotto gli 8 bit/campione. Su quale aspetto dobbiamo dunque agire per comprimere ulteriormente la sorgente ? La Fig. 3.2 rappresenta il segnale vocale limitato nella banda B = 4 kHz con il valore di alcuni suoi campioni.

Si nota che il valore di due campioni consecutivi non è normalmente molto diverso, come se in qualche modo il campione successivo risentisse del valore del precedente. Questo aspetto indica una certa quantità di memoria nella sorgente di segnale, nel senso che il valore corrente di un campione è fortemente correlato con un certo numero di campioni precedenti, e indica anche una via pratica per comprimere ulteriormente la sorgente come desiderato.

La Fig. 3.3 mostra lo schema di codifica PCM differenziale (DPCM). Come si nota, la quantizzazione non avviene direttamente sulla sequenza x[n], ma sulla sequenza e[n] ottenuta come differenza tra x[n] e una sequenza $\hat{x}[n]$ il cui campione n-esimo è ottenuto come *predizione* di x[n] a partire dai campioni precedenti di una versione di x[n], che abbiamo chiamato $x_Q[n]$, che contiene un errore di quantizzazione. Infatti $\hat{x}[n] = x[n] - e[n]$; d'altronde e[n] viene quantizzato, ottenendo $e_Q[n]$ e, come si vede dallo schema,

$$x_Q[n] = \hat{x}[n] + e_Q[n] = x[n] - e[n] + e_Q[n] = x[n] - q[n] \quad , \quad q[n] \stackrel{\bigtriangleup}{=} e[n] - e_Q[n]$$

CHE COS'È LA "CODIFICA DI SORGENTE" ? 81



Figura 3.4 Decodificatore (A)DPCM

dove q[n] è proprio l'errore di quantizzazione.

Di fatto, come sarà chiaro tra poco, il segnale $x_Q[n]$ è proprio la *ricostruzione* di x[n] che il decodificatore DPCM è in grado di produrre. La predizione garantisce buoni risultati proprio in virtù dell'alta correlazione tra campioni consecutivi di x[n], e quindi il segnale errore di predizione e[n] è un segnale mediamente "piccolo" (cioè con valore efficace minore di quello di x[n]) e che può essere quindi quantizzato su un numero minore di bit.

Per avere un'idea di come è possibile "predire" il valore del segnale, e di come la correlazione tra campioni adiacenti influisce sulla possibilità di compressione del segnale, analizziamo il caso del "predittore lineare ottimo ad un passo". In questo caso, il segnale predetto $\hat{x}[n]$ viene ricavato come combinazione lineare del campione di segnale ricostruito

82 RIDONDANZA ED EFFICIENZA - COMPRIMEREIDAI

immediatamente precedente:

$$\hat{x}[n] = ax_Q[n-1] = a(x[n-1] - q[n-1])$$
(3.3)

dove il valore *a* viene determinato in modo che l'errore quadratico medio di predizione sia minimo:

$$\varepsilon \stackrel{\Delta}{=} E\{|\hat{x}[n] - x[n]|^2\} = \min$$
(3.4)

Sviluppando i conti si ha

$$\varepsilon = E\{|ax[n-1] - aq[n-1] - x[n]|^2\}$$

= $E\{a^2x^2[n-1] + a^2q^2[n-1] + x^2[n] - 2ax[n]x[n-1]\}$
= $a^2(R_x[0] + \sigma_a^2) - 2aR_x[1] + R_x[0]$ (3.5)

dove abbiamo fatto l'ipotesi che x[n] sia un processo stazionario, e ne abbiamo indicato con $R_x[m] \stackrel{\Delta}{=} E\{x[n]x[n-m]\}$ la sequenza d'autocorrelazione, e dove abbiamo indicato con σ_q^2 la potenza dell'errore di quantizzazione, errore che si suppone anche indipendente dal valore del segnale. La (3.5) è una funzione quadratica in *a* che viene minimizzata per il seguente valore ottimo:

$$a_{OTT} = R_x[1] / (R_x[0] + \sigma_q^2)$$
(3.6)

Se evidentemente i campioni del segnale fossero incorrelati, si avrebbe a = 0, il segnale sarebbe non predicibile e la codifica DPCM perderebbe di significato !

In generale, il predittore è una funzione causale di N campioni precedenti del segnale da codificare, $\hat{x}[n] = \mathcal{F}(x[n-1], x[n-2], \dots, x[n-N])$, spesso un semplice filtro FIR

$$\hat{x}[n] = \sum_{k=1}^{N} a_k x[n-k]$$

con i coefficienti da ottimizzare come sopra.

La ricostruzione del segnale avviene come in Fig. 3.4, in cui si ricostruisce il segnale con errore di quantizzazione $x_Q[n]$: il filtro di predizione ricostruisce infatti il valore corrente di $\hat{x}[n]$ a partire dai campioni precedenti di $x_Q[n]$ esattamente come nel codificatore, e al valore predetto viene poi aggiunto l'errore di predizione quantizzato $e_Q[n]$ in ingresso al decodificatore stesso ricostruito dal flusso dai ricevuto. Si vede che il codificatore contiene sostanzialmente una replica del decodificatore per ricostruire il segnale quantizzato a partire dal quale si calcola la predizione.

Il motivo per cui il predittore del codificatore riceve i campioni $x_Q[n]$ piuttosto che gli x[n] disponibili nel codificatore è che il codificatore e il decodificatore, nella predizione devono essere perfettamente "allineati". Se il codificatore usasse x[n], i valori predetti $\hat{x}[n]$ sarebbero diversi da quelli del decodificatore che non ha x[n] - per evitare questo disallineamento anche il codificatore è costretto ad usare una quantità che il codificatore ha disponibile, cioè $x_Q[n]$.

Nel caso particolare del segnale vocale, è necessario rendere il predittore *adattativo*, perché il segnale *non* è stazionario - le caratteristiche della voce cambiano sostanzialmente quando si emettono vocali (segnale periodico molto predicibile) a consonanti come la "s" o la "f" nel qual caso il segnale è più *noise-like*. Allora i coefficienti del filtro predittore vengono adattati con una certa cadenza alle condizioni variabili del segnale vocale (10 ms). In questo modo la predizione è sempre ottimale e si ha una sostanziale riduzione del numero di bit necessari per quantizzare e[n].

Naturalmente, oltre a $e_Q[n]$, si devono codificare nel flusso di uscita anche i valori di detti coefficienti, che però incidono marginalmente sulla velocità di informazione complessiva. La codifica ADPCM (Adaptive Differential PCM) è stata standardizzata nella norma ITU G.721 per una velocità di trasmissione di 32 kbit/s, realizzando quindi un rapporto di compressione 2 rispetto al PCM standard, e con una qualità soggettiva del segnale praticamente coincidente. Questa codifica viene utilizzata nella telefonia fissa per raddoppiare il numero di conversazioni smaltibile dalla rete (dato che si dimezza il bit-rate). Viene anche adottata nei telefoni cordless DECT per la comunicazione tra la cornetta e la base.

I tre esempi di codifica di sorgente secondo criteri diversi (probabilità, fedeltà, memoria) saranno paradigmatici delle tecniche che possono essere sviluppate attraverso lo studio e l'applicazione sistematica della teoria dell'informazione.

3.2 Codifica senza perdita di una sorgente senza memoria

Possiamo caratterizzare una sorgente di informazione digitale come un dispositivo o un sistema che produce una sequenza di variabili aleatorie discrete S[n] (o processo aleatorio a tempo discreto) i cui valori, detti simboli, appartengono ad un certo spazio campione finito $\Omega = \{s_1, s_2, \dots, s_M\}$ chiamato alfabeto. I vari simboli sono quindi i vari valori assunti dalle v.a. della successione in corrispondenza della particolare realizzazione della successione aleatoria. Omettendo la dipendenza dal tempo, identificheremo la sorgente con il nome delle v.a. della successione, e cioè S.

La velocità di emissione dei simboli di S si misura naturalmente in simboli/s o baud. L'alfabeto Ω potrebbe essere binario $\{0,1\}$ quando la sorgente è un generico *file* di dati, potrebbe essere l'alfabeto internazionale $\{a, b, c, d, \dots, x, y, z, \}$ compreso lo spazio se la sorgente è un testo, $\{0, 1, \dots, 255\}$ se la sorgente è un'immagine in B/N e il simbolo rappresenta il livello di grigio di un pixel.

L'emissione di un simbolo da parte di una sorgente di informazione digitale (d'ora in poi, brevemente: sorgente) corrisponde dunque al verificarsi di un evento, in particolare all'evento $\{S[n] = s_m\}$ (in breve, e con abuso di notazione, evento s_m) per un qualche $m, 1 \leq m \leq M$. In una sorgente senza memoria S, gli atti di emissione dei simboli sono indipendenti e la caratterizzazione della sorgente stessa è completa quando sono note le varie proprietà di presentazione dei vari simboli dell'alfabeto, supposte stazionarie

$$p_m \stackrel{\Delta}{=} \Pr\{S[n] = s_m\} \tag{3.7}$$

Le sorgenti di informazione possono essere dunque di due tipi: i) sorgente senza memoria, se le v.a. S[n] sono mutuamente indipendenti; ii) sorgente con memoria (causale o non causale) se S[n] è statisticamente dipendente da qualche S[m], $m \neq n$ (risp. m < n o anche m > n). Per quanto visto nel paragrafo precedente il codificatore vocale è una sorgente con memoria, mentre tipicamente i bit che scorrono in un collegamento Internet possono considerarsi una sorgente binaria di informazione senza memoria.

Vediamo dunque come si può procedere per codificare una sorgente senza memoria. Per prima cosa bisogna stabilire un nuovo alfabeto, detto di codice, sul quale i simboli di sorgente saranno appunto rimappati. Indichiamo con $\Delta = \{d_1, d_2, ..., d_L\}$ questo nuovo alfabeto (chiamato a volte dizionario) - normalmente Δ è binario per poter facilmente trasmettere o memorizzare il flusso risultante. Un codice \mathcal{M} sarà allora una corrispondenza che associa a gruppi di simboli di Ω gruppi di simboli di Δ , chiamati *parole di codice*. In particolare, un codice a blocco mappa ciascuno dei singoli simboli s_m di Ω (m =

1, 2, ..., M) in una ben determinata e fissa sequenza di simboli di Δ , cioè in una parola di codice fissa che chiameremo μ_m , come nell'esempio seguente di un ipotetico alfabeto extraterrestre a 4 sole lettere:

- Alfabeto di sorgente: $\Omega = \{a, b, c, d\}$
- Alfabeto di codice: $\Delta = \{0, 1\}$

• Codice:
$$\mathcal{M}_1$$
:
$$\begin{cases} s_1 = a \to \mu_1 = 0\\ s_2 = b \to \mu_2 = 10\\ s_3 = c \to \mu_3 = 11\\ s_4 = d \to \mu_4 = 10 \end{cases}$$

Si nota che questo codice è del tipo a *lunghezza variabile* perché le parole di codice hanno appunto lunghezza variabile. La codifica ASCII per rappresentare i caratteri alfanumerici è un altro esempio di codifica di sorgente (a lunghezza fissa) con alfabeto di sorgente a 127 simboli e codifica con parole di 7 simboli binari. Utilizzando il codice, è possibile codificare una qualunque *stringa* (cioè sequenza finita) di simboli di sorgente per inviarli su di un link Internet o memorizzarli in un file e successivamente rileggerli.

In generale, si desidera poter ricostruire *esattamente* (cioè *senza perdita*) la stringa di sorgente a partire dalla stringa codificata. Da questo punto di vista, il codice (3.2) è inutilizzabile perché *singolare*: uno stesso blocco di simboli di codice (cioè una stessa parola di codice) è in corrispondenza con *due* diversi simboli di sorgente, ed è quindi impossibile decodificare il codice. L'operazione di decodifica è la mappa inversa al codice che permette cioè di risalire dalle parole di codice ai blocchi di sorgente. Un codice non singolare potrebbe essere:

$$\mathcal{M}_2: \begin{cases} a \to 0\\ b \to 10\\ c \to 11\\ d \to 00 \end{cases}$$

che però non è ancora decodificabile. Infatti il blocco 00 è ambiguo: può riferirsi a una successione di due lettere a di sorgente o a un singolo d. Ciò che si ricerca nella codifica di sorgente è un codice a decodifica univoca, che cioè non crea ambiguità nella ricostruzione del messaggio di sorgente.

Non è semplice controllare se un codice è univocamente decodificabile, e l'esempio appena fatto lo dimostra. Elenchiamo infatti la codifica secondo questo codice di tutte le stringhe di sorgente di lunghezza K = 2: Abbiamo costruito l'estensione del codice alla lunghezza K = 2 di sorgente - non è un nuovo codice, ma è banalmente il prolungamento del codice dato (in generale, possiamo estendere fino alla lunghezza generica K). Si nota che questo nuovo codice è singolare: i simboli ad e da vengono mappati nella stessa parola 000. Questo indica che si possono incontrare problemi nella decodifica del codice M_2 come già accennato. Se ne conclude che un codice è *univocamente decodificabile* se (e, si può dimostrare, solo se) la generica estensione K-esima del codice stesso è nonsingolare per ogni K - una condizione concettualmente semplice ma assai difficile da verificare...

Esaminiamo ora due distinti codici univocamente decodificabili per la consueta sorgente quaternaria:

$$\mathcal{M}_3: \begin{cases} a \to 00 \\ b \to 01 \\ c \to 11 \\ d \to 10 \end{cases} e \quad \mathcal{M}_4: \begin{cases} a \to 00 \\ b \to 01 \\ c \to 011 \\ d \to 0111 \end{cases}$$

Per \mathcal{M}_3 , non appena ricevo uno qualunque dei simboli di codice sono in grado di procedere alla decodifica e ricostruire il simbolo di sorgente; viceversa, se con \mathcal{M}_4 ricevo il simbolo 01, non posso procedere alla decodifica immediata, ma devo attendere uno o due bit finché non ricevo un altro 0. Il primo dei due codici è istantaneo, mentre il secondo non lo è. Questa caratteristica di avere ritardo di decodifica piccolo o nullo è raccomandabile in tutte quelle applicazioni cosiddette "in tempo reale", come la trasmissione telefonica, in cui un ritardo eccessivo sul segnale può diventare fastidioso.

> $aa \rightarrow 00$ $ab \rightarrow 010$ $ac \rightarrow 011$ $ad \rightarrow 000$ $ba \rightarrow 100$ $bb \rightarrow 1010$ $bc \rightarrow 1011$ $bd \rightarrow 1000$

 $ca \rightarrow 110$

 $cb \rightarrow 1110$

 $cc \rightarrow 1111$

 $cd \rightarrow 1100$

 $da \rightarrow 000$

 $db \rightarrow 0010$ $dc \rightarrow 0011$

 $dd \rightarrow 0000$

Esiste una condizione necessaria e sufficiente per con-
trollare se un codice è univocamente decodificabile e
istantaneo: la generica parola di codice non deve com-
parire come prefisso di un' altra qualunque parola. Il
codice \mathcal{M}_3 viola questa condizione: la parola associata
a b è infatti il prefisso di ben due altri simboli del codice
stesso. Un codice che rispetta la condizione del prefisso
si chiame codice a prefisso. Un risultato fondamentale ri-
guardo i codici a prefisso (istantanei) è la disuguaglianza
di Kraft che riguarda la lunghezza delle varie parole di
codice.

Indichiamo come sempre con L il numero di simboli dell'alfabeto di codice, e indichiamo con $l_m, m =$ 1, ..., M le lunghezze misurate in simboli delle varie parole di codice μ_m associate agli M simboli di sorgente. Per M_3 si ha L = 2, $l_1 = 2$, $l_2 = 2$, $l_3 = 3$ e $l_4 = 4$. Un codice istantaneo univocamente decodificabile può avere parole di codice con lunghezza $l_m, m = 1, ..., M$ se e solo se

$$\sum_{m=1}^{M} L^{-l_m} \le 1$$
 (3.8)

Convincersi di questa relazione è semplice nel caso

di codice binario, cioè con L = 2. Supponiamo di aver ordinato le lunghezze delle parole in modo che $l_1 \leq l_2 \leq ... \leq l_M$. Posso raffigurare tutte le possibili parole di codice (di lunghezza variabile) su di un albero binario di ordine l_M costruito a partire da un'unica radice seguendo una regoletta ricorsiva come mostrato in Fig. 3.5 (a) per $l_M = 5$: per ogni nodo aggiungo un ramo "verso l'alto" e contrassegno il nodo terminale con la parola del nodo "padre" seguita da un 1, e un ramo verso il basso contrassegnando il nodo con la parola "padre" seguita da uno 0. La parola "radice" è ovviamente nulla, e in totale l'albero ha 2^{l_M} nodi terminali (foglie).

Una parola di codice è dunque un qualche nodo di questo albero, e l'albero è particolarmente utile per visualizzare la regola del prefisso: se vogliamo costruire un codice a prefisso, dobbiamo evitare che una parola di codice abbia un nodo "figlio" di qualunque ordine (cioè a qualunque distanza verso le foglie dell'albero) che risulti ancora una parola di

85



Figura 3.5 Albero di Kraft

codice. Allora se scelgo una certa parola di codice di lunghezza l_m , l'albero non potrà più avere le $2^{l_M}/2^{l_m} = 2^{l_M-l_m}$ foglie che derivano dalla parola scelta, come mostrato in Fig. 3.5 (b) per la parola 01 con lunghezza 2 (in rosso): vengono cancellate (tutte le) $2^{5-2} = 8$ foglie (in grigio) che hanno 01 come prefisso. Poiché non posso cancellare dall'albero più foglie di quante ce ne siano in totale, ripetendo il ragionamento per ciascuna parola del codice, dovrà succedere che

$$2^{l_M - l_1} + 2^{l_M - l_2} + \dots + 2^{l_M - l_M} = \sum_{m=1}^M 2^{l_M - l_m} \le 2^{l_M}$$
(3.9)

ovvero, semplificando, la disuguaglianza di Kraft. La generalizzazione al codice *L*-ario è immediata.

Il ragionamento dimostra la necessità della disuguaglianza, ma ragionando in modo "inverso" si arriva facilmente anche a dimostrare la sufficienza. Naturalmente questa condizione stabilisce che la costruzione di un codice con parole di certe lunghezze, in particolare troppo *corte*, è impossibile. Vedremo nel prossimo paragrafo l'utilità di questo risultato parlando di codifica a minima lunghezza.

Torniamo a un problema già enunciato. Possiamo cercare di eliminare in qualche modo la ridondanza di una sorgente evitando di trasmettere quella frazione η di simboli "inutili" ed utilizzare il collegamento o il supporto di memorizzazione nella maniera più efficiente possibile? Evidentemente la soluzione alla questione consiste in un opportuno schema di *codifica di sorgente senza perdita*. Il criterio di scelta di tale codice sarà la minimizzazione della lunghezza delle parole di codice necessarie per codificare la sorgente, in modo che il file codificato abbia la minima dimensione possible o il link digitale per la trasmissione in tempo reale richieda la minima velocità.

A che cosa è proporzionale la dimensione del file risultante dalla codifica di una stringa di sorgente sufficientemente lunga? Ovviamente alla lunghezza *media* delle parole prodotte dal codificatore. Questo parametro è dato da

$$l_{med} \stackrel{\Delta}{=} \sum_{m=1}^{M} p_m l_m \tag{3.10}$$

espresso in cifre binarie che vengono in media associate ad ogni simbolo di sorgente. Naturalmente, un codice con lunghezza media delle parole minima richiede la minima capacità di trasmissione richiesta alla rete per comunicare l'informazione di una data sorgente. Se la sorgente emette a R_s baud, la velocità media di emissione dopo la codifica è $R_s^{cod} = R_s l_{med}$ cifre binarie/s.

Nascono due domande:

- 1. Qual è la lunghezza minima che si può ottenere ?
- 2. Come è possibile ottenerla?

In particolare, la disuguaglianza di Kraft 3.8 fa intuire che non è possibile ridurre arbitrariamente la lunghezza delle parole di codice, altrimenti si rischia di perdere la caratteristica di univoca decodificabilità - sembra esistere un limite fisico al di sotto del quale non è possibile scendere. Per rispondere a questa domanda dobbiamo aprire un'ampia parentesi, giustificata dall'importanza del tema della codifica di sorgente per aumentare l'efficienza nella comunicazione o memorizzazione dell'informazione - uno dei punti chiave identificati da Shannon.

88 RIDONDANZA ED EFFICIENZA - COMPRIMEREIDATI

3.3 Shannon Information

Nel paragrafo precedente abbiamo liberamente fatto uso di termini come *informazione*, *velocità di informazione*, *capacità* senza darne una definizione precisa. Dobbiamo a questo punto uscire dall'ambiguità e precisare tutti questi termini e concetti.

Il concetto di informazione nasce cercando di stabilire il grado di *incertezza* che viene rimosso dal verificarsi di un certo evento. Se la squadra della Juventus incontra la squadra del Pontedera in amichevole, possiamo modellare questo accadimento con un esperimento aleatorio che ha tre possibili eventi elementari (risultati), {Vittoria Juve}, {Pareggio} e {Vittoria Pontedera}. Qual è il grado di incertezza, e quanta informazione apporta il verificarsi di uno dei due possibili risultati ? L'evento {Vittoria Juve}, che ha la più alta probabilità di verificarsi, apporta ben poca informazione perchè rimuove ben poca incertezza, e ad esso forse non viene neanche dedicato un trafiletto in cronaca sportiva dei giornali nazionali. Se al contrario il Pontedera batte la Juventus, {Vittoria Pontedera} apporta *grande* informazione e magari viene riportato in prima pagina.

Dunque l'informazione di un certo evento A, che indicheremo con In(A), è una *misura* del grado di incertezza rimosso dal verificarsi di tale evento. Per convenienza e consistenza con questa definizione empirica, l'informazione deve obbedire a certe regole:

- $\ln(A) \ge 0$
- In(A) varia in maniera inversamente proporzionale a Pr(A)
- In(A) = 0 quando Pr(A) = 1, cioè l'evento certo non porta informazione
- Per due eventi indipendenti $A_1 \in A_2$, $\ln(A_1 \cap A_2) = \ln(A_1) + \ln(A_2)$

Una definizione conveniente che ha queste proprietà (in particolare la terza) è la seguente:

$$\ln(A) \stackrel{\triangle}{=} \log\left(1/\Pr(A)\right) = -\log\left(\Pr(A)\right) \tag{3.11}$$

La base di calcolo dei logaritmi non è essenziale. Se si utilizza la base 2, come è ormai universalmente in uso, l'informazione viene misurata in *bit* (binary unit). Agli albori della teoria dell'Informazione venivano usati logaritmi naturali, e l'unità veniva chiamata *nat*. Dunque:

$$\ln(A) \stackrel{\Delta}{=} -\log_2\left(\Pr(A)\right) \tag{3.12}$$

Stimando in 1 su un milione il numero di spose o sposi che all'altare dicono "NO", il lettore ricavi il numero di bit di informazione apportati da quest'evento (che talvolta merita la prima pagina del giornale locale).

Esempio 3.13

Dimostriamo la necessità della misura logaritmica dell'informazione secondo le 4 proprietà di cui sopra. Per due eventi indipendenti $A \in B$ aventi probabilità $Pr(A) = p_A \in Pr(B) = p_B$, sappiamo che la legge matematica ancora da determinarsi μ che lega la probabilità alla misura d'informazione deve godere della proprietà $\mu(p_A \cdot p_B) = \mu(p_A) + \mu(p_B)$. Deriviamo questa relazione membro a membro rispetto a p_A :

$$p_B \cdot \mu'(p_A \cdot p_B) = \mu'(p_a) \tag{3.13}$$

dove $\mu'(p) \stackrel{\triangle}{=} d\mu(p)/dp$. Deriviamo adesso la (3.13) *rispetto a* p_B :

$$\mu'(p_A \cdot p_B) + p_B p_A \cdot \mu''(p_A \cdot p_B) = 0 \implies \mu'(p) + p\mu''(p) = 0 \implies (p\mu'(p))' = 0$$
(3.14)

ovvero $p\mu'(p) = \gamma$, dove γ è una qualunque costante. Aggiungiamo la condizione al contorno $\mu(1) = 0$ (terza proprietà) e risolviamo l'equazione differenziale:

$$p\frac{d\mu}{dp} = \gamma \Rightarrow d\mu = \gamma \frac{dp}{p} \Rightarrow \int_{1}^{p} d\mu = \gamma \int_{1}^{p} \frac{dp}{p}$$
 (3.15)

e, tenendo conto della condizione al contorno, troviamo la soluzione $\mu(p) = \gamma \ln(p)$. Tenendo conto che $0 \le p \le 1$ e della prima proprietà, deve essere $\gamma < 0$ ma arbitrario. La soluzione finale è allora $\mu(p) = -\log(p)$ ove il log può essere in qualunque base.

3.4 Quantità di informazione di una sorgente senza memoria

È semplice a questo punto calcolare la quantità di informazione mediamente apportata dall'emissione di un singolo simbolo di sorgente:

$$H(S) \stackrel{\triangle}{=} \sum_{m=1}^{M} p_m \ln(s_m) = \sum_{m=1}^{M} p_m \log\left(\frac{1}{p_m}\right) = -\sum_{m=1}^{M} p_m \log_2(p_m) \tag{3.16}$$

Questa *informazione media* valore, che è anche chiamato *entropia* della sorgente, caratterizza il contenuto informativo medio di un singolo simbolo di S, ed è univocamente determinato dalla legge di probabilità della sorgente stessa.

Se S emette alla cadenza di R_s baud (simboli/s), la velocità di informazione (entropy rate) della sorgente è $R_b = R_s \cdot H(S)$ bit/s. Il nome di entropia è stato dato da Claude Shannon a questa misura di informazione da egli stesso formalizzata negli anni '40 dopo gli studi pionieristici di Hartley degli anni '20. L'appellativo si rifà al concetto di entropia di un sistema fisico. In meccanica statistica, Boltzmann scoprì che l'entropia termodinamica è proporzionale (attraverso la omonima costante k) al logaritmo del numero N degli stati che il sistema fisico può occupare. Se gli stati sono equiprobabili, N è anche pari all'inverso della probabilità di occupazione di ogni stato (1/N, appunto), consistentemente alla definizione di Shannon. D'altronde, se gli stati sono equiprobabili, il sistema è nello stato di massimo disordine e, da nozioni di fisica elementare, l'entropia è notoriamente massima: questa quantità dà un'indicazione del grado di *disordine* o *incertezza* del sistema.

Studiamo adesso un caso particolarmente semplice: la *sorgente binaria* senza memoria. Diciamo per semplicità che $s_1 = 0$ e $s_2 = 1$. Essa è caratterizzata dalla probabilità p di presentazione del simbolo 0 P(0) = p, la quale determina anche la P(1) = 1 - p. Il calcolo dell'entropia è immediato:

$$H(S) = p \log\left(\frac{1}{p}\right) + (1-p) \log\left(\frac{1}{1-p}\right) \stackrel{\triangle}{=} \eta(p)$$
(3.17)

In seguito useremo anche la seguente relazione (facilmente verificabile):

$$\frac{d\eta(p)}{dp} = \eta'(p) = \log\left(\frac{1-p}{p}\right) \quad 0 \le p \le 1$$
(3.18)

89



Figura 3.6 Entropia della sorgente binaria

Dal grafico di $\eta(p)$ riportato nella Fig. 3.6 si possono notare alcune proprietà dell'entropia di una sorgente binaria, che possono essere facilmente estese ad una sorgente M-aria qualunque:

- $H(S) \ge 0$ (banale);
- H(S) = 0 se e solo se una tra le probabilità dei simboli è pari a 1. In questo caso la sorgente emette con certezza sempre quel certo simbolo, e quindi non apporta alcun contenuto informativo;
- H(s) è invariante rispetto a permutazioni delle probabilità dei simboli da questo punto di vista gode di una proprietà di simmetria;
- H(S) ≤ log(M) e l'uguale vale solo nel caso in cui tutte le probabilità dei simboli sono uguali (e sono dunque uguali a 1/M);

La sorgente equiprobabile è quella a maggior contenuto informativo: qualunque sbilanciamento nelle probabilità di emissione dei simboli porta ad una diminuzione dell'entropia. Se infatti le probabilità sono sbilanciate, alcuni simboli sono più "certi" di altri e il ridotto ammontare di informazione di ciascuno di questi porta globalmente ad una diminuzione di informazione media. Per dimostrare la proprietà che $H(S) \leq \log(M)$, dobbiamo ricavare un piccolo "lemma" sulle distribuzioni discrete di probabilità noto come disuguaglianza di Gibbs. Se ho due leggi di probabilità $p_m e q_m, m = 1, \ldots, M$, allora

$$\sum_{m=1}^{M} p_m \log\left(\frac{1}{p_m}\right) \le \sum_{m=1}^{M} p_m \log\left(\frac{1}{q_m}\right)$$
(3.19)

Infatti è ben noto dall'analisi matematica che $\ln(x) \le (x-1)$ (Fig. 3.7) e che l'uguaglianza vale solo per x = 1, ove le due curve sono tangenti. Poniamo allora $x = q_m/p_m$; segue che $\ln(q_m/p_m) \le (q_m/p_m - 1)$. Se moltiplico entrambi i membri di questa relazione per p_m e

QUANTITÀ DI INFORMAZIONE DI UNA SORGENTE SENZA MEMORIA 91





sommo su m ottengo

$$\sum_{n=1}^{M} p_m \ln\left(\frac{q_m}{p_m}\right) \le \sum_{m=1}^{M} (q_m - p_m) \Rightarrow$$

$$\sum_{m=1}^{M} p_m \ln\left(\frac{q_m}{p_m}\right) \le 0 \Rightarrow$$

$$\sum_{m=1}^{M} p_m \log\left(\frac{q_m}{p_m}\right) \le 0 \qquad (3.20)$$

ove chiaramente il secondo membro si annulla perché la somma delle due distribuzioni di probabilità vale per entrambe 1, e il segno del primo membro non cambia se da logaritmo in base *e* passiamo a quelli in base 2 come nella definizione di entropia. Il passaggio finale è poi banale.

Esempio 3.14

Se è vero che

$$\sum_{m=1}^{M} p_m \log\left(\frac{q_m}{p_m}\right) \le 0$$

allora anche

$$\sum_{m=1}^{M} p_m \log\left(\frac{p_m}{q_m}\right) \ge 0$$

Questa seconda espressione viene utilizzata in statistica per valutare quanto una certa legge di probabilità $\{p_m\}$ si avvicina ad una legge-prototipo $\{q_m\}$ e viene indicata con il nome di *divergenza di Kullback-Leibler*:

$$D_{KL}(p||q) \stackrel{\triangle}{=} \sum_{m=1}^{M} p_m \log\left(\frac{p_m}{q_m}\right) \ge 0$$
(3.21)

92 RIDONDANZA ED EFFICIENZA - COMPRIMEREIDAII

m	q_m	$p_m^{(1)}$	$p_{m}^{(2)}$
1	$1 \cdot 10^{-7}$	0,002413931	0,094056733
2	0,0000063	0,046366715	0,046978491
3	0,0001701	0,038960623	0,078982638
4	0,0025515	0,043802808	0,093429256
5	0,0229635	0,026206271	0,057075785
6	0,1240029	0,166344282	0,036994862
7	0,3720087	0,395708934	0,454060442
8	0,4782969	0,280196436	0,138421793
D(p q)	0	1,010426223	3,54912057

Tabella 3.1 Leggi di probabilità e Divergenza di K-L



Maggiore è la divergenza, maggiore è lo scostamento rispetto alla legge-prototipo $\{q_m\}$ - divergenza 0 significa al contrario identità.

Ad esempio, da dati sperimentali potrei avere ricavato le due misure di probabilità $p_m^{(1)}$ e $p_m^{(2)}$ di una sorgente a M = 8 livelli mostrati nelle colonne 1-2 in Tab. 3.1, e vorrei stabilire quale delle due si avvicina di più ad un andamento Bernoulli con probabilità 0.1-0.9

$$q_m = \binom{7}{m-1} (0.1)^{m-1} (0.9)^{7-m} , \ m = 1, \dots, 8$$

che deriva da un modello teorico e i cui valori sono riportati in colonna 3. I dati sono anche mostrati graficamente in Fig. 3.8, dove si nota che la legge di probabilità più vicina alla Bernoulli è chiaramente la 1, così come efficacemente indicato dalla divergenza nella Tab. 3.1.

La divergenza di K-L non è una "distanza", nel senso che non gode della proprietà di simmetria $(D(p||q) \neq D(q||p))$, è però efficace come misura globale di scostamento, così come è stata utilizzata in quest'esempio.

Come usiamo questo "lemma" per dimostrare che $H(S) \leq \log(M)$? Basta porre $q_m = 1/M$ per tutti gli m (cioè la distribuzione di una sorgente equiprobabile). Nella disuguaglianza di Gibbs, il primo membro è di fatto l'entropia H(S), per cui si ha

$$H(S) \le \sum_{m=1}^{M} p_m \log(M) = \log(M) \sum_{m=1}^{M} p_m = \log(M)$$
(3.22)

Una sorgente *M*-aria porta dunque al massimo $\log(M)$ bit/simbolo, e produce informazione alla velocità massima di $R_b = \log(M) \cdot R_s$ bit/s. Una generica sorgente senza memoria avrà un contenuto informativo non superiore a questo, e potrà essere caratterizzata con un parametro ρ , $0 \le \rho \le 1$, chiamato ridondanza

$$\rho \stackrel{\Delta}{=} 1 - H(S) / \log(M) \tag{3.23}$$

Il significato di ρ è chiaro per la sorgente binaria. Se questa emette alla velocità di R_s simboli binari al secondo, la sua effettiva velocità di informazione è di $R_b = H(S) \cdot R_s$ bit/s, mentre la velocità massima di informazione che potrebbe supportare è $R_{b,M} = H_{\max} \cdot R_s = \log(2) \cdot R_s = R_s$ bit/s. Dunque

$$\frac{R_b}{R_{b,M}} = \frac{H(S) \cdot R_s}{R_s} = H(S) = 1 - \rho$$
(3.24)

Dei simboli binari di sorgente, soltanto la frazione $H(S) = 1 - \rho$ "porta" effettivamente informazione, mentre la restante frazione ρ è ridondante, cioè apparentemente inutile ai fini della trasmissione dell'informazione. Una sorgente con probabilità fortemente sbilanciate ha $H(S) \ll H_{\text{max}}$, ed è quindi fortemente ridondante (ρ vicino a uno). Viceversa, una sorgente simmetrica ha ridondanza zero.

Da questa osservazione scaturisce naturale il concetto di *compressione* di una sorgente: se $H(S) < H_{max}$, parte dei simboli di sorgente appaiono "inutili" per la trasmissione o la memorizzazione dell'informazione, ed è forse possibile trovare una maniera di "rimappare" i simboli di sorgente in modo da fare a meno di emettere la frazione ρ di simboli ridondanti. Per compiere quest'operazione, come vedremo, e specialmente quando si ha a che fare con sorgenti binarie, è consuetudine "raggruppare" più simboli consecutivi emessi dalla sorgente in blocchi di k simboli e considerare questi blocchi come prodotti da una *nuova* sorgente legata alla originaria. Questa nuova sorgente si indica con $S^{(k)}$, e il suo alfabeto è costituito dagli M^k blocchi distinti ordinati di k simboli M-ari che possiamo formare. Ad esempio, nel caso dell'ottava estensione di una sorgente binaria i bit vengono raggruppati 8 a 8 in byte, e l'alfabeto è, con ovvia notazione,

$$\Omega^{(8)} = \{\underbrace{00000000, 0000001, 00000010, \dots, 11111110, 1111111}_{256 \text{ valori}}\}$$
(3.25)

Poiché i vari simboli emessi dalla sorgente originaria senza memoria sono indipendenti, la distribuzione di probabilità di $S^{(k)}$ è facile a ricavarsi (esercizio per il lettore). Nell'esempio di cui sopra, se P(0) = p e P(1) = q = 1 - p, le probabilità sono

(0)

$$p_m^{(8)} = p^8 = p^7(1-p) = p^7(1-p) \dots p^7(1-p)^7 = (1-p)^8$$

$$(3.26)$$

94 RIDONDANZA ED EFFICIENZA - COMPRIMEREIDAII

Altrettanto facile, stante l'indipendenza e considerando la definizione logaritmica di informazione, è dimostrare che l'entropia della sorgente estesa è data da

$$H\left(S^{(k)}\right) = \underbrace{H(S) + H(S) + \dots + H(S)}_{k \text{ volte}} = k \cdot H(S)$$
(3.27)

Il concetto di estensione sarà utile nel prossimo paragrafo a proposito della codifica ottimale di una sorgente senza memoria.

3.5 Codici (ottimi) per sorgente senza memoria

A questo punto possiamo riprendere il problema della codifica efficiente di sorgente, ed esaminiamo il caso semplificato di alfabeto di codice binario (il più usato in pratica) e con un esempio pratico. Consideriamo la consueta sorgente con alfabeto quaternario $\Omega = \{a, b, c, d\}$ e con le probabilità dei simboli come di seguito:

$$P(a) = 1/2$$

$$P(b) = 1/4$$

$$P(c) = 1/8$$

$$P(d) = 1/8$$
(3.28)

(3.29)

L'entropia di questa sorgente è H(S) = -(1/2)(-1) - (1/4)(-2) - (1/8)(-3) - (1/8)(-3) = 1.75 bit/simbolo. Adottiamo il codice a prefisso (che soddisfa la disuguaglianza di Kraft)

$$\begin{array}{l} a \to 0 \\ b \to 10 \\ c \to 110 \\ d \to 111 \end{array}$$

con il che la lunghezza media delle parole di codice è

$$l_{med} = (1/2)(1) + (1/4)(2) + (1/8)(3) + (1/8)(3) = 1.75$$
 bit (3.30)

Dunque per questo codice la lunghezza delle parole, misurata in bit, uguaglia esattamente il contenuto informativo medio in bit della parola stessa (cioè l'entropia), e la capacità richiesta a un sistema di comunicazione per trasmettere l'informazione della sorgente sarà proprio quella necessaria e sufficiente. Si può intuire che non è possibile ridurre ulteriormente la lunghezza delle parole, pena la mancata codifica (e quindi eventualmente trasmissione) di una parte dell'informazione della sorgente - non per niente questo codice soddisfa la disuguaglianza di Kraft *al limite dell'* = !

Nasce quindi la congettura: *per codici binari, la lunghezza media minima delle parole di codice è non inferiore all'entropia della sorgente*. Per confermare o smentire quest'affermazione, definiamo una distribuzione fittizia di probabilità come segue:

$$q_m \stackrel{\Delta}{=} \frac{2^{-l_m}}{\sum\limits_{m=1}^{M} 2^{-l_m}} = \frac{2^{-l_m}}{\Gamma}$$
(3.31)
che risulta evidentemente normalizzata a 1 e sempre positiva. Applicando la disuguaglianza di Gibbs, si ottiene immediatamente

$$H(S) \le \sum_{m=1}^{M} p_m \log\left(\frac{1}{q_m}\right) = \sum_{m=1}^{M} p_m \left[\log\left(2^{l_m}\right) + \log(\Gamma)\right]$$
(3.32)

ovvero

$$H(S) \le \sum_{m=1}^{M} p_m l_m + \log(\Gamma) = l_{med} + \log(\Gamma)$$
(3.33)

Per la disuguaglianza di Kraft, se il codice deve essere univocamente decodificabile, allora $\Gamma \leq 1$, quindi $\log(\Gamma) \leq 0$ e quindi a maggior ragione $l_{med} \geq H(S)$ come ipotizzato. Questo risultato vale anche per codici *L*-ari, purché l'entropia della sorgente venga divisa per $\log(L)$, che è poi il contenuto informativo massimo di un singolo simbolo di codice.

Ma come possiamo raggiungere questo limite per il codice di una generica sorgente? Confrontiamo le definizioni di H(s) e di lunghezza media l_{med} di un codice:

$$H(S) = \sum_{m=1}^{M} p_m \log\left(\frac{1}{p_m}\right) , \ l_{med} = \sum_{m=1}^{M} p_m l_m$$
(3.34)

Se riuscissimo a fare $l_m = \log(1/p_m)$ avremmo risolto il problema... Naturalmente questo è impossibile perché l_m deve essere intera. Possiamo però provare a definire un codice con

$$l_m = \left\lceil \log\left(\frac{1}{p_m}\right) \right\rceil , \quad m = 1, \dots, M$$
(3.35)

approssimando per eccesso, in modo da non violare la relazione (3.33) con l'entropia, ma cercando di avere comunque una l_{med} vicina il più possibile all'entropia stessa per costruzione.

Ha senso un codice con queste lunghezze di parola? Osserviamo che, da (3.35) segue

$$\log\left(\frac{1}{p_m}\right) \le l_m < \log\left(\frac{1}{p_m}\right) + 1 \tag{3.36}$$

ovvero,

 $2^{\log(p_m)-1} < 2^{-l_m} < 2^{\log(p_m)}$

e, considerando solo i primi due termini e sommando su m

$$2^{-l_m} \le p_m \Rightarrow \sum_{m=1}^M 2^{-l_m} \le \sum_{m=1}^M p_m = 1$$
 (3.37)

Queste lunghezze soddisfano dunque la disuguaglianza di Kraft e consentono di costruire un codice a prefisso - la nostra scelta delle lunghezze di parola ha senso.

Un codice con questa proprietà è quello di Shannon-Fano. Per costruirlo, si comincia col fissare le lunghezze delle parole come in (3.35); si considera poi la lunghezza (o le lunghezze) minima(e) e si costruiscono le parole di questo gruppo come la rappresentazione in base 2 sulla(e) lunghezza(e) data - questo numero (in base 2) è la parola di codice. In quest'ambito il codice è sicuramente a prefisso perchè le parole sono tutte della stessa lunghezza. Esaurito questo gruppetto, si passa a quello di lunghezza immediatamente superiore usando come prefisso il numero che sarebbe spettato se avessimo avuto un'ulteriore parola di lunghezza inferiore (e quindi si ottiene di nuovo una codifica a prefisso), e usando i

96 RIDONDANZA ED EFFICIENZA - COMPRIMEREIDAII

restanti bit per procedere da capo nella numerazione ripartendo da 0. La disuguaglianza di Kraft garantisce che questo procedimento funziona sempre e non si resta mai senza "numero" per una qualche parola.

Esempio 3.15

Costruiamo il codice di Shannon-Fano per la sorgente a 5 simboli qui di seguito:

P(a) = 0.05 P(b) = 0.25 P(c) = 0.25 P(d) = 0.15 P(e) = 0.3(3.38)

L'entropia di questa sorgente è pari circa a 2.15 bit/simbolo. Dalla (3.35)si ha

$$l_1 = 2, l_2 = 2, l_3 = 2, l_4 = 3, l_5 = 5$$

e le parole sono

 $\begin{cases} e : l_1 = 2 \to 00 \\ b : l_2 = 2 \to 01 \\ c : l_3 = 2 \to 10 \\ d : l_4 = 3 \to 110 \\ a : l_5 = 5 \to 11100 \end{cases}$

La lunghezza media di questo codice è

 $l_{med} = 0.3 \cdot 2 + 0.25 \cdot 2 + 0.25 \cdot 2 + 0.15 \cdot 3 + 0.05 \cdot 5 = 2.3$ bit/simbolo

Quanto efficiente è veramente questo codice? Riusciamo a calcolarne in generale la prestazione in termini di l_{med} ? Di nuovo partendo dalla (3.35), ma stavolta moltiplicando tutti i membri per p_m e sommando su m (cioè in pratica calcolando i valori attesi) si ha:

$$\sum_{m=1}^{M} p_m \log\left(\frac{1}{p_m}\right) \le \sum_{m=1}^{M} p_m l_m < \sum_{m=1}^{M} p_m \left(\frac{1}{p_m}\right) + \sum_{m=1}^{M} p_m$$
(3.39)

ovvero

$$H(S) \le l_{med} < H(S) + 1$$
 (3.40)

e cioè la lunghezza media del codice di Shannon-Fano *approssima l'entropia a meno di un bit* - abbiamo ricavato un *bound* alla prestazione del codice. Per l'esempio appena visto del codice di Shannon-Fano, si ha un'entropia di sorgente pari a 2.15 bit e una lunghezza media $l_{med} = 2.3$ cifre binarie. Siamo dunque in grado di valutare l'efficienza di compressione di questo codice relativamente all'entropia.

Esiste per caso un codice migliore di quello di Shannon-Fano? Costruiamo un albero a M foglie come segue. Cominciamo con l'ordinare i simboli di sorgente a probabilità decrescente e assegnamo alle M foglie dell'albero tali probabilità. Procediamo poi ad accorpare le due probabilità più piccole facendo confluire due rami dell'albero che partono



Figura 3.9 Codifica di Huffman

da queste foglie in un unico nodo, che contrassegniamo con la probabilità somma. Otteniamo così un set di M - 1 nodi ai quali applichiamo (ricorsivamente) il medesimo procedimento per ottenere un set di M - 2 nodi e così via. Guardando l'albero, vediamo che a partire dalla sorgente M - aria originaria S, all'*i*-esimo passo di riduzione, con *i* che scorre al contrario $i = M - 1, \ldots, 1$, si viene a creare una nuova "sorgente ridotta" ad *i* simboli che possiamo chiamare S_i , e che rispetto a quella del passo precedente S_{i+1} ha un numero di simboli ridotto di 1 (dove si deve intendere che $S_M \triangleq S$): in Fig. 3.9, l'accorpamento di *a* e *b* dà il nodo a probabilità 0.2, e così via. Nel ridurre la sorgente, contrassegnamo anche i rami creati nel procedimento con i simboli 0 e 1 seguendo una convenzione che manterremo per tutta la costruzione: il ramo che esce dal nodo a probabilità più bassa viene etichettato con 1, quello che esce dal nodo a probabilità più alta con 0. Iteriamo questa procedura finché non giungiamo alla radice a probabilità unitaria ("sorgente" S_1), ottenendo l'albero di Fig. 3.9.

Fatto questo, procediamo ad etichettare i nodi dell'albero, un po' come già visto nella Fig. 3.5: partendo dalla radice, "visitiamo" l'albero, cioè procediamo (a ritroso) dalla radice verso le foglie (i nodi di partenza, cioè i simboli della sorgente). Se prendiamo un ramo verso l'alto, etichettiamo il nodo di arrivo con la parola del nodo padre aggiungendo in coda uno 0, se prendiamo il ramo verso il basso aggiungiamo 1. Quando arriviamo ad una foglia, otteniamo la parola di codice che associamo al simbolo di sorgente corrispondente.

Questo procedimento è la *codifica di Huffman*. L'albero contiene anche i codici di Huffman (generati con il medesimo criterio) per tutte le sorgenti "ridotte" S_i che man mano si sono create. L'algoritmo di codifica può essere implementato in maniera iterativa molto semplicemente, ed è quindi efficiente per applicazioni in tempo reale. Idem dicasi per il procedimento di decodifica, che può essere ridotto a una visita efficiente di un albero. Il codice di Huffman è impiegato in molti casi in cui si desidera comprimere una sorgente di informazione, ad esempio come codice cosiddetto "entropico" finale nel formato JPEG di memorizzazione delle immagini su file.

L'idea base nella costruzione del codice (a prefisso) di Huffman è la consueta: ai simboli più probabili (più frequenti) vengono associate parole di codice corte, mentre le parole più lunghe sono riservate ai simboli meno frequenti. Questa strategia permette chiaramente di minimizzare la lunghezza media delle parole. Infatti la lunghezza media del codice appena trovato è pari a

$$l_{med} = 0.3 \cdot 2 + 0.25 \cdot 2 + 0.25 \cdot 2 + 0.15 \cdot 3 + 0.05 \cdot 3 = 2.2 \tag{3.41}$$

molto vicina al limite teorico inferiore (e migliore di quella di Shannon-Fano). Possiamo quantificare l'efficienza del codice binario con il parametro

$$\eta_c \stackrel{\Delta}{=} H(S)/l_{med}$$
 (3.42)

che è tanto più vicino a 1 quanto migliore è il codice, e rappresenta in un certo senso l'efficienza nella compressione della sorgente. Il lettore discuta il caso particolare in cui la sorgente è tale per cui si trova una codifica avente esattamente $l_m = -\log(p_m)$ e dia un esempio di tale sorgente con relativa codifica.

La proprietà saliente del codice di Huffman è che rappresenta, data la sorgente, il codice senza perdita che ha, tra tutti i possbili, la l_{med} minima - riportiamo per completezza nell'Appendice A la dimostrazione di questa proprietà, dimostrazione molto elegante ma un po' lunga che appesantirebbe lo sviluppo del nostro discorso.

Nella pratica, vengono utilizzati anche codici differenti da quelli appena introdotti, che però hanno caratteristiche particolari. Ad esempio, nella codifica senza perdita audio .flac, dopo uno stadio differenziale con predizione del tutto analogo a quello visto nella sez. 3.1.3, il residuo e[n] viene codificato senza perdita (anziché quantizzato con perdita come nell'ADPCM) con il codice di Golomb-Rice che non richiede tabelle (e quindi è vantaggioso per sorgenti con M grande), che ha un'implementazione algoritmica molto semplice, ma che risulta ottimale solo per particolari andamenti della legge di probabilità dei simboli di sorgente (geometrica), che si incontrano appunto solo in certe applicazioni.

3.5.1 Codifica della sorgente estesa

Già dal nostro esempio risulta che il codice di Huffman, benché ottimale, non raggiunge in generale l'entropia della sorgente. Inoltre, dal punto di vista pratico, sia Huffman che Shannon-Fano sono codici che ben si adattano a una sorgente a molti livelli, ma sono incapaci di codificare una sorgente binaria. Discutiamo qui di seguito come risolvere questi due problemi con uno steso approccio, legato alla *codifica della sorgente estesa*.

Immaginiamo di avere a che fare con una sorgente, anche binaria, estesa all'ordine k e costruiamo un (nuovo) codice di Shannon-Fano per questa sorgente. Scegliendo la lunghezza delle M^k parole secondo la (3.35) e applicando la (3.40) a questa nuova sorgente si ha

$$H(S^{(k)}) \le l_{med}{}^{(k)} < H(S^{(k)}) + 1 \quad \Rightarrow \quad k \cdot H(S) \le l_{med}{}^{(k)} < k \cdot H(S) + 1 \quad (3.43)$$

ove evidentemente $l_{med}^{(k)}$ è la lunghezza media della codifica della sorgente estesa, espressa in (c.b)·(k simboli). Dividendo per k si ha infine

$$H(S) \le l_{med}{}^{(k)}/k < H(S) + 1/k$$
 (3.44)

Poiché adesso ogni parola di codice è associata ad un blocco di k simboli di sorgente, la quantità $l_{med}^{(k)}/k$ rappresenta una sorta di "lunghezza media equivalente" per singolo simbolo della sorgente originaria non estesa, ed è analoga alla l_{med} di quest'ultima (anche se numericamente in generale diversa). Se k è sufficientemente grande, si vede dalla (3.44) che $l_{med}^{(k)}/k$ si avvicina all'entropia della sorgente con accuratezza arbitraria. È chiaro che se questa proprietà vale per il codice di Shannon-Fano, a maggior ragione deve valere per il codice di Huffman che, per qualunque ordine dell'estensione di sorgente, garantisce comunque una l_{med} inferiore a quella di Shannon-Fano.

La ricetta per codificare efficientemente è dunque: codifica sorgenti più estese possibile e giungerai arbitrariamente vicino alla minima lunghezza. Questo enunciato, in forma piuttosto volgarizzata, rappresenta il cosiddetto primo teorema di Shannon della codifica o teorema della codifica priva di disturbo. Questo approccio è utilizzato anche in pratica: nel compressore LZW77 che vedremo in seguito, la stringa (il file) binario da comprimere viene considerato *byte per byte*, adottando cioè una estensione di sorgente con K = 8.

Esempio 3.16

Partiamo da una sorgente binaria con $p_1 = p = 0.1$ e $p_2 = q = 1 - p = 0.9$. L'entropia di questa sorgente è $H(S) \simeq 0,469$ bit/simbolo. Nella Fig. 3.10 possiamo vedere l'evoluzione della quantità $l_{med}^{(k)}/k$ (nonché le altre quantità di sorgente e di codice) per il codice di Shannon-Fano al variare di k. Come si nota, $l_{med}^{(k)}$ si avvicina sempre di più ad $H(S^{(k)})$ come previsto dalla (3.43).

Il primo teorema della codifica cozza purtroppo con la necessità di tenere basso il ritardo di codifica/decodifica e, soprattutto, di tenere bassa la complessità delle stesse operazioni. Estendere una sorgente all'ordine k significa far crescere esponenzialmente il numero di parole di codice distinte e quindi la complessità del codec (codificatore/decodificatore).

3.5.2 Codifica Aritmetica

Come esempio di ulteriore algoritmo (asintoticamente) ottimale per la compressione di sorgenti senza memoria, vediamo brevemente nei suoi principi fondamentali la codifica aritmetica (arithmetic coding). Vi sono molte varianti di questa tecnica, alcune delle quali anche coperte da brevetto, ma non vi è una implementazione di riferimento e/o standard. Il vantaggio rispetto al codice di Huffman è che può (più) facilmente essere reso *adattativo*: l'algoritmo può partire a codificare la stringa di sorgente senza avere bisogno delle informazioni sulla legge di probabilità della sorgente stessa.

Ma vediamo una cosa per volta, partendo dalla versione non adattativa, per la quale cioè si presuppongono note le probabilità a-priori di sorgente p_m . Il principio base della codifica è semplice: si cerca di associare alla stringa di sorgente un numero decimale in base 10 β costruito opportunamente, la rappresentazione binaria del quale costituirà poi la codifica binaria della stringa di sorgente originaria. Vediamo la costruzione di questo numero reale attraverso un esempio semplice. Consideriamo una sorgente ternaria senza memoria con alfabeto $\Omega \equiv \{s_0 = a, s_1 = b, s_2 = c\}$ e con distribuzione di probabilità $p_0 = 0.2, p_1 = 0.5, p_2 = 0.3$. Rappresentiamo poi (in verticale) il segmento dell'asse reale (0,1) e stacchiamo su questo tre intervalli adiacenti la cui ampiezza rappresenta le probabilità della sorgente data, come si vede in Fig. 3.11 (a). I punti intermedi di questa suddivisione rappresentano i valori assunti dalla funzione distribuzione di probabilità della sorgente data, funzione che, essendo la sorgente discreta, ha un andamento "a gradinata", cioè costante a tratti e non decrescente. Supponiamo adesso che la stringa da codificare sia bac. Poiché il primo simbolo è una b, prendiamo in considerazione l'intervallo centrale del segmento (0,1)di ampiezza $\Delta_1 = p_b = 0.5$ (che corrisponde appunto al simbolo b perché ne rappresenta la probabilità) e "zoomiamo" questo intervallo risuddividendolo in M=3 parti di ampiezza ancora proporzionale alle probabilità di sorgente, come mostrato in Fig. 3.11 (b). Ripetiamo poi il procedimento: al secondo step cioè zoomiamo e consideriamo l'intervallo inferiore di ampiezza $\Delta_2 = p_b p_a = 0.1$ poiché il secondo simbolo di sorgente è a, e concludiamo la codifica, dopo un'ulteriore operazione di zoom e suddivisione, considerando l'intervallo superiore di ampiezza $\Delta_3 = p_b p_a p_c = 0.03$ perché il simbolo finale è c, ottenendo la

100 RIDONDANZA ED EFFICIENZA - COMPRIMEREIDAII



Figura 3.10 Codice di Shannon-Fano per una sorgente binaria estesa fino all'ordine k = 5



Figura 3.11 Codifica Aritmetica - primo step





situazione in Fig. 3.12. Come si vede, abbiamo identificato un "sub-intervallo finale" dell'intervallo (0,1) di partenza, in questo caso (0.27,0.3). Il numero reale da associare alla stringa di sorgente è adesso il punto medio dell'intervallo finale: $\beta = 0.285$.

Dobbiamo adesso capire come trovare la codifica binaria di β . La questione è molto semplice: si devono prendere le prime *L* cifre significative (dopo la virgola) della rappresentazione binaria di β , ove $L = \lceil \log_2(1/\Delta_3) \rceil + 1$. Nel nostro caso L = 7, $(\beta)_2 = 0.010010001...$ e la codifica è 0100100. Il decodificatore procede adesso in maniera inversa. Dalla stringa codificata può ricostruire un numero reale decimale $\overline{\beta}$ che rappresenta il "troncamento" di β come sopra. Nel nostro caso, convertendo la stringa 0100100 si ha $\overline{\beta} = (0.0100100)_2 = (1/4 + 1/32)_{10} = 0.28125$. E' facile dimostrare (per esercizio) che anche se $\overline{\beta} \neq \beta$, avendo troncato su L cifre binarie, comunque $\overline{\beta}$ è compreso all'interno dell'intervallo finale della procedura di codifica, e questo è il punto importante. Il valore $\overline{\beta}$ infatti rappresenta il "puntatore" alla stringa di sorgente, e per questo la decodifica è immediata: il valore 0.28125 al primo step punta all'intervallo corrispondente a *b* (vedi Fig. 3.13), nel secondo step, dopo la prima zoomata che il decodificatore deve effettuare, punta al primo intervallo corrispondente ad *a* e ovviamente nel terzo step, dopo la seconda zoomata, punta all'intervallo superiore, corrispondente a *c* - fine della decodifica.

Come per tutti gli algoritmi di codifica di sorgente, dobbiamo adesso discutere il fattore di compressione e la velocità di elaborazione. Purtroppo la procedura descritta, quando viene implementata per lavorare in tempo reale, risulta piuttosto lenta perché non si basa su di una tabella, ma sull'esecuzione in tempo reale di un algoritmo. Detto questo, cerchiamo di valutare il fattore di compressione del codice. Per far questo, ragioneremo in senso "ergodico", cioè valuteremo la l_{med} in modo leggermente diverso da quanto visto finora. Se pensiamo di codificare una stringa molto lunga di sorgente, composta da N simboli consecutivi con $N \to \infty$, allora è chiaro che per l'ergodicità della sorgente, la lunghezza media del

37



Figura 3.13 Decodifica Aritmetica

codice in termini di cifre binarie/simbolo si può valutare come $l_{med} = \lim_{N \to \infty} (L/N)$, cioè il numero totale L di cifre binarie che codifica tutta la stringa, fratto il numero di simboli di sorgente nella stringa.

Nel nostro codice abbiamo visto che l'ampiezza finale Δ_N del sub-intervallo che codifica la stringa è $\Delta_N = \prod_{n=1}^N p_{m[n]}$ ove $m[n] \in \{0, ..., M-1\}$ è l'indice dell'n-esimo simbolo della stringa di sorgente. Consideriamo ora la quantità

$$\log(\Delta_N) = \sum_{n=1}^{N} \log(p_{m[n]}) \tag{3.45}$$

Possiamo riorganizzare la sommatoria raggruppando tutti i termini che contengono p_0 e che saranno in numero pari a un certo N_0 , tutti quelli, in numero N_1 contenenti p_1, \ldots , tutti quelli, in numero N_{M-1} contenenti p_{M-1} , ottenendo

$$\log(\Delta_N) = \sum_{n=1}^{N} \log(p_{m[n]}) = N_0 \log(p_0) + N_1 \log(p_1) + \dots + N_{M-1} \log(p_{M-1})$$
(3.46)

Se N è molto grande, approssimando le probabilità con le frequenze relative otteniamo facilmente

$$\log\left(\frac{1}{\Delta_N}\right) = N \sum_{m=0}^{M-1} \frac{N_m}{N} \log\left(\frac{1}{p_m}\right) \cong N \sum_{m=0}^{M-1} p_m \log\left(\frac{1}{p_m}\right) = N \cdot H(S) \quad (3.47)$$

A questo punto, la lunghezza media del codice per simbolo di sorgente si può valutare come

$$l_{med} = \lim_{N \to \infty} \left(\frac{L}{N}\right) = \lim_{N \to \infty} \left(\frac{\lceil \log_2(1/\Delta_N) \rceil + 1}{N}\right) = \lim_{N \to \infty} \left(\frac{\lceil NH(S) \rceil}{N}\right) = H(S)$$
(3.48)

Dunque, asintoticamente, il codice raggiunge l'entropia della sorgente ! Per lunghezza finita, resta un termine di inefficienza come nel codice di Shannon-Fano, che però tende a svanire per stringhe di sorgente lunghe.

Come abbiamo già accennato, e al contrario dei codici di Huffman e di Shannon-Fano, non è necessario costruire una tabella di codice per effettuare la codifica: il codice è algoritmico nel senso che la parola di codice viene calcolata a tempo di esecuzione senza consultare alcuna tabella. Resterebbe il problema della latenza perché nella nostra descrizione la stringa di cifre binarie codificate viene prodotta quando *tutta* la stringa di sorgente è stata elaborata In realtà, le implementazioni pratiche della codifica aritmetica producono le cifre binarie codificate ad ogni "zoomata", cioè dopo la ricezione di ogni nuovo simbolo, realizzando una codifica in tempo reale a bassa complessità.



Figura 3.14 Codifica Aritmetica Adattativa

3.5.3 Codifica Aritmetica Adattativa

Come è possibile rendere il codice aritmetico adattativo, cioè funzionante anche in assenza di informazioni a-priori sulla distribuzione della sorgente? La risposta è semplice: partendo con l'ipotesi di una sorgente massimamente incerta (cioè con simboli equiprobabili) e aggiornando man-mano le probabilità di presentazione dei simboli di sorgenti con le quali effettuare lo step di codifica successivo.

Ancora nel caso di una sorgente ternaria, facciamo lo stesso esempio del caso precedente di stringa *bac* da codificare. La Fig. 3.14 (a) mostra lo stato del codificatore al primo step e la (b) mostra ciò che succede al secondo step. Come si nota, dal primo al secondo step le probabilità presunte dei simboli sono cambiate tenendo conto delle nuove frequenze di presentazione: al passo iniziale il codificatore parte presumendo che i simboli si siano presentati un numero di volte $N_a = N_b = N_c = 1$ per un totale di $N_{tot} = 3$ estrazioni, in modo che risultino equiprobabili. Nei passi successivi, i valori di N_a, N_b, N_c e N_{tot} vengono aggiornati secondo le effettive presentazioni passo-passo dei vari simboli: dopo il primo step $N_a = N_c = 1$, $N_b = 2$ e naturalmente $N_{tot} = 4$: questo giustifica i nuovi valori 1/4,2/4,1/4 delle probabilità. La situazione finale del codificatore è quella di Fig. 3.15, dalla quale segue che $\beta = 49/120$.

Trascurando il troncamento a β , è chiaro il funzionamento del decodificatore: esso deve partire con la suddivisione equiprobabile di Fig. 3.14 (a), sulla quale $\beta = 49/120$ identifica chiaramente b; il decodificatore stesso, avendo osservato b aggiorna di conseguenza le probabilità dei simboli e compie il primo step come in Fig. 3.15 (b). In questo nuovo stato, il valore 49/120 porta chiaramente alla decodifica di a e al conseguente aggiornamento delle probabilità come in Fig. 3.15 (b); l'ultimo step di decodifica con 49/120 è immediato.

Naturalmente il codificatore adattativo deve superare una certa fase iniziale di "training" nella quale il rapporto di compressione non è ottimale - questo accade finché le probabilità di sorgente stimate non si sono stabilizzate attorno ai valori effettivi, dopodiché il grado di compressione comincia ad essere vicino all'ideale.

3.6 Sorgenti di informazione con memoria

Già nell'esempio della codifica vocale, abbiamo visto un esempio di sorgente con memoria. Il segnale campionato a 4 kHz e rappresentato su 8 bit si può considerare come prodotto da una sorgente con M=256 simboli (i possibili valori degli 8 bit) in cui i vari simboli non sono indipendenti. Se infatti la sorgente avesse queste caratteristiche, il segnale sarebbe rumore bianco e non un segnale vocale !



Figura 3.15 Decodifica Aritmetica Adattativa



Figura 3.16 Esempi di sorgente di informazione: (a) con memoria, (b) senza memoria

Un altro esempio è dato dalla codifica di un'immagine. Consideriamo l'immagine in Fig. 3.16 (a) della modella *Lena* universalmente adottata per esempi di elaborazione delle immagini. Essa è costituita da 256x256 pixel ciascuno su 256 livelli di grigio. Supponendo di scandire l'immagine partendo dal pixel in alto a sinistra e procedendo da sinistra a destra e dall'alto verso il basso con "ritorno a capo" alla fine di ogni riga, si ottiene una stringa di 65536 simboli appartenenti ad un alfabeto a M = 256 valori che rappresenta la nostra sorgente di informazione S[n]. Si nota che, specialmente nelle zone di sfondo, il valore di due pixel adiacenti è in generale molto simile, e quindi ci sarà un'alta correlazione statistica tra i simboli della sorgente. La sorgente è anche in questo caso *con memoria* e su questa caratteristica si può far leva per tentare di codificarla in modo ottimale. Per confronto, vediamo in Fig. 3.16 (b) l'aspetto di una sorgente equivalente senza memoria: tutt'altra cosa! Osserviamo che, nel caso della codifica dell'immagine, $H_{max}=8$ bit/simbolo. Le codifiche di immagine più efficienti realizzano schemi di codice con lunghezza media l_{med}

La trattazione del caso generale di sorgente con memoria è complicato, perché la caratterizzazione completa di tal sorgente richiederebbe in teoria la conoscenza della famiglia delle *probabilità congiunte*

$$p_{m_1,m_2,\dots,m_N} \stackrel{\Delta}{=} \Pr\{S[n] = s_{m_1}, S[n-1] = s_{m_2},\dots,S[n-N+1] = s_{m_N}\} \quad (3.49)$$

per qualunque valore di N. Alternativamente, sarebbe necessario conoscere la descrizione di primo ordine, come per la sorgente senza memoria, e in più la famiglia delle *probabilità*

condizionate

$$\Pr\{S[n] = s_{m_1} | S[n-1] = s_{m_2}, \dots, S[n-N+1] = s_{m_N}\}$$
(3.50)

sempre per tutti gli N.

Esaminiamo meglio questo concetto di informazione condizionata: qual è l'informazione portata all'istante n dal simbolo s_i condizionatamente all'aver osservato il simbolo s_k all'istante n-1 ? Qual è cioè l'informazione dell'evento $\{S[n] = s_i | S[n-1] = s_k\}$? La risposta è semplice: abbreviando la notazione,

$$\ln(s_i|s_k) = \log\left(\frac{1}{P(s_i|s_k)}\right) \tag{3.51}$$

Questo porta con se' anche la definizione di informazione congiunta di due simboli:

$$\ln(s_i, s_k) = \log\left(\frac{1}{P(s_i, s_k)}\right) = \log\left(\frac{1}{P(s_i|s_k)P(s_k)}\right)$$
(3.52)

per cui si può scrivere la relazione interessante

$$\ln\left(s_{i}|s_{k}\right) = \ln\left(s_{i}, s_{k}\right) - \ln\left(s_{k}\right)$$

$$(3.53)$$

L'informazione condizionata è pari all'informazione congiunta dei due simboli diminuita dell'informazione portata dal simbolo precedente s_k che, essendo stato osservato ed essendo quindi noto con certezza, non è più (a posteriori) portatore di alcuna informazione. Se la sorgente è senza memoria, i simboli sono indipendenti e la probabilità condizionata $P(s_i|s_k)$ è pari alla marginale $P(s_i)$. In questo caso, l'informazione condizionata è pari alla In (s_i) e si ritrova la relazione familiare che, per eventi indipendenti, l'informazione congiunta è pari alla somma delle informazioni marginali.

Proseguiamo su questa strada e vediamo che succede se consideriamo le informazioni medie nella (3.53), cioè mediate sulle probabilità congiunte $p(s_i, s_k)$ del secondo ordine della sorgente. Si ottiene:

$$\sum_{i=1}^{M} \sum_{k=1}^{M} P(s_i, s_k) \log\left(\frac{P(s_k)}{P(s_i, s_k)}\right) = \sum_{i=1}^{M} \sum_{k=1}^{M} P(s_i, s_k) \log\left(\frac{1}{P(s_i, s_k)}\right) - \sum_{i=1}^{M} \sum_{k=1}^{M} P(s_i, s_k) \log\left(\frac{1}{P(s_k)}\right)$$
(3.54)

A secondo membro si riconoscono facilmente $H(S[n], S[n-1]) \in -H(S[n-1])$: la sommatoria su *i* del secondo termine marginalizza infatti la probabilità congiunta e dà come risultato $P(s_k)$. Lavoriamo invece il primo membro:

$$\sum_{i=1}^{M} \sum_{k=1}^{M} P(s_i, s_k) \log\left(\frac{P(s_k)}{P(s_i, s_k)}\right)$$
$$= \sum_{i=1}^{M} \sum_{k=1}^{M} P(s_i, s_k) \log\left(\frac{P(s_k)P(s_i)}{P(s_i, s_k)}\right) + \sum_{i=1}^{M} \sum_{k=1}^{M} P(s_i, s_k) \log\left(\frac{1}{P(s_i)}\right)$$
(3.55)

Come sopra, il secondo termine a secondo membro è semplicemente H(S[n]), e quindi si può scrivere

$$H\left(S[n], S[n-1]\right)$$

106 RIDONDANZA ED EFFICIENZA - COMPRIMEREIDATI

$$= H(S[n]) + H(S[n-1]) - \sum_{i=1}^{M} \sum_{k=1}^{M} P(s_i, s_k) \log\left(\frac{P(s_i, s_k)}{P(s_k)P(s_i)}\right)$$
(3.56)

Il termine aggiuntivo prende il nome di *informazione mutua* tra S[n] e S[n-1] e misura chiaramente il grado di (in)dipendenza tra i due valori della sorgente. E' facile vedere che se la sorgente è senza memoria si ha banalmente H(S[n], S[n-1]) = H(S[n]) + H(S[n-1]), altrimenti l'entropia congiunta è sempre minore della somma delle marginali perché l'informazione mutua

$$I(S[n], S[n-1]) \stackrel{\Delta}{=} \sum_{i=1}^{M} \sum_{k=1}^{M} P(s_i, s_k) \log\left(\frac{P(s_i, s_k)}{P(s_k)P(s_i)}\right)$$
(3.57)

è sempre non negativa conseguentemente alla diseguaglianza di Gibbs (3.19). Questo implica che una stringa di una sorgente con memoria porta in generale meno informazione di una a pari lunghezza di una sorgente senza memoria con le stesse statistiche marginali; la sorgente con memoria è quindi "più comprimibile" di una senza memoria .

Il calcolo generale dell'entropia di una sorgente con memoria di lunghezza arbitraria, e quindi la caratterizzazione del suo contenuto informativo, è semplice in principio ma estremamente complessa nei casi pratici. Limitandoci a sorgenti stazionarie, ciò che si deve fare è considerare l'estensione k-esima $S^{(k)}$ della sorgente e calcolare l'entropia di questa:

$$H(S^{(k)}) = H(S[n], S[n-1], ..., S[n-k+1]) = \sum_{m_1, m_2, ..., m_k}^{M} p_{m_1, m_2, ..., m_k} \log\left(\frac{1}{p_{m_1, m_2, ..., m_k}}\right)$$
(3.58)

Dividendo per la lunghezza di estensione k si ottiene l'entropia per simbolo equivalente (diciamo commensurabile) a quella di una sorgente a singoli simboli; l'entropia della sorgente originaria non è altro che il limite per $k \to \infty$ di questa entropia per simbolo:

$$H(S) \stackrel{\Delta}{=} \lim_{k \to \infty} \frac{H(S^{(k)})}{k}$$
(3.59)

Naturalmente, se la sorgente è senza memoria, $H(S^{(k)}) = kH(S)$ e si ricade nella definizione già nota.

3.6.1 Sorgenti Markoviane

Un caso particolarmente utile e fortunato di sorgente con memoria è quello in cui la sorgente viene modellata come una catena di Markov, cioè un processo Markoviano di ordine 1, in cui la conoscenza di tutta la "storia" della sorgente equivale alla conoscenza del solo simbolo immediatamente precedente quello attuale. In questo caso, condizionare a valori della sorgente più lontani di un passo è superfluo, e la catena è quindi caratterizzata dalla matrice delle probabilità di transizione¹

$$\mathbf{\Pi} = \{p_{ik}\} \quad , \quad p_{ik} \stackrel{\Delta}{=} \Pr\{S[n] = s_i | S[n-1] = s_k\}$$
(3.60)

¹La catena deve essere ergodica cioè non possono esistere stati del processo che, se raggiunti, impediscono il successivo raggiungimento di qualche altro stato. Questo garantisce che la catena tocca tutti quanti gli stati permessi e si stabilizza su di una distribuzione di probabilità asintotica (vedi testo)

Infatti le statistiche congiunte di qualunque ordine possono essere ricavate dalla marginale di ordine 1 e dalla $Pr\{S[n] = s_i | S[n-1] = s_k\}$ (brevemente, $p(s_i | s_k)$:

$$p(s_{i_1}, s_{i_2}, \dots, s_{i_N}) = p(s_{i_1}|s_{i_2})P(s_{i_2}|s_{i_3})\dots p(s_{i_{N-1}}|s_{i_N})p(s_N)$$
(3.61)

Già nella notazione si vede che abbiamo fatto l'ipotesi di probabilità di transizione non dipendenti dal tempo n, e cioè di catena omogenea o stazionaria.

Dalla matrice di transizione si possono calcolare le probabilità cosiddette di occupazione asintotica dei singoli stati, cioè le probabilità di presentazione dei vari simboli dopo che la catena si è "messa in moto" da molto tempo: raccogliendo queste probabilità nel vettore $\mathbf{p} \stackrel{\triangle}{=} [p_1, p_2, ..., p_M]^T$ si deve avere

$$\mathbf{\Pi p} = \mathbf{p} \quad \text{con il vincolo} \quad \sum_{m=1}^{M} p_i = 1 \tag{3.62}$$

Questo garantisce infatti che la distribuzione di probabilità degli stati resta costante nel tempo, esaurita una prima fase di transitorio dipendente dallo stato iniziale.

Sorge adesso il problema generale di capire se e come è possibile codificare una sorgente con memoria in maniera più efficiente di una senza memoria. Procediamo come di consueto attraverso un esempio. Consideriamo una sorgente ternaria di Markov con alfabeto $\{a, b, c\}$ e matrice di transizione

$$\mathbf{\Pi} = \begin{bmatrix} 1/3 & 1/4 & 1/4 \\ 1/3 & 1/2 & 1/4 \\ 1/3 & 1/4 & 1/2 \end{bmatrix}$$
(3.63)

Le probabilità di transizione si calcolano risolvendo il sistema

$$\begin{cases} \frac{1}{3}p_{a} + \frac{1}{4}p_{b} + \frac{1}{4}p_{c} = p_{a} \\ \frac{1}{3}p_{a} + \frac{1}{2}p_{b} + \frac{1}{4}p_{c} = p_{b} \\ \frac{1}{3}p_{a} + \frac{1}{4}p_{b} + \frac{1}{2}p_{c} = p_{c} \\ p_{a} + p_{b} + p_{c} = 1 \end{cases} \Rightarrow \begin{cases} -\frac{2}{3}p_{a} + \frac{1}{4}p_{b} + \frac{1}{4}p_{c} = 0 \\ \frac{1}{3}p_{a} - \frac{1}{2}p_{b} + \frac{1}{4}p_{c} = 0 \\ p_{a} + p_{b} + p_{c} = 1 \end{cases}$$
(3.64)

che, come indicato, non è insolubile in quanto la terza equazione è combinazione delle prime due e può essere eliminata. La soluzione è

$$p_a = \frac{3}{11} , \ p_b = p_c = \frac{4}{11}$$
 (3.65)

Forti di questo risultato, possiamo usare questa strategia di codifica: se la sorgente è nello stato a, per codificare il simbolo emesso usiamo un codice di Huffman costruito sulla base delle probabilità di transizione dallo stato a, che rappresentano le probabilità di emissione di un simbolo condizionato all'essere la sorgente in a. Idem dicasi per gli stati b e c. Riassumendo questa situazione stato per stato si ha:

Stato a:

$$P(a|a) = 1/3, P(b|a) = 1/3, P(c|a) = 1/3 \quad \text{Codice: } b \to 10 \quad l^a_{med} = 5/3 \text{ bit}$$

$$c \to 11$$

Stato b:

108 RIDONDANZA ED EFFICIENZA - COMPRIMEREIDAII

$$P(a|b) = 1/4, P(b|b) = 1/2, P(c|b) = 1/4$$
 Codice: $b \to 0$ $l_{med}^b = 3/2$ bit $c \to 11$

Stato c:

$$P(a|c) = 1/4, P(b|a) = 1/4, P(c|a) = 1/2$$
 Codice: $b \to 11$ $l_{med}^a = 3/2$ bit $c \to 0$

La lunghezza media totale si può calcolare come "media delle lunghezze medie condizionate" usando le probabilità asintotiche degli stati, cioè

$$l_{med} = p_a l_{med}^a + p_b l_{med}^b + p_c l_{med}^c = 17/11 \cong 1.5454 \ bit \tag{3.66}$$

Paragoniamo questa lunghezza con quella minima (cioè l'entropia) ottenibile da una sorgente senza memoria S_{sm} avente probabilità dei simboli pari alle probabilità asintotiche della sorgente con memoria data (la cosiddetta sorgente *aggiunta* di quella data):

$$H(S_{sm}) = p_a \log\left(\frac{1}{p_a}\right) + p_b \log\left(\frac{1}{p_b}\right) + p_c \log\left(\frac{1}{p_c}\right) \approx 1.5726$$
(3.67)

che è maggiore della lunghezza media (certamente non minima) del semplice schema di codifica adottato ! Come già sapevamo, la conclusione è che le sorgenti con memoria sono in generale più comprimibili di quelle senza memoria.

Dobbiamo quindi cercare di calcolare l'entropia di una sorgente con memoria S_{cm} per capire fino a che punto possiamo comprimerla. L'entropia di S_{cm} condizionata a un particolare stato è presto ricavata come media dell'informazione condizionata già introdotta:

$$H(S_{cm}|s_k) = \sum_{m=1}^{M} p(s_m|s_k) \ln(s_m|s_k) = \sum_{m=1}^{M} p(s_m|s_k) \log\left(\frac{1}{p(s_m|s_k)}\right)$$
(3.68)

Anche qui, come per le lunghezze medie del codice, per ottenere l'entropia "globale" della sorgente dobbiamo mediare le entropie condizionate sulla distribuzione (asintotica) di probabilità degli stati:

$$H(S_{cm}) = \sum_{k=1}^{M} p(s_k) H(S_{cm}|s_k) = \sum_{k=1}^{M} p(s_k) \sum_{m=1}^{M} p(s_m|s_k) \log\left(\frac{1}{p(s_m|s_k)}\right)$$
$$= \sum_{k=1}^{M} \sum_{m=1}^{M} p(s_k) p(s_m|s_k) \log\left(\frac{1}{p(s_m|s_k)}\right) = \sum_{k=1}^{M} \sum_{m=1}^{M} p(s_m,s_k) \log\left(\frac{1}{p(s_m|s_k)}\right)$$
(3.69)

Secondo questa definizione, l'entropia della sorgente dell'esempio precedente risulta pari a

$$H(\Omega_m) = 1 + \frac{1}{3}\log(\frac{1}{3}) \cong 1.5283 \text{ bit/simbolo}$$
 (3.70)

La codifica di Huffman "per stato" giunge dunque molto vicino al minimo teorico. Ovviamente si può dimostrare che l'entropia della sorgente di Markov con memoria è sempre minore della cosiddetta sorgente "adiacente", cioè della sorgente senza memoria con probabilità dei simboli pari alla probabilità asintotica della sorgente di Markov, e che è già stata introdotta nell'esempio di codifica.



Figura 3.17 Scansione di un'immagine B/N

Esempio 3.17

Torniamo all'esempio della scansione di un'immagine. Abbiamo modellato la scansione di un testo stampato come una sorgente senza memoria sbilanciata, cioè con molti più simboli 0 rispetto ai simboli 1: $p_1 = 0.9$, $p_2 = 0.1$. Se però dobbiamo passare allo scanner una qualunque immagine B/N, (Fig. 3.17) immaginando ancora di rappresentarla con 1 c.b./pixel, in generale il numero di 1 e di 0 stavolta sarà sostanzialmente bilanciato. È compribile questa sorgente? Come già osservato, un'immagine è in generale una sorgente con memoria, e infatti le stringhe risulterebbero del tipo

cioè con lunghe stringhe di 1 e di 0 (aree di colore omogeneo sull'immagine), ma con un numero globalmente identico di 1 e 0.

Il modello corretto è stavolta una sorgente di Markov, con probabilità di transizione del tipo

$$p(0|0) = p(1|1) = 0.9$$
, $p(0|1) = p(1|0) = 0.1$ (3.71)

cioè la tendenza della sorgente è quella di "rimanere" in uno stato una volta raggiunto e cambiare stato sporadicamente - con questo si producono le lunghe stringhe di simboli identici. La sorgente è poi simmetrica, cioè le stringhe di 1 e 0 sono mediamente della stessa lunghezza (la matrice di transizione è simmetrica).

Calcoliamo le probabilità asintotiche:

$$\begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} , \quad p_1 + p_2 = 1$$
(3.72)

prendendo la prima equazione dall'equazione asintotica e sostituendo $p_2 = 1 - p_1$ si ha $0.9p_1 + 0.1(1 - p_1) = p_1$ da cui facilmente $p_1 = p_2 = 0.5$ come ci si attendeva. L'entropia di sorgente è infine

$$H_{\infty}(S) = p_1 H(S|0) + p_2 H(S|1) = 0.5\eta(0.1) + 0.5\eta(0.1) = \eta(0.1)$$
(3.73)

Nonostante le probabilità marginali di 0 e 1 siano 0.5, la sorgente porta assai meno informazione rispetto alla $H_{max} = 1$ come già sappiamo.

110 RIDONDANZA ED EFFICIENZA - COMPRIMEREIDAII

Non esiste una strategia generale ottima di codifica delle sorgenti con memoria. Una possibilità è quella di considerare l'estensione k-esima della sorgente, purché naturalmente se ne conosca la descrizione statistica, cioè le probabilità congiunte di ordine k dei nuovi simboli "estesi". Per una sorgente a catena di Markov, questa descrizione è sempre calcolabile attraverso la matrice di transizione e la "regola della catena". La nuova sorgente può poi essere approssimata come senza memoria e può essere codificata ad es. con un codice di Huffman. Questa procedura, come abbiamo già visto nel caso delle sorgenti senza memoria, è asintoticamente ottima (cioè ottima per $k \to \infty$), ma comporta un aumento esponenziale della complessità.

Un'altra strategia generale di compressione (chiamata codifica Δ) ricalca lo schema del codificatore ADPCM del paragrafo 1. Se si riesce a costruire, sfruttando le proprietà di memoria della sorgente, un buon predittore $\hat{S}[n]$ del simbolo corrente S[n], conviene considerare il segnale differenza o segnale errore di predizione $E[n] = S[n] - \hat{S}[n]$. Pensando a sorgenti binarie, se il predittore ha buone caratteristiche, E[n] sarà costituito da lunghe sequenze di 0 intercalati da qualche 1 qua e là, e può quindi essere efficientemente codificato con un codice a lunghezza delle ripetizioni. La strategia Δ è particolarmente utile per sorgenti con memoria "lunga", cioè non modellabili come semplici catene di Markov.

Esempio 3.18

Riprendiamo la sorgente dell'Esempio 17 e cerchiamo di comprimerla con la strategia " Δ ". Visto che la sorgente tende a "permanere" su di un livello, usiamo come predittore per il valore successivo, semplicemente il valore precedente: $\hat{S}[n] = S[n-1]$, adottando una codifica banalmente differenziale. Qual è la statistica di $E[n] = S[n] - \hat{S}[n]$? Vediamo:

$$\Pr \{ E[n] = 0 \} = \Pr \{ S[n] = S[n-1] \} =$$

 $= \Pr \{S[n] = 0 | S[n-1] = 0\} p_1 + \Pr \{S[n] = 1 | S[n-1] = 1\} p_2 = 0.9$

e quindi E[n] risulta come ci attendevamo sbilanciato verso lo 0, dando origine a stringhe (più) facilmente comprimibili.

Vedremo in seguito altre strategie di "preprocessing" della sorgente per renderla facilmente comprimibile, come la trasformata di Burroughs-Wheeler.

3.7 Codifica Universale (Adattativa) per sorgenti con memoria

Tutti gli schemi di codifica fin qui presentati soffrono di un difetto fondamentale: sono basati su una conoscenza a priori delle caratteristiche di sorgente. Il codice di Huffman per sorgenti senza memoria è basato sulla conoscenza delle probabilità dei simboli, e quello "per stato" per sorgenti Markoviane necessita anche della conoscenza delle probabilità di transizione. Sorgenti a memoria più lunga di quelle Markoviane necessiterebbero addirittura della conoscenza di probabilità congiunte o condizionate di ordine superiore per comprimere al meglio. Perfino la codifica delle ripetizioni RLE non può prescindere dalla conoscenza delle ripetizioni stesse.

Viceversa, nelle applicazioni pratiche la statistica della sorgente assai spesso non è nota o è variabile nel tempo. È necessario quindi uno schema di codifica in grado di funzionare con qualunque sorgente, cioè *universale*. Poiché una tale codifica è in grado di adattarsi a

sorgenti con varie caratteristiche, potremmo chiamarla (ma il termine in questo contesto è meno diffuso) *adattativa*. Una delle tecniche di compressione più diffuse è la cosiddetta LZ, cioè Lempel-Ziv (cognomi degli inventori) della fine degli anni '70. Varie versioni della LZ sono usate in pratica. Lo standard di codifica senza perdita per la rappresentazione di immagini in formato GIF è basato su questa codifica, e anche il formato di memorizzazione compressa dei file "*.zip" per sistema operativo Windows si basa su di una variante della codifica LZ chiamata LZ77, molto simile alla LZ. L'algoritmo "*.zip" (sviluppato da P. Katz a metà degli anni '80) fa seguire alla codifica universale LZ77 una codifica di Shannon-Fano, e per far questo usa una stima delle probabilità dei simboli ricavata come by-product dalla LZ77 stessa.

Il Lempel-Ziv e le sue modificazioni basano il proprio funzionamento su di una tabella di codice che associa a parole di lunghezza variabile della sorgente (cosiddette *frasi*) parole di codice di lunghezza fissa. È questo un modo di funzionamento opposto a quello del codice di Huffman. La corrispondenza è fatta in modo che frasi di sorgente "lunghe" e "frequenti" vengano etichettate con una parola di codice più corta. Questa strategia di compressione senza perdite è particolarmente efficace in sorgenti che presentano sequenze che ricorrono spesso immutate, anche se ripetute senza regolarità. Naturalmente sorgono due problemi: i) come costruire la tabella, e ii) come costruita in maniera adattativa segmentando opportunamente la sorgente in frasi, e ii) la tabella può essere ricostruita dal decodificatore semplicemente sulla base dell'osservazione del messaggio codificato.

Facciamo un esempio per capire come il LZW procede, limitandoci a sorgenti e codici binari, e supponendo di costruire una tabella di 16 parole di codice a 4 bit. La tabella parte con le due parole di codice iniziali 0000 e 0001 corrispondenti ai simboli (caratteri) di sorgente 0 e 1.

Supponiamo ora che la sequenza da codificare sia (nessuno scandalo, è una sequenza come un'altra...)

Il codificatore comincia dunque a scandire la sequenza da codificare per segmentarla (operazione di *parsing*), cercando *la più corta frase non ancora memorizzata in tabella*. Una volta trovata, questa viene inserita in tabella e ad essa viene assegnata come parola di codice la prima parola in tabella non ancora assegnata (naturalmente su 4 bit). Nell'esempio, la prima frase (sottosequenza) è 01, e quindi il primo passo di codifica è:

01:	0010
1:	0001
0:	0000

Questa frase *non* viene codificata con la parola appena inserita in tabella, ma come la concatenazione della codifica della parola già presente seguita dalla codifica dell'ultimo carattere (bit) della nuova frase: 0000**001**. Questa è la parola di codice che viene prodotta. Questo procedimento prosegue fino al riempimento della tabella, cosicché il parsing della stringa d'ingresso, la tabella, e le parole di codice sono le seguenti (fino a riempimento tabella):

0:	0000	1010 :	1000
1:	0001	10101 :	1001
01:	0010	010101 :	1010
010:	0011	0101010 :	1011
10:	0100	101010 :	1100
101 :	0101	1010101 :	1101
0101:	0110	01010101 :	1110
01010:	0111	010101010 :	1111

0000**0001**,0010**0000**,0011**0000**,0100**0001**,0011**0001**, 0110**0000**,0101**0000**,1000**0001**,0111**0001**, 1010**0000**,1001**0000**,1100**0001**,1011**0001**,1110**0**

A questo punto, la codifica procede in modo puramente tabellare sfruttando la tabella ormai fissa e non più adattativa.

Il motivo per cui la prima parola di codice non è quella della frase appena scoperta, ma la concatenazione di una parola preesistente e un nuovo carattere deriva dal funzionamento del decodificatore. Quest'ultimo inizia la decodifica con le sole parole di inizializzazione 0000 e 0001 corrispondenti a 0 e 1 e quindi non potrebbe decodificare la parola nuova - può far affidamento solo su parole già presenti e può quindi immediatamente decodificare la prima frase, cioè 01. Accorgendosi che 01 non è in tabella il decodificatore la inserisce e da lì in poi segue passo passo l'evoluzione della tabella di codifica, che viene automaticamente ricostruita.

Si nota che la segmentazione riesce a "trovare" sequenze *lunghe* tipiche della sorgente che vengono codificate con parole *più corte* e costituiscono il nocciolo della compressione. In questo senso il codice è adattativo e si adegua automaticamente alle caratteristiche statistiche della sorgente stessa, incluso lo sfruttamento della memoria. Nella codifica, il rapporto di compressione sarà eccellente perché verranno principalmente usate le ultime entries della tabella, per le quali la parola di codice è(assai) più corta di quella di sorgente. Lo LZ77 osserva in tempo reale il rapporto di compressione (che dopo il transitorio iniziale si stabilizza); se per caso il rapporto tende a peggiorare, resetta la tabella e riparte da capo. Questo può accadere ad esempio quando in un file da comprimere si ha prima del testo, con certe proprietà statistiche, e poi un'immagine di un'illustrazione, con caratteristiche diverse, per la quale la tabella fino a quel momento utilizzata non è ottimale.

Nelle implementazioni commerciali (il formato . zip per Windows o l'algoritmo "compress" per Linux), come simboli elementari di sorgente vengono considerati gruppi di 8 bit consecutivi (byte) e la tabella è quindi in partenza riempita dei 256 possibili valori di tali simboli. Le parole di codice sono lunghe 12 bit per un dizionario di 4096 simboli di codice, dei quali i primi 256 (che riempiono solo 8 dei 12 bit) sono appunto i simboli elementari di sorgente.

3.8 Da "Source Compression" a "Data Compression": La trasformata di Burrows-Wheeler

Gli algoritmi "classici" per la compressione di sorgente hanno generato un'infinità di tecnologie efficienti più moderne per la "data compression", applicate in una moltitudine di ambiti nel campo dell'ICT, che hanno a loro volta generato ulteriori tecnologie e algoritmi di



data parsing, efficient data representation/storage eccetera che appartengono decisamente al mondo del'informatica e che sono al di là dello scopo di queste note. Vogliamo però descrivere uno dei progenitori di questi algoritmi, che non è a stretto rigore un compressore, ma è invece una tecnologia "trasversale" di *preprocessing* che facilita il compito dei successivi veri algoritmi di (compressione, parsing, storage) che si rivelino necessari: la *Burrows-Wheeler transform* (BWT), che come vedremo è molto efficace nel pre-processare sorgenti con memoria, e in particolare caratterizzate da parole (cioè stringhe anche lunge di simboli di sorgente) che ricorrono immutate frequentemente.

La BWT si applica ad un stringa (finita) di simboli di sorgente lunga N e rende un'altra stringa di simboli di sorgente ancora lunga N ottenuta da una *permutazione* della stringa originaria. La permutazione, in quanto tale, è invertibile e quindi dalla trasformata si puo ricostruire la stringa originale. Supponiamo che la nostra sorgente abbia alfabeto $\Omega = \{a, h, i\}$ e che si debba preprocessare la stringa

ahiahiahiahiahi

Appendiamo alla stringa un simbolo di "end-of-file (EOF)" che non appartiene a Ω , ad esempio # che dobbiamo intendere come precedente nell'alfabeto il primo simbolo $s_0 = a$ (come un " s_{-1} "). A questo punto immaginiamo di formare una matrice di simboli di dimensione $(N + 1) \times (N + 1)$ ottenuta inserendo in prima riga la stringa data (compreso il # finale), e nelle righe successive gli shift ciclici a sinistra di una posizione della riga precedente. Dico "immaginiamo" perché la matrice non viene in realtà calcolata dalle implementazioni della BWT - l'operazione è concettuale. Otterremo quanto segue, tabella di sinistra:

> ahiahiahiahiahi# #ahiahiahiahiahi hiahiahiahiahi#a ahi#ahiahiahiahi iahiahiahiahi#ah ahiahi#ahiahiahi ahiahiahiahi#ahi ahiahiahi#ahiahi hiahiahiahi#ahia ahiahiahiahi#ahi iahiahiahi#ahiah ahiahiahiahiahi# ahiahiahi#ahiahi hi#ahiahiahiahia hiahiahi#ahiahia hiahi#ahiahiahia iahiahi#ahiahiah hiahiahiahiahi#a ahiahi#ahiahiahi hiahiahi#ahiahia hiahi#ahiahiahia hiahiahiahi#ahia iahi#ahiahiahiah iahiahiahiahi#ah ahi#ahiahiahiahi i#ahiahiahiahiah hi#ahiahiahiahia iahi#ahiahiahiah i#ahiahiahiahiah iahiahi#ahiahiah #ahiahiahiahiahi iahiahiahi#ahiah

Adesso dobbiamo ordinare la matrice per righe, considerando le righe come parole di sorgente e ordinandole in ordine lessicografico (alfabetico), tenendo anche conto di #, ottenendo la tabella di destra sopra mostrata. La prima riga contiene la stringa originaria preceduta da #, e la prima colonna contiene tutti i caratteri della stringa disposti in ordine alfabetico - ma queste non hanno niente a che vedere con la BWT. La trasformata è infatti la stringa ottenuta leggendo l'*ultima colonna* della matrice riordinata, rimuovendo il simbolo di escape #, ma aggiungendo il numero della posizione dove questo si trovava:

iiiiihhhhhaaaaa5

114 RIDONDANZA ED EFFICIENZA - COMPRIMEREIDAII

Sembra incredibile, ma la BWT è invertibile. Prima di dimostrarlo, notiamo la particolarità dell'output: le ricorrenze della sorgente sono state individuate e raggruppate - in questo modo si può applicare un banale algoritmo di compressione come il RLE e ottenere globalmente un compressore semplicissimo e molto efficace.

Invertiamo la BWT, cominciando con il ricostruire l'ultima colonna della matrice riordinata (aggiungendo # nel posto 5) e con il ricostruire subito dopo anche la prima colonna che si ottiene riordinando alfabeticamente tutti i caratteri dell'ultima:

#xxxxxxxxxxxxxxxi	i#	#a	#axxxxxxxxxxxxxi
axxxxxxxxxxxxxi	ia	ah	ahxxxxxxxxxxxxxi
axxxxxxxxxxxxxi	ia	ah	ahxxxxxxxxxxxxxi
axxxxxxxxxxxxxi	ia	ah	ahxxxxxxxxxxxxxi
axxxxxxxxxxxxxi	ia	ah	ahxxxxxxxxxxxxxi
axxxxxxxxxxxx#	#a	ah	ahxxxxxxxxxxxx#
hxxxxxxxxxxxxxa	ah	hi	hixxxxxxxxxxxxxa
hxxxxxxxxxxxxxa	ah	hi	hixxxxxxxxxxxxxx
hxxxxxxxxxxxxx	ah	hi	hixxxxxxxxxxxxxa
hxxxxxxxxxxxxx	ah	hi	hixxxxxxxxxxxxx
hxxxxxxxxxxxxxa	ah	hi	hixxxxxxxxxxxxxa
ixxxxxxxxxxxx	hi	i#	iaxxxxxxxxxxxx
ixxxxxxxxxxxxx	hi	ia	iaxxxxxxxxxxxx
ixxxxxxxxxxxx	hi	ia	iaxxxxxxxxxxxx
ixxxxxxxxxxxx	hi	ia	iaxxxxxxxxxxxxx
ixxxxxxxxxxxxx	hi	ia	iaxxxxxxxxxxxxx

A questo punto ho anche tutte le *coppie* di caratteri della sorgente, considerando che le righe sono cicliche. Mettendo in ordine alfabetico le coppie, ottengo le prime *due* colonne, e così via. Al termine, la prima riga, tolto il primo carattere, sarà la stringa originaria. Gli algoritmi di trasformata diretta e inversa non necessitano della costruzione esplicita delle matrici, ma ottengono le varie sequenze giocando ciclicamente con gli indici della stringa - sono entrambi di complessità *lineare* con N, rendendo la BWT molto efficiente.

Chiaramente, la BWT è un algoritmo di preprocessing: in se' non comprime i dati, ma li predispone a una compressione semplice - è usato in vari SW di compressione e come preprocessore per l'allineamento di sequenze di DNA in bioinformatica. Se vogliamo, la BWT evidenzia la grande distanza tra l'impianto "concettuale" chiarissimo e potentissimo della Information Theory, e le implementazioni scaltre che vengono da un approccio empirico ma efficace come quello descritto.

3.9 E adesso? Codifica di sorgente con perdita

Tutte le codifiche di sorgente fin qui esaminate sono senza perdita (lossless) e, messe alla prova, garantiscono nella maggioranza dei casi rapporti di compressione significativi ma non drammatici (attorno al 50% per un *file* di testo). Per questo sono molto usate anche strategie di compressione della sorgente *con perdita* (lossy) in cui il flusso di bit di informazione non può essere ricostruito esattamente dal decodificatore. Queste strategie consentono però di ottenere rapporti di compressione (molto) maggiori degli schemi senza perdita.

Molte tecniche di compressione della sorgente con perdita sono basate, come vedremo, su di una serie di criteri empirici fortemente dipendenti dall'applicazione. Esiste un contesto formale in cui queste tecniche possono essere accomunate e formalizzate ? Affronteremo la

E ADESSO? CODIFICA DI SORGENTE CON PERDITA 115



Figura 3.18 Comunicazione su di un collegamento con disturbo

questione nel Cap. 5 e scopriremo che la risposta è positiva, ma prima di esaminare questo contesto dovremo fare una lunga digressione sulla trasmissione dell'informazione in modo affidabile su canali con disturbo. Questo argomento, interessante di per se', è collegato con la questione della codifica con perdita. Lo schema elementare della trasmissione affidabile su di un canale con disturbo è infatti quello mostrato in Fig. 3.18, in cui l'informazione di sorgente giunge all'utilizzatore parzialmente distorta. Questa "distorsione" (in senso non tecnico) dell'informazione può essere dovuta a disturbi incontrollabili e indesiderabili introdotti dall'esterno (rumore, interferenza ecc.), come nel caso ad esempio delle trasmissioni radio. Ma uno schema di questo tipo si adatta anche al caso in cui il "canale" è un procedimento di codifica/decodifica che intenzionalmente perturba (in maniera controllata) l'informazione di sorgente di sorgente in modo da ridurne la velocità di trasmissione o l'occupazione su file, come già discusso. Lo studio del problema della trasmissione affidabile su di un canale disturbato che affronteremo nel paragrafo successivo permetterà di introdurre concetti che saranno anche alla base del problema della compressione di sorgente con perdita d'informazione che perseguiremo nel Cap. 5.



CAPITOLO 4

ALLA RICERCA DELL'AFFIDABILITÀ PERDUTA



"Rappresentazione del satellite Voyager 1 che è stato lanciato nel 1977 e che trasmette tuttora dati di esplorazione del sistema solare"

-Communications against all odds

Il satellite Voyager 1 è l'oggetto che riesce a trasmettere affidabilmente dati digitali da 140 a 1400 bit/s sulla massima distanza mai coperta da un oggetto costruito dall'uomo: attualmente (2020) ha sorpassato i confini del sistema solare e si trova a circa 150 unità

Comunicazioni Digitali, I Edizione. di M. Luise Copyright © 2023 University of Pisa

118 ALLA RICERCA DELL'AFFIDABILITÀ PERDUTA

astronomiche (22×10^9 km) dalla Terra. Il motto di questo capitolo è infatti: comunicare in ogni avversità - garantire cioè l'affidabilità dei dati ricevuti su di un collegamento nonostante le varie fonti di disturbo.

4.1 Comunicazioni Digitali su di un "canale rumoroso"

Nei paragrafi precedenti abbiamo stabilito come caratterizzare il contenuto informativo di una sorgente d'informazione, e abbiamo esaminato alcune tecniche per rendere una sorgente il meno ridondante possibile, con o senza perdita di fedeltà, ci siamo cioè occupati di come rendere *efficiente* il link di comunicazione, impegnandolo con la minima quantità di dati possible.

Detto questo, ricordiamo che lo scopo di un sistema di comunicazione è quello di trasferire una certa quantità di informazione da una sorgente S[n] ad un utilizzatore in un certo intervallo di tempo. Spesso tale trasmissione deve avvenire in tempo reale in modo che la replica della successione di simboli di sorgente ricevuta dall'utilizzatore sia quanto più possibile sincrona con quella originaria. È questo il caso della trasmissione di segnali vocali per una comunicazione telefonica, o segnali video in una videoconferenza. Altre volte è invece ammissibile che la trasmissione avvenga con (grande) ritardo o in tempo virtuale, come nel caso del trasferimento di un file di dati (ad esempio una transazione bancaria) tra due server, in cui la sincronia è del tutto inessenziale, oppure nel caso della diffusione broadcast di materiale registrato (film, spettacoli ecc.). In ogni caso, a seconda del tipo di sorgente il sistema deve garantire una certa qualità del servizio (QoS, Quality of Service) espressa normalmente in termini di rapporto segnale/rumore, o tasso d'errore binario (BER, Bit Error Rate) o ritardo massimo, o probabilità di perdita di pacchetti eccetera. Per molti motivi, non è possibile in generale ricevere l'informazione di sorgente inalterata a causa di disturbi, interferenze, imperfezioni o limitazioni degli apparati e così via. Bisogna dunque accettare nella comunicazione un certo ammontare di perdita di informazione dovuta alla trasmissione su di un mezzo fisico.

Un problema analogo è quello della memorizzazione del messaggio di sorgente su un mezzo di capacità limitata. Il disturbo è costituito in questo caso dagli errori di scrittura/lettura sul supporto di memoria (tipicamente un hard-disk) che possono occasionalmente verificarsi. Per il problema della memorizzazione è anche chiaro un ulteriore aspetto: il mezzo deve offrire una capacità di memorizzazione adeguata al messaggio di sorgente che si intende registrare. Tutti sanno ad esempio che è impossibile memorizzare su di un CD standard, di capacità inferiore ai 700 Mbyte, un intero film di durata media ad alta qualità - diventa necessario un supporto più capace come il DVD. Limitazione di capacità si hanno anche nel caso della trasmissione in tempo reale. Se la sorgente emette ad una velocità netta informativa (dopo compressione) di R_b bit al secondo, il canale deve essere in grado di offrire una capacità commisurata, che dipende primariamente dalla banda fisicamente disponibile, ma anche, come vedremo, dall'intensità del disturbo.

Dunque abbiamo delineato i due problemi fondamentali della trasmissione dell'informazione, e cioè la *affidabilità* misurata da una certa QoS per una certa Capacità. Vedremo nei prossimi paragrafi l'interrelazione tra questi due aspetti, e le strategie che possono essere adottate per sfruttare un dato canale in maniera ottimale.



Figura 4.1 Sorgente originale (a), con *BER* intermedia (b), e con *BER* alta (c)

4.2 Il canale di comunicazione nella Teoria dell'Informazione

4.2.1 Modelli di Canale

Per avere una rappresentazione visuale di che cosa significhi imporre una "qualità del servizio" mostriamo in Fig. 4.1 l'immagine di prova "Lena" così come prodotta dalla sorgente (a) e in due casi di trasmissione con perdita d'informazione. Nel prima caso (b), i bit costituenti l'informazione sono errati in maniera aleatoria con probabilità 10^{-4} , mentre nel secondo (c) la BER è $5 \cdot 10^{-4}$. L'alterazione della sorgente è evidente.

La situazione che ha condotto a questo fenomeno è già stata modellata nella Fig. 3.18, in cui il messaggio di sorgente viene inviato attraverso un canale di trasmissione che introduce un disturbo più o meno rilevante. Se tale disturbo è di forte intensità si ha una situazione come quella di Fig. 4.1 (c), che necessita di qualche forma di contromisura. Prima di poter capire che tipo di contromisure possono essere adottate (i codici a protezione d'errore) dobbiamo procedere allo studio della modellizzazione del canale di trasmissione sopra accennata.

Come di consueto, partiamo con un esempio: la trasmissione dei simboli di una sorgente binaria attraverso modulazione 16-QAM su canale AWGN (Additive White Gaussian Noise), come mostrato in Fig. 4.2 (a). Tale schema descrive lo "strato fisico" (physical layer) del collegamento digitale; volendo costruire modelli di questo collegamento un poco più astratti abbiamo varie opzioni, mostrate ulteriormente in Fig. 4.2: immaginando di prescindere dall'effettiva struttura di ciò che è racchiuso all'interno del blocco tratteggiato, nella Fig. 4.2 (b) abbiamo un canale a ingresso binario e uscita binaria; in Fig. 4.2 (c) abbiamo un

ALLA RICERCA DELL'AFFIDABILITÀ PERDUTA



Figura 4.2 Diversi modelli del link digitale 16-QAM con rumore

canale a 16 simboli di ingresso e 16 in uscita, e infine in 4.2 (d) abbiamo ingresso binario e uscita "continua" (chiamata anche soft in contrapposizione all'uscita hard prodotta dal decisore a soglia). Dovremo quindi cercare di analizzare nel seguito modelli di questo tipo che prescindono dall'implementazione dello strato fisico, e considerano solo il trattamento dell'informazione ai terminali di ingresso/uscita, cioè modelli di tipo end-to-end.

Cominciamo dunque con un caso più semplice di quello appena esaminato, e precisamente una trasmissione BPSK con disturbo AWGN. Dai fondamenti delle telecomunicazioni, siamo in grado di ricavare facilmente la probabilità di errore P(E) sul generico bit di sorgente, calcolando come passo intermedio le probabilità di errore condizionate ai due possibile valori di sorgente P(E|0) e P(E|1):

$$P(E|0) = P(\hat{S}[n] = \hat{1}|S[n] = 0) \stackrel{\triangle}{=} P(\hat{1}|0) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$
$$P(E|1) = P(\hat{S}[n] = \hat{0}|S[n] = 1) \stackrel{\triangle}{=} P(\hat{0}|1) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$
(4.1)

dove $\hat{S}[n]$ è la versione ricostruita al ricevitore ("stimata") del simbolo di sorgente trasmesso S[n], e dove si nota la "simmetria" del problema indotta dall'uguaglianza delle due probabilità condizionate.

Sulla base della conoscenza di queste quantità, e seguendo il modus operandi appena delineato, possiamo "dimenticare" l'effettiva struttura fisica del sistema (modulazione, livello di rumore etc.) e passare ad un modello più astratto di canale con disturbo che rappresentiamo graficamente in Fig. 4.3 (a). Il significato di questa rappresentazione è chiaro: se la sorgente emette un simbolo "0", verrà ricevuto (erroneamente) un simbolo "1" con probabilità P(1|0) e verrà ricevuto (correttamente) un simbolo " $\hat{0}$ " con probabilità $P(\hat{0}|0) = 1 - P(\hat{1}|0)$. Analogamente per il simbolo trasmesso "1". La caratterizzazione del

120



Figura 4.3 Rappresentazione dei canali BSC (a) e BEC (b)

canale richiede dunque la conoscenza delle 4 probabilità condizionate $P(\hat{S}[n] = s_m |S[n] =$ $s_k \stackrel{\Delta}{=} P(s_m | s_k), m, k = 1, 2.$

Più in generale, l'uscita del canale può avere un alfabeto con un numero di simboli diverso da quello di sorgente, quando si vogliono modellare situazioni diverse da quella appena vista. Come secondo esempio pensiamo all'invio di un pacchetto di dati in una rete IP, senza considerare stavolta la tecnologia di strato fisico. In qualunque nodo della rete, prima di inoltrare il pacchetto al nodo successivo o all'utilizzatore finale, si controlla l'integrità del pacchetto da possibili errori nella comunicazione attraverso il parity check sull'apposito campo CRC (Cyclic Redundancy Check) del preambolo. Se il parity check fallisce, il pacchetto viene semplicemente scartato e rimosso dalla rete - abbiamo una cancellazione (erasure) del pacchetto. Se invece il parity check è soddisfatto, il pacchetto viene inoltrato o utilizzato perché privo di errori. Questa situazione è rappresentata nello schema di Fig. 4.3 (b), ove si nota che *non* compaiono probabilità del tipo $P(\hat{1}|0) \circ P(\hat{0}|1)$ perché sono ritenute uguali a 0 - il canale non contempla eventi errore. Compare invece esplicitamente un terzo esito della trasmissione del bit, e cioè la cancellazione del bit stesso perché appartenente a un pacchetto scartato dal parity check - sia che fosse tra quelli del pacchetto ricevuti correttamente, sia che fosse tra quelli che hanno fatto fallire il check. Il modello del canale con cancellazione (BEC, Binary Erasure Channel) è mostrato in Fig. 4.3 (b), dove $P_{CE} \stackrel{\triangle}{=} P(CE|0) = P(CE|1)$ è la probabilità di una *Channel Erasure* sul livello rispettivamente 0 o 1.

Un ulteriore esempio di canale con diversi alfabeti di ingresso e uscita è quello rappresentato da un modem BPSK, in cui il segnale analogico ricevuto viene digitalizzato (cioè campionato e quantizzato) su 3 bit, come indicato in Fig. 4.4. Allora l'uscita del canale C[n] ha un alfabeto $\Theta = \{c_1, c_2, ..., c_8\}$ a 8 simboli (gli otto possibili valori dei tre bit).

Generalizzando, un canale di comunicazione con disturbo discreto senza memoria è caratterizzato dalla matrice $M \times L$ delle probabilità condizionate

$$\mathbf{P} = \{P(c_k|s_m)\}, \ k = 1, ..., K; \ m = 1, ..., M$$
(4.2)

che sono anche chiamate probabilità di canale. Il canale classico della teoria delle modulazioni non codificate con AWGN è un canale cosiddetto con decisioni a soglia (harddecisions) o "quadro" per cui K = M (è il caso della Fig. 4.2 (b)). In questo caso si calcola

121



Figura 4.4 Collegamento BPSK con soft-output quantizzato su 3 bit

facilmente la probabilità totale di errore:

$$P(E) = \sum_{m=1}^{M} P(E|s_m) P(s_m) = \sum_{m=1}^{M} [1 - P(c_m|s_m)] P(s_m)$$
(4.3)

dove naturalmente le probabilità a priori $P(s_m)$ sono note dalla descrizione della sorgente. Il lettore dica come è fatta la matrice **P** di un canale "quadro" privo di errori e come invece è quella di un canale con disturbo di gran lunga prevalente sul segnale. L'esempio associato alla trasmissione BPSK su canale con AWGN definisce il BSC in cui le due probabilità di errore condizionate sono uguali.

4.2.2 Equivocazione, Informazione Mutua, Irrilevanza

Tornando al caso generale, le probabilità associate ai simboli di sorgente s_m determinano il contenuto informativo della sorgente, $H(S[n]) = -\sum_{m=1}^{M} P(s_m) \log(P(s_m))$. D'altronde, a partire dal vettore colonna \mathbf{p}_s delle probabilità di sorgente (o di ingresso, note al ricevitore) si può facilmente calcolare il vettore colonna \mathbf{p}_c delle probabilità dei simboli di canale (o di uscita) attraverso la matrice delle probabilità di canale:

$$\mathbf{p}_c = \mathbf{P}\mathbf{p} \tag{4.4}$$

la matrice **P** ha infatti il significato di una matrice di transizione. Questa nuova distribuzione di probabilità determina a sua volta il contenuto informativo dei simboli C[n] all'uscita dal canale:

$$H(C) = \sum_{k=1}^{K} P(c_k) \log\left(\frac{1}{P(c_k)}\right)$$
(4.5)

Esplicitando la (4.4), notiamo che si tratta in pratica di una formulazione vettoriale del teorema della probabilità totale: , $P(c_k) = \sum_{m=1}^{M} P(c_k|s_m)P(s_m), k = 1, ..., K.$

Che relazione c'è tra le due informazioni (entropie) H(S) e H(C)? Qual è l'effetto della trasmissione sul canale con disturbo sul contenuto informativo della sorgente? Nel rispondere a questa domanda adotteremo il punto di vista del ricevitore che conosce le probabilità a priori della sorgente ma che, osservato un certo valore di C[n], non è del tutto certo (per effetto dei disturbi) su quale sia il valore di S[n] effettivamente trasmesso: c'è un'*incertezza* residua su S[n] anche dopo che è stata osservata l'uscita del canale. Supponendo di aver osservato il valore c_k di canale, questa incertezza condizionata è l'entropia della sorgente condizionata al particolare valore c_k :

$$H(S|c_k) = \sum_{m=1}^{M} P(s_m|c_k) \log\left(\frac{1}{P(s_m|c_k)}\right)$$
(4.6)

Considerando tutti i possibili simboli di canale e mediando si ottiene la definizione di incertezza media di uscita o *equivocazione*:

$$H(S|C) = \sum_{k=1}^{K} P(c_k) \sum_{m=1}^{M} P(s_m|c_k) \log\left(\frac{1}{P(s_m|c_k)}\right)$$
$$= \sum_{k=1}^{K} \sum_{m=1}^{M} P(s_m, c_k) \log\left(\frac{1}{P(s_m|c_k)}\right)$$
(4.7)

Prima di osservare l'uscita del canale, l'incertezza del ricevitore sul simbolo trasmesso è pari a quella "a priori", cioè H(S). *Dopo* l'osservazione dell'uscita del canale, l'incertezza residua è dunque H(S|C). Se ad esempio il canale è privo di disturbo, tutte le incertezze $H(S|c_m)$ sono nulle e così anche H(S|C): l'incertezza sui simboli di sorgente è nulla perché, senza disturbo, non c'è possibilità di equivocare. Se al limite il canale è massimamente disturbato, C[n] è del tutto indipendente da S[n], le probabilità dei simboli di sorgente del canale, e quindi H(S|C) = H(S) e l'equivocazione è massima !

La differenza tra l'informazione di sorgente e l'equivocazione di canale H(S) - H(S|C), (che vale H(S) per canale senza disturbo oppure 0 per canale "solo disturbo"), ha un significato importante. Viene chiamata *informazione mutua* e rappresenta la quantità di informazione che il canale è in grado di far transitare nonostante la presenza del disturbo:

$$I(S,C) \stackrel{\triangle}{=} H(S) - H(S|C) \tag{4.8}$$

Dimostreremo che $I(S, C) \ge 0$, ma visto che anche $H(S|C) \ge 0$ in quanto media di entropia, è chiaro che l'informazione che effettivamente transita nel canale è sempre *minore* dell'informazione della sorgente a causa della presenza del disturbo, come ci si poteva attendere. Lavoriamo adesso la definizione di informazione mutua:

$$I(S,C) = \sum_{m=1}^{M} P(s_m) \log\left(\frac{1}{P(s_m)}\right) + \sum_{k=1}^{K} \sum_{m=1}^{M} P(s_m, c_k) \log(P(s_m | c_k))$$
$$= \sum_{k=1}^{K} \sum_{m=1}^{M} P(s_m, c_k) \log\left(\frac{1}{P(s_m)}\right) + \sum_{k=1}^{K} \sum_{m=1}^{M} P(s_m, c_k) \log(P(s_m | c_k))$$
$$= \sum_{k=1}^{K} \sum_{m=1}^{M} P(s_m, c_k) \log\left(\frac{P(s_m | c_k)}{P(s_m)}\right)$$
$$= \sum_{k=1}^{K} \sum_{m=1}^{M} P(s_m, c_k) \log\left(\frac{P(s_m, c_k)}{P(s_m)P(c_k)}\right) = I(C, S)$$
(4.9)

Dalla (4.9) si nota la formale simmetria in S[n] e C[n] della definizione di I(S, C) = I(C, S), per cui possiamo anche scrivere:

$$I(S,C) = H(S) - H(S|C) = H(C) - H(C|S) = I(C,S)$$
(4.10)

da cui si ricava una sorta di "bilancio" informativo tra l'ingresso e l'uscita rappresentato in Fig. 4.5:

$$H(C) = H(S) - H(S|C) + H(C|S) = I(S,C) + H(C|S)$$
(4.11)



Figura 4.5 Flusso informativo sul canale rumoroso

L'interpretazione è interessante: il contenuto informativo dell'uscita del canale è costituito da una parte "utile" I(S, C) data dall'aliquota dell'informazione di sorgente transitata dal canale, più una seconda componente irrilevante H(C|S) (chiamata infatti *irrilevanza di canale* o irrilevanza *tout-court*) che è sì informazione, ma di provenienza estranea alla sorgente, e cioè generata dal disturbo. Se, al limite, il canale è "solo disturbo", l'informazione mutua è nulla, l'irrilevanza è massima, l'entropia di canale è costiutia dalla sola irrilevanza, e può anche essere maggiore di quella di sorgente (si pensi a simboli di canale uniformemente distribuiti per effetto di un disturbo prevalente) !

Dovremmo a questo punto esaminare alcune proprietà formali della mutua informazione in relazione all'*entropia congiunta* di $S \in C$, definita come segue:

$$H(S,C) \stackrel{\Delta}{=} \sum_{k=1}^{K} \sum_{m=1}^{M} P(s_m, c_k) \log\left(\frac{1}{P(s_m, c_k)}\right)$$
(4.12)

Non insisteremo su queste proprietà, che possono essere facilmente ricavate dal lettore o dalla lettrice come esercizio: ad esempio, trovare

$$I(S,C) = H(S) + H(C) - H(S,C)$$
(4.13)

e darne un'interpretazione. Sfruttando poi la disuguaglianza di Gibbs (3.19), dimostrare che $I(S, C) \ge 0$ e dire quando I(S, C) = 0.

4.2.3 Canali notevoli

Vediamo come calcolare in maniera semplice l'informazione mutua del canale binario simmetrico con probabilità di transizione P(E). Indicando con p la probabilità a priori del simbolo 0, $p \stackrel{\triangle}{=} P(0)$ si ha:

$$I(S,C) = H(S) - H(S|C) = H(C) - H(C|S)$$
(4.14)

Studiamo il termine H(C|S). Si ha

$$H(C|S) = H(C|0)P(0) + H(C|1)P(1) = H(C|0)p + H(C[n])(1-p)$$
(4.15)

D'altronde, le entropie condizionate di cui sopra sono semplici da calcolare, perché, una volta assegnato 0 come simbolo d'ingresso, si ottiene in uscita il simbolo 1 con probabilità P(E) e il simbolo 0 con probabilità 1 - P(E). Questa entropia condizionata è quindi quella di una sorgente binaria con probabilità di un simbolo pari a P(E), che abbiamo definito nella (3.17) con η (P(E)). Questo stesso ragionamento, per la simmetria del canale, vale anche condizionando al simbolo di sorgente 1 e quindi

$$H(C|S) = \eta(P(E)) P(0) + \eta(P(E)) P(1) = \eta(P(E))$$
(4.16)

e non dipende dalle caratteristiche (probabilità) della sorgente. Inoltre, l'uscita del canale C è binaria, e quindi $H(C) = \eta \left(P(\hat{0}) \right) = \eta \left(P(\hat{1}) \right)$. Usando il teorema della probabilità totale,

$$P(0) = (1 - P(E)) \cdot p + P(E) \cdot (1 - p)$$
(4.17)

e quindi

$$I(S,C) = \eta \left([1 - P(E)] \cdot p + P(E) \cdot (1 - p) \right) - \eta \left(P(E) \right)$$
(4.18)

Altrettanto semplice è il calcolo dell'informazione mutua per il BEC con probabilità di sorgente di nuovo $p \in 1 - p$:

$$I(S,C) = H(S) - H(S|C) = \eta(p) - H(S|C)$$
(4.19)

Studiamo H(S|C):

$$H(S|C) = H(S|\hat{0})P(\hat{0}) + H(S|E)P(E) + H(S|\hat{1})P(\hat{1})$$
(4.20)

Si vede immediatamente che $H(S|\hat{0}) = H(S|\hat{1}) = 0$ perché quando viene ricevuto uno $\hat{0}$ o un $\hat{1}$ non vi è alcuna incertezza sul simbolo inviato dalla sorgente (che sono certamente rispettivamente 0 o 1), e quindi l'entropia (condizionata) è uguale a 0. Quando viene ricevuto E, l'incertezza è quella di una sorgente binaria con probabilità

$$P(0|E) = \frac{P(E|0)P(0)}{P(E)}$$
(4.21)

avendo utilizzato la regola di Bayes. D'altronde, per la probabilità totale,

$$P(E) = P(E|0)P(0) + P(E|1)P(1) = p_{CE}p + p_{CE}(1-p) = p_{CE}$$
(4.22)

quindi $P(0|E) = p_{CE} \cdot p/p_{CE} = p, H(S|E) = \eta(p)$, e, riassumendo,

$$I(S,C) = H(S) - H(S|C) = \eta(p) - p_{CE}H(S|C) \cdot \eta(p) = (1 - p_{CE})\eta(p) \quad (4.23)$$

L'esempio più noto di canale *asimmetrico* è infine il cosiddetto "canale Z" il cui diagramma di transizione è rappresentato nella Fig. 4.6. Una situazione di questo tipo si riscontra nelle comunicazioni su fibra ottica, nelle quali (in regime di shot-noise) l'errore sul livello 1 è molto più probabile che sul livello 0, e quindi nel modello la $P(\hat{1}|0)$ viene considerata nulla. La quantità p_z rappresenta la probabilità di errore sul livello 1, per cui la probabilità totale di errore è

$$P(E) = p_z(1-p)$$
(4.24)

Cerchiamo di trovare l'informazione mutua del canale Z, valutata come

$$I(S,C) = H(C) - H(C|S)$$
(4.25)

126 ALLA RICERCA DELL'AFFIDABILITÀ PERDUTA



Figura 4.6 Diagramma del Canale Z

Poiché C è binaria sui livelli $\hat{0}$ e $\hat{1}$, sappiamo che $H(C) = \eta \left(P(\hat{0}) \right)$. D'altronde

$$P(\hat{0}) = p \cdot P(\hat{0}|0) + (1-p) \cdot P(\hat{0}|1) = p + p_z(1-p)$$
(4.26)

Passando al calcolo dell'irrilevanza,

$$H(C|S) = p \cdot H(C|0) + (1-p) \cdot H(C|1) = (1-p) \cdot H(C|1) = (1-p)\eta(p_z)$$
(4.27)

Infatti, condizionando al simbolo 0 di sorgente, non c'è alcuna incertezza sul valore dell'uscita dal canale, e quindi H(C|0) = 0. Condizionando invece al livello 1, l'uscita è binaria con probabilità p_z , $1 - p_z$. Riassumendo,

$$I(S,C) = \eta \left(p + p_z(1-p) \right) - (1-p)\eta(p_z)$$
(4.28)

4.3 Capacità di Shannon di un canale di comunicazione con disturbo

Dal punto di vista della nostra modellistica, assegnare un canale di trasmissione significa assegnare la matrice (di transizione) delle probabilità di canale $\mathbf{P} = \{P(c_k|s_m)\}$, k = 1, ..., K; m = 1, ..., M. Torniamo a esaminare l'espressione dell'informazione mutua I(S, C), cioè dell'aliquota di informazione di sorgente che il canale riesce a far transitare d verso l'utilizzatore:

$$I(S,C) = \sum_{k=1}^{K} \sum_{m=1}^{M} P(s_m, c_k) \log\left(\frac{P(s_m, c_k)}{P(s_m)P(c_k)}\right)$$
$$= \sum_{k=1}^{K} \sum_{m=1}^{M} P(c_k|s_m)P(s_m) \log\left(\frac{P(c_k|s_m)}{P(c_k)}\right)$$
$$= \sum_{k=1}^{K} \sum_{m=1}^{M} P(c_k|s_m)P(s_m) \log\left(\frac{P(c_k|s_m)}{\sum_{i=1}^{M} P(c_k|s_i)P(s_i)}\right)$$
(4.29)

La mutua informazione dipende naturalmente dal canale attraverso le probabilità di transizione (che sono note), ma anche dalla sorgente, attraverso la distribuzione di probabilità $P(s_m)$. Essa potrà dunque, a parità di canale, essere maggiore o minore a seconda delle caratteristiche della sorgente.

Questa osservazione ci permette di dare una definizione importante, e cioè quella di capacità di canale c o capacità di Shannon. Questa quantità è pari alla massima informazione

mutua che può essere trasferita dal canale, al variare delle caratteristiche della sorgente:

$$c \stackrel{\triangle}{=} \max_{\mathbf{p}} I(S, C) = \max_{\mathbf{p}} \sum_{k=1}^{K} \sum_{m=1}^{M} P(c_k | s_m) P(s_m) \log \left(\frac{P(c_k | s_m)}{\sum_{m=1}^{M} P(c_k | s_m) P(s_m)} \right)$$
(4.30)

dove, come già visto, $\mathbf{p} \stackrel{\triangle}{=} [P(s_1), P(s_2), ..., P(s_M)]^T$. Se un dato canale deve essere utilizzato in modo massimamente efficiente, la sorgente deve in un certo senso essere "adattata" al canale in modo da sfruttarlo al meglio. La massimizzazione di *c* rispetto a **p** può essere in generale ottenuta con il metodo dei moltiplicatori di Lagrange, poiché si deve rispettare il vincolo $\sum_{m=1}^{M} P(s_m) = 1$ (nonché ovviamente $P(s_m) \ge 0 \forall m$), ma questa strada si rivela in generale complicata. Nel paragrafo successivo vedremo qualche esempio ricavato in modo elementare per casi particolari.

4.3.1 Capacità dei canali notevoli

Per calcolare la capacità di Shannon del BSC dobbiamo massimizzare rispetto a *p* l'informazione mutua (4.18):

$$c_{BSC} = \max_{p} \left\{ \eta \left((1 - P(E)) \cdot + P(E) \cdot (1 - p) \right) - \eta \left(P(E) \right) \right\} = \max \left\{ \eta \left((1 - P(E)) \cdot + P(E) \cdot (1 - p) \right) \right\} - \eta \left(P(E) \right)$$
(4.31)

Poiché l'irrilevanza non dipenda dalle probabilità di sorgente, l'informazione mutua è massima quando è massima H(C), che è l'entropia di una sorgente binaria e vale al massimo l bit/simbolo quando i simboli di uscita dal canale sono equiprobabili. In conclusione, la capacità del canale binario simmetrico con probabilità di errore P(E) sembra essere

$$c_{BSC} = 1 - \eta \left(P(E) \right) \text{ bit/simbolo}$$

$$(4.32)$$

Diciamo "sembra" perché dobbiamo verificare che tale max venga effettivamente raggiunto - in altre parole, qual è la legge di probabilità della sorgente che permette di ottenere questa capacità ? COme abbiamo già detto, quella che consente di ottenere probabilità dei simboli di canale identiche: P(C[n] = 0) = P(C[n] = 1) = 1/2. Ma, come è facile verificare, questo si ottiene se e solo se p = 1/2, cioè con una sorgente equiprobabile.

Studiando la (4.32), notiamo che aumentando la P(E) da 0 a 1/2 (canale da privo di rumore a "solo rumore"), progressivamente la capacità di canale cala da 1 a 0. Il lettore/lettrice spieghi perché, aumentando la P(E) oltre 1/2, la capacità tende a ri-aumentare, e provi a generalizzare il concetto di canale simmetrico anche al caso "quadro" non binario. Par il REC il calcolo della capacità ancora più semplica:

Per il BEC il calcolo della capacitàè ancora più semplice:

$$c_{BEC} = \max\left\{=(1 - p_{CE})\eta\left(p\right)\right\} = 1 - p_{CE}$$
(4.33)

che si ottiene di nuovo quando p = 1 - p = 1/2 (conseguenza del fatto che anche il BEC è un canale *simmetrico*). La capacità di Shannon è in questo caso coincidente con il *throughput* della rete a pacchetto quando si interpreta la p_{CE} come la probabilità di perdita di un pacchetto.

Possiamo riunire BSC e BEC reintroducendo in quest'ultimo gli eventi errore, ammettendo cioè che la possibilità di rilevare gli errori sui bit componenti un pacchetto del CRC sia



Figura 4.7 Canale binario simmetrico con errori e cancellazioni BSC/E

limitata, e che quindi con probabilità p_e vengano inoltrati/utilizzati anche alcuni pacchetti contenenti qualche errore - chiameremo questo canale BSC/E, il cui modello è mostrato in Fig. 4.7. Poiché il canale è simmetrico, diamo ormai per scontato che la capacità si realizzi con simboli di sorgente equiprobabili (H(S) = 1) come indicato in figura; in tali condizioni, considerando di nuovo la simmetria, si trovano facilmente le probabilità di uscita:

$$P(\hat{0}) = P(\hat{1}) = \frac{1}{2}(1 - p_{CE}) , \ P(E) = p_{CE}$$

dalle quali poi si ricava

$$c_{BSC/E} = H(S) - H(S|C) = 1 - H(S|\hat{0})P(\hat{0}) - H(S|E)P(E) - H(S|\hat{1})P(\hat{$$

$$= 1 - 2H(S|\hat{0})P(\hat{0}) - H(S|E)P(E) = 1 - (1 - p_{CE})H(S|\hat{0}) - p_{CE}H(S|E)$$
(4.34)

D'altronde,

$$P(0|\hat{0}) = \frac{P(\hat{0}|0)P(0)}{P(\hat{0})} = 1 - \frac{p_e}{1 - p_{CE}} , \quad P(0|E) = \frac{P(E|0)P(0)}{P(E)} = 1/2$$

per cui si conclude che

$$c_{BSC/E} = 1 - (1 - p_{CE})\eta\left(\frac{p_e}{1 - p_{CE}}\right) - p_{CE} = (1 - p_{CE})\left[1 - \eta\left(\frac{p_e}{1 - p_{CE}}\right)\right]$$
(4.35)

e il risultato è comprensibile: resta la perdita secca di capacità pari a $1 - p_{CE}$ derivante dalla cancellazioni, non però sulla capacità unitaria del canale ideale senza errori, ma sulla capacità del BSC con probabilità aggiornata alla frazione di pacchetti non perduti, cioè $p_e/(1 - p_{CE})$.

Cerchiamo infine di ricavare la capacità del canale "Z" la cui informazione mutua è

$$I(S,C) = \eta \left(p + p_z(1-p) \right) - (1-p)\eta(p_z)$$
(4.36)

Per trovare la capacità, calcoliamone la derivata rispetto a p:

$$\frac{dI(S,C)]}{dp} = \eta' \left(p + p_z(1-p) \right) \left(1 - p_z \right) + \eta(p_z)$$



Figura 4.8 Confronto tra le capacità dei canali BSC e Z

Ricordando la (3.18) e uguagliando a zero abbiamo,

$$\log \frac{(1-p)(1-p_z)}{p+p_z(1-p)} = -\frac{\eta(p_z)}{1-p_z}$$
(4.37)

(4.38)

dalla quale, dopo qualche conto,

$$p = p_{cap} = \frac{1 - p_z (1 + 2^{-\beta})}{(1 - p_z) \cdot (1 + 2^{-\beta})} \ , \ \beta \stackrel{\triangle}{=} \frac{\eta(p_z)}{1 - p_z}$$

Questo valore corrisponde a una capacità

$$c_{Z} = \eta \left(p_{cap} + p_{z} (1 - p_{cap}) \right) - (1 - p_{cap}) \eta(p_{z})$$

= $\eta \left(\frac{1}{1 + 2^{-\beta}} \right) - \frac{\beta}{1 + 2^{\beta}} = \eta \left(\frac{1}{1 + 2^{-\frac{\eta(p_{z})}{1 - p_{z}}}} \right) - \frac{\frac{\eta(p_{z})}{1 - p_{z}}}{1 + 2^{\frac{\eta(p_{z})}{1 - p_{z}}}}$ (4.39)

Se p_z è piccolo, come accade in pratica, allora $2^{\pm\beta} \simeq 1$, $\beta \simeq \eta(p_z)$ e quindi la capacità è approssimativamente

$$c_Z \simeq 1 - \frac{1}{2}\eta(p_z) \tag{4.40}$$

che rassomiglia formalmente a quella del BSC. Le due capacità sono messe a confronto nella Fig. 4.8, e si nota che il canale Z, non prevedendo errori sul livello 0, è superiore al BSC.

Esempio 4.19

Riprendiamo in considerazione l'esempio che ci ha portato ad introdurre il BSC, e cioè la comunicazione con un modem BPSK non codificata a filtro adattato che opera su di un canale AWGN. Sappiamo che in questo caso il canale end-to-end è BSC con probabilità di errore $p_e = Q(\sqrt{2E_s/N_0})$. Possiamo allora mettere in relazione diretta il rapporto segnale rumore E_s/N_0 con la capacità sfruttabile a valle del modem (cioè considerando il canale come un BSC):

$$c_{BPSK} = 1 - \eta(p_e) = 1 - \eta \left[Q\left(\sqrt{\frac{2E_s}{N_0}}\right) \right]$$
(4.41)





Questa relazione viene rappresentata in Fig. 4.9 e mostra che si può ottenere una capacità non trascurabile anche per rapporti segnale-rumore insospettabilmente bassi, fino a -10 dB. Riprenderemo tale considerazione quando discuteremo la capacità del canale con rumore Gaussiano e il piano delle efficienze di Shannon.

Esempio 4.20

Un canale binario con interferenza è descritto dalla relazione

$$C[n] = S[n] + W[n]$$

(4.42)

ove la sorgente S è binaria con livelli ± 1 , e binario è anche il rumore W[n]. Quest'ultimo, indipendente dalla sorgente S, assume però i livelli $\pm \alpha$ con equiprobabilità. Possiamo interpretare W[n] come un termine di *interferenza* simile al segnale utile, ma di ampiezza diversa.

L'alfabeto di canale è a 4 livelli $\{c_0 = -1 - \alpha, c_1 = -1 + \alpha, c_2 = +1 - \alpha, c_3 = +1 + \alpha\}$, ma evidentemente la capacità non potrà mai essere maggiore di 1 bit/c.u.. Le probabilità di canale sono mostrate in Fig. 4.10 (a), ove si nota che il canale è simmetrico. La capacità sarà dunque c = 1 - H(S|C) ammettendo che la distribuzione di sorgente che la raggiunge è p = 1 - p = 1/2. Prima di calcolare l'equivocazione, osserviamo che, essendo la sorgente equiprobabile, e così anche i livelli del rumore, i quattro simboli in uscita al canale si presentano tutti con probabilità 1/4. Calcoliamo ora l'equivocazione:

$$H(S|C) = \frac{1}{4} \sum_{k=0}^{3} H(S|c_k) p(c_k) = \frac{1}{4} \sum_{k=0}^{3} \eta[p(0|c_k)] p(c_k)$$
(4.43)

Se $\alpha \neq \pm 1$ è noto al ricevitore, le incertezze $\eta[p(0|c_k)]$ sono tutte *nulle*, e la capacità del canale è *unitaria*: il canale è ideale. Nonostante l'interferenza, dal livello ricevuto è sempre possibile risalire con certezza al simbolo trasmesso.

Se invece $\alpha = 1$, cioè l'interferenza è strutturalmente *identica* alla sorgente, il canale diventa come in Fig. 4.10 (b) con i simboli c_2 e c_3 che vengono a coincidere con un


Figura 4.10 Canale binario con interferenza

unico simbolo $\bar{c} = 0$ avente $p(\bar{c}) = 1/2$, e le cose cambiano: si ha evidentemente ancora $H(S|c_0) = H(S|c_3) = 0$, ma

$$p(0|\bar{c}) = \frac{p(\bar{c}|0)p(0)}{p(\bar{c})} = \frac{1/2 \cdot 1/2}{1/2} = \frac{1}{2}$$
(4.44)

e quindi

$$c = 1 - H(S|C) = 1 - \eta \left(p(0|\bar{c}) \right) p(\bar{c}) = 1 - \frac{1}{2}\eta \left(\frac{1}{2}\right) = \frac{1}{2} \text{ bit/c.u.}$$
(4.45)

La capacità si riduce a 1/2 bit/c.u. perché il canale è con probabilità 1/2 ideale (quando l'interferenza è "costruttiva) e con probabilità 1/2 inutile (quando l'interferenza è "distruttiva", cioè opposta al bit di sorgente). Stessa cosa accade se $\alpha = -1$. La lettrice o il lettore possono discutere l'analogia di questo esempio con il caso del BEC.

4.4 Il secondo teorema della codifica (o teorema della codifica con disturbo) di Shannon

4.4.1 Generalità sulla Codifica di canale

Nel paragrafo precedente, abbiamo "astrattamente" definito la quantità c come la massima I(S, C) tra uscita e ingresso del canale, e a questa quantità abbiamo dato il nome di "capacità" di canale, suggerendo un certa interpretazione della quantità stessa - perché? Inoltre, il calcolo della capacità del BSC sembra portare con se' un paradosso. Infatti abbiamo trovato che c non è mai superiore a 1 bit, mentre la sorgente che realizza la capacità, che ha distribuzione uniforme, genera esattamente 1 bit/simbolo, cioè un'informazione maggiore della capacità (?!) stessa. Dobbiamo dare una risposta chiara a queste due domande/dubbi, precisando il significato di c.

Prendiamo come esempio per la nostra discussione il BSC, e cerchiamo di considerarlo con la sua capacità *c separatamente* rispetto alla sorgente. Da un lato abbiamo quest'ultima, che genera informazione al ritmo di H(S) bit/simbolo, e può eventualmente essere codificata a minima lunghezza secondo le tecniche già esaminate. Dall'altro lato abbiamo il canale,



Figura 4.11 Collegamento digitale con codifica di canale

con una capacità di c bit/simbolo. Che succede se la nostra sorgente è tale per cui $H(S) \ge c$ e cerchiamo di trasmettere questa informazione sul canale? Stiamo violando qualche legge fondamentale? Inoltre, sempre consideriamo il BSC, siamo portati a dire: la probabilità di errore P(E) offerta dal canale non dipende certo dalla sorgente, e resta *immutata* sia che $H(S) \ge c$ ma anche viceversa. È possibile migliorare l'affidabilità del canale?

La risposta a tutte le domande di cui sopra sta nella nozione di *codifica di canale* che *segue* quella di sorgente e *precede* l'invio sul canale rumoroso, come mostrato in Fig. 4.11. Questa "architettura" di sistema è tipica della stragrande maggioranza dei link digitali del moderno Internet. Esistono numerosi schemi di codificatori "di canale" - vedremo le specifiche tecnologie utilizzate in pratica nel Cap. 6. In generale, nella codifica di canale, i bit in ingresso al canale rumoroso vengono "protetti" dal disturbo aggiungendo ulteriori simboli di *ridondanza* ricavati dai bit di informazione stessi, il cui scopo è quello di permettere al decodificatore all'uscita del canale di rilevare e correggere eventuali errori prodottisi a causa del disturbo. Consideriamo l'esempio di Fig. 4.11 in cui i bit di sorgente vengono raggruppati in *blocchi* di lunghezza K, e il codificatore produce in corrispondenza di ciascuno di questi blocchi di sorgente, altri blocchi di codice di cifre binarie di lunghezza N > K. La ridondanza è

$$\eta = (N - K)/N = 1 - K/N \tag{4.46}$$

mentre il cosiddetto tasso di codifica (coding rate) è per definizione

$$r \stackrel{\triangle}{=} 1 - \eta = K/N < 1 \tag{4.47}$$

Come vedremo tra un attimo (e in molto maggior dettaglio nel Cap. 6), la presenza di questa ridondanza permette al decodificatore di *rilevare* e *correggere* eventuali errori nel blocco di codice introdotti dal BSC - purché ovviamente il numero di errori sia inferiore a un certo limite determinato dal tipo di codifica stesso.

4.4.2 Codifica di Canale per Negati

Immaginiamo dunque di voler *migliorare* la qualità di un collegamento digitale su di un canale BSC avente probabilità di errore (o BER in pratica) pari alla consueta p_e . La cosa più semplice da fare è utilizzare un *codice a ripetizione*: volendo inviare un bit di informazione b[k], lo inviamo due volte, tre volte, ..., N volte per accertarsi che il ricevitore (meglio) "capisca" ciò che viene trasmesso. Applicando questa codifica, dobbiamo come sempre distinguere chiaramente tra *bit* e *simboli*: utilizzando un codice con N - 1 ripetizioni, l'*n*-esimo *simbolo binario codificato a*[*n*] è

$$a[n] = b[n//N]$$
 (4.48)



Come migliorare la BER di un link digitale con la codifica di canale Figura 4.12

dove l'operatore // indica il risultato della divisione intera di n per N. In (4.48), n è l'indice del simbolo codificato che scorre al (coded) symbol rate $R_c = 1/T_c$ (il pedice c sta per *codificato*), mentre k = n//N è l'indice di bit che scorre al (più lento) bit rate $R_b = 1/T_b$, come rappresentato nella Fig. 4.12 (a). Si comprende facilmente che se siamo forzati a mantenere costante la velocità di segnalazione di a[n] sul canale, l'effetto del codice a ripetizione è quello di *diminuire* il bit rate di informazione netto del fattore N: $R_b = R_c/N$, dato che gli N-1 bit simboli binari) che aggiungiamo appositamente al bit di sorgente non apportano alcuna ulteriore effettiva informazione - rappresentano invece ridondanza, visto che il loro scopo è solo quello di articolare meglio il messaggio inviato.

La ridondanza che è stata introdotta dalla codifica di canale viene quantificata attraverso il cosiddetto tasso di codifica (coding rate)

$$r \stackrel{\triangle}{=} R_b/R_c < 1 \tag{4.49}$$

e/o la ridondanza del codice

$$\eta \stackrel{\triangle}{=} \frac{R_c - R_b}{R_c} = 1 - r \tag{4.50}$$

Nel nostro caso, abbiamo aggiunto N-1 bit (simboli) ridondanti ad ogni bit di informazione, e quindi r = 1/N e $\eta = 1 - 1/N$. Più piccolo è il tasso, più vicina a 1 è la ridondanza, e il contenuto relativo di informazione di ogni simbolo codificato diventa sempre minore.

È intuitivo comprendere che al diminuire del *coding rate* aumenta l'affidabilità del dato ricevuto - vediamo allora di quantificare questo miglioramento, calcolando la BER end-to-end P_e , cioè la probabilità di rivelare in modo errato il bit di informazione b[k] dopo codifica e docodifica. Per valutare questo parametro dobbiamo quindi specificare il criterio applicato dal *decodificatore* che riceve i simboli (binari) d[n] all'uscita del BSC di Fig. 4.12, per ricavare la stima $\hat{b}[k]$ del k-esimo bit di informazione.

Qual è il decoder ottimale per il codice a ripetizione binario? La risposta viene dalla teoria della stima a massima verosimiglianza ed è semplice, nell'ipotesi di trasmissione indipendente da simbolo a simbolo. Il risultato è il cosiddetto decodificatore a maggioranza: a partire da una "parola" (blocco) di simboli ricevuti d[kN], d[kN+1], ..., d[kN+N-1]che codificano il bit b[k], il decodificatore fornisce $\hat{b}[k] = 0$ se il numero di simboli pari a 0 nel blocco ricevuto è maggiore del simbolo di 1, viceversa b[k] = 1. In un codice a ripetizione normalmente N è dispari, e quindi non vi sono ambiguità nella decodifica.

133

134 ALLA RICERCA DELL'AFFIDABILITÀ PERDUTA

La giustificazione del criterio a maggioranza è semplice. Se tutti i simboli ricevuti nella parola sono identici, o non c'è stato alcun errore sul canale, oppure ce ne sono stati N (cioè *tutti* i simboli ricevuti sono errati). Il decoder non può ovviamente distinguere tra questi due casi, ma è chiaro che il secondo, che provoca un errore di decodifica, è a probabilità molto bassa. In tutti gli altri casi in cui vi sono meno di N errori, il decoder si accorge che non vi è accordo tra i simboli ricevuti e che quindi qualcosa è andato storto: il decoder "*rivela* fino a N - 1 errori di canale". Purtroppo, se la maggioranza dei simboli ricevuti è in errore, anche la decisione finale su b[k] sará errata - anche se c'è stata *rivelazione* di errore, non è possibile procedere alla *correzione* dell'errore stesso. Si possono correggere errori di canale finché rappresentano la minoranza, cioè fino a un numero di (N - 1)/2. In questo caso, la decisione del decoder sarà corretta e gli errori di canale saranno stati corretti.

Nell'ipotesi di trasmissione sul canale in condizioni indipendenti di ogni simbolo codificato, il calcolo della probabilità di errore end-to-end P_e dopo decodifica è semplice. Chiamiamo N_E la variabile aleatoria che rappresenta il numero di simboli errati in una parola di codice ricevuta $d[kN], d[kN + 1], ..., d[kN + N - 1], 0 \le N_E \le N$; allora

$$P_e = Pr\{N_E > (N-1)/2\}$$
(4.51)

Gli N_E errori si producono indipendentemente, e quindi la legge di probabilità di N_E è *binomiale* con probabilità p_e (la probabilità di errore del BSC equivalente):

$$\Pr\{N_E = n\} = \binom{N}{n} p_e^n (1 - p_e)^{N-n} \quad 0 \le n \le N$$
(4.52)

per cui la probabilità di errore, cioè la probabilità di avere più di (N-1)/2 errori, è pari alla somma della probabilità di avere un numero di errori pari a (N+1)/2, (N+1)/2+1, ..., N (eventi incompatibili):

$$P_e = \sum_{n=(N+1)/2}^{N} {\binom{N}{n}} p_e^n (1-p_e)^{N-n}$$
(4.53)

Se $p_e < 1/2$, vedremo che $P_e < p_e$ e quindi il codice ha effettivamente ridotto la probabilità di errore sul bit di informazione. In particolare, il nostro BSC potrebbe essere il modello di una comunicazione con segnalazione binaria su canale con disturbo AWGN e ricezione con filtro adattato (Fig. 4.12 (b)), per la quale come sappiamo $p_e = Q(\sqrt{2E_s/N_0})$. Nella figura 4.13 mostriamo le curve di BER end-to-end P_e in funzione del rapporto segnale-rumore E_s/N_0 e per diversi valori della lunghezza di ripetizione N.

Vediamo che, fissato E_s/N_0 e quindi la qualità del BSC, la BER può diventare *arbitraria-mente piccola* purché si aumenti adeguatamente il numero N di ripetizioni (ovvero si riduca proporzionalmente il tasso r = 1/N) - una buona proprietà per un codice a correzione d'errore! A che prezzo? Supponiamo che la trasmissione sul canale fisico sia realizzata con un modem la cui velocità di segnalazione sul canale R_c sia fissa e non modificabile. Se dobbiamo aumentare molto N per migliorare la BER codificata P_e , cioè diminuire molto r perché la BER non protetta p_e è molto alta, otteniamo un bit rate di informazione netto $R_b = r \cdot R_c$ sempre più piccolo - abbiamo aumentato l'affidabilità alle spese della capacità (bit-rate).

Esempio 4.21

Immaginiamo di applicare il codice a ripetizione al BSC di Fig. **??** (b). Possiamo valutare in modo semplice il miglioramento a valle del codice? Proviamo a partire da un BSC



Figura 4.13 BER del codice a ripetizione

non buono ma neanche pessimo con $p_e \ll 1$ (diciamo, $p_e=0.01$). Il termine dominante nella BER (4.53) è il primo della sommatoria, cioè quello che presenta la minima potenza di p_e ; possiamo anche in ogni termine presumere che $(1 - p_e) \simeq 1$. Secondo questa approssimazione,

$$P_e \simeq \binom{N}{\frac{N+1}{2}} p_e^{(N+1)/2} \tag{4.54}$$

e usando la formula di Stirling per approssimare il coefficiente binomiale otteniamo

$$P_e \simeq \frac{2^N}{\sqrt{\pi(N+1)/2}} \ p_e^{(N+1)/2} = \sqrt{\frac{2p_e}{\pi(N+1)}} \ (2\sqrt{p_e})^N \tag{4.55}$$

Si vede chiaramente che non appena $2\sqrt{p_e} < 1$, la BER a valle del codice diminuisce esponenzialmente secondo N/2, cosicché è possibile, come già anticipato, ottenere valori arbitrariamente piccoli della BER per un adeguatamente grande valore di N.

Immaginiamo che si debba invece mantenere costante il bit-rate netto R_b . È necessario allora aumentare la velocità di segnalazione dei simboli codificati $R_c = R_b/r$, al costo di aumentare la banda necessaria sul canale fisico per supportare il bit-rate dato R_b . Ad esempio, usando un impulso di segnalazione di tipo SRRC e alfabeto binario, la banda (a radiofrequenza) necessaria è $B_{RF} = (1+\beta)R_c = (1+\beta)R_b/r$, evidentemente incrementata rispetto al caso non codificato con r = 1. Torneremo su quest'argomento nel paragrafo 4.6.

136 ALLA RICERCA DELL'AFFIDABILITÀ PERDUTA

4.4.3 Il Secondo Teorema di Shannon

Rimandiamo adesso lo studio (più) dettagliato dei codici di canale al Cap. 6 e poniamoci una domanda fondamentale, collegata al tema della capacità di Shannon dal quale siamo partiti: èsempre vero che per diminuire la BER dobbiamo diminuire il coding rate e quindi l'efficienza spettrale? Esistono codici migliori del banale codice a ripetizione usato come esempio? Nella televisione digitale via satellite DVB-S(2) il rapporto E_s/N_0 al ricevitore è sfavorevole (pochi dB), ma l'applicazione è molto esigente dal punto di vista della BER: una buona ricezione del video digitale richiede BER= 10^{-10} , una condizione indicata come *Quasi-Error-Free* (QEF). È proprio necessario arrivare a tassi bassissimi per garantire una BER accettabile?

Per capirne di più, consideriamo di nuovo la trasmissione di una sorgente senza memoria sul nostro BSC avente capacità $c = 1 - \eta (P(E))$, e supponiamo di rispettare una condizione che chiameremo di "buon pilotaggio" e cioè $r \leq c$. In questo modo, i simboli binari codificati portano ciascuno una quantità di informazione pari ad rH(S), e quindi anche supponendo di inviare al canale, prima del codificatore, una sorgente (binaria) perfettamente compressa cioè con $H(S) = H_{MAX} = 1$, l'informazione associata a ciascuno dei simboli in ingresso al canale potrà sempre essere $\leq c$ scegliendo un tasso r adeguato.

Come costruire un codice con r pari ad un certo valore desiderato? Cerchiamo di generalizzare la nozione del codice a ripetizione attraverso l'introduzone di un codice a *blocco*, che cioè mappa sequenze di bit di sorgente **b** lunghe K simboli (chiamati blocchi) in sequenze binarie di canale (blocchi) a lunghe N simboli, N > K (immaginiamo b ed a come vettori *riga* di simboli binari). Definire il codice significa mappare le 2^{K} possibili sequenze (parole) di sorgente **b** di lunghezza K in un sottoinsieme di cardinalità 2^{K} (il cosiddetto *codebook* Γ) dell'insieme completo delle 2^N parole binarie di lunghezza N con K/N = r < 1, r tasso di codifica, così come rappresentato nella Fig. 4.14. Scegliere le "poche" 2^{K} parole di codice **a** all'interno dello spazio con le "molte" 2^{N} parole di canale permette di "spaziare" le parole di codice in modo che risultino molto "lontane" tra di loro, cioè molto diverse, e quindi in modo che, in presenza di qualche errore di ricezione causato dal disturbo che altera qualcuno dei simboli inviati, siano ancor riconoscibili, proprio perché "lontane" e quindi "diverse". Il codice a ripetizione è un codice a blocco con K = 1 e quindi r = K/N = 1/N come già sappiamo. Scegliendo K ed N opportunamente e scegliendo opportunamente il codebook si possono realizzare codici a protezione d'errore al tasso rdesiderato e anche molto efficienti, come vedremo nel Cap. 6.

Una peculiare strategia di definire un codice come in Fig. 4.14 è quella immaginata da Shannon nella dimostrazione del cosiddetto *teorema della codifica di canale*: invece di preoccuparci di distanziare al massimo le parole del codebook, scegliamo invece *casualmente* tali parole, in particolare seguendo la distribuzione di probabilità dei simboli che realizza la capacità del canale utilizzato - una scelta che sembra andare in direzione opposta a quella di cercare di trovare una strategia affinché il codice sia il migliore possible. Il codice "random" così ottenuto potrà essere buono o meno buono con una certa probabilità, intendendo con "buono" il fatto che effettivamente garantisca una probabilità di errore $P(E_m)$ sui bit di sorgente più piccola di un certo valore ε prefissato dalla mia applicazione.

La motivazione di questa scelta apparentemente bizzarra è però la seguente: si può dimostrare che, purché $r \leq c$, il valore atteso $E\{P(E_m)\}$ della probabilità di errore di tutti questi codici casuali è "buono", cioè risulta minore del valore ε che mi sono prefissato, non appena K (e quindi anche N) è sufficientemente grande, con K/N = r < c fissato, e indipendentemente da quanto ε possa essere piccolo. Diciamo "valore atteso" perché questa $P(E_m)$, al variare del codice estratto randomicamente, è essa stessa una... variabile



Figura 4.14 Costruzione di un codice a blocchi

aleatoria. Se la *media* $E\{P(E_m)\}$ su tutti i codici di queste $P(E_m)$ gode di questa proprietà, deve esistere *almeno uno* tra tutti i codici che singolarmente gode della stessa proprietà, cioè che $P(E_m) < \varepsilon$!

Abbiamo dunque dimostrato che, fissato $r \leq c$, esiste certamente un codice a protezione d'errore con tale tasso che permette di ridurre la BER sui bit di sorgente ad un valore arbitrariamente piccolo - cioè garantisce una *comunicazione affidabile*. Dimostrata l'esistenza di questo fantomatico codice, l'ingegnere dell'ICT deve cercare di avvicinarsi il più possibile a questa prestazione-limite con schemi di codifica di canale praticamente implementabili oggetto del prossimo capitolo.

La dimostrazione formale del secondo terorema di Shannon della codifica è lunga e complicata, e la riportiamo nell'Appendice B in versione "debole", cioè per r < c e per il canale BSC - consigliamo di (ri)prenderla in considerazione (solo) dopo lo studio dettagliato del Cap. 6 sui codici a blocco.

Esiste anche una versione "inversa" del teorema: se la condizione di "buon pilotaggio" non viene rispettata, se cioè r > c, allora non esiste alcun codice che può ridurre la $P(E_m)$ arbitrariamente - esisterà un "floor" di $P(E_m)$ che risulterà irriducibile, qualunque sia la tecnologia impiegata. In particolare, il teorema *inverso* della codifica di canale dice che se ci si "accontenta" di una BER target $P(E_m) = p_T$ finita assegnata, allora è possibile aumentare il rate del codice fino al valore

$$r = \frac{c}{1 - \eta(p_T)} \tag{4.56}$$

che risulta naturalmente maggiore della capacità "affidabile" c.

La velocità di produzione dell'informazione (information rate) R_b di una sorgente con entropia H(s) bit/simbolo che emette simboli a una cadenza $R_s = 1/T_s$ simboli/s è ovviamente $R_b = H(S)/T_s$, che rappresenta il bit-rate netto (informativo) di sorgente. Considerando la Fig. 4.15, ci si convince che la condizione di "buon pilotaggio" si può anche esprimere come

$$R_b \le \mathcal{C} \tag{4.57}$$

dove $C = c \cdot R_c$ è la capacità di Shannon in bit/s quando il canale viene utilizzato con una cadenza di R_c simboli codificati/s. Dunque il teorema della codifica di canale giustifica l'importanza che fino a questo momento abbiamo dato al concetto di informazione mutua e

138 ALLA RICERCA DELL'AFFIDABILITÀ PERDUTA



Figura 4.15 Velocità di Informazione e Capacità C

capacità di un canale rumoroso, perché stabilisce un vincolo ben concreto alla realizzazione di una comunicazione affidabile.

Esempio 4.22

Il teorema inverso della codifica di canale è utile per stabilire qual è la *performance* limite in termini di BER che una certa tecnologia può fornire, ammettendo che comunque una certa BER residua dopo la decodifica di canale possa restare, dipendentemente dall'applicazione. Ad esempio, la voce digitalizzata è ancora accettabile con BER= 10^{-3} , un video HD richiede invece una BER= 10^{-10} , quindi i requisiti di affidabilità possono essere molto diversi e comunque *finiti*.

Facciamo riferimento a un canale BSC implementato attraverso un modem BPSK con capacità data dalla (4.41). Tenendo conto che $E_s = rE_b$, dal teorema inverso (4.56) e appunto dalla (4.41) possiamo scrivere

$$=\frac{1-\eta\left[Q\left(\sqrt{\frac{2rE_b}{N_0}}\right)\right]}{1-\eta(BER)}$$

per cui la curva-limite si ottiene ricavando la BER come segue:

$$BER = \eta^{-1} \left(1 - \frac{1 - \eta \left[Q\left(\sqrt{\frac{2rE_b}{N_0}}\right) \right]}{r} \right)$$
(4.59)

(4.58)

che va naturalmente ricavata per via numerica, visto che la funzione η^{-1} non è nota analiticamente. Il risultato è mostrato in Fig. 4.16, nella forma di diverse curve-limite di BER per diversi tassi di codifica r. Il valore di E_b/N_0 per il quale le curve diventano "verticali" corrisponde al caso di BER arbitrariamente piccola, cioè fornisce il valore per E_b/N_0 per il quale la capacità c è proprio pari al valore di r caratteristico della curva stessa.

4.5 II canale AWGN

4.5.1 Canali "hard" e "soft"

Fino a questo momento, abbiamo considerato un modello molto astratto di un canale di trasmissione discreto, cioè con sorgente finita e simboli di uscita scelti anch'essi da un alfabeto finito. È questo il caso dei sistemi delineati nelle Figg. 4.2 (a) e (b) in cui ha avuto luogo la "rigenerazione" del messaggio discreto. Possiamo però considerare la "variabile

IL CANALE AWGN 139



Figura 4.16 Curve-limite sul canale BSC/BPSK

di decisione" z[n] prima della "rigenerazione" come in Fig. 4.2 (c) e modellare quindi il canale come avente ingresso binario (il bit di sorgente n-esimo) e uscita continua (nel senso delle teoria dei segnali, cioè ad ampiezza continua). Come già accennato, questa situazione viene anche indicata con l'appellativo di uscita "soft" contrapposta all'uscita "hard" del decisore a soglia. È intuitivo il fatto che il canale con uscita "soft" può rivelarsi più promettente dal punto di vista della capacità di canale. Infatti ridurre una grandezza da "infiniti" livelli come nel caso di uscita soft a due soli livelli come per l'uscita hard sembra apportare una qualche perdita di informazione. Questo è pienamente confermato dalla pratica se aggiungiamo al nostro schema di trasmissione anche un codificatore di canale. COme vedremo enl paragrafo 6.2, a parità di codice, ad esempio convoluzionale, è possibile decodificare il messaggio di sorgente all'uscita del canale soft o di quello hard, con algoritmi di decodifica di complessità confrontabile (algoritmi di Viterbi). Ebbene, la decodifica ad ampiezza continua mostra una robustezza nei confronti del rumore circa 2 dB maggiore della decodifica a soglia, nel senso che la probabilità di errore sui simboli decodificati "soft" con un dato livello di rumore è pari a quella ottenuta con decodifica "hard", ma un livello di rumore di 2 dB inferiore.

Questa osservazione induce anche a riconsiderare il canale "lato trasmettitore": se usiamo un modello di canale meno astratto, riconosciamo che ciò che in realtà viene inviato per trasmettere l'informazione della sorgente è una *forma d'onda* continua nelle ampiezze e nel tempo ottenuta con un qualche metodo di codifica e modulazione. Considerare forme d'onda anziché simboli è particolarmente significativo se il canale fisico di trasmissione pone delle limitazioni sulla larghezza di banda a disposizione per la comunicazione, un aspetto del tutto assente dal modello più astratto di canale discreto. Per arrivare a trattare la situazione di trasmissione di segnali su di un canale a banda limitata bisogna procedere per gradi, partendo dalla considerazione (semplificata) di un canale a tempo discreto e ampiezze di ingresso/uscita continue (soft). Se la sorgente e il canale sono entrambi stazionari, possiamo restringerci al caso di trasmissione isolata (one-shot) di un unico simbolo (continuo) di sorgente.

Una situazione rappresentativa di alcuni problemi pratici (trasmissione via satellite, su fibra ottica con limitazioni di rumore termico ecc.) è quella delineata in Fig. 4.17 in cui la



Figura 4.17 Additive White Gaussian Noise (AWGN) a tempo discreto

trasmissione del simbolo (continuo) di sorgente S[n] viene disturbata dalla presenza di una componente di rumore Gaussiano additivo W[n], cosicchè l'uscita del canale risulta C[n] = S[n] + W[n]. La considerazione di una grandezza continua costringe a un ripensamento della definizione di informazione da associare a tale grandezza. Cerchiamo, come spesso accade, di estendere a questo scenario la definizione già nota per il caso discreto tramite un procedimento al limite. La nostra sorgente continua stazionaria senza memoria S[n] sarà adesso caratterizzata da una funzione densità di probabilità $f_S(s)$. "Modelliamo" questa sorgente come una sorgente discreta $S_{\Delta}[n]$ a infiniti valori equispaziati di un certo Δs . L'entropia è

$$H(S_{\Delta}) = \sum_{m=-\infty}^{\infty} \left(f_S(m\Delta s)\Delta s \right) \log\left(\frac{1}{f_S(m\Delta s)\Delta s}\right)$$
$$= \sum_{m=-\infty}^{\infty} f_S(m\Delta s) \log\left(\frac{1}{f_S(m\Delta s)}\right) \cdot \Delta s + \log\left(\frac{1}{\Delta s}\right) \sum_{m=-\infty}^{\infty} f_S(m\Delta s) \cdot \Delta s \quad (4.60)$$

Facendo ora $\Delta s \rightarrow 0$, le due sommatorie tendono a risultati finiti. In particolare la seconda sommatoria tende banalmente a 1. Purtroppo però, la quantità $\log(1/\Delta s)$ tende a $+\infty$, e la definizione così formulata diventa inutilizzabile. Qual è il motivo di questo "fallimento" del nostro procedimento al limite ? A una sorgente con un numero finito di livelli è associata informazione finita che può essere rappresentata con un numero finito di bit pari, per definizione, all'entropia H(S). È immediato rendersi conto allora che a una sorgente continua con "infiniti" livelli è associata un'informazione che può essere rappresentata solo con un numero infinito di cifre, e quindi essa stessa infinita. Come uscire da quest'impasse ?

Nel calcolo della capacità di canale o comunque dell'informazione mutua sorgentecanale si ha a che fare con una *differenza* tra entropie (o grandezze simili), o meglio tra l'entropia di sorgente e l'equivocazione: I(S, C) = H(S) - H(S|C). Immaginiamo allora di discretizzare anche l'uscita del canale C[n] con un passo Δc , e scriviamo l'espressione dell'informazione mutua (4.9) adattando le definizioni delle probabilità come già fatto nella (4.60):

$$I(S,C) = \sum_{k} \sum_{m} f_{SC}(m\Delta s, k\Delta c) \Delta s \Delta c \cdot \log\left(\frac{f_{SC}(m\Delta s, k\Delta c) \Delta s \Delta c}{f_S(m\Delta s) \Delta s \cdot f_C(k\Delta c) \Delta c}\right)$$

dove naturalmente $f_{SC}(s,c)$ è la densità di probabilità congiunta traS[n] e C[n]. Proseguendo con il calcolo

$$= \sum_{k} \sum_{m} f_{SC}(m\Delta s, k\Delta c) \log\left(\frac{f_{SC}(m\Delta s, k\Delta c)}{f_{S}(m\Delta s) \cdot f_{C}(k\Delta c)}\right) \Delta s\Delta c$$
$$\xrightarrow{\Delta c, \Delta s \to 0} \int \int f_{SC}(s, c) \log\left(\frac{f_{SC}(s, c)}{f_{S}(s) \cdot f_{C}(c)}\right) dsdc \tag{4.61}$$

n	C[n] = 2.3503858+W[n]
0	2.3502098
1	2.3502087
2	2.3500982
3	2.3502089
4	2.3504389
5	2.3501468
6	2.3500938

Tabella 4.1 Rappresentazione euristica dell'informazione mutua I(S, C)

La sorpresa è che il risultato è stavolta perfettamente ben definito e *finito*, e costituisce la nuova definizione dell'informazione mutua I(S, C) per canali a ingresso e uscita continue

$$I(S,C) \stackrel{\triangle}{=} \int \int f_{SC}(s,c) \log\left(\frac{f_{SC}(s,c)}{f_S(s) \cdot f_C(c)}\right) dsdc$$
(4.62)

e che è la generalizzazione immediata della (4.9) di partenza.

Come dobbiamo interpretare questo risultato, cioè che l'informazione mutua del canale soft-input soft-output è finita? Immaginiamo di inviare un certo valore *soft* s_0 di S[n], cioè un numero reale con infinite cifre (decimali o binarie a seconda della rappresentazione) e quindi portatore, in teoria, di un numero infinito di bit di informazione - e anche l'uscita del canale C[n], essendo soft, sarà portatore di infinita entropia H(C). Nel nostro caso di canale AWGN (ma questo ragionamento è generalizzabile) e *condizionatamente* a $S[n] = s_0$, abbiamo che $C[n] = s_0 + W[n]$ e quindi, osservando che s_0 è un valore noto, l'entropia condizionata $H(C|S = s_0)$ è essenzialmente pari alla H(W) ed è quindi ancora *infinita*. Si intuisce allora che volendo valutare la I(S, C) = H(C) - H(C|S) ci troviamo in una situazione indeterminata del tipo $\infty - \infty$ che puòtranquillamente dare un risultato finito.

Esempio 4.23

Immaginiamo di fissare un qualunque valore di sorgente s_0 , ad esempio $s_0 = 2.3503858...$ e ripetiamo molte volte l'invio sul canale AWGN di $S[n] = s_0$, in un caso in cui $\sigma_w \ll s_0$; otteniamo una serie di valori come quelli di Tab. 4.1 nei quali si nota che le prime cifre di C[n] (le più significative) sono "stabili", mentre le cifre successive (le meno significative) variano in maniera aleatoria a causa del disturbo (perché le più significative non variano?). La tabella è in pratica la rappresentazione delle quantità che abbiamo appena discusso: tutti i valori di C[n] permettono di valutarne l'incertezza, cioè l'entropia H(C) che è infinta perchè colpisce le infinite cifre decimali meno significative; le cifre "stabili", in numero finito, rappresentano l'informazione certa ma *finita* sul valore della sorgente che si ottiene osservando C[n], cioè la I(S, C); le cifre meno significative, del tutto incerte, rappresentano infine la H(C|S), cioè l'incertezza residua sull'uscita tolta l'informazione in ingresso (le cifre stabili).

Se adesso applichiamo alla definizione (4.62) la stessa catena di passaggi della (4.9), ma in senso inverso, otteniamo facilmente la seguente relazione:

$$I(S,C) = \int f_S(s) \log\left(\frac{1}{f_S(s)}\right) ds - \int \int f_{SC}(s,c) \log\left(\frac{1}{f_{S|C}(s)}\right) ds dc$$

142 ALLA RICERCA DELL'AFFIDABILITÀ PERDUTA

$$\stackrel{\triangle}{=} h(S) - h(S|C) \tag{4.63}$$

dove abbiamo definito due nuove quantità, formalmente analoghe alla H(S) e alla H(S|C), che chiamiamo entropia *differenziale* ed equivocazione *differenziale*. Analogamente definiremo h(C) e h(C|S).

Perchè chiamiamo differenziali le quantità appena definite ? In generale, se vogliamo valutare la differenza tra l'entropia H(S) e quella H(T) di una seconda sorgente T[n], iniziando dalle entropie delle due sorgenti discretizzate $S_{\Delta}[n]$ e $T_{\Delta}[n]$ (e usando lo stesso passo Δs per entrambe), otteniamo:

$$H(S_{\Delta}) - H(T_{\Delta}) =$$

$$\sum_{m=-\infty}^{\infty} f_S(m\Delta s) \log\left(\frac{1}{f_S(m\Delta s)\Delta s}\right) \cdot \Delta s + \log\left(\frac{1}{\Delta s}\right) \sum_{m=-\infty}^{\infty} f_S(m\Delta s) \cdot \Delta s$$

$$= \sum_{m=-\infty}^{\infty} f_T(m\Delta s) \log\left(\frac{1}{f_T(m\Delta s)\Delta s}\right) \cdot \Delta s + \log\left(\frac{1}{\Delta s}\right) \sum_{m=-\infty}^{\infty} f_T(m\Delta s) \cdot \Delta s \quad (4.64)$$

Se $\Delta s \rightarrow 0$, i due termini divergenti si elidono, e il risultato è che la differenza delle due entropie è

$$\lim_{\Delta s \to 0} \left\{ \sum_{m=-\infty}^{\infty} f_S(m\Delta s) \log\left(\frac{1}{f_S(m\Delta s)}\right) \Delta s \right\}$$
$$- \lim_{\Delta s \to 0} \left\{ \sum_{m=-\infty}^{\infty} f_T(m\Delta s) \log\left(\frac{1}{f_T(m\Delta s)}\right) \Delta s \right\}$$
$$= \int_{-\infty}^{\infty} f_S(s) \log\left(\frac{1}{f_S(s)}\right) ds - \int_{-\infty}^{\infty} f_T(s) \log\left(\frac{1}{f_T(s)}\right) ds$$
(4.65)

è cioè finita e calcolabile attraverso le entropie differenziali. La parte illimitata dell'entropia è comune alle due sorgenti per il solo fatto che sono *soft* e non è differente tra le due - la parte individuale che le caratterizza, e che rimane nella differenza, è proprio quella differenziale.

Per riassumere, allo scopo di procedere a valutazioni di informazione mutua e da questa di capacità di canale nel caso di quantità *soft*, si deve usare l'*entropia differenziale*

$$h(S) \stackrel{\triangle}{=} \int_{-\infty}^{\infty} f_S(s) \log\left(\frac{1}{f_S(s)}\right) ds$$
(4.66)

con l'accortezza che questa, contrariamente al caso di sorgente discreta, *non* rappresenta l'informazione generata dalla sorgente, come vedremo successivamente.

Esempio 4.24

Calcoliamo l'entropia differenziale di una sorgente S[n] uniformemente distribuita nell'intervallo (finito) $[s_1; s_2]$. Per definizione,

$$h(S) = \int_{s_1}^{s_2} f_S(s) \log\left(\frac{1}{f_S(s)}\right) ds = \int_{s_1}^{s_2} \frac{1}{s_2 - s_1} \log\left(s_2 - s_1\right) ds = \log(s_2 - s_1)$$
(4.67)

Se $s_2 - s_1 < 1$, notiamo che h(S) < 0, e quindi, come anticipato, concludiamo che l'entropia differenziale *non* puòrappresentare il contenuto informativo della sorgente. Poniamoci adesso un secondo problema: tra tutte le densità di probabilità a supporto finito, cioè diverse da zero solo nel'intervallo $[s_1; s_2]$, qual è quella avente h(S) massima? Dobbiamo cioè trovare il massimo di

$$\int_{s_1}^{s_2} f_S(s) \log\left(\frac{1}{f_S(s)}\right) \, ds$$

sotto la condizione di normalizzazione $\int f_S(s) ds = 1$ (altrimenti la questione non ha senso).

La soluzione si trova attraverso il calcolo delle variazioni. Formiamo la funzione Lagrangiana

$$\mathcal{L}(f_S, \lambda) = \int_{s_1}^{s_2} f_S(s) \log\left(\frac{1}{f_S}(s)\right) ds + \lambda \left(\int_{s_1}^{s_2} f_S(s) ds - 1\right)$$
$$= \int_{s_1}^{s_2} \left(f_S(s) \log\left(\frac{1}{f_S(s)}\right) + \lambda f_S(s)\right) ds - \lambda \tag{4.68}$$

e, per trovarne il massimo, deriviamo formalmente rispetto ad f_S , intesa come una incognita, e uguagliamo a 0:

$$\int_{s_1}^{s_2} \left(-1 + \log\left(\frac{1}{f_S(s)}\right) + \lambda \right) \, ds = 0$$

Stante l'arbitrarietà di λ , si deve avere

$$-1 + \log\left(\frac{1}{f_S(s)}\right) + \lambda = 0 \Rightarrow f_S(s) = 2^{\lambda - 1}$$

cioè la densità di probabilità deve essere *costante* sull'intervallo di supporto. Usando il vincolo di normalizzazione $\int f_S(s) ds = 1$, si ottiene la densità uniforme che già abbiamo studiato.

Il risultato dell'esempio 24 non sorprende: la densità uniforme è quella che massimizza l'incertezza sul valore della sorgente, e in un certo senso è l'equivalente continua dei livelli equiprobabili di una sorgente discreta.

Esempio 4.25

Ricaviamo adesso il valore dell'entropia differenziale per una sorgente Gaussiana con media η_S e varianza σ_S^2 , per la quale

$$f_s(s) = \frac{1}{\sqrt{2\pi\sigma_S^2}} \exp\left(-\frac{(s-\eta_S)^2}{2\sigma_S^2}\right)$$

Dalla definzione,

$$h(S) = \int_{-\infty}^{\infty} f_s(s) \cdot \left(\frac{1}{2}\log(2\pi\sigma^2 + \frac{(s-\eta_S)^2}{2\sigma_S^2}\log(e)\right) ds$$

144 ALLA RICERCA DELL'AFFIDABILITÀ PERDUTA

$$= \frac{1}{2}\log(2\pi\sigma_S^2) + \frac{\sigma_S^2}{2\sigma^2}\log(e) = \frac{1}{2}[\log(2\pi\sigma_S^2) + \log(e)] = \log\left(\sqrt{2\pi e\sigma_S^2}\right) \quad (4.69)$$

Notiamo che h(S) è tanto maggiore quanto maggiore è la varianza della sorgente, e che (soprattuto) è *indipendente dal particolare valore medio* η_S .

Nell'esempio precedente, abbiamo analizzato il caso di supporto finito, cioè un vincolo relativo all'intervallo di variazione della sorgente - in pratica, una limitazione sul valore *di picco* del segnale. In altri casi però è più significativo un vincolo sulla *potenza* della sorgente stessa. Qual è la allora la densità di probabilità che massimizza l'entropia differenziale senza particolari limitazioni di ampiezza sul segnale, ma con un vincolo di potenza σ^2 (supponendo valor medio nullo) ?

Procediamo come nell'esempio precedente con il calcolo della variazioni, tenendo conto adesso di *due* vincoli:

$$\int_{-\infty}^{\infty} f_S(s) \log\left(\frac{1}{f_S(s)}\right) ds = \max \quad \text{con} \quad \int_{-\infty}^{\infty} s^2 f_S(s) ds = \sigma^2 \quad \text{e} \quad \int_{-\infty}^{\infty} f_S(s) ds = 1 \tag{4.70}$$

e la Lagrangiana è

$$\mathcal{L}(f_s, \lambda, \mu) = \int_{-\infty}^{\infty} \left[f_S(s) \log\left(\frac{1}{f_S(s)}\right) + \lambda s^2 f_S(s) + \mu f_S(s) \right] ds = \max \quad (4.71)$$

Formalmente "derivando" rispetto a f_S sotto il segno di integrale si ottiene

$$\int_{-\infty}^{\infty} \left[-1 - \log(f_S(s)) + \lambda s^2 + \mu \right] ds = 0$$
(4.72)

ovvero, tenendo conto dell'arbitrarietà di λ e μ .

$$-1 - \log(f_S(s)) + \lambda s^2 + \mu = 0 \quad \Rightarrow \quad f_S(s) = \exp(-1 + \mu) \exp(\lambda s^2) \quad (4.73)$$

cioè esponenziale quadratica. Applicando i vincoli, si trova naturalmente

$$f_S(s) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{s^2}{2\sigma^2}\right) \tag{4.74}$$

cioè la familiare distribuzione Gaussiana !

L'entropia differenziale h(S) delle sorgenti continue gode di alcune proprietà notevoli: ad esempio, l'entropia congiunta h(S, Z) di due sorgenti indipendenti è pari alla somma h(S) + h(Z). Inoltre, come dimostrato in particolare nell'Esempio 25, h(S) è invariante rispetto a cambiamenti del (solo) valor medio della sorgente (la dimostrazione generale è elementare: calcolare la h(Z) con Z[n] = a + S[n], a costante nota)

L'entropia differenziale della sorgente Gaussiana gioca un ruolo fondamentale nel calcolo della capacità del canale Gaussiano (a banda limitata) che affronteremo di seguito.

4.5.2 Il canale Gaussiano a tempo discreto

Estendiamo adesso al canale continuo i concetti già introdotti per il canale discreto, in particolare vediamo la (nuova) definizione di capacità. Un canale a ingresso e uscita continua è caratterizzato non appena si conosce la densità di probabilità $f_{C|S}(c|s)$ dell'uscita del

canale condizionata a un certo valore dell'ingresso del canale stesso. Nel canale additivo Gaussiano di Fig. 4.17, sappiamo che il rumore è a media nulla e varianza σ_W^2 , si ha banalmente

$$f_{C|S}(c|s) = \frac{1}{\sqrt{2\pi\sigma_W^2}} e^{-\frac{(c-s)^2}{2\sigma_W^2}}$$
(4.75)

Già sappiamo che

Ι

$$I(S,C) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{SC}(s,c) \log\left(\frac{f_{SC}(s,c)}{f_S(s) \cdot f_C(c)}\right) dsdc$$
$$(S,C) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_C(c|s) f_S(s) \log\left(\frac{f_C(c|s)}{\int_{-\infty}^{\infty} f_C(c|s) f_S(s) ds}\right) dsdc$$
(4.76)

La capacità di canale si troverà naturalmente variando la distribuzione di probabilità della sorgente $f_S(s)$ in modo che l'informazione mutua risulti massima:

$$c \stackrel{\Delta}{=} \max_{f_S} I(S, C) = \max_{f_S} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_C(c|s) f_S(s) \log \left(\frac{f_C(c|s)}{\int_{-\infty}^{\infty} f_C(c|s) f_S(s) \, ds} \right) dcds$$
(4.77)

Per questa capacità di canale vale ancora il secondo teorema della codifica nella stessa forma della sorgente discreta.

Trovare una soluzione di questa equazione per il caso generale è arduo. Per il canale Gaussiano additivo stazionario però le cose sono abbastanza semplici. Sappiamo infatti che

$$I(S,C) = h(S) - h(S|C) = I(C,S) = h(C) - h(C|S)$$
(4.78)

Ma h(C|S) è facilmente calcolabile: condizionatamente ad un valore noto della sorgente s_0 , l'uscita del canale è

$$C[n] = s_0 + W[n] (4.79)$$

cioè s_0 è il *valor medio* dell'uscita del canale C[n] che risulta condizionatamente Gaussiana con una entropia differenziale che *non* dipende dal valor medio:

$$h(C|s_0) = \log\left(\sqrt{2\pi e \sigma_W^2}\right) \tag{4.80}$$

Mediando ulteriormente su S e qualunque sia la densità di probabilità (incognita) di S[n], il risultato non cambia, per cui

$$h(C|S) = \log\left(\sqrt{2\pi e \sigma_W^2}\right) \tag{4.81}$$

che ovviamente non dipende dalle caratteristiche della sorgente (il lettore dimostri che questa proprietà vale per tutti i canali additivi). Dunque

$$c_{AWGN} \stackrel{\triangle}{=} \max_{f_S} I(S,C) = \max_{f_S} \left[h(C) - h(C|S) \right] = \max_{f_S} h(C) - \log\left(\sqrt{2\pi e \sigma_W^2}\right)$$
(4.82)

146 ALLA RICERCA DELL'AFFIDABILITÀ PERDUTA

Dobbiamo dunque cercare il massimo della h(C), sapendo che la varianza di C[n], dato che S[n] e W[n] sono indipendenti, è $\sigma_C^2 = \sigma_S^2 + \sigma_W^2$. Dall'Esempio 25 sappiamo che il massimo di h(C) tra tutte le densità di probabilità aventi varianza σ_C^2 , si ha se e quando l'uscita C[n] è Gaussiana, e vale $\log \sqrt{2\pi e \sigma_C^2}$. Dobbiamo però chiederci se tale condizione è effettivamente raggiungibile - la risposta è semplice perché, stante W[n] Gaussiana e indipendente da S[n], si ottiene una C[n] Gaussiana quando (anche) l'ingresso S[n] è Gaussiano (che è quindi la nostra *capacity-achieving distribution*).

Nel nostro modello di canale di Fig. 4.17 sappiamo anche che la potenza della sorgente, per motivi di fisica plausibilità, è nota e vale σ_S^2 . Dai ragionamenti appena fatti, concludiamo facilmente che

$$c_{AWGN} = \log\left(\sqrt{2\pi e \left(\sigma_S^2 + \sigma_W^2\right)}\right) - \log\left(\sqrt{2\pi e \sigma_W^2}\right) = \frac{1}{2}\log\left(1 + \frac{\sigma_S^2}{\sigma_W^2}\right) \quad \text{(bit/c.u.)}$$

$$(4.83)$$

È interessante discutere i casi-limite: se il rumore ha una varianza (potenza) che è molto grande nei confronti di quella della sorgente (segnale), si nota che la capacità di canale tende a zero e si arriva alla consueta situazione di canale inutile. Viceversa, se il rumore ha una varianza che è molto piccola, al ,limite tendente a zero, il canale tende a diventare ideale, il rapporto σ_S^2/σ_W^2 tende a crescere illimitatamente, e così anche il suo logaritmo. La conclusione è che la capacità di canale cresce indefinitamente, contrariamente al caso del canale discreto in cui essa resta comunque limitata dalla dimensione dell'alfabeto di canale anche per il caso ideale. Tale risultato non ci sorprende, al contrario è un effetto dell'aver supposto che l'ingresso del canale è un valore continuo con, in teoria, "infinite cifre decimali".

Come già accennato, sarebbe teoricamente possibile in assenza di rumore codificare una stringa di bit arbitrariamente lunga in un unico valore con un numero di decimali arbitrariamente grande. Si potrebbe poi trasmettere tale valore (simbolo) in un unico atto di trasmissione e recuperare tale valore al ricevitore in maniera esatta e senza possibilità di equivocazione. In tal modo sarebbe possibile trasmettere una quantità di informazione arbitrariamente grande. La presenza del rumore pone naturalmente un limite di "accuratezza" nel recuperare il valore continuo trasmesso: alcune delle cifre decimali meno significative diventano incerte, e la quantità di informazione trasmissibile diventa limitata. L'accuratezza è tanto migliore quanto più grande è la potenza (varianza) di segnale nei confronti della potenza (varianza) di rumore. Dunque la capacità, e cioè la qualità del canale, resta legata al *rapporto segnale-rumore* SNR (signal-to-noise-ratio):,

$$c_{AWGN} = \frac{1}{2} \log \left(1 + \text{SNR} \right) \text{(bit/c.u.)}$$
(4.84)

Il canale Gaussiano può avere una capacità anche di parecchi bit/c.u., in funzione del SNR al ricevitore. Tale capacità dovrà essere sfruttata, come vedremo meglio nel seguito, attraverso opportuni metodi di Modulazione e Codifica (MOD/COD).

Esempio 4.26

Introduciamo nel canale AWGN di Fig. 4.17 una variabilità aleatoria con statistica di Rayleigh dell'ampiezza del segnale utile (indipendente dal disturbo W[n], secondo la

modellistica del canale wireless con fading piatto che vedremo in dettaglio nel paragrafo 7.1.3:

$$C[n] = A \cdot S[n] + W[n] \tag{4.85}$$

dove la densità di probabilità dell'ampiezza a è

$$f_A(a) = 2a \cdot \exp(-a^2)$$
, $a > 0$ (4.86)

avente potenza media $E\{A^2\} = 1$. Questo modello si riferisce a un caso in cui l'ampiezza del segnale utile è (molto) lentamente variabile nel tempo rispetto al symbol rate, con la statistica assegnata. Possiamo intendere allora la capacità di Shannon anch'essa come una quantità variabile nel tempo: in certi periodi si rivelerà soddisfacente, in altri no, a seconda del valore che l'ampiezza del canale ha in quel periodo di tempo. Fissato un certo valore dell'ampiezza $A = \alpha$, la capacità *condizionata* a questo valore è

$$c(\alpha) = \frac{1}{2} \log \left(1 + \alpha^2 \frac{\sigma_S^2}{\sigma_W^2} \right) = \frac{1}{2} \log \left(1 + \alpha^2 \cdot \text{SNR} \right)$$
(4.87)

Una volta definito un certo livello c_0 di capacità "limite", potremmo essere interessati a calcolare la *frazione di tempo* entro la quale c è soddisfacente, cioè nella quale $c \ge c_0$. Calcolando in modo *ergodico* la frazione di tempo come probabilità, troviamo immediatamente

$$\Pr\{c \ge c_0\} = \Pr\left\{\frac{1}{2}\log\left(1 + A^2 \text{SNR}\right) \ge c_0\right\}$$
$$= \Pr\left\{A \ge \sqrt{\frac{2^{2c_0} - 1}{\text{SNR}}}\right\} = 1 - \frac{2^{2c_0} - 1}{\text{SNR}}$$
(4.88)

Questa quantità è la probabilità della *disponibilità* del collegamento, e il suo complemento a 1, cioè $(2^{2c_0} - 1)/SNR$ è invece la probabilità di *fuori servizio*. Il valore SNR è il rapporto segnale-rumore sul canale AWGN equivalente (cioè privo di fading, per il quale $A \equiv 1$), ma rappresenta anche il valore di SNR *medio* sul canale con fading, tenendo conto che $E\{A^2\} = 1$.

4.5.3 Il canale Gaussiano con ingresso binario (BIAWGN)

Torniamo adesso a un modello di sistema BPSK disturbato da AWGN in cui abbiamo una modulazione *binaria*, e consideriamo come uscita del canale il valore soft della variabile di decisione campionata z[n] all'uscita del filtro adattato (canale Binary-Input AWGN, BIAWGN). Aggiungiamo però un codificatore di canale binario con tasso di codifica r come indicato in Fig. 4.18. Il canale è di nuovo Gaussiano *additivo* a tempo discreto, però di un tipo diverso da tutti quelli presi in considerazione finora: è un canale a *ingresso binario* e *uscita continua*. La capacità di canale si può ricavare di nuovo come

$$c_{BIAWGN} = \max_{f_S} \left[h(C) - h(C|S) \right] = \max_{f_S} h(C) - \log\left(\sqrt{2\pi e \sigma_W^2}\right) \tag{4.89}$$

ove σ_W^2 è la varianza della componente di rumore di canale. Se si usa una trasmissione con sagomatura radice di Nyquist in trasmissione e ricezione e si normalizza l'ampiezza dei



Figura 4.18 Canale AWGN con ingresso binario (BIAWGN)

simboli trasmessi al valore 1, si trova facilmente che (vedi (2.103))

$$\sigma_W^2 = \left(2E_s/N_0\right)^{-1} = \left(2rE_b/N_0\right)^{-1} \tag{4.90}$$

Per calcolare h(C) dobbiamo ora ricavare $f_C(c)$, e dobbiamo poi procedere alla massimizzazione rispetto alla probabilità dei simboli di sorgente. Possiamo dare per scontato, vista la simmetria della densità di probabilità del rumore e dei livelli della modulazione, che la massimizzazione della h(C) relativamente alla distribuzione dei simboli di sorgente si realizza quando questi sono equiprobabili, per cui

$$f_C(c) = \frac{1}{2} f_C(c|0) + \frac{1}{2} f_C(c|1) = \frac{1}{2} \frac{1}{\sqrt{2\pi\sigma_W^2}} \left[\exp\left(-\frac{(c+1)^2}{2\sigma_W^2}\right) + \exp\left(-\frac{(c-1)^2}{2\sigma_W^2}\right) \right]$$
$$= \frac{1}{2} \sqrt{\frac{E_s/N_0}{\pi}} \left[\exp\left(-\frac{E_s}{N_0}(c+1)^2\right) + \exp\left(-\frac{E_s}{N_0}(c-1)^2\right) \right]$$
(4.91)

quina

$$c_{BIAWGN} = -\int_{-\infty}^{\infty} \frac{1}{2} \sqrt{\frac{E_s/N_0}{\pi}} \left[\exp\left(-\frac{E_s}{N_0}(c+1)^2\right) + \exp\left(-\frac{E_s}{N_0}(c-1)^2\right) \right] \\ \cdot \log\left(\frac{1}{2} \sqrt{\frac{E_s/N_0}{\pi}} \left[\exp\left(-\frac{E_s}{N_0}(c+1)^2\right) + \exp\left(-\frac{E_s}{N_0}(c-1)^2\right) \right] \right) dc \\ - \frac{1}{2} \log\left(\frac{\pi e}{E_s/N_0}\right) \ bit/c.u.$$
(4.92)

Questa capacità non potrà mai ovviamente essere maggiore di 1. Il suo andamento è mostrato in Fig. 4.19, dove si nota il miglioramento rispetto al già noto canale BSC con matched filter in cui l'uscita soft viene predecisa e ridotta a binaria prima della decodifica di canale - la differenza per rapporti E_s/N_0 intermedi è di circa 2 dB a parità di capacità.

Per avere una trasmissione con affidabilità alta quando si usano sorgenti perfettamente compresse (cioè ad entropia unitaria), il tasso del codice dovrà essere minore di questa capacità di canale, cioè si dovrà avere $r \leq c_{BIAWGN}$. Nel caso-limite di uguaglianza, ricordando che $E_s = rE_b$, avremo

$$r + \int_{-\infty}^{\infty} \frac{1}{2} \sqrt{\frac{rE_b/N_0}{\pi}} \left[\exp\left(-r\frac{E_b}{N_0}(c+1)^2\right) + \exp\left(-r\frac{E_b}{N_0}(c-1)^2\right) \right]$$

IL CANALE AWGN 149



Figura 4.19 Capacità del canale BI-AWGN a confronto con quella del BSC/BPSK

$$\cdot \log\left(\frac{1}{2}\sqrt{\frac{rE_b/N_0}{\pi}} \left[\exp\left(-r\frac{E_b}{N_0}(c+1)^2\right) + \exp\left(-r\frac{E_b}{N_0}(c-1)^2\right)\right]\right) dc$$
$$+ \frac{1}{2}\log\left(\frac{\pi e}{rE_b/N_0}\right) = 0 \tag{4.93}$$

Notiamo che in questo caso-limite r coincide con l'efficienza spettrale: se infatti la segnalazione avviene alla velocità di R_s simboli di codice al secondo, la velocità di informazione di sorgente è $R_b = r \cdot R_S$ bit/s. Usando la banda minima per la segnalazione, cioè la banda di Nyquist si ha anche (per un segnale passa-banda modulato) $B = R_s$, e quindi $R_b/B = r$.

La relazione appena ricavata, in cui l'integrazione deve essere calcolata numericamente, definisce implicitamente la curva-limite sul piano di Shannon mostrata in Fig. 4.20 a tratto spesso, confrontata con la curva di capacità del canale Gaussiano analogico (tratto fino) analizzato precedentemente. Dalla curva di Fig. 4.20 possiamo anche ricavare la tabella di valori-limite di E_b/N_0 (con r = c) che garantiscono comunicazione affidabile, da confrontarsi con gli stessi ricavabili per bassa BER dalla Fig. 4.16 e pure riportati in tabella. Il canale con uscita "soft" (BIAWGN) è più efficiente del canale pre-deciso in modo hard

r	$(\mathbf{E_b}/\mathbf{N_0})_{\mathbf{min}}$	$(\mathbf{E_b}/\mathbf{N_0})_{\mathbf{min}}$	
	BIAWGN (dB)	BSC/BPSK (dB)	
1/4	-0.80	0.97	
1/3	-0.49	1.21	
1/2	0.19	1.77	
2/3	1.06	2.51	

Tabella 4.2 Valori di E_b/N_0 limite per diversi canali e tassi di codifica

(BSC/BPSK) di più di 1.5 dB.



4.5.4 Il canale AWGN complesso e la capacità delle costellazioni I/Q

Abbiamo visto che il canale BIAWGN è intrinsecamente limitato a una capacità massima di 1 bit/c.u. in quanto l'ingresso è forzatamente binario. Stesso ragionamento si deve applicare nel caso in cui dobbiamo calcolare la capacità di un modem operante su AWGN per il quale il formato di segnale (cioè la costellazione) è stabilita in anticipo (QPSK, 16-QAM ecc.). Dovemmo cioè ricavare curve di capacità simili alla (4.93), per un canale a ingresso digitale pari ad una particolare costellazione, e uscita soft derivante dall'AWGN. Il calcolo è molto simile a quello della (4.93), dobbiamo però tenere conto che la capacità va valutata sull'*equivalente in banda base* del canale, cioè ragionando su di un segnale I/Q complesso.

Senza nessun vincolo sulla costellazione d'ingresso, il calcolo del canale a tempo discreto AWGN *complesso* SISO è semplice:

$$C_{I/Q} = \log(1 + \text{SNR}) \text{ (bit/simbolo)}$$
(4.94)

un valore cioè *doppio* rispetto alla (4.84). Il risultato si giustifica molto facilmente: le due componenti I/Q del segnale complesso rappresentano *due* canali di comunicazione indipendenti (perchè le componenti I/Q dell'AWGN sono indipendenti) e quindi la capacità totale è la *somma* delle capacità parziali.

In pratica, il canale AWGN complesso viene utilizzato scegliendo un opportuno formato COD/MOD, cioè una codifica a protezione d'errore seguita da una modulazione I/Q. Sul canale vengono inviati i simboli I/Q appartenenti ad una costellazione ad M punti, sulla quale vengono opportunamente mappati (ad esempio tramite la tecnologia Bit-Interleaved Coded Modulation, BICM) i simboli binari codificati ottenuti da un codice binario con tasso r. In questo caso, i simboli I/Q portano ciascuno una quantità di informazione pari a $r \cdot \log(M)$ e quindi la condizione di buon pilotaggio è

$$r \cdot \log(M) \le \mathcal{C}_{I/Q} \to M^r \le 1 + \text{SNR}$$
 (4.95)

Se, come spesso accade, $SNR \gg 1$ ed esprimendo SNR in dB, abbiamo

$$r \cdot N_b \le \frac{\mathrm{SNR}(dB)}{3} \tag{4.96}$$

IL CANALE AWGN



Figura 4.21 Capacità di varie costellazioni I/Q su canale AWGN

dove N_b è il numero di simboli binari codificati associati ad ogni punto di costellazione.

Nel caso di canale AWGN complesso con ingresso I/Q digitale, cioè con una costellazione preassegnata, il calcolo è del tutto simile a quello descritto nella (4.93), ma l'integrazione per calcolare le varie entropie differenziali va svolta in generale sul piano complesso perchè la $f_C(c)$ è una "Gaussian mixture" equiprobabile a due dimensioni con medie condizionate pari ai vari punti della costellazione I/Q. Il risultato però è semplice, rassomiglia parecchio alla curva di Fig. 4.20 ed è mostrato nella Fig. 4.21. Come per il BIAWGN, la capacità non cresce indefinitamente ma "satura" al valore log(M) ove M è naturalmente il numero di simboli della costellazione, e comunque tale capacità deve risultare non inferiore a $r \cdot \log(M)$.

4.5.5 Il canale Gaussiano a tempo continuo

Generalizziamo adesso la formula di capacitá del canale AWGN a tempo discreto. Se la velocità di segnalazione sul canale è $R_s = 1/T_s$ baud, la capacità in termini di bit/s è naturalmente

$$\mathcal{C}_{AWGN} = R_s \cdot c_{AWGN} = \frac{1}{2T_s} \log \left(1 + \frac{\sigma_S^2}{\sigma_W^2} \right) \quad \text{(bit/s)} \tag{4.97}$$

e la condizione di comunicazione affidabile diventa

$$R_b = R_s \cdot N_b \le R_s \cdot c_{AWGN} = \mathcal{C}_{AWGN} \tag{4.98}$$

ove N_b è il numero di bit per simbolo del metodo MOD/COD adottato.

Partendo da qui, siamo in grado di trattare il caso di trasmissione con forme d'onda continue nel tempo, il cosiddetto *waveform channel* rappresentato in Fig. 4.22. Come è possibile estendere il risultato(4.83) al nostro caso, cioè passare da un contesto a tempo discreto ad uno a tempo continuo equivalente? Dobbiamo presumere che i due casi siano

151



Figura 4.22 Canale AWGN a banda limitata

in relazione di campionamento/interpolazione secondo le relazioni (2.70)-(2.87), e che ovviamente, per rendere i due contesti del tutto equivalenti, tutti i segnali a tempo continuo coinvolti siano *a banda limitata B*, campionati con una frequenza di campionamento $f_c = 2B$. In particolare, anche il rumore sarà Gaussiano additivo bianco a banda limitata, caratterizzato cioè da una densità spettrale di potenza $S_W(f) = N_0/2$ costante nella banda *B* del segnale e 0 al di fuori, e quindi da una potenza $P_W = \sigma_W^2 = N_0 B$. Per quanto riguarda il segnale, sappiamo che la capacità si realizza quando la sorgente è Gaussiana, e stavolta anche limitata in banda. In questo caso, essa può essere rappresentata, secondo il teorema del campionamento, da 2*B* campioni al secondo che presenteranno tutti la medesima potenza $P_S = \sigma_S^2$ (sorgente stazionaria). Questa situazione è dunque equivalente ad una trasmissione a tempo discreto in cui la cadenza di segnalazione (uso del canale) si identifica con la frequenza di campionamento dei segnali: $f_c = 2B = 1/T_s$. Applicando la formula già trovata per il tempo discreto, se ne conclude

$$C_{AWGN} = \frac{1}{2T_s} \log\left(1 + \frac{\sigma_S^2}{\sigma_W^2}\right) = B \log\left(1 + \frac{P_S}{N_0 B}\right) \quad \text{(bit/s)} \tag{4.99}$$

Questo è il risultato fondamentale di Shannon sulla capacità del canale AWGN a banda limitata. Anche se, purtroppo, non è un risultato di carattere costruttivo, serve comunque da pietra di paragone per valutare l'efficacia dei sistemi di comunicazione che si possono realizzare in pratica, nel senso che nessun sistema può operare con "piccole" probabilità di errore al di sopra della capacità testé calcolata. Si può notare che la quantità $P_S/(N_0B)$ rappresenta il cosiddetto rapporto segnale-rumore; la capacità del canale Gaussiano è dunque proporzionale alla banda disponibile (e questo non è sorprendente, visto che aumentando la banda si può aumentare la velocità di segnalazione), ma anche al logaritmo del rapporto segnale-rumore cioè, in pratica, al rapporto segnale rumore espresso in dB (per lo meno quando questo è \gg 1). Aumentare la banda a parità di livello del rumore è sempre la maniera migliore per aumentare la capacità del canale proprio per questa proporzionalità logaritmica che rende più difficile aumentare la capacità aumentando la potenza in trasmissione.

4.6 Efficienza dei metodi di modulazione e codifica

La formula della capacità di canale permette anche di misurare la bontà dei metodi di modulazione e codifica (brevemente, MOD/COD) impiegati sul canale Gaussiano. Riprendiamola:

$$\mathcal{C}_{AWGN} = B \log \left(1 + \frac{P_S}{N_0 B} \right) \tag{4.100}$$



Figura 4.23 Efficienze delle tecnologie MOD/COD

In generale, sappiamo che il "buon pilotaggio" richiede $R_b \leq C$ (R_b velocità netta di informazione in bit/s), dalla quale

$$\frac{R_b}{B} \le \log\left(1 + \frac{R_b}{B} \frac{P_S T_b}{N_0}\right) \Rightarrow \frac{R_b}{B} \le \log\left(1 + \frac{R_b}{B} \frac{E_b}{N_0}\right)$$
(4.101)
$$\frac{E_b}{N_0} \ge \frac{2^{R_b/B} - 1}{R_b/B}$$
(4.102)

ovvero

dove
$$E_b$$
 è l'energia per bit ricevuta. Questa relazione può essere interpretata come una "partizione" del piano $(E_b/N_0, R_b/B)$, chiamato anche piano di Shannon, e il limite è la curva descritta da

$$\frac{E_b}{N_0} = \frac{2^{R_b/B} - 1}{R_b/B} \tag{4.103}$$

La zona che rispetta il "buon pilotaggio" viene considerata "raggiungibile", l'altra "non raggiungibile", come indicato in Fig. 4.23.

Il parametro R_b/B , si misura in (bit/s)/Hz e denota l'*efficienza spettrale* di uno schema di MOD/COD. Maggiore è questa quantità, minore sarà la banda richiesta al sistema per trasmettere un flusso con una data velocità di informazione. Un errore di nomenclatura in cui si cade spesso, specialmente nel "mondo Internet", è quello di chiamare "banda" la velocità di informazione R_b che è misurata in bit/s e dipende solo dalle caratteristiche del flusso informativo. La vera banda *B* misurata in Hz è una caratteristica del *segnale fisico* utilizzato per veicolare il flusso informativo, e dipende anche dal metodo di MOD/COD (costellazione e tasso della codifica di canale). La Fig. 4.24 rappresenta una costellazione 16QAM con il relativo *mapping* dei 4 bit associati a ciascun simbolo s_i , i = 1, ..., 16. Usando un codice di canale, i valori indicati in figura sono i *simboli binari* a[n] prodotti dal codificatore di canale piuttosto che i bit di informazione b[k] (vedi Fig. 4.12).





Nella Fig. 4.20 è ad esempio riportato il punto caratteristico di un MOD/COD BPSK con il consueto codice convoluzionale K = 7 del DVB con decodifica "soft"; si nota che il punto si trova piuttosto distante dalla curva-limite. Qual è il MOD/COD indicato dal simbolo quadrato che giunge assai più vicino a questa curva ? La risposta a questa domanda sarà data nel prossimo capitolo, quando esamineremo i più performanti schemi di codifica di canale che raggiungono quasi, come dimostra la Fig. 4.20 il limite di Shannon. Vediamo adesso come sistematizzare questo tipo di confronto a un qualunque MOD/COD.

In generale, per un formato di modulazione lineare I/Q con M punti nella costellazione, velocità di segnalazione R_s symb/s, e con sagomatura di trasmissione SRRC (Square-Root Raised Cosine) avente roll-off factor α , il bit-rate di informazione è $R_b = R_s \cdot \log(M) \cdot r$ (r tasso del codice), e la banda (radio) è $B = (1 + \alpha)R_s$. Dunque

$$\frac{R_b}{B} = \frac{r \cdot \log(M)}{1 + \alpha} \tag{4.104}$$

Nel caso ottimale di $\alpha = 0$, cioè con segnalazione alla banda di Nyquist, l'efficienza spettrale è pari a rlog(M), cioè proprio il numero di bit (di informazione) N_b associati ad ogni simbolo di costellazione.

Un caso limite interessante è quello di segnali cosiddetti *spread-spectrum*, per i quali la banda a disposizione è molto più grande del bit-rate sul collegamento (cioè $B \gg R_b$): la relazione della capacità dice che è possibile una trasmissione con equivocazione arbitrariamente piccola (cioè in pratica senza errori) purché

$$\frac{E_b}{N_0} \ge \left(\frac{E_b}{N_0}\right)_{\min} \stackrel{\triangle}{=} \lim_{R_b/B \to 0} \frac{2^{R_b/B} - 1}{R_b/B} = \ln 2 \cong -1.6 \text{dB}$$
(4.105)

Per confrontare l'efficienza di un formato MOD/COD con la capacità di canale, si comincia fissando un valore finito della P(E) sul bit ritenuta sufficientemente piccola per l'applicazione, e si calcola poi il valore di E_b/N_0 necessario per ottenere tale prestazione con il MOD/COD assegnato, valore che chiameremo *efficienza energetica*. Se fissiamo in base ai requisiti di un'applicazione $P(E) = 10^{-5}$, dalla curva del ricevitore ottimo su AWGN per una QPSK non codificata si ottiene il valore $E_b/N_0 = 9.6$ dB. Questo MOD/COD trasmette N_b2 bit/simbolo (costellazione quaternaria) e quindi l'efficienza spettrale è $R_b/B=2$. La modulazione QPSK senza nessuna codifica corrisponde dunque al punto (2,9.6) sul piano di Shannon, come indicato. La modulazione BPSK corrisponderà al punto (1,9.6) e si allontana quindi dalla curva limite nella direzione orizzontale, cioè in quella dell'*efficienza spettrale*.

Esempio 4.27

Valutiamo i punti rappresentativi di diverse MOD/COD usate in pratica.

i) Una QPSK con codice convoluzionale avente tasso di codifica r = 1/2 ha la stessa efficienza spettrale della BPSK senza codifica di canale, cioè 1 bit/simbolo. Per il codice convoluzionale standard r = 1/2, K = 7 delle trasmissioni televisive DVB si ha un "guadagno di codifica" pari a 5.2 dB per una $P(E) = 10^{-5}$. Ciò significa che il punto rappresentativo di una QPSK codificata con tale codice è (1,4.4).

ii) Per una modulazione 8PSK con codifica a traliccio di Ungerböck, l'efficienza spettrale è quella di una QPSK non codificata, cioè 2 bit/s/Hz, ma c'è un guadagno nella codifica a traliccio che, per il codificatore a 32 stati ottimo è di circa 3.2 dB. Il punto rappresentativo è dunque (2,6.4).

iii) Le modulazioni di tipo M-QAM con costellazione quadrata incrementano l'efficienza spettrale della trasmissione, ma penalizzano l'efficienza energetica di circa 6 dB per ogni quadruplicazione dei punti della costellazione. Sulla Fig. 4.23 abbiamo indicato il punto (4,13.5) rappresentativo della 16-QAM.

Scegliendo opportunamente il formato MOD/COD si può "percorrere" tutta la curva di capacità adattando il formato alle caratteristiche del canale: maggiore efficienza spettrale se il rapporto segnale-rumore è elevato, minore efficenza spettrale se il livello di protezione deve essere maggiore perché lo SNR non è buono. In questo modo si implementa la cosiddetta tecnologia ACM (Adaptive Coding and Modulation) tipica degli standard di comunicazione digitale moderni.

4.6.1 Separazione Sorgente/Canale

A chiusura di questo capitolo sulla codifica di canale, ripensando alla codifica di sorgente del capitolo precedente, sembra che manchi tra i due argomenti un anello di congiunzione. Il dover introdurre una certa quantità di ridondanza per costruire un codice di canale può sembrare contrastante all'esigenza, ampiamente sottolineata nel capitolo precedente, di rimuovere il più possibile la ridondanza di sorgente. Una domanda del tutto legittima è infatti la seguente: prima di rimuovere completamente la ridondanza di sorgente, non sarebbe più giusto o conveniente toglierne una quantità *controllata*, in modo da lasciarne un "giusto" residuo che adatti la sorgente al canale con disturbo? Questo equivale al dire che si dovrebbe/potrebbe fare una codifica di sorgente e canale *congiunta* (joint source/channel coding) - in ultima analisi infatti ciò che si richiede per una trasmissione affidabile è $H(S) \leq c$.

La risposta a questa domanda è *NO*, *non conviene*. A parte la difficoltà pratica di sviluppare algoritmi congiunti che siano ottimali sia nei riguardi della sorgente che del canale, nel caso semplice che abbiamo qui discusso di sorgenti e canali senza memoria, non vi è alcuna necessità di codificare congiuntamente: possiamo prima (attraverso il teorema della codifica senza perdita) rappresentare fedelmente con ridondanza tendente a zero l'informazione di sorgente e poi, fatto questo, possiamo comunicare affidabilmente tale rappresentazione sul canale rumoroso con un opportuno codice di canale. In altre parole,

156 ALLA RICERCA DELL'AFFIDABILITÀ PERDUTA

la codifica congiunta ottima sorgente/canale è... la codifica *disgiunta*. Questa proprietà rimane anche per casi più complicati e/o realistici, cioè sorgenti e canali con memoria: è sempre possibile arrivare alla comunicazione fedele dell'informazione di sorgente su canale rumoroso con affidabilità arbitraria semplicemente usando codifiche separate. Benché espresso in forma colloquiale, questo è un *terzo* teorema di Shannon (implicitamente dimostrato attraverso i precedenti due) che prende il nome di "Teorema della separazione sorgente-canale".



CAPITOLO 5

DISTORC3R3 MA NON TROPPO - CODIFICA DI SORGENTE CON PERDITA



Leonardo da Vinci (1452-1519), "La Gioconda", riproduzione facente uso di caratteri stampabili dell'alfabeto ASCII

-Popular ASCII Art

L'immagine in "arte ASCII" riproduce con un certo grado di fedeltà (o distorsione che

Comunicazioni Digitali, I Edizione. di M. Luise Copyright © 2023 University of Pisa 158 DISTORC3R3 MA NON TROPPO - CODIFICA DI SORGENTE CON PERDITA

dir si voglia) il celebre quadro di Leonardo. Per certi usi questa riproduzione può essere più che sufficiente, e certamente rappresenta la sorgente di informazione con molti meno caratteri...

5.1 Perché codificare con perdita?

Già nella sez. 3.1 abbiamo evidenziato che spesso il bit-rate netto di sorgenti digitali come l'HD video è troppo grande per essere trattato senza riduzione. La codifica senza perdita (talvolta chiamata entropica) permette di ridurre la dimensione dei dati da trattare, ma non consente di realizzare grandi rapporti di riduzione - al massimo qualche unità e in casi particolari e fortunati.

Un diverso approccio è quello di procedere alla codifica di sorgente con perdita di informazione (lossy) che non è (più) soggetto al vincolo dell'entropia e consente di ottenere forti tassi di compressione. In particolare, da questo Capitolo emergerà la proprietà generale, già chiara intuitivamente, di poter guadagnare in tasso di compressione della sorgente a patto di sacrificare la fedeltà nella rappresentazione dell'informazione (cioè aumentare la perdita di informazione).

A questo scopo saranno indispensabili i concetti di informazione mutua ed equivocazione che sono stati introdotti a proposito della comunicazione/memorizzazione affidabile su canale con disturbo nel Cap. 4.

Codifica di Sorgente con una Misura di Distorsione 5.2

Se indichiamo con $\Omega = \{s_1, s_2, ..., s_M\}$ l'alfabeto di sorgente e con $\Delta = \{d_1, d_2, ..., d_L\}$ l'alfabeto di codice (dizionario), il codice è una corrispondenza $\omega : \Omega \to \Delta$ che associa a gruppi di simboli di Ω gruppi di simboli di Δ . Esempi di questa situazione sono i codici di Huffman o il Lempel-Ziv universale. Ma con questi codici il fattore di compressione è limitato dall'entropia della sorgente. Lo scopo della codifica con perdita è quello di trovare un codice la cui lunghezza media è inferiore all'entropia della sorgente, in modo che l'informazione possa essere trasmessa (o memorizzata) con una richiesta di capacità ridotta, ma con perdita di fedeltà nella rappresentazione dell'informazione di sorgente.

Quando si procede ad una codifica con perdita diventa allora indispensabile introdurre una misura della perdita di fedeltà associata alla codifica. L'immagine "Lena" (riportata in Fig. 5.1 (a)) può essere considerata come una sorgente di informazione discreta con simboli a 256 valori (i valori quantizzati su 8 bit dei livelli di luminosità di ogni pixel). L'immagine è rappresentata su 256×256 pixels, e in totale necessita di $256 \times 256 \times 8 = 524288$ cifre binarie (65536 byte). Tale sorgente può essere codificata su di una stringa di bit ben più corta con perdita accettabile come in Fig. 5.1 (b); intuitivamente, volendo ridurre ancora la lunghezza della rappresentazione si incrementerà la perdita di fedeltà fino a livelli inaccettabili. È chiaro che per stabilire che cosa è accettabile e che cosa non lo è si deve introdurre una *misura di distorsione* che ci permetta di giungere a corrette conclusioni.

Esaminiamo la questione della quantizzazione di un segnale e consideriamola come una codifica per ridurre la rappresentazione della sorgente originaria. Se indichiamo con S[n] il campione *n*-esimo di segnale e con D[n] = Q(S[n]) il risultato dell'operazione di quantizzazione, abbiamo una distorsione pari a

$$d[n] = D[n] - S[n]$$
(5.1)



Figura 5.1 Esempio di codifica con perdita

La *misura* della distorsione è in questo caso la potenza media (varianza) dell'errore di quantizzazione (la nostra distorsione) d[n] tra il valore del segnale codificato (quantizzato) e il valore originale:

$$\bar{d} \stackrel{\triangle}{=} E\left\{ \left(D[n] - S[n] \right)^2 \right\}$$
(5.2)

(5.3)

Ammettendo di quantizzare con un passo di quantizzazione q, sappiamo che risulta sotto certe ipotesi

$$\bar{d} = \frac{q^2}{12} = \frac{2A}{12 \cdot 2^{N_b}}$$

ove 2A è la dinamica del quantizzatore ed N_b è il numero di bit nella conversione. Chiaramente, all'aumentare di N_b diminuisce la distorsione totale \bar{d} , ma aumenta la lunghezza della rappresentazione della sorgente perchè aumenta il numero di bit/simbolo. La definizione appropriata di *distorsione* dipende fortemente dall'applicazione, ma in generale una distorsione intesa come errore quadratico medio si rivela di applicabilità quasi universale, ed è usata in campo audio e dell'elaborazione delle immagini (almeno in prima approssimazione).

Per poter calcolare una misura di distorsione totale di una codifica con perdita si devono distinguere le funzioni di *codifica* e *decodifica*. Nella codifica senza perdita, queste due operazioni sono perfettamente speculari e la decodifica è direttamente determinata dalla codifica stessa (trasformazione invertibile). Per le codifiche con perdita viceversa, la decodifica *non* è *univoca* proprio per la possibilità di scambiare fedeltà e complessità tipica delle codificato MPEG: la ricostruzione della sorgente può essere di migliore o di peggiore qualità a seconda del decodificatore che si adopera (software o hardware, a basso o alto costo ecc.).

La codifica è dunque una mappa ω da $\Omega = \{s_1, s_2, ..., s_M\}$ in un qualche alfabeto intermedio del tutto inessenziale, il cui scopo è quello di trasmettere o di memorizzare l'informazione. La decodifica è poi una seconda applicazione \mathcal{D} da questo alfabeto intermedio a $\Delta = \{d_1, d_2, ..., d_L\}$, cioè l'alfabeto di codice. Spesso Δ è scelto, per semplicità, come un sottoinsieme di Ω (si pensi al caso banale della riduzione del numero di bit dei valori di un segnale con una quantizzazione più rozza).

Limitandoci dunque al caso di sorgenti discrete senza memoria, la distorsione è una *funzione non negativa* della coppia (simbolo di sorgente, simbolo di codice):

$$d(s_m, d_k) \tag{5.4}$$

160 DISTORC3R3 MA NON TROPPO - CODIFICA DI SORGENTE CON PERDITA



Figura 5.2 Analogia tra compressione con perdita e trasmissione su canale rumoroso

che è tanto maggiore quanto più il simbolo di codice d_k è "lontano" o "meno fedele" in qualche senso rispetto al simbolo di sorgente s_m . Per il caso della (ri)quantizzazione, potremmo scegliere banalmente

$$d(s_m, d_k) = |s_m - d_k|^2$$
(5.5)

a patto di interpretare i simboli di sorgente e di codice come rappresentazioni in base due di valori di segnale. Una volta assegnata la coppia ω , \mathcal{D} è possibile caratterizzare la *distorsione media* del processo di codifica/decodifica. Conoscendo $\omega \in \mathcal{D}$ possiamo infatti, a partire dalle probabilità dei simboli di sorgente p_m , calcolare le probabilità condizionate dei simboli di codice

$$P(d_k|s_m) \tag{5.6}$$

e da queste la distorsione media

$$\bar{d}(\omega, \mathcal{D}) = \sum_{m=1}^{M} \sum_{k=1}^{L} P(s_m, d_k) d(s_m, d_k) = \sum_{m=1}^{M} \sum_{k=1}^{L} P(s_m) P(d_k | s_m) d(s_m, d_k)$$
(5.7)

L'operazione di codifica-decodifica è in pratica "riassunta" nei particolari valori delle probabilità $P(d_k|s_m)$. Questo sottolinea l'analogia tra la *codifica con perdita* e la *comunicazione su canale con disturbo* che abbiamo introdotto nel Cap. 4: entrambi i problemi sono caratterizzati dalla stessa descrizione, e cioè in quel caso la matrice delle "probabilità di canale", nel nostro caso la matrice delle "probabilità di decodifica" (matrice di decodifica). Questa analogia è ulteriormente ribadita dalla Fig. 5.2 dove il "disturbo" che nel canale appare esplicitamente è di fatto la distorsione introdotta dal codec (codificatore-decodificatore).

5.2.1 Curva Tasso-Distorsione

Stabilita adesso la misura di distorsione totale della sorgente, dobbiamo imporre il criterio di fedeltà. Dobbiamo cioè richiedere che questa distorsione sia minore di un valore \bar{d}_{max} richiesto dall'applicazione. Questa condizione potrà essere verificata o meno, a seconda dello schema di codifica-decodifica adottato: al limite, con una codifica senza perdita, la distorsione $\bar{d}(\omega, D)$ è nulla. Ciò che che determina il raggiungimento di quest'obiettivo,

cioè l'avere

$$d(\omega, \mathcal{D}) \le d_{\max} \tag{5.8}$$

è naturalmente la maggiore o minore fedeltà del codec, che, nella nostra descrizione probabilistica basata sulla "matrice di decodifica", si riflette in ultima analisi sui valori delle probabilità $P(d_k|s_m)$ che ω , \mathcal{D} implicano. Una quantizzazione troppo rozza ad esempio, porta errori di quantizzazione troppo alti che non consentono di soddisfare la condizione di una distorsione media accettabile.

Chiameremo *ammissibili* tutti quei codec, cioè tutti quegli insiemi di valori $\{P(d_k|s_m)\}$, m = 1, ..., M, k = 1, ..., L, per cui la distorsione è sufficientemente piccola:

$$\overline{d}\left(\{P(d_k|s_m)\}\right) \le d_{\max} \tag{5.9}$$

Per ciascuno dei set di $\{P(d_k|s_m)\}$ ammissibili si ha una *informazione mutua* tra la sorgente e la sua ricostruzione imperfetta pari come sempre a

$$I(S,D) = H(S) - H(S|D) =$$

$$\sum_{n=1}^{M} \sum_{k=1}^{L} P(s_m) P(d_k|s_m) \log \left(\frac{P(d_k|s_m)}{\sum_{m=1}^{M} P(s_m) P(d_k|s_m)}\right)$$
(5.10)

Questa informazione mutua, che naturalmente è sempre inferiore all'informazione della sorgente H(S), rappresenta la parte "residua" di H(S) stessa, cioè quella che è necessario preservare nell'operazione di codifica in modo da non sorpassare la distorsione prefissata - in ultima analisi rappresenta anche la "dimensione del file" sul quale eventualmente la sorgente codificata verrà memorizzata. Per ottimizzare il codec vorremmo allora che questa informazione mutua, rispettando il vincolo di distorsione, fosse la più *piccola* possibile, in modo da comprimere il più possibile la sorgente.

Per sorgenti digitali, definiamo il tasso di codifica

$$\eta_c \stackrel{\triangle}{=} \frac{I(S,D)}{H(S)} \le 1 \tag{5.11}$$

che risulterà tanto più piccolo quanto più la codifica è efficiente, e rappresenta appunto il fattore di riduzione nella rappresentazione della sorgente. Per caratterizzare la possibilità di codificare la sorgente con una data distorsione si definisce allora la *funzione tasso-distorsione* come

$$R(d_{\max}) \stackrel{\triangle}{=} \min_{\{P(d_k|s_m)\} \text{ ammissibili}} \eta_c = \min_{\{P(d_k|s_m)\} \text{ ammissibili}} \frac{I(S,D)}{H(S)} \le 1$$
(5.12)

Il teorema di Shannon della codifica con perdita dice che, fissata la distorsione massima ammissibile d_{\max} , è possibile trovare uno schema di codifica di sorgente con perdita avente tasso $R \ge R(d_{\max})$, mentre non è possibile scendere sotto tale valore del tasso per il livello di distorsione fissata. Il teorema risulta chiaro se si pensa che la funzione tasso-distorsione viene proprio ottenuta dalla minimizzazione dell'informazione mutua, individuando così quella condizione minima indispensabile al raggiungimento del livello di distorsione prefissata.

La funzione tasso-distorsione dice dunque qual è il minimo tasso di codifica per la sorgente data, ovvero qual è il massimo rapporto di compressione $1/R(d_{\max})$ per la sorgente, una volta fissata la massima distorsione tollerabile. Naturalmente, si può dimostrare



Figura 5.3 Andamento generale della curva tasso-distorsione

che $R(d_{\text{max}})$ è decrescente con d_{max} , e questo spiega perché in generale alti rapporti di compressione si possono realizzare soltanto con alti valori della distorsione. L'andamento tipico di una funzione tasso-distorsione è mostrata in Fig. 5.3. Volendo distorsione nulla, è indispensabile avere tasso unitario, perché la codifica deve interamente preservare l'informazione di sorgente. È interessante invece notare che esiste un certo valore di distorsione media limite, quello cui tende la curva quando si abbassa arbitrariamente il tasso stesso. Tale valore si ottiene supponendo informazione mutua nulla, cioè D[n] indipendente da S[n] e corrisponde ad una sorta di "codifica pessima" o di quella condizione che, nel caso di trasmissione su canale con disturbo, avevamo chiamato canale inutile.

5.2.2 Curve Tasso-Distorsione Notevoli

5.2.2.1 Sorgente binaria con distanza di Hamming Come esempio semplice di calcolo di una curva tasso-distorione consideriamo la consueta sorgente binaria senza memoria, e supponiamo di volerla comprimere mediante una strategia con perdita. L'alfabeto del decodificato sarà di nuovo binario, e il criterio di bontà della ricostruzione, e cioè la definizione di funzione di distorsione si basa semplicemente sulla diversità del bit decodificato rispetto a quello di sorgente, cioè si può considerare come distorsione un *errore di decodifica*. Questo è un criterio universale che prescinde dall'aspetto semantico della sorgente, cioè dal tipo di informazione che essa produce e dal modo in cui essa viene percepita dall'utilizzatore (audio, video, dati, voce ecc.).

Con questo criterio avremo

$$d(s_m, d_k) = s_m \oplus d_k \tag{5.13}$$

che è poi la familiare "distanza di Hamming". La distorsione media è

$$\bar{d}(\omega, D) = \sum_{m=1}^{M} \sum_{k=1}^{L} P(s_m) P(d_k | s_m) d(s_m, d_k)$$
$$= p P(1|0) + (1-p) P(0|1) = \Pr\{D[n] \neq S[n]\}$$
(5.14)



Figura 5.4 Curva tasso-distorsione per sorgente binaria con distorsione BER

pari cioè alla probabilità totale di un errore di decodifica, e vogliamo che sia inferiore ad una distorsione prefissata, cioè una probabilità di erronea decodifica prefissata P_e : $d(\omega, D) \leq d_{\max} = P_e$. Si noti l'analogia dell'operazione di codifica-decodifica con quella di trasmissione su di un canale rumoroso !

Come nel caso del canale rumoroso la minimizzazione diretta della mutua informazione è infattibile. Osserviamo però che

$$I(S,D) = H(S) - H(S|D) = \eta(p) - H(S|D) = \eta(p) - H(E|D)$$
(5.15)

dove $E[n] \stackrel{\triangle}{=} S[n] \oplus D[n]$ è l'errore di decodifica. L'uguaglianza tra le entropie condizionate viene dal fatto che, assegnato D[n], conoscere E[n] o conoscere S[n] è esattamente la stessa cosa perché dall'uno si passa in maniere biunivoca all'altro (sottolineiamo, assegnato D). D'altronde, come per tutte le entropie condizionate, $H(E|D) \leq H(E)$ e quindi

$$I(S, D) = \eta(p) - H(E|D) \ge \eta(p) - H(E)$$
(5.16)

Poiché inoltre, per costruzione, $Pr\{D[n] \neq S[n]\} \leq d_{max} = P_e$ (con $P_e \leq 0.5$), segue che

$$I(S,D) \ge \eta(p) - H(E) \ge \eta(p) - \eta(P_e)$$
(5.17)

Si può dimostrare (esercizio per il lettore) che l'uguaglianza si ottiene nel caso in cui il codec è tale per cui

$$\Pr\{D[n] = 0\} = \frac{p - d_{\max}}{1 - 2d_{\max}} = \frac{p - P_e}{1 - 2P_e}$$
(5.18)

In conclusione

$$R(d_{\max}) = \frac{\eta(p) - \eta(d_{\max})}{\eta(p)} = 1 - \frac{\eta(d_{\max})}{\eta(p)}$$
(5.19)

Questo risultato vale naturalmente finchè $d_{\max} \le p \le 0.5$. Quando $d_{\max} = p$ il tasso di compressione è nullo e si ha il punto-limite della curva (perché ? Come si realizza la codifica "a tasso 0" in questo punto ?). La relativa curva tasso-distorsione è mostrata in Fig. 5.4 per due diverse sorgenti.

164 DISTORC3R3 MA NON TROPPO - CODIFICA DI SORGENTE CON PERDITA

5.2.2.2 Sorgente Gaussiana con errore quadratico Vediamo adesso di comprimere una sorgente Gaussiana a media nulla e con varianza σ_S^2 , cioè in pratica rappresentare il valore reale della sorgente S[n], in teoria ad infinite cifre decimali, su di un numero *finito* di bit. Come misura della distorsione utilizziamo il consueto errore quadratico medio

$$\bar{d} \stackrel{\triangle}{=} \mathbb{E}\left\{ |D[n] - S[n]|^2 \right\}$$
(5.20)

Sappiamo che, per quantità "soft"

$$I(S,D) = h(S) - h(S|D) = \sqrt{2\pi e\sigma_S^2} - h(S|D) = \sqrt{2\pi e\sigma_S^2} - h(E|D)$$
(5.21)

ove $E[n] \stackrel{\triangle}{=} D[n] - S[n]$ è l'errore di decodifica, e dove l'uguaglianza vale secondo lo stesso criterio già visto per la variabile binaria (assegnato D, l'incertezza su S è pari a quella su E). Seguendo la falsariga di quanto visto per la variabile binaria,

$$I(S,D) = \log\left(\sqrt{2\pi e \sigma_S^2}\right) - h(E|D) \ge \log\left(\sqrt{2\pi e \sigma_S^2}\right) - h(E)$$
(5.22)

Consideriamo adesso il vincolo sulla distorsione, in particolare imponendo il valore limite $\bar{d} = d_{max}$. Dalla (5.20) è chiaro che $\bar{d} = d_{max} = \sigma_E^2$, quindi stiamo di fatto imponendo il valore della varianza dell'errore. Sotto questo vincolo, la h(E) è massima, e quindi la I(S, D) (5.21) è minima, quando E[n] è Gaussiano con la prescritta varianza d_{max} . In questo caso

$$I(S,D) \ge \log\left(\sqrt{2\pi e \sigma_S^2}\right) - \log\left(\sqrt{2\pi e d_{max}}\right) = \frac{1}{2}\log\left(\frac{\sigma_S^2}{d_{max}}\right)$$
(5.23)

Prima di concludere il calcolo dobbiamo chiederci se il bound è effettivamente raggiungibile, cioè se esiste una $f_{D|S}(d|s)$ che permette di ottenere l'uguaglianza nella (5.23). Osserviamo che il bound introdotto nella (5.22) vale con l'uguaglianza quando E[n] è *indipendente* da D[n]. Questa condizione, insieme con l'avere E[n] Gaussiano, si realizza sicuramente se possiamo esprimere D[n] nella forma D[n] = S[n] + E[n] con E[n], appunto, indipendente da S[n] e quindi anche da D[n] - abbiamo una perfetta analogia con il canale rumoroso AWGN. In questo caso,

$$f_{D|S}(d|s) = \frac{1}{\sqrt{2\pi d_{max}}} e^{-\frac{(d-s)^2}{2d_{max}}}$$
(5.24)

In questa versione per sorgenti "soft" non ha senso normalizzare la I(S, D) rispetto alla h(S) perché quest'ultima *non* rappresenta il contenuto informativo della sorgente (cioè la "dimensione" del file in formato raw, che per la sorgente soft sarebbe infinita). La curva tasso distorsione è allora data semplicemente dalla minima I(S, D):

$$R(d_{\max}) = \frac{1}{2} \log \left(\frac{\sigma_S^2}{d_{max}}\right) \quad \text{bit/symb} \quad , \quad d_{max} \le \sigma_S^2 \tag{5.25}$$

Invertendo questa relazione, si ottiene

$$\frac{\sigma_S^2}{d_{max}} = 2^{2R} \quad \rightarrow \quad \frac{\sigma_S^2}{d_{max}} \bigg|_{dB} = 2R \log_{10}(2) \simeq 6R \tag{5.26}$$

rappresentata in alcuni casi esemplificativi in Fig. 5.5. Possiamo interpretare la quantità σ_S^2/d_{max} come il *rapporto segnale-distorsione*, e la relazione (5.26) diventa allora la familiare *legge dei 6 dB* che si riscontra in pratica anche nella conversione analogico-digitale: aumentare di un bit la lunghezza della rappresentazione del valore di sorgente (binary wordlength) porta ad un aumento di 6 dB del rapporto segnale-distorsione.





5.2.3 Dalla teoria alla pratica

Nel resto del Capitolo vedremo alcuni schemi pratici popolarissimi, e direi "di bandiera" della compressione di sorgente con perdita, in particolare per la compressione di immagini, di video, e di audio ad alta qualità. Più che costituire un riferimento o una dare descrizione dettagliata, l'intento è quello di far capire la complessità insita in questi algoritmi quando la misura di distorsione è *percettiva*, cioè strettamente legata alle caratteristiche della percezione umana. Il destinatario finale dell'informazione semantica collegata alle immagini, video o audio è nella grande maggioranza dei casi un essere umano (escluse applicazioni di visione o sorveglianza automatica audio/video con rilevamento di eventi, ad esempio), e quindi la misura della distorsione del codec è data dall'opinione che, in media, l'utente ha della bontà della ricostruzione della sorgente.

Per questi schemi, il contenuto *semantico* della sorgente (cioè il *tipo* di informazione che rappresenta) è, come vedremo, fondamentale. Non ha senso cioè cercare di comprimere attraverso un algoritmo JPEG-like un file audio con un brano musicale, perchè il tipo di compressore JPEG, con la misura di distorsione strettamente legata alla qualità percepita di un'immagine, è totalmente inadatto per un flusso audio.

5.3 Codifica delle immagini digitali in formato JPEG

Uno dei più popolari esempi di codifica con perdita è il formato di memorizzazione delle immagini standardizzato nei primi anni '90 dal comitato congiunto ISO/ITU-T chiamato JPEG (Joint Photographic Expert Group). Per dare un'idea delle tecniche utilizzate nello standard JPEG ci limiteremo a esaminare la codifica di immagini B/N a 256 livelli di grigio. L'alfabeto di sorgente ha dunque M=256 possibili simboli (ciascuno dei 256 livelli di grigio di ogni pixel) corrispondenti a una $H_{max} = 8$ bit/simbolo. Per convenienza, consideriamo la sorgente come bidimensionale, cioè i vari simboli S[n, m] sono identificati dagli indici n ed m corrispondenti alla posizione del pixel sull'immagine, ordinati rispettivamente da sinistra a destra e dall'alto verso il basso come in Fig. 5.6

La codifica JPEG impiega più stadi di compressione, dei quali i finali sono basati sulle stesse codifiche binarie senza perdita già viste nei paragrafi precedenti, e saranno

166 DISTORC3R3 MA NON TROPPO - CODIFICA DI SORGENTE CON PERDITA



Figura 5.6 Numerazione dei pixel di un'immagine



Figura 5.7 Quantizzazione rozza senza criterio percettivo - 4 bit/pixel (a), 2 bit/pixel (b)

esaminati in breve successivamente. I primi stadi di compressione sono invece *con perdita* e conferiscono grande efficienza di compressione. Essi sono basati su considerazioni di carattere *percettivo* (perceptual coding): l'occhio umano, nel vedere un'immagine è molto sensibile a cambiamenti nella luminosità dell'immagine su scala relativamente grande, corrispondenti a componenti di *frequenza spaziale* bassa nella decomposizione di Fourier dell'immagine stessa. Viceversa, la sensibilità a dettagli "fini" apportati dall frequenze spaziali alte nello spettro dell'immagine è inferiore.

Come è possibile sfruttare questo dato di fatto ? Già sappiamo che con la "quantizzazione rozza" di un segnale nella sua forma originale la degradazione di qualità è forte ed evidente (vedi Fig. 5.7) e questo si verifica sia per le immagini che, come vedremo, anche per i segnali audio e video. Le considerazioni di cui sopra (evidentemente suffragate da misurazioni di carattere *soggettivo*) suggeriscono allora che si può effettuare una *quantizzazione selettiva in frequenza* con una perdita di fedeltà modesta e controllabile. La quantizzazione non avviene direttamente sul segnale in modo indiscriminato, ma sulla *trasformata* del segnale stesso. In particolare, le componenti di "bassa frequenza" che sono le più importanti nella sintesi del segnale sono codificate con alta fedeltà, e cioè con un alto numero di bit. Viceversa, le componenti ad "alta frequenza" che stabiliscono i "dettagli" del segnale subiscono una quantizzazione molto più rozza con un contributo modesto di distorsione dell'immagine. In
CODIFICA DELLE IMMAGINI DIGITALI IN FORMATO JPEG 167



Figura 5.8 Segmentazione JPEG dell'immagine

un certo senso, le basse frequenze portano più informazione per l'occhio e quindi richiedono più bit nell'operazione di codifica.

Preservare le componenti a bassa frequenza nello spettro e distorcere solo quelle ad alta frequenza è anche una maniera di sfruttare le "correlazioni" tra pixel adiacenti nell'immagine, cioè la "memoria" della sorgente. Questo forte grado di correlazione si riflette infatti sulla forma dello spettro di tipo "passa basso" che viene appunto preservata dallo schema di quantizzazione adottato.

Altra particolarità saliente dell'algoritmo JPEG è la "modularità" o caratteristica di "degradazione progressiva" (graceful degradation): aumentando il fattore di compressione (fino a 10 e più) si ottiene una progressiva e controllata perdita di fedeltà. Maggiore compressione si ottiene infatti "risparmiando" sempre più nella quantizzazione delle componenti ad alta frequenza, perdendo cioè sempre più le informazioni sul dettaglio dell'immagine, ma senza apprezzabilmente alterare lo scenario globale.

L'algoritmo di compressione comincia con il *segmentare* l'immagine in blocchi di 8x8 pixel. Questa dimensione è un compromesso tra la necessità di ingrandire i blocchi per meglio sfruttare le correlazioni tra pixel, senza far crescere troppo la complessità dell'elaborazione. Prendendo come esempio la codifica di "Lena" schematizzata in Fig. 5.8, ogni blocco corrisponderà e 64 byte di sorgente (8 bit/pixel, B/N). Si procede poi al calcolo della trasformata di questo segmento di sorgente. Trattandosi di un'immagine, la trasformata sarà bidimensionale, e non è una trasformata discreta di Fourier (DFT) standard. Il JPEG ha optato per la trasformata "coseno" discreta più adatta all'elaborazine delle immagini (DCT, Discrete Cosine Transform), per la quale esiste un algoritmo veloce e che produce valori reali immediatamente quantizzabili.

Per un generico segnale monodimensionale x[n] di durata N, la DCT si ottiene dalla DFT di un segnale $x_e[n]$ di lunghezza 2N ottenuto prolungando il segnale originario con i campioni "allo specchio" attorno all'istante temporale N - 1/2, cioè come in Fig. 5.9 in

168 DISTORC3R3 MA NON TROPPO - CODIFICA DI SORGENTE CON PERDITA





cui

$$x_e[n] = \begin{cases} x[n] & n = 0, ..., N-1 \\ x[2N-1-n], & n = N, ..., 2N-1 \end{cases}$$
(5.27)

Calcolando la DFT $X_e[u]$ di $x_e[n]$ si ha:

$$\bar{X}_{e}[u] = \frac{1}{2N} \sum_{n=0}^{2N-1} x_{e}[n] e^{-j2\pi \frac{nu}{2N}} = \frac{1}{2N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{nu}{2N}} + \frac{1}{2N} \sum_{n=N}^{2N-1} x[2N-1-n] e^{-j2\pi \frac{nu}{2N}} = \frac{1}{2N} e^{j\frac{\pi u}{2N}} \sum_{n=0}^{N-1} x[n] e^{-j\pi \frac{(2n+1)u}{2N}} + \frac{1}{2N} e^{j\frac{\pi u}{2N}} \sum_{n=0}^{N-1} x[n] e^{j\pi \frac{(2n+1)u}{2N}} = \frac{1}{N} e^{j\frac{\pi u}{2N}} \sum_{n=0}^{N-1} x[n] \cos\left(\frac{(2n+1)\pi u}{2N}\right)$$
(5.28)

Per evitare che compaia il termine $\exp\{j\pi u/2N\}$ (che non dipende dalla sequenza da trasformare) si può immaginare di "ritardare" la sequenza $x_e[n]$ di mezzo campione in modo che risulti simmetrica rispetto allo zero (e al punto (2N - 1)/2), e quindi con trasformata reale pari a

$$\bar{X}'_{e}[u] = \bar{X}_{e}[u]e^{-j\frac{\pi u}{2N}} = \frac{1}{N}\sum_{n=0}^{N-1} x[n]\cos\left(\frac{(2n+1)\pi u}{2N}\right)$$
(5.29)

Naturalmente, questa trasformata può essere considerata solo per u = 0, 1, ..., N - 1 in quanto, per qualunque sequenza x[n], si ha $\bar{X}'_e[N] = 0$, $\bar{X}'_e[n] = -\bar{X}'_e[2N-n]$ e quindi i valori per $u \ge N$ sono sempre ricavabili dai primi N. Infine, per normalizzazione (in particolare per ottenere una trasformazione tempo-frequenza *ortonormale* o *unitaria*) si moltiplicano per $\sqrt{2N}$ tutte le funzioni coseno oscillanti alle varie frequenze tranne quella oscillante alla frequenza u=0 che ha potenza *doppia* delle altre e deve essere moltiplicata solo per \sqrt{N} :

$$\bar{X}[u] = \sqrt{(2 - \delta[u])N} \cdot \bar{X}'_e[u] = \sqrt{\frac{2 - \delta[u]}{N}} \sum_{n=0}^{N-1} x[n] \cos\left(\frac{(2n+1)\pi u}{2N}\right)$$



Figura 5.10 Funzioni-base della DCT

$$u = 0, 1, \dots, N - 1 \tag{5.30}$$

ove $\delta[u]$ è la sequenza impulsiva unitaria.

Stante l'ortonormalità della trasformazione, la relazione inversa a quella data si trova immediatamente (la dimostrazione è lasciata per esercizio al lettore e si ricava facilmente esprimendo $x_e[n]$ come antitrasformata di $\bar{X}_e[u]$):

$$x[n] = \sum_{u=0}^{N-1} \sqrt{\frac{2-\delta[u]}{N}} \bar{X}[u] \cos\left(\frac{(2n+1)\pi u}{2N}\right)$$
(5.31)

Tornando al JPEG, per N=8 si ha

$$\bar{X}[u] = \sqrt{\frac{2 - \delta[u]}{8}} \sum_{n=0}^{7} x[n] \cos\left(\frac{(2n+1)\pi u}{16}\right) , \quad u = 0, 1, ..., 7$$
(5.32)

Passando poi al caso bidimensionale, se con S[n, m] si indica il valore della luminosità del pixel di coordinate n, m in un generico blocco (cioè il simbolo di sorgente n,m), la DCT è data da

$$\bar{S}[u,v] \stackrel{\triangle}{=} \frac{\sqrt{2-\delta[u]}\sqrt{2-\delta[v]}}{8} \sum_{n=0}^{7} \sum_{m=0}^{7} S[n,m] \cos\left(\frac{(2n+1)\pi u}{16}\right) \cos\left(\frac{(2m+1)\pi v}{16}\right)$$
$$u,v = 0,...,7 \tag{5.33}$$

Le funzioni "base" prodotto di coseni della DCT sono mostrate in Fig. 5.10 (quantizzate su 256 livelli di grigio).

Le componenti con $u \in v$ piccole, cioè quelle di bassa frequenza spaziale, sono quelle che hanno i coefficienti della DCT più grandi e portano più informazione. Ad esempio, la Fig. 5.11 mostra Lena codificata e decodificata usando solo il coefficiente $\overline{S}[0,0]$ e la relativa funzione base e denota che già con il solo coefficiente "continuo" della DCT si ha una parvenza dell'immagine originale. La Tab. 5.1 mostra inoltre i valori della DCT dei tre blocchi nell'immagine di Lena corrispondenti alle zone "soppraciglio", "occhio", "naso" indicate in Tab. 5.1 ed evidenzia la predominanza dei coefficienti di ordine inferiore.

170 DISTORC3R3 MA NON TROPPO - CODIFICA DI SORGENTE CON PERDITA



Figura 5.11Immagine ricostruita con la sola componente (0,0) della DCT

Original Group	DCT Spectrum	Quantization Error	
a. Eyebrow	d. Eyebrow spectrum	g. Using 10 bits	
231 224 224 217 217 203 189 196	174 19 0 3 1 0 -3 1	0 0 0 0 -1 0 0 0	
210 217 203 189 203 224 217 224	52 -13 -3 -4 -4 -4 5 -8	-1 0 0 0 0 0 0 -1	
196 217 210 224 203 203 196 189	-18 -4 8 3 3 2 0 9	0 0 0 0 0 0 0	
210 203 196 203 182 203 182 189	5 12 -4 0 0 -5 -1 0	0 0 0 0 0 0 0 0	
203 224 203 217 196 175 154 140	1 2 -2 -1 4 4 2 0	0 0 0 0 0 0 0 0	
182 189 168 161 154 126 119 112	-1 2 1 3 0 0 1 1	0 0 1 0 0 0 1 0	
175 154 126 105 140 105 119 84	-2 5 -5 -5 3 2 -1 -1	0 0 0 0 0 0 0 0	
154 98 105 98 105 63 112 84	3 5 -7 0 0 0 -4 0	0 0 0 0 0 0 0 0	
b. Eye	e. Eye spectrum	h. Using 8 bits	
42 28 35 28 42 49 35 42	70 24 -28 -4 -2 -10 -1 0	0 -3 -1 -1 1 0 0 -1	
49 49 35 28 35 35 35 42	-53 -35 43 13 7 13 1 3	1 0 -1 -1 0 0 0 -1	
42 21 21 28 42 35 42 28	23 9 -10 -8 -7 -6 5 -3	-1 -2 1 0 -2 0 -2 -2	
21 35 35 42 42 28 28 14	6 2 -2 8 2 -1 0 -1	-1 -2 -1 2 0 2 0 1	
56 70 77 84 91 28 28 21	-10 -2 -1 -12 2 1 -1 4	0 -2 1 0 0 1 0 0	
70 126 133 147 161 91 35 14	3 0 0 11 -4 -1 5 6	0 -4 -1 0 1 0 0 0	
126 203 189 182 175 175 35 21	-3 -5 -5 -4 3 2 -3 5	0 -2 0 1 -1 -1 1 -1	
49 189 245 210 182 84 21 35	3 0 4 5 1 2 1 0	-1 -3 1 1 1 -3 -2 -1	
c. Nose	f. Nose spectrum	i. Using 5 bits	
154 154 175 182 189 168 217 175	174 -11 -2 -3 -3 6 -3 4	-13 -7 1 4 0 0 10 -2	
154 147 168 154 168 168 196 175	-2 -3 1 2 0 3 1 2	-22 6 -13 5 -5 2 -2 -13	
175 154 203 175 189 182 196 182	3 0 -4 0 0 0 -1 9	-9 -15 0 -17 -8 8 12 25	
175 168 168 168 140 175 168 203	-4 -6 -2 1 -1 4 -10 -3	-9 16 1 9 1 -5 -5 13	
133 168 154 196 175 189 203 154	1 2 -2 0 0 -2 0 -5	-20 -3 -13 -16 -19 -1 -4 -22	
168 161 161 168 154 154 189 189	3 -1 3 -2 2 1 1 0	-11 6 -8 16 -9 -3 -7 6	
147 161 175 182 189 175 217 175	3 5 2 -2 3 0 4 3	-14 10 -9 4 -15 3 3 -4	
175 175 203 175 189 175 175 182	4 -3 -13 3 -4 3 -5 3	-13 19 12 9 18 5 -5 10	

 Tabella 5.1
 Segmento dell'immagine, sua DCT, ed errore di quantizzazione

CODIFICA DELLE IMMAGINI DIGITALI IN FORMATO JPEG 171

1	a. 1	Lov	v co	m	orce	sio	n
1	1	1	1	1	2	2	4
1	1	1	1	ı	2	2	4
t	1	1	1	2	2	2	4
t	1	1	1	2	2	4	8
1	1	2	2	2	2	4	8
2	2	2	2	2	4	8	8
2	2	2	4	4	8	8	16
4	4	4	4	8	8	16	16

 Tabella 5.2
 Tabelle di quantizzazione





La terza colonna di Tab. 5.1 mostra l'effetto della quantizzazione sulle componenti della DCT: queste vengono ridotte al numero di bit indicato, e poi viene calcolata la DCT inversa per ricostruire il blocco nell'immagine. I valori di tabella rappresentano la differenza tra il valore originale di sorgente dei pixel e quello ricostruito in questa maniera. Si nota che la perdita di fedeltà è progressiva e non è uniforme nello spazio. Naturalmente, nella Tab. 5.1 non è stato applicato l'artificio cardine del JPEG, che consente di ottenere alti rapporti di compressione e bassa distorsione, cioè la quantizzazione *non uniforme* in frequenza (quantizzazione percettiva). Sappiamo che il passo di quantizzazione può essere preso molto maggiore per le alte frequenze che per le basse frequenze, in modo che addirittura molti coefficienti "in basso a destra" nelle tabelle degli spettri di Tab. 5.1 vengano completamente annullati e non vengano addirittura codificati. In questo modo possiamo variare il "punto di lavoro" del compressore sulla curva tasso-distorsione di Fig. 5.3.

Il grado di compressione, e cioè il livello di quantizzazione può essere variato a seconda delle esigenze imponendo delle "tabelle di quantizzazione" come quelle di Tab. 5.2. Queste tabelle rappresentano il passo di quantizzazione frequenza per frequenza: chiamando $\Delta[u, v]$ il generico elemento di questa tabella, il valore della DCT quantizzata $\bar{S}_q[u, v]$ si ottiene come

$$\bar{S}_q[u,v] = \operatorname{int}\left(\frac{\bar{S}[u,v]}{\Delta[u,v]}\right)$$
(5.34)

Tenendo conto che comunque $\bar{S}[u, v] < 256$, è chiaro che $\Delta[u, v]=1$ significa "quantizzazione massimamente fine", mentre laddove si ha $\Delta[u, v]=256$ si impone di fatto l'annullamento del coefficiente quantizzato corispondente. È chiaro che il decodificatore deve sapere quale tabella di quantizzazione il codificatore sta usando.

I valori frequenziali quantizzati così ottenuti vengono poi "serializzati" in modo da costruire un flusso binario che può essere trasmesso o memorizzato su file. La "componente

DISTORC3R3 MA NON TROPPO - CODIFICA DI SORGENTE CON PERDITA



continua" dell'immagine, cioè il coefficiente $\bar{S}[0,0]$ è codificato in modo differenziale da blocco a blocco 8x8 poiché risulta lentamente variabile, cioè fortemente correlato, sull'immagine. Per i restanti coefficienti, il percorso di scansione è "a zig-zag diagonale" come in Fig. 5.12 in modo che i coefficienti ad alta frequenza su pochi bit (o annullati del tutto) vengono raggruppati alla fine della stringa. In questo modo, è possibile codificare la stringa serializzata con codifica delle ripetizioni, in modo da comprimere ancora la sorgente. Il passo finale è poi quello di applicare una codifica di Huffman per ottenere la stringa codificata finale (in realtà, codifica delle ripetizioni e di Huffman sono combinate in un'unica codifica di Huffman modificata di cui lo standard assegna la tabella).

Le procedure di codifica e di decodifica sono riassunte in forma di schema a blocchi nella Fig. 5.13. Il flusso di bit codificato ottenuto come sopra viene poi formattato in modo gerarchico per la memorizzazione come in Fig. 5.14. Un frame è tutta l'immagine da codificare, uno scan è la scansione di uno dei tre colori fondamentali (solo uno per immagini in B/N), un segment è un gruppo di blocchi, e ogni block è la codifica di un blocco 8x8 dell'immagine. Nell'intestazione (frame header) si trovano informazioni ausiliarie: dimensioni dell'immagine, numero di colori, tabella di quantizzazione (quindi grado di

172

CODIFICA DEI SEGNALI VIDEO DIGITALI IN FORMATO MPEG 173



Figura 5.15 Diversi livelli di distorsione

distorsione) ecc., nello *scan header* abbiamo il numero dei colori, la tabella del codice di Huffman per ogni colore e così via.

Un esempio di codifica JPEG a qualità, e quindi tasso, variabile è mostrato in Fig. 5.15 relativamente alla zona degli occhi dell'immagine "Lena". L'immagine (a) è l'originale a 8 bit/pixel, mentre la (b) e la (c) sono codificate rispettivamente con 0.8 e 0.18 (!) bit/pixel. Il compromesso di Fig. 5.3 tra tasso e distorsione è evidente.

5.4 Codifica dei segnali video digitali in formato MPEG

Il formato MPEG (Motion Picture Expert Group) è in realtà una *famiglia* di standard per la codifica di segnali digitali audio e video con qualità variabile e variabile bit-rate. Lo standard MPEG-1 è del 1993 e prevede codifica video a qualità medio-bassa (confrontabile con il formato analogico di videoregistrazione VHS) con audio stereo di qualità medio/alta, in modo da ottenere una velocità di trasmissione di 1.5 Mbit/s. Lo standard MPEG-2 (ITU-T H.262) emanato nel 1995 prevede invece qualità simile al video analogico PAL o superiore con audio stereo o multicanale 5.1 di alta qualità per una velocità di trasmissione tipica di 6 Mbit/s (nello standard variabile a seconda della qualità desiderata da 3 a 15 Mbit/s), e viene usato negli standard televisivi di trasmissione DVB (Digital Video Brodcasting). Le estensioni più recenti di questo standard sono l'MPEG-4/H.264 AVC (Advanced Video Coding) emanato nel 1999 per lo sviluppo dell'HDTV e lo HEVC (High-Efficiency Video Coding) del 2013 necessario per guadagnare un fattore pari circa al 50% sull'H.264-MPEG4 ed è tipico dell'UHDTV 4k.

La Fig. 5.16 rappresenta un segnale video costituito da una *sequenza temporale di immagini fisse*. Nello standard televisivo europeo con risoluzione standard (SDT) e aspetto 4:3, il segnale è campionato al ritmo di 25 immagini/s chiamate quadri (*frame*) e con una risoluzione di 720 × 576 pixel (standard CCIR 601). Ad ogni pixel sono associati 3 valori di intensità dei 3 colori primari Rosso, Verde e Blu (RGB) ciascuno rappresentato su 8 bit (i consueti 256 livelli) per un flusso totale lordo di informazione di $25 \times 720 \times 576 \times 3 \times 8 = 248.832$ Mbit/s, una velocità più che ragguardevole. Per l'HDTV e la UHDTV con aspetto panoramico 16:9 aventi risoluzione rispettivamente 1920×1080 e 3840×2160 pixel sappiamo già (sez. 3.1) che arriviamo a 1.2 e (con HDR) 6.2 Gbit/s. Queste velocità evidenziano chiaramente la necessità di una tecnica di compressione video molto efficiente, da ricercarsi nella categoria delle compressioni con perdita.

Il primo passo per la compressione del flusso di informazione si basa su considerazioni percettive e riguarda la struttura dell'immagine all'interno di ogni quadro. Dai 3 valori RGB si può passare (pixel per pixel) con la seguente trasformazione lineare, e quindi senza



Figura 5.16 Quadri in un flusso video

perdita di informazione, a tre valori equivalenti chiamati YUV

$$Y = 0.299R + 0.587G + 0.114B$$
$$U = 0.493(B - Y)$$
$$V = 0.877(R - Y)$$

Il segnale Y è chiamato *luminanza* e porta l'informazione dell'immagine ridotta in B/N. U e V costituiscono il segnale di *crominanza* che individua su di un diagramma bidimensionale il colore dei vari pixel. L'occhio umano è sensibile a dettagli fini dell'immagine per quanto riguarda la luminanza e molto meno per quel che riguarda la crominanza. I segnali U e V possono quindi essere filtrati per ridurne la banda spaziale e sottocampionati (cioè decimati) lungo le linee orizzontali e/o verticali per ridurne il bit-rate. In fase di decodifica, le componenti U e V sottocampionate verranno poi interpolate per ripristinare la risoluzione originale.

Il criterio di sottocampionamento è denotato con le sigle 4:2:2 per indicare sottocampionamento di un fattore 2 solo sulle linee orizzontali, e 4:2:0 per indicare sottocampionamento verticale e orizzontale dello stesso fattore. Ad esempio, nello standard DVB è usato il criterio 4:2:0 che porta la velocità di informazione a 25x720x576x8+25x360x288x2x8=124 Mbit/s, ancora molto grande. Le successive operazioni di compressione si basano sui criteri già esaminati per il formato JPEG per quanto riguarda la compressione delle immagini all'interno di ogni quadro, cioè per la codifica *intra-quadro*, e su concetti simili a quelli già esaminati a proposito della compressione delle sorgenti con memoria mediante tecniche ADPCM per quanto riguarda la codifica della successione delle varie immagini, cioè la codifica *inter-quadro*. In particolare, la codifica intra-quadro avviene quasi esattamente come nel JPEG, cioè attraverso DCT su blocchi di pixel, quantizzazione percettiva variabile in frequenza, scansione a zig-zag e codifica a lunghezza variabile di Huffman modificata. Dovremo invece esaminare più da vicino le tecniche di codifica ADPCM produce una stima del quadro corrente a partire dalla conoscenza dei quadri passati.

In un programma video tipico, si possono avere sequenze di immagini anche molto lunghe fortemente correlate: ciò accade finché la telecamera non cambia inquadratura o finché "carrella" molto lentamente sulla stessa scena. Questo suggerisce appunto di usare una modalità di codifica temporale di tipo *differenziale* in cui la codifica intra-quadro è



Figura 5.17 Predizione del moto

applicata non ai singoli fotogrammi, ma alle differenze tra il quadro attuale e la sua propria predizione calcolata sulla base dei quadri precedenti. Sappiamo già che questa procedura è efficiente nel caso di sorgente con forte correlazione temporale, come nel caso presente. Una maniera di effettuare la predizione è suggerita da una considerazione elementare: le immagini consecutive in una sequenza video correlata (cioè con camera quasi-stazionaria) sono costituite da uno sfondo stazionario o molto lentamente variabile con qualche forma in primo piano in movimento più veloce. La predizione viene quindi effettuata cercando di prevedere dove ogni "sottoblocco" di pixel dell'immagine (di dimensione dimensione 16×8 ad esempio in MPEG2) si sposterà nel quadro successivo. Questa procedura è schematizzata in Fig. 5.17.

Il movimento stimato del blocco viene ricavato dal codificatore correlando il sottoblocco del quadro all'istante t all'interno di una "finestra" di ricerca nel quadro t - 1 in modo da ricavare un vettore di moto (motion vector) che dovrà in decodifica essere applicato al blocco nel quadro t-1 per predire il quadro t. Nel codificatore, viene costruita in questa maniera la predizione del blocco t e viene ricavato l'errore di predizione da codificare intra-quadro. Come già visto nella sez. 3.1.3, l'immagine "di riferimento" all'istante t - 1 per il calcolo del vettore di moto e dell'errore di predizione non è la vera immagine da codificare, ma quella prodotta allo stesso istante dal decodificatore, in modo che quest'ultimo (che non ha a disposizione la vera immagine, ma solo la versione decodificata) possa correttamente avvalersi della predizione. Quindi nel codificatore si ha una replica del decodificatore per calcolare gli errori di predizione. L'effetto della predizione del moto si comprende appieno dall'esame di Fig, 5.18. L'immagine (a) è il quadro t sulla base del quale sono stati calcolati i vettori di moto rappresentati sul quadro t - 1 (b). L'immagine (c) è la differenza tra i quadri t e t-1 senza applicare la compensazione del moto, mentre l'immagine (d) è l'immagine errore che risulta usando la compensazione del moto, applicando cioè i vettori dell'immagine (b). La riduzione della varianza della predizione è evidente, e quindi la codifica mediante DCT sarà molto più efficiente.

Lo schema di codifica e decodifica risultante è rappresentato in Fig. 5.19 nel quale la predizione temporale (inter-quadro) e la codifica d'immagine DCT tipo JPEG (intra-quadro) sono combinate come già sottolineato. Il blocco FS è una memoria di quadro che mantiene il quadro precedente per predizione o decodifica. Questo schema è del tutto analogo a quello, basato sullo stesso criterio benché applicato a un segnale audio monodimensionale, di Fig. 3.3-3.4. La quantizzazione di quello è sostituita dalla codifica percettiva del quadro con quantizzazione selettiva di questo; il predittore a filtro FIR è sostituito dal predittore

DISTORC3R3 MA NON TROPPO - CODIFICA DI SORGENTE CON PERDITA



Figura 5.18 Immagine (a), Vettori di moto (b), Differenza senza predizione (c), Differenza con predizione (d)

video con i vettori di moto.

Per poter "editare" un programma video o comunque per poter accedere il programma da qualunque punto, si devono prevedere però dei "punti di ingresso" nella sequenza in corrispondenza dei quali sia possibile decodificare immediatamente le immagini senza avere a disposizione le informazioni sui quadri passati. Se infatti ci limitassimo a leggere un quadro "errore" rispetto al precedente ci limiteremmo a vedere qualcosa di simile alla Fig. 5.18 (d) senza poter ricostruire correttamente l'immagine. Questa fa sì che i quadri MPEG vengano divisi in tre "tipi" che determinano il modo di predizione utilizzabile. Il tipo "I" (Intra) è codificato senza predizione e senz'alcun riferimento a quadri precedenti, ma usando solo DCT intra-quadro e quindi in modo meno efficiente. Le immagini "I" sono intercalate periodicamente nel programma video e costituiscono i punti di ingresso appena menzionati. Le immagini di tipo "P" usano predizione rispetto ai precedenti quadri I o P e possono essere decodificati solo a partire da questi. È chiaro però che la codifica di un'immagine P richiede una quantità di informazione molto minore rispetto a una I. Ne segue che il flusso di bit generato dal codificatore MPEG è un flusso a velocità variabile. Questo spiega la presenza del Video Buffer (VB) in Fig. 5.17 il cui scopo è quello di "regolarizzare" la velocità del flusso dati in modo da poter essere trasmesso a velocità costante, e che può agire sulla procedura di codifica riducendo il bit-rate in casi di possibile traboccamento del buffer¹.

Esiste un terzo tipo di quadro, precisamente il "B" ove B sta per "bidirezionale": la predizione del moto all'istante t viene effettuata in modo non-causale (cioè diversamente da

¹Questa variabilità nelle velocità "istantanee" d'informazione rende però i flussi MPEG molto adatti ad una multiplazione statistica su reti IP.

176

CODIFICA DEI SEGNALI VIDEO DIGITALI IN FORMATO MPEG



Figura 5.19 Compressione video mediante predizione adattativa

quanto descritto finora) tenendo conto cioè sia del quadro P o I precedente (t-1) sia del P o I successivo (t + 1), e quindi la compressione di questi quadri è massimamente efficiente. Per non violare la causalità in decodifica però, bisogna alterare l'ordine naturale dei quadri: l'immagine B all'istante t deve essere trasmessa dopo il quadro t+1 che è servito per la predizione e che quindi servirà per la decodifica. Questo porta al concetto di gruppo di immagini (GOP, group of pictures), cioè di una struttura preordinata di immagini I, P e B in un ordine ben preciso adatto alla decodifica. Un esempio di GOP in ordine naturale potrebbe essere:

$$B_1 B_2 I_3 B_4 B_5 P_6 B_7 B_8 P_9 B_{10} B_{11} P_{12}$$

che deve essere trasmesso per non violare la causalità come segue

$$I_3B_1B_2P_6B_4B_5P_9B_7B_8P_{12}B_{10}B_{11}$$

Questa permutazione nell'ordine temporale, rappresentata sinteticamente per un caso differente in Fig. 5.20, introduce naturalmente un certo ritardo temporale la cui rilevanza deve essere valutata a seconda dell'applicazione. La Fig. 5.21 dettaglia maggiormente il criterio di segmentazione del flusso video MPEG (diremmo in termini antiquati, il tipo di scansione dell'immagine). Come si nota, ogni immagine all'interno di un GOP è "affettata" in più slice (fette) costituita da un certo numero di macroblocchi. Ogni macroblocco è costituito da 2x2 blocchi di 8x8 pixel cui viene applicata la DCT. Il formato risultante dei dati è quello di Fig. 5.22

Naturalmente lo standard MPEG comprende molti più dettagli di quanto non sia stato esposto in questo breve paragrafo, che è solamente relativo ad argomenti strettamente legati alla codifica di sorgente.

177

178 DISTORC3R3 MA NON TROPPO - CODIFICA DI SORGENTE CON PERDITA



Figura 5.20 Esempio di GOP



robieck contains loui fuinnairce and two chroinnairce oxo biecks of quantised bic ricolandik

Figura 5.22 Formato-base del file MPEG

5.5 Codifica dei segnali audio digitali in formato MP3

Lo standard MPEG, come già accennato, prevede anche la codifica di un segnale audio ad alta qualità stereo o multicanale che viene multiplato con il segnale video. Le codifiche audio MPEG possono però essere impiegate anche autonomamente per trasmissione e memorizzazione efficiente. Un esempio è il popolarissimo formato di codifica audio MP3. Questa sigla è la contrazione di MPEG-1 Layer 3, cioè il più efficiente dei 3 modi (layer) di codifica audio standardizzati all'interno dello MPEG-1. Senza entrare per il momento nei dettagli dei 3 modi, le prestazioni in termini di tasso di compressione sono le seguenti:

Compression Ratio	Bit-Rate
1:4	by Layer 1 (corresponds with 384 kbps for a stereo signal)
1:61:8	by Layer 2 (corresponds with 256192 kbps for a stereo signal)
1:101:12	by Layer 3 (corresponds with 128112 kbps for a stereo signal)

Tabella 5.3 MPEG1 Audio Layers

Ricordiamo che il flusso netto di un segnale audio stereo codificato PCM a standard CD senza compressione è di circa 1.4 Mbit/s. In particolare, il formato MP3 può ottenere tassi di compressione variabili e naturalmente qualità variabile, come è sintetizzato nella tabella che segue:

	and the second second second second	1. A		
sound quality	bandwidth (kHz)	mode	bit-rate (kbit/s)	compression ratio
telephone sound	2.5	mono	8	96:1
better than short-wave	4.5	mono	4.516	48:1
better than AM radio	7.5	mono	32	24:1
similar to FM radio	11	stereo	5664	2624:1
near-CD	15	stereo	96	16
CD	>15	stereo	112128	1412:1

 Tabella 5.4
 Qualità e rapporto di compressione di alcuni formati audio

Gli standard MPEG-1 di codifica audio con perdita, e in particolare il Layer 3 del formato MP3, si basano ancora una volta sulla codifica percettiva, implementando (ancora una volta...) una *quantizzazione variabile in ambito frequenziale*. Le caratteristiche psicoacustiche su cui si basano sono sostanzialmente due: i) la diversa sensibilità dell'apparato uditivo a suoni di diversa frequenza e ii) l'effetto di *mascheramento* di componenti frequenziali intense a una data frequenza sulle componenti più deboli alle frequenze vicine.

Per quanto riguarda il primo aspetto, studi di psicoacustica già negli anni '30 avevano messo in evidenza l'esistenza di una soglia di udibilità dei suoni variabile al variare della frequenza (curve di Fletcher-Munson), mostrata in Fig. 5.23. Questa grandezza è misurata in dB di pressione acustica (SPL, Sound Pressure Level), cioè dB rispetto alla pressione di riferimento pari a 20 Pa. Immaginando di eseguire un'analisi spettrale di un segnale audio dato, ciò che conta per rappresentare il livello delle componenti di segnale frequenza per frequenza è la differenza del valore dello spettro di ampiezza del segnale rispetto a questa curva. Quindi, componenti molto vicine a questa soglia o addirittura al di sotto di questa possono essere ignorate o quantizzate piuttosto rozzamente. Questo presuppone che il codificatore esegua un'analisi spettrale del segnale in tempo reale e, sulla base della



Figura 5.23 Soglia di udibilità dell'orecchio umano

curva di soglia, decida quali componenti scartare e quali codificare con più o meno bit a seconda della distanza dalla soglia stessa. Con questo criterio di quantizzazione percettiva si realizzano già apprezzabili fattori di compressione. Studi più recenti (anni '80) hanno anche portato alla identificazione di 24 *sottobande* dello spettro audio dette *critiche* sulla base delle quali l'orecchio analizza i segnali. Questi ragionamenti spiegano la struttura di massima del codificatore MPEG Layer 1-2 mostrata in Fig. 5.24 (a). Il primo passo nella codifica è l'analisi spettrale in tempo reale a 32 sottobande (un numero "comodo" vicino al 24 delle bande critiche dell'orecchio) con un banco di filtri paralleli ciascuno ottenuto per semplice traslazione frequenziale da un unico prototipo passa-basso H(z). In parallelo, il segnale viene anche analizzato con una FFT standard per ricavarne il contenuto energetico nella varie sottobande e, sulla base della curva di udibilità appena descritta, calcolare il criterio di quantizzazione (cioè di *allocazione dei bit*) in ogni sottobanda.

Per ottimizzare l'allocazione dei bit, e quindi aumentare ulteriormente il grado di compressione della sorgente rispetto al criterio della soglia di udibilità, viene sfruttato anche l'effetto di *mascheramento spettrale*. Se nello spettro del segnale audio esiste una componente con forte intensità, essa induce un "rialzamento locale" nella soglia di udibilità delle componenti adiacenti, come mostrato in Fig. 5.25. Tutte le componenti che ricadono sotto questa soglia rialzata possono allora essere ignorate nel processo di codifica, e quelle sopra soglia possono essere codificate con un numero di bit "piccolo", in modo che il relativo rumore di quantizzazione resti comunque *sotto* la soglia e sia sostanzialmente inudibile. È chiaro che questa strategia induce una allocazione di bit per la quantizzazione delle componenti frequenziali che varia in tempo reale sulla base delle caratteristiche "locali" dei vari segmenti di segnale audio che vengono analizzati.

Una volta calcolato il numero di bit da allocare in ciascuna sottobanda, si procede alla quantizzazione delle componenti nelle sottobande e alla relativa formattazione dei dati. Naturalmente, insieme ai bit di informazione devono essere multiplate anche le informazioni ausiliarie relativamente al grado di quantizzazione utilizzato in ogni sottobanda. Il decodificatore di Fig. 5.24 (b) ha una struttura molto più semplice: si limita ad estrarre dal segnale codificato le informazioni ausiliarie sulla quantizzazione, e ricostruisce il segnale con un banco di filtri di sintesi duale di quello di analisi nel codificatore, usando il passo di quantizzazione per ogni sottobanda ricavato dalle informazioni ausiliarie.

CODIFICA DEI SEGNALI AUDIO DIGITALI IN FORMATO MP3 181









Nell'MP3, cioè nello standard di Layer 3, la risoluzione dell'analisi spettrale viene aumentata fino a 576 sottobande (coefficienti) ricavati in due stadi in cascata di analisi. Come si nota dalla Fig. 5.26, il banco di 32 filtri dei Layer 1-2 viene realizzato con una struttura polifase che permette di effettuare, insieme al filtraggio, la decimazione (sottocampionamento) dei segnali in uscita di un fattore 32. Le uscite del banco di filtri vengono ulteriormente analizzate con una trasformata discreta coseno (DCT) con numero di coefficienti variabile da 6 a 18 a seconda del maggiore o minore grado di compressione e di fedeltà che si vuole ottenere. Dopo l'allocazione dei bit, è prevista infine una codifica entropica di Huffman (che non è mostrata in Fig. 5.24), per aumentare al massimo il tasso totale di compressione. Altri fattori sfruttabili per aumentare la compressione sono la codifica congiunta dei due canali stereo (che hanno fortissima correlazione mutua) che qui non considereremo.

Infine, menzioniamo che il formato audio/video MPEG4 (file extension .mp4) prevede un perfezionamento della codifica mp3 chiamato AAC (Adcanced Audio Coding). La principale differenza rispetto a mp3 è che l'analisi spettrale a sottobande viene effettuata solamente attraverso una DCT, piuttosto che da filtro polifase+DCT - il principio di funzionamento resta però il medesimo, con variazioni ai parametri dell'algoritmo che lo rendono più flessibile e performante.

Ulteriori approfondimenti sul tema della codifica con perdita sono al di fuori dello

182 DISTORC3R3 MA NON TROPPO - CODIFICA DI SORGENTE CON PERDITA



Figura 5.26 Banco di filtri di analisi

scopo di queste note e appartengono più al dominio del signal processing in generale e del multimedia signal processing in particolare che al filone dell'Information Theory.



CAPITOLO 6

CODIFICA DI VANALE CODIFICA YI CANALE CODIFICA DI CANALE



"Ripetere Aiuta"

-Motto latino

Questa celebre locuzione latina, di incerta origine, è alla base della forma più semplice di codifica di canale, e cioè il codice a ripetizione.

Comunicazioni Digitali, I Edizione. di M. Luise Copyright © 2023 University of Pisa

6.1 Capacità di Shannon e Fondamenti della Codifica di Canale

Lo scopo di questo capitolo è quello di discutere i principali schemi di codifica con prestazioni il più vicino possibile al limite di Shannon (4.103). Il teorema della codifica di canale visto in 4.4.3 non dice sfortunatamente come trovare questi codici; quasi 80 anni di ricerca in questo campo sono culminati però nella scoperta (o riscoperta) negli anni '90 delle famiglie di codici con decodifica iterativa, cioè i codici turbo ed LDPC e più recentemente ('10) dei codici polari. Ripercorreremo la sviluppo storico della tecnologia, partendo dai codici a blocco lineari, proseguendo con i convoluzionali, introducendo poi i turob/LDPC per finire con i codici polari.

6.1.1 Codici a Blocco e Decodifica Ottima

Un codice a blocco binario (N, K) con tasso r = K/N, K < N, è composto da due ingredienti. Per prima cosa, si deve identificare l'insieme delle 2^K stringhe, blocchi, o *parole di codice* \mathbf{a}_i , $i = 0, 1, ..., 2^K - 1$ composte da N simboli binari codificati. Tale insieme Γ è il *dizionario (codebook)* del codice. Il dizionario sarà costituito da una scelta opportuna di queste 2^K parole tra tutte le possibili 2^N stringhe binarie di lunghezza N. Successivamente, si deve definire la *mappa* che il codificatore userà per associare ciascuno dei 2^K blocchi di sorgente di lunghezza K (blocks) \mathbf{b}_i , $i = 0, 1, ..., 2^K - 1$, alle parole di codice preselezionate, esattamente come abbiamo fatto per il codice a ripetizione.

Una volta definito il codice come sopra, le varie parole di codice associate a blocchi di bit di sorgente vengono inviate su di un canale con disturbo e da quest'ultimo occasionalmente alterate. Il *decodificatore di canale* riceve questo input disturbato e cerca di identificare la parole del dizionario che verosimilmente è stata inviata, e risale al blocco di bit di informazione che erano associati a tale parola - in altre parole, esegue la *decodifica* del blocco.

La decodifica del semplice codice (N, 1) a ripetizione è semplice, ed è anche soggetta ad una simpatica interpretazione *geometrica* che ci insegnerà come progettare buoni codici e buoni decodificatori. Supponiamo di usare soltanto N = 3 ripetizioni. Dopo trasmissione sul canale BSC, ciò che otteniamo è un blocco di N = 3 simboli binari che può avere uno qualunque tra i $2^N = 8$ valori binari su 3 cifre. Sappiamo che non tutti questi valori appartengono al dizionario di codice, che in questo caso contiene solo le parole 000 e 111.

Possiamo rappresentare questa situazione nella Fig. 6.1 dove vediamo uno spazio N-dimensionale i cui punti rappresentano parole binarie N-dimensionali. I punti evidenziati in rosso sono le parole di codice, gli altri punti sono parole che possono essere ricevute in uscita dal BSC ma che contengono errori. Ad esempio, supponiamo di inviare il bit di informazione 1 (e quindi la parola di codice 111), e che si verifichi un errore sul primo simbolo codificato, cosicché la parola ricevuta sia 011. Il decodificatore a maggioranza deciderà correttamente perché tra tutte le possibile parole nel dizionario, notiamo dalla Fig. 6.1 che 111 (corrispondente alla presunta trasmissione di 1) è la *più vicina* a ciò che è stato ricevuto. Se al contrario si producono *due* errori di canale, cosicché riceviamo, ad esempio, 010, la parola più vicina diventa 000, e si produce anche un errore di decodifica.

Cerchiamo di formalizzare questa interpretazione geometrica, e indichiamo con:

$\mathbf{b} = [b[0], b[1], \dots, b[K-1]]$	blocco di codice
$\mathbf{a} = [a[0], a[1], \dots, b[N-1]]$	blocco di informazione
$\mathbf{d} = [d[0], d[1], \dots, d[N-1]]$	blocco in uscita al canale



Figura 6.1 Rappresentazione geometrica del codice a ripetizione con N = 3

Il compito del decodificatore è chiaro: osservare **d** (in generale \neq **a** perché contenente errori), e trovare la particolare parola di codice $\hat{\mathbf{a}}$ tra tutte quelle del dizionario Γ , la cui *probabilità a posteriori* $p(\hat{\mathbf{a}}|\mathbf{d})$ è massima (decodifica a massima probabilità a posteriori, MPA):

$$\hat{\mathbf{a}} = \underset{\tilde{\mathbf{a}} \in \Gamma}{\operatorname{argmax}} p(\tilde{\mathbf{a}} | \mathbf{d})$$
(6.1)

Il blocco di informazione decodificato $\hat{\mathbf{b}}$ si ottiene poi invertendo la mappa del codice a partire da $\hat{\mathbf{a}}$. Usando il teorema di Bayes, $p(\tilde{\mathbf{a}}|\mathbf{d}) = p(\mathbf{d}|\tilde{\mathbf{a}})p(\tilde{\mathbf{a}}/p(\mathbf{d}); \mathbf{d}'$ altronde $p(\mathbf{d})$ non dipende da $\tilde{\mathbf{a}}$, e tutte le $\tilde{\mathbf{a}}$ sono equiprobabili perchè tutti blocchi di informazione cui sono associate sono anch'essi equiprobabili (i bit di sorgente sono equiprobabili e indipendenti). Quindi il criterio originale MPA di decodifica (6.1) diventa il criterio equivalente di *massima verosimiglianza*, MV:

$$\hat{\mathbf{a}} = \operatorname*{argmax}_{\tilde{\mathbf{a}} \in \Gamma} p(\mathbf{d}|\tilde{\mathbf{a}})$$
(6.2)
$$\hat{\mathbf{a}} = \operatorname*{argmax}_{\tilde{\mathbf{a}} \in \Gamma} \ln p(\mathbf{d}|\tilde{\mathbf{a}})$$
(6.3)

Oltre ai blocchi **b**, **a**, **d** il cui significato abbiamo già discusso, conviene introdurre anche il vettore (blocco) degli errori di canale **e**, di dimensione N, che indica in che posizione il blocco ricevuto **d** presenta degli errori, cioè differisce dal blocco trasmesso **a**:

$$\mathbf{e} = \mathbf{a} \oplus \mathbf{d} \; , \; \mathbf{d} = \mathbf{a} \oplus \mathbf{e} \tag{6.4}$$

Poiché tali errori sono prodotti da un BSC, la probabilità che il generico elemento di **e** sia uguale a 1(0) è $p_e(1 - p_e)$, e inoltre gli elementi di **e** sono indipendenti perché il BSC è senza memoria.

Riprendiamo adesso il nostro criterio (6.3). Proprio perché gli errori di canale sono indipendenti

$$p(\mathbf{d}|\tilde{\mathbf{a}}) = p(d[0]|\tilde{a}[0]) \cdot p(d[0]|\tilde{a}[0]) \cdot \dots \cdot p(d[N-1]|\tilde{a}[N-1])$$
(6.5)

Osserviamo anche che il blocco **d** è l'input del nostro algoritmo di decodifica, cioè l'uscita del canale ricevuta, mentre **ã** "scandisce" tutto il dizionario. Tenendo conto della (6.4), è chiaro che nella (6.5) $p(d[n]|\tilde{a}[n]) = p(e[n]|\tilde{a}[n]) = p(e[n]) \forall n$, l'ultima uguaglianza giustificata dal fatto che gli errori sono indipendenti dal segnale. Allora la verosimiglianza (6.5) si ricava facilmente:

$$p(\mathbf{d}|\tilde{\mathbf{a}}) = p(\mathbf{e}|\tilde{\mathbf{a}}) = (p_e)^{N_{eq}} \cdot (1 - p_e)^{N_{neq}}$$
(6.6)

ovvero

dove p_e è la probabilità di un singolo errore di canale e dove abbiamo usato una notazione particolare: N_{eq} indica "il numero di elementi di **ã** che sono uguali ai corrispondenti elementi di **d**", mentre N_{neq} indica "il numero di elementi di **ã** che sono diversi dai corrispondenti elementi di **d**". Naturalmente, $N_{neq} = N - N_{eq}$ per cui

$$\ln p(\mathbf{d}|\tilde{\mathbf{a}}) = N_{neq} \ln p_e + (N - N_{neq}) \ln(1 - p_e)$$
(6.7)

Per maggior precisione, la quantità N_{neq} che indica il numero di elementi per i quali i due vettori binari **d** e $\tilde{\mathbf{a}}$ si differenziano è chiamata la *distanza di Hamming* $d_H(\mathbf{d}, \tilde{\mathbf{a}})$ tra **d** e $\tilde{\mathbf{a}}$.¹

Usando questa notazione,

$$\ln p(\mathbf{d}|\tilde{\mathbf{a}}) = -d_H(\mathbf{d}, \tilde{\mathbf{a}}) \ln \frac{1 - p_e}{p_e} + N \ln(1 - p_e)$$
(6.8)

Il secondo termine nella (6.8) non dipende da $\tilde{\mathbf{a}}$, e quindi, indipendentemente dal particolare valore della BER sul canale, il criterio (algoritmo) di decodifica sarà

$$\hat{\mathbf{a}} = \operatorname*{argmax}_{\tilde{\mathbf{a}} \in \Gamma} p(\mathbf{d}|\tilde{\mathbf{a}}) = \operatorname*{argmin}_{\tilde{\mathbf{a}} \in \Gamma} d_H(\mathbf{d}, \tilde{\mathbf{a}}) =$$
(6.9)

Tra tutte le parole di codice, il decodificatore deve trovare quella la cui distanza di Hamming dal blocco ricevuto **d** è minima - elettamente ciò che fa il decodificatore a maggioranza del codice a ripetizione: la nostra interpretazione geometrica inziale è stata pienamente confermata da un criterio teorico di ottimalità.

Questo criterio di "Trova la più vicina" è ottimale anche nel caso di trasmissione su canale AWGN con campioni di rumore indipendenti da simbolo a simbolo:

$$r[n] = a_{\pm}[n] + w[n] \quad \Rightarrow \mathbf{r} = \mathbf{a}_{\pm} + \mathbf{w} \tag{6.10}$$

dove $a_{\pm}[n]$ è la versione rimappata su di un alfabeto simmetrico ± 1 del simbolo di codice a[n]. Nell'esempio del codice a ripetizione abbiamo genericamente parlato di "distanza" alludendo sia alla *distanza di Hamming* che adesso conosciamo, ma anche alla distanza *Euclidea* nello spazio \mathbb{R}^N , anche perché nel sottospazio costituito dai vertici del cubo di Fig. 6.1 due punti vicini con Hamming sono anche vicini Euclidei.

Se l'ingresso del decoder è un vettore binario **d** (*hard-decoding*), l'unica distanza che ha senso è quella di Hamming. Ma se al contrario il decodificatore, come mostrato in Fig. 6.2, osserva un canale AWGN con filtro adattato e campionatore (*soft-decoding*), il blocco ricevuto $\mathbf{r} = [r[0], r[1], ..., r[N - 1]]$ è ad *ampiezza continua* e il punto N-dimensionale che lo rappresenta si troverà in *qualunque posizione* in \mathbb{R}^N , non necessariamente in una delle 2^N posizioni "binarie" ai vertici di un ipercubo come nell'hard-decoding. In ogni caso, il decoder deve trovare la parola di codice che risulta più vicina a ciò che è stato ricevuto, stavolta nel senso delle minimizzazione della distanza Euclidea

$$d_E(\mathbf{r}, \tilde{\mathbf{a}}) \stackrel{\triangle}{=} \|\mathbf{r} - \tilde{\mathbf{a}}\| = \sum_{n=0}^{N-1} (r[n] - \tilde{a}[n])^2$$
(6.11)

Esempio 6.28

¹La distanza di Hamming tra un generico vettore binario e il vettore nullo, cioè il numero di simboli pari a 1 nel vettore, viene chiamata *peso di Hamming* del vettore dato



Figura 6.2 Decodifica di un codice con ingresso soft

Il problema della decodifica soft è, omettendo qualche dettaglio preliminare,

$$\hat{\mathbf{a}} = \operatorname*{argmax}_{\tilde{\mathbf{a}} \in \Gamma} \ln f_{\mathbf{R}}(\mathbf{r}|\tilde{\mathbf{a}}) \tag{6.12}$$

(6.14)

dove $f_{\mathbf{R}}(\mathbf{r}|\tilde{\mathbf{a}})$ è la densità di probabilità (pdf) del vettore ricevuto condizionata a un particolare valore della parola di codice di tentativo del decoder. Con questo condizionamento, il valore di r[n] in ingresso al decoder é pari a $r[n] = \tilde{a}[n] + w[n]$ dove w[n] è una v.a. Gaussiana media nulla e varianza σ^2 . Tenendo conto che w[n] e w[m] sono statisticamente indipendenti con $n \neq m$, la pdf congiunta di \mathbf{r} è Gaussiana multidimensionale a componenti incorrelate:

$$f_{\mathbf{R}}(\mathbf{r}|\tilde{\mathbf{a}}) = \frac{1}{(2\pi\sigma^2)^{N/2}} \prod_{n=0}^{N-1} \exp\left\{-\frac{(r[n] - \tilde{a}[n])^2}{2\sigma^2}\right\}$$
(6.13)

cosicché

$$\ln f_{\mathbf{R}}(\mathbf{r}|\tilde{\mathbf{a}}) = C - \frac{1}{2\sigma^2} \sum_{n=0}^{N-1} \left(r[n] - \tilde{a}[n] \right)^2$$

dove $C = -N/2 \ln(2\pi\sigma^2)$ non dipende da **a**̃. Dunque si vede chiaramente che massimizzare ln $f_{\mathbf{R}}(\mathbf{r}|\mathbf{\tilde{a}})$ è equivalente a minimizzare la distanza Euclidea (6.11), indipendentemente dal valore della varianza di rumore.

Possiamo adesso chiederci: data una certa lunghezza delle parole N, qual è il "miglior" codice? Il criterio di decodifica a minima distanza suggerisce una risposta ovvia: quello per il quale le parole di codice sono "massimamente sparpagliate" cosicché risulta intuitivamente più difficile commettere errori di decodifica. Riprendendo in considerazione la Fig. 6.1, notiamo che 000 e 111 sono effettivamente le parole più lontane che si possono scegliere con N = 1. Un altro buon codice sarebbe stato quello con le parole 011 e 100, ad esempio (il decoder non sarebbe così semplice come quello a maggioranza).

Ciò che effettivamente conta nel progetto del codice è dunque la *distanza minima* tra due parole del codice. Questo parametro determina direttamente la BER del codice, perché risulta strettamente legato alla frequenza degli eventi errore nella decodifica. Se la distanza minima è grande, confondere una certa parola di codice con un'altra (la più vicina) richiede un disturbo relativamente grande, sicuramente pià grande di quello che causa errori in un codice in cui le parole hanno minima distanza maggiore.

Esempio 6.29

Ricaviamo la struttura del decoder ottimo per il codice a ripetizione sul canale AWGN soft-output dell'esempio 28.

Dobbiamo trovare la regola di decisione (decodifica) che individua quale tra le due possibili parole di codice 00...00 and 11...11 con il consueto mapping di a[n] su $a_{\pm}[n] \in \{\pm 1\}$ presenta la minima distanza Euclidea (6.11) rispetto al vettore ricevuto **r**:

$$b[0] = 0$$

$$\sum_{n=0}^{N-1} (r[n] - (-1))^2 \stackrel{<}{>} \sum_{n=0}^{N-1} (r[n] - (+1))^2 \qquad (6.15)$$

$$\hat{b}[0] = 1$$

Calcolando i quadrati ed eliminando i termini comuni ai due lati della diseguaglianza otteniamo

$$b[0] = 0$$

$$\sum_{n=0}^{N-1} r[n] < 0$$

$$\hat{b}[0] = 1$$
(6.16)

Il decodificatore *combina* tutti gli N soft inputs r[n] appartenenti ad un unico blocco (possiamo interpretare questa operazione come *smoothing*, cioè filtraggio, del rumore), ed esegue una decisione a soglia finale rispetto al livello 0. Lasciamo al lettore la verifica che questo decodificatore a ingresso soft, a parità di valore di E_s/N_0 , fornisce prestazioni di BER migliori rispetto al decodificatore a maggioranza (4.53).

Perchè esistono molte famiglie diverse di codici che si utilizzano in pratica? La questione principale sta negli algoritmi di decodifica. Secondo quanto appena discusso, sembra abbastanza semplice trovare buoni codici, nel senso di buoni *dizionari* contenenti parole lontane tra di loro. Tutt'altra cosa è trovare buoni *algoritmi di decodifica* che possano fornire buone prestazioni con *complessità ragionevole*. Parliamo di complessità perchè un decodificatore MV ottimo a minima distanza è sempre teoricamente implementabile calcolando tutte le 2^K distanze dal vettore ricevuto **r** ad ogni parola di codice **a** $\in \Gamma$ e trovando la minima. Tale decodificatore ha una complessità esponenziale in K che diventa proibitiva non appena K raggiunge qualche decina - improponibile.

Quindi progettare un codice significa sì trovare un buon codebook con buone proprietà metriche, ma soprattuto trovare un codebook che, attraverso particolari proprietà matematico/strutturali, permetta una algoritmo di *decodifica semplice*. Storicamente i primi esempi di (buoni) codici che seguono questa filosofia sonio i codici *parity check* e/o gli *Hamming codes*, inventati negli anni '50 e primi esemplari della grande classe dei *codici a blocco algebrici lineari* che introduciamo nella prossima sezione.

6.1.2 Codici a Blocco Algebrici

Sappiamo dalla precedente sezione, che il codice a ripetizione con K = 1, N = 3 e tasso r = 1/3, può correggere fino ad un singolo errore di canale in un blocco di codice. Questa performance potrebbe essere anche "troppo buona" nel senso che potremmo avere errori più sporadici di uno ogni tre simboli e/o potremmo riscontrare che il tasso di codifica di 1/3 è troppo basso (troppa ridondanza aggiunta). Potremmo chiederci allora come costruire un codice con buone performance di BER ma con un coding rate maggiore.

Un esempio semplice che va in questo senso è il vecchio codice a *doppio controllo di parità* (il cosiddetto *codice prodotto* inventato da Elias negli anni '50) che è ad oggi

0	1	1	1		0	1	1	1	1	
0	1	1	1		0	0	1	0	1	
0	0	1	0	(a)	1	1	1	0	1	(h)
1	1	1	0	(a)	1	1	1	0	1	(0)
1	Ο	0	1		1	0	0	1	0	
т	0	0	T		0	0	1	0		

Esempio di codice a doppio controllo di parità Figura 6.3

ancora in uso. Supponiamo di avere un blocco di sorgente di lunghezza $K = M^2$, in modo che si possano organizzare i bit di informazione in una tabella (matrice) quadrata come in Fig. 6.3 (a), dove $K = 4^2 = 16$. Il codificatore aggiunge alla matrice di bit i parity check orizzontali (risp. verticali) sulla destra (risp. al di sotto) della tabella. Ogni parity check orizzontale (verticale) si ottiene dalla somma modulo-2 degli elementi della riga (colonna) della matrice, in modo che la nuova riga (colonna) comprendente il check abbia un numero *pari* di 1 (donde il nome). La tabella finale è quella di Fig. 6.3 (b) che rappresenta la parola di codice a[n], $n = 0, ..., M^2 + 2M - 1$ da inviare sul canale. I bit di sorgente vengono inviati per primi nel loro ordine naturale, e i check vengono inviati successivamente in un qualunque ordine standardizzato, in modo che il codice è sistematico (cioè a[n] = b[n], n = 0, ..., K - 1), e la lunghezza del blocco di codice è $N = M^2 + 2M = 24$. Il coding rate è $r = M^2/(M^2 + 2M)$ che risulta molto vicino a 1 quando M (ovvero K) è grande. Nel nostro esempio, r = 2/3.

Il meccanismo di correzione (decodifica) del nostro codice è molto semplice e chiaro. Il decodificatore ricostruisce una tabella simile a quella di Fig. 6.3 (a) a partire dalla parola hard-detected **d**. In particolare, dopo aver riempito la parte sistematica $M \times M$ con i primi K simboli ricevuti, ricalcola localmente le parità e aggiunge questi valori (modulo-2) ai simboli di parità hard-detected ricevuti contenuti nella seconda parte di d stesso (N-K=2M simboli). Il simboli così risultanti (parità ricevute \oplus parità ricalcolate) costituiscono la cosiddetta sindrome. Se non vi sono errori nel blocco ricevuto, le parità ricalcolate localmente sono uguali a quelle ricevute e tutti i simboli della sindrome sono 0; il decodificatore non fa alcuna azione e inoltra i bit sistematici presumendo che siano corretti.

Supponiamo ora che vi sia un singolo errore su di un simbolo di sorgente come in Fig. 6.4) (a) (in grassetto): alcuni simboli della sindrome sono adesso diversi da 0 perchè le parità ricalcolate sono diverse da quelle ricevute (ancora Fig. 6.4) (a)). Si nota che i simboli della sindrome uguali a 1 marcano le "coordinate" del bit in errore, che può così essere identificato e corretto. Se il singolo errore si è verificato su di un bit di parità come in Fig. 6.4) (b), allora c'è un *solo* simbolo di sindrome uguale a 1, cosicché il decoder sa che l'errore è sul bit di parità ma che i bit di informazione sono in realtà corretti: non c'è bisogno di alcuna azione ulteriore.

Possiamo generalizzare il codice a controllo di parità? E in generale, possiamo costruire codici "buoni" e allo stesso tempo semplici da decodificare?

6.1.2.1 Definizione di codice algebrico lineare Cerchiamo di dare una risposta a queste domande introducendo il *campo* algebrico binario $\Omega_2 \stackrel{\triangle}{=} (\{0,1\},+,\cdot)$ cioè l'insieme delle cifre binarie con le operazioni modulo-2 di somma e prodotto. La definizione di Ω_2 induce anche l'ulteriore definizione dello spazio vettoriale finito N-dimensionale Ω_2^N : gli elementi

0	1	1	1	0		0	1	1	1	0	
0	0	1	0	0		0	0	1	0	1	
1	0	1	0	1	(a)	1	1	1	0	0	(b)
1	0	0	1	0		1	0	0	1	0	
0	1	0	0			0	0	0	0		





Figura 6.5 Definitione del codebook Γ con N = 6 e K = 3

di Ω_2^N sono le 2^N stringhe $\mathbf{v} = [v[0], v[1], ..., v[N-1]], v[n] \in \Omega_2$ che considereremo come vettori riga.

Un codice lineare a blocco (N, K) con K < N è un sottospazio vettoriale K-dimensionale Γ (il dizionario) di Ω_2^N contenente 2^K elementi (le parole di codice), come rappresentato nella Fig. 6.5 dove le parole di codice e di canale vengono rappresentate come punti in un ipotetico spazio Ω_2^N mappato sul piano. Con la usuale notazione $\mathbf{b_i} = [b_i[0], b_i[1], ..., b_i[K-1]]$ per il blocco di K bit di informazione, associamo ad ogni $\mathbf{b_i}$ una parola (blocco) $\mathbf{a_i} = [a_i[0], a_i[1], ..., a_i[N-1]] \in \Gamma, i = 0, ..., 2^K - 1$. Il tasso del codice è per costruzione r = K/N.

Che significa dire che Γ è un sottospazio vettoriale di Ω_2^N ? Significa che possiamo identificare K elementi linearmente indipendenti $\mathbf{g}_0, \mathbf{g}_1, ..., \mathbf{g}_{K-1} \in \Omega_2^N$ che rappresentano una *base lineare* di Γ , e cioè che ogni elemento (parola di codice) di Γ può essere decomposta come una combinazione lineare degli elementi della base. Sfruttando questa decomposizione possiamo immediatamente stabilire una mappa tra il generico blocco di informazione \mathbf{b} e la corrispondente parola di codice \mathbf{a} :

$$\mathbf{a} = b[0] \cdot \mathbf{g}_0 + b[1] \cdot \mathbf{g}_1 + \dots + b[K-1] \cdot \mathbf{g}_{K-1}$$
(6.17)

Poiché $\mathbf{g}_0, \mathbf{g}_1, ..., \mathbf{g}_{K-1}$ rappresentarono una base del sottospazio, blocchi di informazione differenti generano parole di codice differenti (a causa dell'unicità della scomposizione su base) e il codice è sicuramente (univocamente) decodificabile in assenza di errori di canale.

Dunque l'algoritmo di codifica (6.17) è il seguente: :

$$\mathbf{a} = \mathbf{b} \cdot \mathbf{G}_{ns} \tag{6.18}$$

dove abbiamo introdotto la matrice generatrice del codice

$$\mathbf{G}_{ns} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \dots \\ \mathbf{g}_{K-1} \end{bmatrix}_{K \times N}$$
(6.19)

Il rango della matrice (rettangolare) \mathbf{G}_{ns} è sicuramente pari a K perchè (tutte) le sue righe rappresentano una base del sottospazio Γ e quindi sono linearmente indipendenti.² Creare il sottospazio vettoriale di Ω_2^N attraverso \mathbf{G}_{ns} significa individuare le 2^K parole di codice, ovvero punti in Ω_2^N che dovranno essere massimamente distanti per fornire un buon codice.

Prima di concentrarci su distanze e decodificatori, osserviamo che il codice definito dalla (6.18), e in particolare la matrice generatrice \mathbf{G}_{ns} è *non sistematico* (donde i pedici). A partire da \mathbf{G}_{ns} è però sempre possibile ottenere un codice equivalente attraverso un procedimento di eliminazione di *Gauss-Jordan*, dopo la quale la matrice non sistematica viene ridotta alla forma *sistematica* seguente:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \stackrel{\bullet}{\bullet} \cdots \quad \stackrel{\bullet}{\bullet} \stackrel{\bullet$$

dove **P** è la cosiddetta matrice di *generazione della parità*. L'equivalenza tra i due codici significa che lo spazio vettoriale generato dalle due matrici è il medesimo, visto che le righe dell'una sono ottenute da combinazioni lineari delle righe dell'altra. Quindi il dizionario è esattamente lo stesso. Ciò che cambia è naturalmente la particolare mappa (corrispondenza) tra le parole di sorgente e quelle di codice, ma le proprietà di distanza restano le stesse.

La matrice sistematica **G** garantisce che il codice è sistematico esso stesso perché i primi K simboli della parola codificata sono la replica in chiaro dei K bit di informazione. Gli ulteriori N - K simboli binari che vengono aggiunti nella codiifca al blocco di codice sono i cosiddetti *bit di parità* che vengono generati dalla matrice **P** e conferiscono ridondanza al codice.

Esempio 6.30

Continuiamo con il banale codice a ripetizione. Per questa codifica, K = 1, e la matrice generatrice di dimensione $1 \times N$ è chiaramente $\mathbf{G} = \mathbf{g}_0 = \mathbf{1}_N$, cioè il vettore *N*-dimensionale costituito da tutti 1. La matrice è (già) sistematica e la matrice di generazione della parità è $\mathbf{P} = \mathbf{1}_{N-1}$.

Ricaviamo adesso la matrice generatrice per il codice double-parity check di Elias per il caso M = 2 e quindi K = 4, N = 8, r = 1/2. Immaginando che i simboli di parità

²Il rango di una matrice è per definizione il massimo numero di righe o colonne della matrice stessa che risultano linearmente indipendenti.

vengono inserito nel blocco a "leggendo" prima la colonna dall'alto in basso e dopo la riga da sinistra a destra, la lettrice può verificare che

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$
(6.21)

6.1.2.2 Matrice di Parity-check Insieme con il concetto di (sotto)spazio vettoriale Γ , nasce il concetto indotto di (sotto)spazio ortogonale (o complemento ortogonale) di Γ , che indichiamo con Γ^{\perp} , di dimensionalità N - K: esso contiene tutti gli elementi di Ω_2^N che risultano ortogonali a tutti gli elementi di Γ . Si dimostra facilmente che ogni elemento di Ω_2^N si può scomporre in modo unico come la somma di un blocco che appartiene a Γ più un altro blocco che appartiene a Γ^{\perp} .

Poiché (anche) Γ^{\perp} è uno spazio vettoriale, possiamo identificarne una base costituita dagli N - K elementi indipendenti $\mathbf{h}_0, \mathbf{h}_1, ..., \mathbf{h}_{N-K-1}$. Presa una qualunque parola di codice **a** in Γ , avremo

$$\mathbf{a} \cdot \mathbf{h}_i = 0$$
, $i = 0, 1, ..., N - K - 1$ (6.22)

Infatti ogni elemento di Γ è per definizione ortogonale a qualunque elemento di Γ^{\perp} , e ciò si verifica se e solo se la generica parola di codice è ortogonale a *tutti* gli elementi della base di Γ^{\perp} stesso. Preso un qualunque elemento **d** di Ω_2^N , in generale $\mathbf{d} \cdot \mathbf{h}_i \neq 0$ se $\mathbf{d} \notin \Gamma$. In generale possiamo scomporte **d** come somma di due componenti, la parola di codice $\mathbf{a} \in \Gamma$ e un termine ortogonale $\mathbf{e} \in \Gamma^{\perp}$, cosicché $\mathbf{d} = \mathbf{a} + \mathbf{e} \operatorname{con} \mathbf{a} \cdot \mathbf{h}_i = 0$ e $\mathbf{e} \cdot \mathbf{h}_i \neq 0$.

Le N - K equazioni (6.22) sono i *parity check* che generalizzano la nozione di controllo della parità che abbiamo già visto per il codice prodotto. Possiamo riassumere tutte queste equazioni come segue:

$$\mathbf{a}\mathbf{H}^T = \mathbf{0}_{N-K} \tag{6.23}$$

dove abbiamo introdotto la matrice di controllo della parità (parity check)

$$\mathbf{H} \stackrel{\triangle}{=} \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \dots \\ \mathbf{h}_{N-K-1} \end{bmatrix}_{(N-K) \times N}$$
(6.24)

La matrice di parity-check può essere utilizzata per effettuare il controllo degli errori sul vettore hard-detected **d** ricevuto all'uscita del canale con disturbo. Se non vi sono errori, $\mathbf{d} = \mathbf{a}$ (per quella certa $\mathbf{a} \in \Gamma$ che è stata inviata sul canale) cosicché

$$\mathbf{d}\mathbf{H}^T = \mathbf{0} \tag{6.25}$$

Se qualcuna della equazioni di controllo della parità falisce, sappiamo con certezza che il canale ha introdotto degli errori, e quindi possiamo svolgere la funzione di *rivelazione dell'errore* - è un po' questo il concetto fondamentale del codice algebrico: far sì che gli

errori introdotti dal canale vengano visti come elementi dello spazio ortogonale Γ^{\perp} , cioè $\mathbf{d} = \mathbf{a} + \mathbf{e}$ come già accennato.

Dobbiamo però capire come ricavare H a partire da G. La cosa divertente è che H viene gratis se la matrice generatrice è in forma sistematica:

$$\mathbf{H}_{(N-K)\times N} = \left[\mathbf{P}_{(N-K)\times K}^{T} | \mathbf{I}_{(N-K)\times (N-K)} \right]$$
(6.26)

Esempio 6.31

Ricaviamo la matrice di parity check del codice a ripetizione. Considerando (6.26) troviamo

$$\mathbf{H}_{(N-1)\times N} = \begin{bmatrix} \mathbf{1}_{(N-1)\times 1}^T | \mathbf{I}_{(N-1)\times (N-1)} \end{bmatrix}$$
(6.27)

che, nel caso particolare di N = 5, è

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{H}^{T} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(6.28)

Il significato delle N - 1 equazioni di parity-check è chiaro: ogni simbolo codificato a[n] con n > 0 deve essere uguale ad a[0].

La matrice di parity-check per il consueto codice (8,4) double parity check è invece banalmente

 $\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$ (6.29)

Prima di passare all'argomento della *decodifica* dei codici algebrici, vogliamo approfondire la questione delle proprietà in termini di distanza di un codice e della sua capacità di correzione degli errori. Sappiamo che il codice a ripetizione ha una distanza minima di Hamming tra le parole di codice pari ad $d_{min} = N$, e corregge fino a $(N-1)/2 = (d_{min}-1)/2$ errori di canale (N dispari). Questa proprietà vale in realtà per *qualunque* codice: il parametro-chiave che determina la capacità di correzione degli errori è proprio la *distanza minima* tra due parole di codice:

$$d_{min} \stackrel{\triangle}{=} \min_{\mathbf{a}_i, \mathbf{a}_k \in \Gamma} d_H(\mathbf{a}_i, \mathbf{a}_k) \tag{6.30}$$

Prendiamo come esempio l'arcinoto codice di Hamming (7,4) a controllo di parità

$$\mathbf{G}_{K \times N} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}_{4 \times 7}$$
(6.31)

$$\mathbf{H}_{N\times(N-K)}^{T} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{7\times3}$$
(6.32)

Come vedremo, questo codice ha distanza minima $d_{min} = 3$ e quindi corregge fino ad 1 errore di canale con un coding rate r = 4/7 - è quindi migliore del codice (8,4) double parity-check di Elias perché corregge lo stesso numero di errori (cioè 1) su 4 bit di sorgente, ma con una ridondanza *minore* (3 bit di check anziché 4).

La definizione di d_{min} (6.30) è chiarissima, ma può essere utilmente semplificata visto che Γ è uno spazio vettoriale, e che quindi combinando linearmente (cioè in pratica sommando) due parole di codice si ottiene una nuova parola di codice. Immaginiamo di aver effettuato la ricerca su tutto il codebook Γ prevista dalla (6.30) e di aver trovato la coppia di parole $\mathbf{a_1}, \mathbf{a_2}$ che stanno alla distanza minima d_{min} . Adesso sommiamo ad entrambe le parole una *stessa* parola di codice qualunque $\bar{\mathbf{a}}$, ottenendo due nuove parole di codice $\mathbf{a_3} = \mathbf{a_1} \oplus \bar{\mathbf{a}}, \mathbf{a_4} = \mathbf{a_2} \oplus \bar{\mathbf{a}}$. E' chiaro che

$$d_H\left(\mathbf{a_3}, \mathbf{a_4}\right) = d_H\left(\mathbf{a_1}, \mathbf{a_2}\right) = d_{min} \tag{6.33}$$

e che quindi in generale esistono *più* coppie di parole a distanza minima. Se poi, in particolare, scegliamo $\bar{\mathbf{a}} = \mathbf{a}_1$ abbiamo che

$$d_{min} = d_H \left(\mathbf{0}, \mathbf{a_5} \right) = w_H \left(\mathbf{a_5} \right) \tag{6.34}$$

dove $\mathbf{a}_5 = \mathbf{a}_2 \oplus \mathbf{a}_1$ è un'altra parola di codice ancora, e $w_H(\mathbf{a}_5)$ ne indica il *peso di Hamming* cioè il numero di elementi nella parola diversi da 0. La conclusione è che la d_{min} si può (anche) ricavare come minima distanza delle parole di codice dal vettore **0**, ovvero minimo *peso di Hamming* delle parole stesse:

$$d_{min} \stackrel{\triangle}{=} \min_{\mathbf{a}_i \in \Gamma} w_H(\mathbf{a}_i) \tag{6.35}$$

Se indichiamo con \mathbf{a}_{min} una parola di codice avente peso d_{min} , poiché (in quanto parola di codice) deve aversi $\mathbf{a}_{min}\mathbf{H}^T = \mathbf{0}_{N-K}$, segue che d_{min} è (anche) pari al minimo numero di righe della matrice di parity-check \mathbf{H}^T che devono essere sommate per dare il vettore $\mathbf{0}_{N-K}$.

I codici introdotti da Richard Hamming negli anni '50 sono stati il primo esempio di codici strutturati e facilmente decodificabili con proprietà di distanza *migliori* di quelle dei codici precedentemente sviluppati con criteri ad-hoc.

6.1.2.3 Decodificatore a sindrome La maniera più semplice e generale di decodificare i codici lineari come il codice di Hamming è basata sulla *decodifica a sindrome*. Come già sappiamo, se vi sono errori prodotti dal canale, in generale $\mathbf{da} \oplus \mathbf{e} \neq \mathbf{a}$ e per rivelare eventuali errori possiamo controllare la parola ricevuta mediante la matrice di controllo di parità:

$$\mathbf{d}\mathbf{H}^T = (\mathbf{a} + \mathbf{e})\mathbf{H}^T = \mathbf{e}\mathbf{H}^T \stackrel{\triangle}{=} \mathbf{s}$$
(6.36)

Il vettore (N - K)-dimensionale **s** risulta in generale non nullo proprio a causa degli errori di canale , viene chiamato la *sindrome* del blocco ricevuto, ed è la base per la rivelazione e correzione degli errori.

Dalla (6.36) vediamo che la sindrome è indipendente dalla particolare parola di codice che è stata trasmessa; è funzione solo del particolare vettore errore introdotto dal canale, e segnala la particolare equazione di controllo che fallisce. Purtroppo però la sindrome non segnala direttamente gli errori di canale, come si capisce facilmente: mentre si hanno 2^N diversi vettori errore (e quindi distribuzioni di errore nella parola ricevuta), ci sono soltanto 2^{N-K} sindromi distinte, per cui una stessa sindrome deve necessariamente corrispondere a vettori errori anche diversi. Di fatto, il procedimento di decodifica del codice (8,4) di Elias mostrato nella Fig. 6.4 è proprio basato sul calcolo della sindrome: i valori nei posti precedentemente occupati dai simboli di check sono proprio i valori della sindrome, che permettono di risalire al vettore errore.

Calcolata la sindrome, dobbiamo procedere con la decodifica, basata sul principio della minima distanza. Per far questo, possiamo preliminarmente immaginare di incolonnare tutte le 2^{K} parole binarie di lunghezza $N \mathbf{a}_{0}, ..., \mathbf{a}_{2^{K}-1}$, partendo da $\mathbf{a}_{0} = \mathbf{0}$. Questa sarà la prima colonna di una matrice \mathbf{D} che riempiremo progressivamente con *tutte le possibili* 2^{N} parole di canale. La prima colonna contiene dunque tutte le parole di codice, intese come le parole di canale ricevute senz'alcun errore, cioè con un vettore errore di tutti zeri $\mathbf{e}_{0} = \mathbf{0}$. Le colonne ulteriori si ottengono aggiungendo alla colonna delle parole di codice un vettore errore \mathbf{e}_{j} (uguale per tutti gli elementi della colonna) con peso di Hamming (cioè numero di errori) sempre più alto: N colonne con tutti gli N possibili vettori da 1 errore, N(N-1)/2 colonne con tutti i possibili doppi errori,..., fino a un totale di 2^{N-K} colonne caratterizzate da 2^{N-K} distinti vettori errore. Dunque l'elemento generico della matrice $\mathbf{d}_{i,j}$ è pari alla somma della *i*-esima parola di codice con il *j*-esimo vettore errore: $\mathbf{a}_{i,j} = \mathbf{a}_{i} \oplus \mathbf{e}_{j}$, e la matrice si presenta come segue:

	$\mathbf{a}_0 \oplus \mathbf{e}_0 = \mathbf{e}_0$	e ₁		$e_{2(N-K)-1}$	
D	$\mathbf{a}_1 \oplus \mathbf{e}_0 = \mathbf{a}_1$	$\mathbf{a}_1 \oplus \mathbf{e}_1$	·	$\cdots \mathbf{a}_1 \oplus \mathbf{e}_{2^{(N-K)}-1}$	(6.27)
$\mathbf{D} =$:		· · .		(0.37)
	$\mathbf{a}_{2^{K}-1} \oplus \mathbf{e}_{0} = \mathbf{a}_{2^{K}-1}$	$\mathbf{a}_{2^{K}-1} \oplus \mathbf{e}_{1}$		$\mathbf{a}_{2^{K}-1} \oplus \mathbf{e}_{2^{(N-K)}-1}$	

Questa matrice $2^K \times 2^{N-K}$ si chiama lo *standard array* del codice, e rappresenta una maniera scaltra di "ordinare" le uscite del canale per facilitare la decodifica. Infatti, ogni colonna della matrice, chiamata *coset*, è caratterizzata dalla medesima sindrome, visto che è caratterizzata dallo stesso vettore errore. Il vettore errore, cioè il primo elemento della colonna, è il *leader* del coset, ed è ciò che ci serve per la decodifica. Infatti non è necessario memorizzare tutto lo standard array - dobbiamo solo associare alle 2^{N-K} sindromi che possiamo incontrare nella decodifica, il rispettivo leader di coset identificato nell'array. Il decodificatore calcola la sindrome $\mathbf{s} = \mathbf{dH}^T$, accede questa tabella e ricava il leader associato alla particolare sindrome calcolata; il valore del leader rappresenta il vettore errore a minimo peso (e quindi massimamente probabile) che più verosimilmente ha causato la parola osservata. Questo leader viene sottratto (cioè sommato modulo-2) al blocco ricevuto per ricostruire la parola di codice più verosimile.

Esempio 6.32

Per il codice di Hamming (7,4), la lista dei leader di coset (vettori errore) ha $2^3 = 8$

196 CODIFICA DI VANALE CODIFICA YI CANALE CODIFICA DI CANALE

elements, cioè il vettore **0** e i 7 vettori di lunghezza 7 e peso 1:



Con questa tabella siamo in grado di procedere alla decodifica di una parola di canale **d** non appena ne calcoliamo la sindrome **s**.

Un esempio di codice a blocco popolare scoperto da Marcel Golay negli anni '60) è il codice (23,12) (il cui tasso r=12/23 è molto vicino a 1/2) con $d_{min} = 7$ che corregge fino

a 3 errori sul blocco di N=23 simboli con matrice di parity-check $\mathbf{H} = \begin{bmatrix} \mathbf{P}^T | \mathbf{I}_{11} \end{bmatrix}$ dove

|--|

Il codice di Golay condivide con il codice Hamming (7,4) una proprietà particolare: sono entrambi codici *perfetti*. Un codice perfetto è tale per cui le distanza tra tutte le parole è la medesima, ed è ovviamente uguale alla distanza minima (di Hamming) del codice. A parte casi banali, questo garantisce che il codice abbia il massimo tasso di codifica r per una determinata distanza minima d_{min} , poiché ha il numero *massimo* di parole di codice per un dato vincolo di distanza minima (vista la distribuzione massimamente regolare all'interno di Ω^N). Infatti il *limite di Hamming* dice che per ogni codice avente $d_{min} \leq 2q + 1$, il numero di parole di codice $|\Gamma|$ è limitato da

$$|\Gamma| \le \frac{2^N}{\sum_{i=0}^q \binom{N}{i}} \tag{6.39}$$

Se $d_{min} = 2q + 1$ e il limite di Hamming vale con l'uguaglianza, il codice è perfetto con tasso $r = \log_2 |\Gamma|/N$. Il codice perfetto banale è... il codice a ripetizione.

6.1.2.4 BER dopo decodifica Sappiamo che le caratteristiche di correzione degli errori di ogni codice sono determinate dalla d_{min} . Supponiamo che il decodificatore osservi il blocco hard-detected **d** proveniente da un BSC con probabilità di errore non protetta p_e . Nella relazione (4.53) siamo riusciti a calcolare esattamente la BER P_e dopo decodifica per il codice a ripetizione. Si dimostra che una relazione simile si applica ad *ogni* codice, identificando con N del repetition coding la d_{min} del generico codice, salvo che stavolta invece del valore esatto, il calcolo rende un limite superiore (upper bound) della BER:

$$P_e \le \sum_{n=(d_{min}+1)/2}^{N} {\binom{N}{n}} p_e^n (1-p_e)^{N-n}$$
(6.40)

L'uguaglianza si verifica soltanto per... i codici perfetti come quello a ripetizione.

Se invece la parola di codice viene trasmessa su di un canale AWGN con ricezione a filtro adattato come in Fig. 6.2, la situazione è più complicata. In questo caso si trova facilmente

un bound *inferiore* alla BER dopo decodifica P_e che però risulta piuttosto accurato ad alti rapporti segnale-rumore:

$$P_e \ge \mathbf{Q}\left(\sqrt{\frac{2E_s}{N_0}d_{min}}\right) = \mathbf{Q}\left(\sqrt{\frac{2E_b}{N_0}d_{min}\cdot r}\right) \tag{6.41}$$

Codici con grande distanza minima riescono a compensare la diminuzione del fattore r nel rapporto segnale-rumore E_s/N_0 intrinseca alla codifica (a parità di E_b/N_0) e apportare complessivamente un *coding gain* pari (asintoticamente) a $r \cdot d_{min}$ - preciseremo il concetto di coding gain nel paragrafo 6.2.4.

6.2 Codici Convoluzionali e Decodifica di Viterbi

I codici a blocco sono tuttora usatissimi in moltissime applicazioni dell'ICT. Ad esempio, i codici di Reed-Solomon (una classe particolare di codici per canali con erasure) sono impiegati nei driver per gli Hard-Disk, nella televisione digitale, nei lettori di CD/DVD eccetera. Dopo l'introduzione dei codici algebrici, la seconda famiglia importante di codici che fu introdotta negli anni '60-'70 èquella dei *codici convoluzional*i con tutte le loro varianti (punctured codes, Ungerböck's codes e così via).

6.2.1 Codifica convoluzionale non sistematica

Come il nome suggerisce, una codificatore concoluzionale è un circuito digitale che esegue la *convoluzione* tra il flusso di dati binari d'ingresso b[k] e una certa risposta impulsiva h[k] del codificatore per costruire i simboli codificati a[n]. Tra le diverse varianti di codificatori usati in pratica, la più semplice è quella dei codificatori *feedforward* (privi di reazione) non-sistematici con tasso r = 1/N, un esempio dei quali è rappresentato in Fig. 6.6 con N = 2. I due segnali binari di uscita $a_0[k] e a_1[k]$ sono prodotti da due filtri FIR lineari. L'unica differenza con lo schema generale di Fig. 2.21 è che nel nostro caso tutti i prodotti e le somme devono intendersi modulo-2, ma la relazione ingresso-uscita è la medesima:

$$a_i[k] = b[k] \cdot h_i[0] + b[k-1] \cdot h_i[1] + \dots + b[k-K+1] \cdot h_i[K-1] \quad i = 0,1 \quad (6.42)$$

Considerando che le operazioni sono binarie,

 $a_i[k] = b[k] \otimes h_i[0] \oplus b[k-1] \otimes h_i[1] \oplus \dots \oplus b[k-K+1] \otimes h_i[K-1] \quad i = 0,1$ (6.43)

Le due risposte impulsive $h_0[m]$ e $h_1[m]$ hanno una lunghezza K che si indica con il nome *constraint length* del codice (in pratica il numero di simboli di ingresso consecutivi che influiscono direttamente sul valore del simbolo di uscita)

La struttura finale del codificatore in Fig. 6.7 si ottiene poi considerando che i) il registro a scorrimento (linea di ritardo) può essere condiviso dai due "filtri binari"; ii) i valori di $h_i[m], m = 0, ..., K - 1$ sono binari: se $h_i[m] = 1$, allora il bit di ingresso ritardato di m colpi di clock è usato direttamente per calcolare l'uscita del codificatore secondo (6.43), e quindi nello schema di Fig. 6.7 inseriamo un collegamento corrispondente - se invece $h_i[m] = 0$ non c'è alcun collegamento; iii) i simboli codificati a[n] si ottengono come interleaving (multiplexing) di $a_0[k]$ e $a_1[k]$, cosicché si ottengono *due* simboli codificati per ogni singolo bit di informazione: $a[n] = a_{|n|_2}[n//2]$.





J



Figura 6.8 Codificatore convoluzionale dello standard DVB per la televisione digitale

La maniera più semplice per denotare la configurazione di un codificatore a tasso r = 1/N è specificare N stringhe binarie che rappresentano i valori di $h_i[m]$, i = 1, ..., N. Per il nostro codificatore a tasso 1/2 di Fig. 6.7, le due stringhe sono rispettivamente 111 e 101. Nella televisione digitale standard ETSI DVB-T viene utilizzato un codice convoluzionale con K = 7 e r = 1/2 il cui schema è mostrato in Fig. 6.8. Le stringhe di configurazione sono 1111001 1011011 che, espresse in base 8 per sinteticità, sono pari a $|171|_8 e |133|_8$.

6.2.2 Codifica convoluzionale sistematica

Oltre alla configurazione in *feedforward* che abbiamo appena descritto, vengono utilizzati anche codificatori con configurazione in *feedback*, specialmente per *concatenare* due codificatori ed ottenere il cosiddetto codice Turbo (sezione 6.3). Come in ogni filtro digitale, una "configurazione in feedback" equivale ad una relazione ingresso-uscita *ricorsiva*. La Fig. 6.9 riporta un esempio di codificatore convoluzionale *sistematico* e *ricorsivo* a tasso 1/2 e constraint length K = 4. Il codificatore è sistematico perché i simboli di posto pari in uscita sono una replica dei bit di informazione, $a[2k] = a_0[k] = b[k]$. Calcolando la funzione di trasferimento $H(z^{-1})$ del circuito digitale, si trova facilmente l'equazione (ricorsiva) che caratterizza i simboli codificati di posto dispari $a[2k + 1] = a_1[k]$:

$$a_1[k] = a_1[k-2] \oplus a_1[k-3] \oplus b[k] \oplus b[k-1] \oplus b[k-3]$$
(6.44)

Il codificatore è adesso IIR, cioè a risposta impulsiva infinita, e la convoluzione viene implementata indirettamente attraverso l'equazione ricorsiva (6.44) piuttosto che usando esplicitamente la risposta impulsiva come in (6.43). La configurazione del codificatore viene specificata con le stringhe binarie che caratterizzano la parte rispettivamente in avanti (feedforward) e in reazione (feedback) del circuito. Per il codificatore di Fig. 6.9, la sezione in avanti è data da 1101 mentre la sezione in feedback è (1)011. In quest'ultima, la prima cifra della stringa indicata tra parentesi è invariabilmente uguale a 1 (perchè...?). In generale, codici generati da codificatori feedforward sono equivalenti, nel senso che abbiamo giá discusso per i codici a blocco, a codici generati da codificatori in feedback.

Strutture di codificatore più complicate (feedforward o feedback) sono quelle con tassi di codifica del tipo $r = M/N \operatorname{con} M \neq 1$. I concetti principali sono gli stessi che abbiamo già introdotto per r = 1/N, e quindi tali codici non verrano presi in considerazione. Osserviamo in chiusura di sezione una differenza fondamentale tra i codici a blocco e i convoluzionali: nei primi, i bit di informazione vengono segmentati in blocchi di lunghezza K e tutto il processing di codifica sl trasmettitore e di de codifica al ricevitore è limitato a questo orizzonte temporale finito del *blocco*. Al contrario, il codificatore convoluzionale (6.43) o (6.44) non ha alcuna delimitazione intrinseca alla sequenza di bit d'ingresso e/o di



Figura 6.9 Codificatore convoluzionale sistematico ricorsivo con r=1/2, K=3 impiegato negli standard 3gpp (UMTS, LTE)

quella di uscita, che possono essere di lunghezza arbitraria N, anche grandissima - possiamo chiamarli codificatori di *sequenza*, e non è possibile identificare un dizionario finito delle parole di codice. Questa differenza si riflette nell'algoritmo di decodifica che introduciamo nella sezione successiva.

6.2.3 Decodifica di un codice convoluzionale

Dopo la discussione sulla codifica convoluzionale, dobbiamo risolvere le due questioni principali successive: i) come possiamo decodificare il codice in maniera ottimale rispetto al disturbo? ii) come possiamo calcolare la BER dopo decodifica, e quindi stabilire se un codice è più performante di un altro (e quindi trovare codici ottimi)? Riguardo il punto i), i codici convoluzionali furono inventati da Elias negli anni '50, e le proprietà di ottimalità furono subito chiare. Ma... nessuno sapeva come decodificarli efficientemente - il decodificatore di Viterbi oggi usato universalmente è stato introdotto nel 1967 !

Il decodificatore ottimo per un codice convoluzionale si ricava con lo stesso approccio che ha portato alla derivazione del criterio MV (6.9). Come già accennato, il problema principale sta nel fatto che la "parola di codice" di un convoluzionale ha in teoria lunghezza illimitata - in pratica molto lunga, altrettanto quanto tutta una trama dati, e una decodifica a minima distanza esaustiva è del tutto impraticabile. La decodifica MV del codice è di fatto la stima di tutta la *sequenza* dei bit d'ingresso, ed é chiamata *maximum-likelihood sequence estimation* (MLSE): il decodificatore deve trovare la sequenza dei simboli di canale $\hat{a}[n]$ (legata uno-a-uno alla sequenza di bit di informazione), tra tutte quelle compatibili con il codice, che presenta la minima distanza (di Hamming o Euclidea) rispetto alla sequenza di valori in uscita dal canale con disturbo (hard o soft). Questo è esattamente ciò che fa il decodificatore di Viterbi (DV).

Per descrivere il funzionamento del DV, cominciamo con l'osservare che il codificatore di Fig. 6.7 è di fatto una macchina sequenziale a stati finiti (macchina di Mealy). Notiamo infatti che l'uscita del codificatore è nota a partire da i) il valore attuale del bit di informazione b[k] e ii) i valori dei precedenti bit b[k-1], ..., b[k-K-1] nella memoria. Ogni volta che un nuovo bit di informazione viene presentato al codificatore, lo stato viene aggiornato secondo la semplice operazione di scorrimento verso destra nel registro di memoria, e l'uscita viene calcolata sulla base del (nuovo) bit e del valore del (nuovo) stato. Dunque lo stato del codificatore è la stringa di (K-1) bit memorizzati nello shift-register; l'ingresso della macchina è il k-esimo di sorgente, e l'uscita è la coppia di simboli codificati. Il numero di stati della macchina è infine $N_s = 2^{(K-1)}$, pari al numero di distinti contenuti

202 CODIFICA DI VANALE CODIFICA YI CANALE CODIFICA DI CANALE





della memoria.

A partire dallo schema del codificatore si può costruire il *diagramma di stato* mostrato in Fig. 6.10: gli stati (e i rispettivi contenuti della memoria) sono rappresentati dai poligoni mentre le transizioni da stato a stato (i vari rami) sono etichettate con il valore dell'ingresso b[k] che provoca la transizione e, tra parentesi quadre, i corrispondenti valori $a_0[k], a_1[k]$ dell'uscita. Dal diagramma di stato si ricava infine il *diagramma a traliccio* che contiene le medesime informazioni del diagramma di stato, ma rappresenta anche *l'evoluzione temporale* del codificatore convoluzionale, aspetto che nel diagramma di stato non si apprezza pienamente.

Nel diagramma a traliccio del codificatore di Fig. 6.7 mostrato in 6.11 (a) abbiamo rappresentato sull'asse orizzontale il tempo, indicizzato dal clock di informazione k, mentre sull'asse verticale abbiamo indicato gli $N_s = 4$ stati del codificatore rappresentati come cerchietti, ed etichettati dal possibile contenuto del registro di memoria al tempo k. I rami che uniscono diversi stati a diversi tempi consecutivi rappresentano le medesime transizioni del diagramma di stato di Fig. 6.10 (con le medesime etichette). Non tutte le transizioni da stato a stato sono possibili, ma solo quelle determinate dalla specifica struttura del codificatore (in particolare, quelle implicate dalla funzione di scorrimento a destra che determina lo stato successivo).

La Fig. 6.11 (b) evidenzia l'evoluzione temporale specifica del codificatore quando l'ingresso è la *particolare* stringa 1010, evoluzione che viene rappresentata da uno specifico *percorso* all'interno del traliccio; come si vede, abbiamo ipotizzata che all'istante k - 2 il codificatore fosse nello stato 00.

Supponiamo come sempre che i nostri bit codificati $a[n] \in \{0, 1\}$ siano ri-mappati sui simboli simmetrici di canale $a_{\pm}[n] \in \{-1, 1\}$ e inviati su canale AWGN con uscita soft. Il decodificatore, una volta ricevuta la sequenza dei 2N campioni $r[n] = a_{\pm}[n] + w[n]$, $n = 0, 1, \ldots, 2N-1$ relativi alla sequenza dei bit di informazione $b[k], k = 0, 1, \ldots, N-1$ con w[n] AWGN, deve trovare il particolare percorso nel trellis (e quindi la particolare


Figura 6.11 Descrizione a traliccio del codificatore di Fig. 6.6 (a); Percorso nel traliccio derivante dalla stringa di ingresso 1010 (b)

sequenza dei bit di informazione b[0], b[1], ..., b[N-1]) che con la massima probabilità (verosimiglianza) ha generato l'osservato. Riprendendo la notazione dell'Esempio 28, lo stimatore a massima verosimiglianza della sequenza $\mathbf{a}[N-1]$ dei 2N simboli codificati è

$$\hat{\mathbf{a}}[N-1] = \operatorname*{argmax}_{\tilde{\mathbf{a}}[N-1]\in \text{Trellis}} \ln f_{\mathbf{R}}(\mathbf{r}|\tilde{\mathbf{a}}[N-1])$$
$$= \operatorname*{argmin}_{\tilde{\mathbf{a}}[N-1]\in \text{Trellis}} \sum_{n=0}^{2N-1} (r[n] - \tilde{a}_{\pm}[n])^2$$
(6.45)

dove abbiamo indicato esplicitamente che $\mathbf{a}[N-1]$ è scelta tra le sequenze appartenenti al codice (cioè al traliccio) - si tratta come sempre di minimizzare una distanza Euclidea.

In linea di principio, il numero di sequenze di codice di lunghezza 2N sulle quali effettuare la minimizzazione in (6.45) è pari a 2^N (tante quanto sono le sequenze di bit di sorgente lunghe N che producono le sequenze codificate), e quindi la complessità della procedura appare *esponenziale* con N. In realtà, il problema si risolve brillantemente con un algoritmo ricorsivo (appunto, l'algoritmo di Viterbi) che esegue la stima a massima verosimiglianza della sequenza di informazione (MLSE) con una complessità *lineare* con N, in particolare proporzionale al numero degli stati N_s del codificatore.

Supponiamo che il decodificatore sia magicamente giunto nella decodifica fino all'istante k - 1, e che questo *non* sia l'istante finale N - 1. Nella ricostruzione del percorso più verosimile, il decodificatore ha portato con sé fino all'istante k - 1, come illustrato in Fig. 6.12 (a), N_s percorsi "candidati", uno per ogni stato, identificati dalle sequenze di informazione candidate $\tilde{\mathbf{b}}_i[k-1] = \left[\tilde{b}_i^{(k-1)}[0], \tilde{b}_i^{(k-1)}[1], ..., \tilde{b}_i^{(k-1)}[k-1]\right], i = 0, 1, ..., N_s - 1$, che sono chiamate le *sopravvisute* (vedremo tra un attimo il perché di questo nome). Ogni sopravvissuta porta con sé una misura, chiamata *metrica* $\delta_i[k-1], i = 0, ..., N_s - 1$ che si è accumulata passo dopo passo fino al tempo attuale k - 1, e che rappresenta la maggiore o minore *verosimiglianza* che la sopravvissuta rappresenti la parte iniziale della sequenza stimata finale. Tali metriche appaiono in Fig.6.12 (a) alla destra del traliccio per ciascuna sopravvissuta.

A partire da questa configurazione, il DV *estende* le sopravvissute al tempo k non appena vengono ricevuti i due campioni rumorosi $r[2k-2] = a_{\pm,0}[k-1] + w[2k-2]$ e $r[2k-1] = a_{\pm,1}[k-1] + w[2k-1]$ (il mapping tra a[n] ed $a_{\pm}[n]$ è $0 \rightarrow -1$ e $1 \rightarrow +1$). Per fare questo, vengono per prima cosa calcolate le *metriche di ramo* sulle transizioni del traliccio:

$$\lambda^{(i,j)}[k] = \left(r[2k-2] - \tilde{a}^{(i,j)}_{\pm,0}[k-1]\right)^2 + \left(r[2k-1] - \tilde{a}^{(i,j)}_{\pm,1}[k-1]\right)^2 \tag{6.46}$$

Le metriche di ramo sono relative alle *transizioni* perché in questa equazione $\tilde{a}_{\pm,0}^{(i,j)}[k-1]$ e $\tilde{a}_{\pm,1}^{(i,j)}[k-1]$ sono rispettivamente i valori dei simboli (ri-mappati) codificati che vengono prodotti dal codificatore quando si trova nello stato *i* al tempo k-1 ed evolve nello stato *j* al tempo k - sono dunque caratteristiche della transizione (i, j). Nella Fig. 6.12 mostriamo al di sopra del traliccio, in ciascun intervallo di bit, i valori disturbati ricevuti r[2k-2] e r[2k-1] che vengono appunto utilizzati per calcolare le metriche di ramo, le quali sono anche indicate in (b) su ognuno dei rami corrispondenti a una transizione dal tempo k-1 al tempo k.

Le metriche di ramo permettono di estendere le sopravvissute, e anche di aggiornare le rispettive metriche accumulate. Consideriamo infatti un generico stato di arrivo al tempo k; vediamo che vi sono sempre due rami che terminano in tale stato provenienti da due stati di



Figura 6.12 Sopravvissute al tempo k - 1 sul traliccio (a); procedura ACS (b)

206 CODIFICA DI VANALE CODIFICA YI CANALE CODIFICA DI CANALE

partenza $i_0 e i_1$ al tempo k - 1. Ad esempio, i rami che terminano nello stato 00 provengono dagli stati 00 e 01 al passo precedente. Allora possiamo costruire due sequenze "candidate" a diventare la nuova sopravvissuta dello stato j al tempo k estendendo semplicemente lungo quei due rami le vecchie sopravvissute rispettivamente degli stati $i_0 e i_1$ del tempo k - 1, come si vede in Fig. 6.12 (b), dove i due rami di estensione che "convergono" in ciascun stato di arrivo sono colorati in nero e in grigio. Adesso il DV i) somma (*adds*) le due metriche di ramo calcolate al tempo k alle metriche accumulate delle due sopravvissute che al tempo k - 1, per ottenere le metriche accumulate *candidate* delle due sopravvissute candidate:

$$\delta_{j}^{0}[k] = \delta_{i_{0}}[k-1] + \lambda^{(i_{0},j)}[k]$$

$$\delta_{i}^{1}[k] = \delta_{i_{1}}[k-1] + \lambda^{(i_{1},j)}[k]$$
(6.47)

Ancora in Fig. 6.12 (b) mostriamo, oltre alle metriche di ramo di ogni transizione, anche le due metriche accumulate candidate $\delta_j^0[k] \in \delta_j^1[k]$ risultanti da (6.47) sulla destra del traliccio. I due colori si riferiscono alle due candidate caratterizzate dalla transizione dello stesso colore.

Dopo l'operazione di *add*, il DV, per ogni stato, ii) confronta (*compares*) le due metriche accumulate candidate $\delta_j^0[k] \in \delta_j^1[k]$ per trovarne la minima, e infine iii) sceglie (*selects*) tra le due candidate, quella che ha accumulato la metrica minima (e che quindi ha la minima distanza dalla sequenza di campioni rumorosi osservati) $\delta_j[k] = \arg \min_i \delta_j^i[k]$. Questa sequenza (percorso) sopravvive dopo la procedura di ACS poiché qualunque futura eventuale estensione del percorso che esce dallo stato j al tempo k avrà comunque una metrica accumulata minore se proveniente dalla sopravvissuta piuttosto che dalla "defunta", che può essere scartata *definitivamente*. È proprio questa operazione di *scelta* che impedisce la crescita esponenziale con k della complessità del decodificatore (come succederebbe nel calcolo esaustivo delle distanze relative a tutte le sequenze prodotte dal codificatore), dato che il numero di sequenze delle quali valutare la distanza resta costante ad ogni passo, e pari (soltanto) ad N_s . Per una sequenza di informazione di lunghezza totale N, la complessita è dunque pari ad $N \cdot N_s$ piuttosto che 2^N !

Alla fine di questo procedimento ACS (add-compare-select), il DV ha nuove sopravvissute $\tilde{\mathbf{b}}_i[k] = \left[\tilde{b}_i^k[0], \tilde{b}_i^k[1], ..., \tilde{b}_i^k[k]\right], i = 0, 1, ..., N_s - 1$, come indicato in Fig. 6.13 (a)-(b). Da notare che l'*i*-esima sopravvissuta (cioè che appartiene allo stato *i*) all'istante *k non* è in generale l'estensione della *i*-esima sopravvissuta all'istante $k - 1^3$. Le sopravvissute vengono in generale "rimescolate" nel senso che quella che era la sopravvissuta di un certo stato diventa poi dopo estensione la prima parte della sopravvissuta di un *altro* stato, come si deve bene in Fig. 6.13 (b) che mostra la configurazione del decodificatore al tempo *k* dopo l'esecuzione di tutti gli ACS per ogni stato.

Quando k = N - 1, si deve effettuare un'ultima scelta tra le sopravvissute, e la "sopravvissuta delle sopravvissute" che ha accumulato la minima metrica $\delta_j[N-1]$ diventa la sequenza di bit di informazione più verosimile prodotta dal decodificatore. Se N è molto grande, memorizzare tutte le sopravvissute dall'inizio della sequenza fino alla fine può risultare infattibile. Se però, ad un certo istante k qualunque, seguiamo le sopravvissute *all'indietro nel tempo* sul traliccio, vediamo che i percorsi tendono a *confluire* entro un

³Questa è la ragione per cui dobbiamo utilizzare un secondo indice (temporale) nei vari bit delle sopravvissute in aggiunta all'identificatore dello stato



Figura 6.13 Confronto tra sopravvissute al tempo k - 1 (a) e sopravvissute al tempo k (b)

208 CODIFICA DI VANALE CODIFICA YI CANALE CODIFICA DI CANALE



Figura 6.14 Confluenza delle sopravvissute nel decodificatore di Viterbi

unico percorso dal quale, indietro nel tempo, tutti quanti i percorsi delle sopravvissute provengono. (Fig. 6.14). Empiricamente, si nota che la confluenza avviene in media tornando indietro nel traliccio di un intervallo temporale pari a $4 \div 5 \cdot K$ bit di informazione (K è la constraint length del codificatore). Di conseguenza, i bit appartenenti all'unica sezione di tutte le sopravvissute possono essere immediatamente resi disponibili dal decoder perchè non cambieranno mai più, dato che qualunque estensione futura di qualunque lunghezza di tutte le sopravvissute avrà comunque quei bit come sezione iniziale. Il decodificatore può quindi operare pressoché in tempo reale, introducendo però un *ritardo di decodifica* (latenza, decision depth) pari proprio a $4 \div 5$ volte la constraint length, in moltissime applicazioni più che tollerabile.

6.2.4 BER Performance

Ricavare la prestazione in termini di BER è, anche per i codici convoluzionali, abbastanza complicato, per cui la parola definitiva è spesso lasciata alle misure o alla simulazioni. Possiamo però anche qui ricavare un limite semplice alle prestazioni, molto simile al (6.41) ricavato per i codici a blocco. Il limite è anche utile per ottimizzare il codice e trovare i "migliori" codici convoluzinali.

Abbiamo visto che il DV tenta di ricostruire il percorso che il codificatore ha preso nel traliccio quando con una certa stringa di bit di informazione in ingresso. Riconsiderando il funzionamento del decodificatore, e in particolare della procedura ACS e dell'accumulo delle metriche, riconosciamo facilmente che quando la componente di disturbo è piccola (ovvero quando $r[2k] \simeq a_{\pm,0}[k] = r[2k+1] \simeq a_{\pm,1}[k]$), il percorso che corrisponde alle effettiva sequenza di bit di informazione accumula una metrica praticamente zero, mentre le metriche degli altri crescono senza limite perché la sequenza di simboli codificati di tentativo $\tilde{a}_{0}^{(i,j)}[k] = \tilde{a}_{1}^{(i,j)}[k]$ in (6.46), a parte casi sporadici, non corrisponde alla sequenza effettiva come invece accade alla " sopravvissuta delle sopravvissute ". Considerando con maggiore attenzione la metrica di ramo (6.46) ci accorgiamo che questa rappresenta di fatto il quadrato della distanza Euclidea tra il campione ricevuto e la coppia di simboli codificati di tentativo $\tilde{a}_{i}^{(i,j)}[k], i = 0, 1$. La metrica accumulata (6.47) ha quindi il significato della distanza euclidea quadrata tra la *sequenza* di campioni di segnale ricevuti r[0], ..., r[2N-1] e ogni sequenza di canale corrispondente alla sopravvissuta. La sopravvissuta finale è quindi quella che ha la minima distanza *complessiva* dai campioni di segnale ricevuti.

Come si produce un errore di decodifica ? Considerando la precedente discussione,



Figura 6.15 Evento errore nel decodificatore di Viterbi che induce 1 errore di bit

avremo un *evento errore* ogniqualvolta, a causa di una componente di rumore rilevante, esiste un percorso nel traliccio più vicino al segnale ricevuto del percorso corretto (e quindi una corrispondente sequenza errata di bit di informazione). Questo è proprio il caso del percorso tratteggiato in Fig. 6.15: il DV prende momentaneamente una strada sbagliata per ricongiungersi poi col percorso corretto dopo un tempo imprevedibile che dipende dall'intensità del rumore e dalle proprietà del traliccio del codice (vedi ancora Fig. 6.15). Durante questo evento errore, il decodificatore produce una sfilza (burst) di errori la cui lunghezza è casuale. Più frequenti sono gli eventi errore, maggiore sarà la BER del decodificatore.

Questi eventi errore sono a probabilità relativamente bassa quanto più i percorsi nel traliccio, ovvero le possibili sequenze codificate, sono "separate" o "lontane", esattamente come nel caso dei codici a blocco. *Lontane* significa "a grande distanza reciproca", cosicché lo scegliere un percorso sbagliato nel traliccio non sarà frequente neanche in condizioni di basso rapporto segnale-rumore. Da qui il parametro che determina la prestazioni di BER del codice: la *minima distanza tra sequenze di codice* compatibili col traliccio. Se questa distanza è piccola, sarà relativamente frequente che il DV imbocchi una strada sbagliata, dando luogo a una BER dopo decodifica non esaltante.

Ottimizzare un codice convoluzionale significa dunque trovare un traliccio per il quale la *minima distanza* e sufficientemente grande. Nel caso di trasmissione su BSC (o comunque con ingresso binario al decodificatore), si deve trovare un traliccio per il quale è grande la *minima distanza di Hamming*. Tale distanza deve essere calcolata in teoria per distanze di *qualunque lunghezza N*, con *N* (anche) arbitrariamente grande. La minima distanza di Hamming tra sequenze arbitrariamente lunghe su di un dato traliccio si indica con d_{free} (distanza libera) e può essere valutata per simulazione o con metodi analitici basati su funzioni di trasferimento.

Dunque la BER di un codificatore convoluzionale con decodifica di Viterbi a ingresso soft è sostanzialmente determinata dalla d_{free} . In particolare sussiste il seguente limite inferiore:

$$P_e \ge N_{avg} \cdot Q\left(\sqrt{r \cdot d_{free}} \frac{2E_b}{N_0}\right) \tag{6.48}$$

dove N_{avg} rappresenta il numero medio di bit di informazione errati in un evento errore che presenta d_{free} . Senza considerare questo parametro (così come fatto per i codici a blocco in (6.41)), vediamo che il guadagno di codifica (asintotico, cioè per alti valori del rapporto

209

K	$(h_0[k])_8$	$(h_1[k])_8$	d_{fre}
3	5	7	5
4	15	17	6
5	23	35	7
6	53	75	8
7	133	171	10
8	247	371	10
9	561	753	12

Tabella 6.1 Tabella dei migliori codici convoluzionali con r=1/2



Figura 6.16 Curve di BER dei codici ottimi con r = 1/2 su canale AWGN

segnale-rumore) è pari a $10 \log_{10}(r \cdot d_{free})$ dB. Un tasso di codifica piccolo deve essere compensato da un più che proporzionale incremento della d_{free} per dare un guadagno complessivo significativo.

In passato sono state condotte indagini esaustive su tutte le possibili configurazioni di codificatore convoluzionale per trovare quelle con la massima d_{free} a parità di complessità di decodifica, cioè N_s ovvero K. Un esempio dei risultati è mostrato in Tab. 6.1 per i codici a tasso 1/2. La tabella fornisce i set di coefficienti $h_i[k]$ (i cosiddetti generatori) ottimi, la corrispondente d_{free} , e il valore di N_{avg} . Per evitare lunghe stringhe binarie, la sequenza dei valori $h_i[k]$ è espressa in base 8 (vedi anche Fig. 6.8): ogni cifra ottale deve essere intesa come la stringa di 3 bit corrispondente.

Un esempio di curve di BER ricavate per simulazione è mostrato in Fig. 6.16 che riporta le performance dei codici ottimi con r=1/2 di Tab. 6.1, incluso quello di Fig. 6.8. Si nota il miglioramento delle prestazioni per complessità (cioè N_s) crescente, in accordo con il crescere della d_{free} .

Questa figura evidenzia chiaramente che cosa abbiamo guadagnato con l'utilizzo del codice: il guadagno di codifica (coding gain) è infatti il fattore (in dB) di cui diminuisce il

rapporto E_b/N_0 rispetto al caso non codificato, volendo mantenere costante la BER sui bit di informazione. In un ricevitore digitale, il livello di rumore N_0 che accompagna il segnale utile è costante ed è dovuto alle varie fonti di disturbo nel ricevitore - la diminuzione di E_b/N_0 indotta dalla codifica può essere sfruttata come una riduzione della potenza ricevuta necessaria per garantire una certa BER richiesta dall'applicazione. La riduzione della potenza ricevuta può essere sfruttata prevedendo una equivalente riduzione della potenza trasmessa e/o una diminuzione della dimensione (guadagno) dell'antenna ricevente in un collegamento wireless, apportando in ogni caso un guadagno su qualche parametro di progetto. Vediamo in Fig. 6.16 che il coding gain del codice con K = 7 è pari a 5.4 dB per una BER= 10^{-5} .

6.2.5 Decodifica Probabilistica ad uscita soft: l'algoritmo BCJR per codici convoluzionali

L'algoritmo di Viterbi non è l'unica maniera di decodificare un codice convoluzionale. Con il DV, otteniamo la stima MV della sequenza di bit di ingresso b[k], cosicché possiamo catalogare l'algoritmo come soft-input, binary-output (SIBO). Successivamente al DV, sono stati sviluppati altri algoritmi che, oltre a fornire una stima digitale (hard) dei bit, rendono anche una quantità *soft* (cioè $\in \mathbb{R}$) associata ad ogni bit di informazione: soft-input, soft output (SISO). Il principale di questi algoritmi è il cosiddetto BCJR (dai nomi degli inventori Bahl, Cocke, Jelinek, Raviv) per effettuare la decodifica di un blocco di lunghezza *finita* di simboli prodotti da un codificatore convoluzionale (diverso quindi da un decodificatore di *sequenza*).

Prima di capire che cosa sia questa informazione soft associata al bit di informazione, richiamiamo un po' di notazione. Come sempre, considereremo un blocco di K bit di informazione $\mathbf{b} = [b[0], b[1], ..., b[K-1]]$ che viene codificato da un codificatore convoluzionale a tasso r per ottenere un blocco di N (con $N = K \cdot r$ intero) simboli codificati $\mathbf{a} = [a[0], a[1], ..., a[N-1]]$. Questi N simboli codifcati a[n] vengono ri-mappati su simboli simmetrici di canale $a_{\pm}[n]$ che sono inviati su di un canale AWGN con segnalazione BPSK e rivelazione a filtro adattato priva di ISI come in Fig. 6.2. L'uscita a tempo di simbolo del filtro adattato è $r[n] = a[n]_{\pm} + w[n], n = 0, 1, ..., N - 1$, con w[n] rumore Gaussiano bianco a tempo discreto con varianza $\sigma^2 = (2E_s/N_0)^{-1}$ (vedi (2.111)). Il blocco soft ricevuto è quindi, con notazione evidente, $\mathbf{r} = \mathbf{a}_{\pm} + \mathbf{w}$. A causa della memoria del codificatore, le componenti di **r** non sono indipendenti. È intuitivo, considerando la struttura del codificatore convoluzionale, capire che a[n] dipende dalla storia del codificatore, e quindi da $a[n-1], a[n-2], \ldots$ Da questo discende però che a[n-1] dipende reciprocamente da a[n], cioè un simbolo codificato, oltre che dal proprio "passato" dipende anche dal proprio... "futuro". Quindi anche per un codificatore convoluzionale ha senso, nel fare la decodifica di un certo bit b[k] tenere conto di *tutto* il segnale osservato, cioè di tutto il blocco **r** sia che i campioni r[n] appartengano al passato sia che appartengano al futuro di b[k] - una cosa che un decodificatore a blocco normalmente fa, ma che il VD non fa proprio perchè lavora in tempo reale e non ha accesso al futuro di b[k] (decodificatore di sequenza causale).

Dunque la decodifica MAP (*maximum a-posteriori probability*) del *blocco* di codice convoluzionale segue il seguente criterio:

$$\hat{b}[k] = \underset{x \in \{0,1\}}{\arg \max} \left(\Pr\{b[k] = x | \mathbf{r}\} \right)$$
(6.49)

L'algoritmo BCJR fa esattamente questo, e fornisce anche come by-product un'indica-

zione della *affidabilità* di ciascuna decisione. L'uscita soft dell'algoritmo per ogni bit di informazione è infatti la seguente:

$$L(b[k]) \stackrel{\triangle}{=} \ln \frac{\Pr\left\{b[k] = 1 | \mathbf{r}\right\}}{\Pr\left\{b[k] = 0 | \mathbf{r}\right\}}$$
(6.50)

Cerchiamo di capire perchè consideriamo (6.50) come l'affidabilità del bit.

Supponiamo che b[k] = 1, e che prima di osservare **r** i due valori 0 e 1 siano a-priori equiprobabili - impossibile decidere. Dopo avere osservato **r**, le cose cambiano: le probabilità di 0 e di 1 cambiano a seconda dell'osservato, diventano le probabilità *a-posteriori*, e saranno in generale diverse tra di loro, anche parecchio sbilanciate. Prendere una decisione riguardo b[k] che possa essere considerata affidabile significa che dopo l'osservazione di **r**, la Pr { $b[k] = 1|\mathbf{r}$ } al numeratore è (ben) più grande della Pr { $b[k] = 0|\mathbf{r}$ } al denominatore; il rapporto è (molto) maggiore di 1 e il ln è positivo e (molto) grande in modulo. Se b[k] iè invece uguale a 0, troviamo similmente che L(b[k]) è negativo ma ancora (molto) grande in modulo. In entrambi i casi, la quantità chiamata *log-a-posteriori-probability-ratio* (LAPPR) L(b[k]) presenta una grande ampiezza (modulo), indicando grande *affidabilità*. Se invece il LAPPR è vicino a zero (piccolo in ampiezza) le due probabilità sono simili e la decisione riguardo b[k] non può essere considerata del tutto sicura. In ogni caso, il *segno* di L(b[k])rappresenta la hard-decision sul bit di informazione presa dal decodificatore: se L(b[k]) > 0, allora $\hat{b}[k] = 1$, viceversa se L(b[k]) < 0.

Esempio 6.33

Il concetto di LAPPR non si riferisce necessariamente a un bit codificato. Supponiamo infatti di effettuare una trasmissione *non* codificata sul solito canale AWGN, $r[n] = a_{\pm}[n] + w[n]$ con $a_{\pm}[n]$ direttamente legato al bit di informazione, $a_{\pm}[n] = 2b[n] - 1$. Il calcolo del LAPPR riguarda solo r[n] perché questo risulta indipendente dagli altri r[m], $m \neq n$, ed è immediato:

$$L(b[n]) = \ln \frac{\Pr\{b[n] = 1 | \mathbf{r}\}}{\Pr\{b[n] = 0 | \mathbf{r}\}} = \ln \frac{\Pr\{b[n] = 1 | r[n]\}}{\Pr\{b[n] = 0 | r[n]\}}$$
$$\ln \left(\frac{f_R(r[n]|b[n] = 1) \Pr\{b[n] = 1\}}{f_R(r[n])} \frac{f_R(r[n])}{f_R(r[n]|b[n] = 0) \Pr\{b[n] = 0\}}\right)$$
$$= \ln \frac{f_R(r[n]|b[n] = 1)}{f_R(r[n]|b[n] = 0)}$$
(6.51)

dove abbiamo usato il teorema di Bayes, abbiamo considerato bit di sorgente equiprobabili, e dove $f_R(\cdot)$ indica la pdf del campione soft ricevuto r[n]. Le due pdf condizionate che compaiono nella (6.51) sono entrambe *Gaussiane* con la stessa varianza $\sigma^2 = (2E_s/N_0)^{-1}$, e con un valor medio pari a $a_{\pm}[n] = 2b[n] - 1$. Usando l'espressione della pdf Gaussiana e calcolando il logaritmo abbiamo

$$L(b[n]) = -\frac{(r[n]-1)^2}{2\sigma^2} + \frac{(r[n]+1)^2}{2\sigma^2} = \frac{2r[n]}{\sigma^2} = \frac{4E_s}{N_0}r[n]$$
(6.52)

Si vede che il criterio di decisione MAP relativamente a b[n] è semplicemente una decisione a soglia su L(b[n]), cioè una hard-decision su r[n] come già (ben) noto.

La descrizione dettagliata dell'algoritmo BCJR per calcolare la LAPPRs su di un blocco di simboli codifcati è parecchio complicata, e viene riportata nell'Appendice C. Ci limitiamo

a osservare che, per ogni bit b[k], il valore di L(b[k]) si ricava da *due* sub-algoritmi ricorsivi simili a un DV che operano sul traliccio. Il primo algoritmo scorre sul blocco ricevuto **r** dall'inizio fino all'istante corrispondente al bit k - 1 per calcolare la dipendenza dei valori di canale dal *passato* di b[k]; il secondo sub-algoritmo ricorsivo scorre il traliccio a *ritroso nel tempo* dalla fine del blocco fino all'istante corrispondente al bit k + 1 per calcolare la dipendenza dei valori di canale dal *futuro* di b[k]. Questi due valori, insieme con un valore analogo alla metrica di ramo del DV relativa ai campioni di canale dell'istante k, vengono combinati per fornire il numeratore e il denominatore di L(b[k]) e quindi la stima di b[k], $\hat{b}[k]$. Secondo questa descrizione, la complessità del BCJR si può valutare come all'incirca il doppio (per la presenza dei *due* algoritmi avanti/indietro) di quella di un DV per lo stesso codice.

Come si confrontano le prestazioni degli algoritmi DV e BCJR ? Molto simili se i bit di informazione sono equiprobabili e indipendenti perchè in tal caso la stima MAP bit a bit del BCJR o MLSE di tutta la sequenza del DV sono sostanzialmente la stessa cosa. Ci possono essere sfumature dovute all'implementazione, troncamenti, ritardo di decodifica ecc., ma se non ci sono particolari requisiti, non si vede alcuna utilità nell'avere un algoritmo che presenta BER equivalente con un'uscita soft inutile e anche doppiamente complesso. Ma, la disponibilità dell'uscita soft diventa essenziale per la decodifica di un *codice turbo*, oggetto della prossima sezione.

6.3 Decodifica Iterativa (Turbo) di Codici Convoluzionali Concatenati

L'anno 1993, con l'invenzione dei codici turbo da parte di Berrou e Glavieux, segna l'inizio dell'era moderna per la codifica di canale. Fino allora, probabilmente il miglior codice di larga applicazione per tassi medio-bassi era ottenuto dalla *concatenazione* di un codice *esterno* algebrico classico con tasso r_o , e di un codice *interno* convoluzionale con tasso r_i , come indicato in Fig. 6.17 (a): lo storico codice della televisione digitale DVB in cui il codice esterno è Reed-Solomon modificato con $r_o = 188/204$ e l'interno è convoluzionale con K = 7 e tasso variabile, ottenuto a partire da quello già visto in Fig. 6.8 con $r_i = 1/2$. Il codice esterno è a *bassa ridondanza* (alto tasso r) e ha la funzione di ridure le (lunghe) sfilze di errori di bit provocate da un evento errore del DV per il codice interno. In questa maniera, con un tasso di codifica globale $r = r_i \cdot r_o$ praticamente uguale a quello del codice interno r_i ($r = 94/204 \simeq 0.46$ se $r_i = 1/2$), e utilizzando tecnologie già consolidate, si ottiene un codice complessivo il cui guadagno di codifica a 10^{-5} per il tasso minimo r = 94/204 si attesta a un buon 7 dB (vedi Fig. 6.17 (b)). Però, il punto rappresentativo della MOD/COD sul piano di Shannon in Fig. 4.23 non è vicinissimo al limite.

6.3.1 Il Codificatore Turbo PCCC

Dunque l'idea di *concatenare* due codici sembrava la vincente. Il decodificatore era però dichiaratamente sub-ottimo: era la cascata di due decoder sì ottimi, ma per i due codici considerati separatamente, senza nessuna interazione tra i due. L'idea fondamentalmente innovativa introdotta da Berrou e Glavieux fu quella della *decodifica iterativa* utilizzando due decodificatori a *uscita soft*.

Prima di spiegare come costruire un turbo codice e il suo decodificatore, riprendiamo in considerazione l'algoritmo BCJR, partendo dalla formulazione del LAPPR:

$$L(b[k]) \stackrel{\triangle}{=} \ln \frac{\Pr\{b[k] = 1 | \mathbf{r}\}}{\Pr\{b[k] = 0 | \mathbf{r}\}} = \ln \frac{f_{\mathbf{R}}(\mathbf{r}|b[k] = 1)}{f_{\mathbf{R}}(\mathbf{r}|b[k] = 0)} + \ln \frac{\Pr\{b[k] = 1\}}{\Pr\{b[k] = 0\}}$$
(6.53)



Figura 6.17 Schema (a) e curve di BER (b) del codice concatenato della televisione digitale DVB

dove abbiamo usato il teorema di Bayes come in (6.51). Il primo termine nella (6.53) è il cosiddetto *Log-rapporto di verosimiglianza* (Log-Likelihood Ratio, LLR), mentre il secondo dipende dalle probabilità *a-priori* dei bit di sorgente ed è usualmente pari a 0. Infatti, con una buona compressione di sorgente, le probabilità dei due livelli binari sono uguali, e il log-rapporto è 0 - questa componente a-priori del LAPPR sembra inutile e di fatto il criterio MAP decade nell'MV.

Se però avessimo in parallelo al decodificatore che fornisce (6.53) un secondo decoder che produce una qualche indicazione addizionale riguardo i bit di informazione, come se questi bit provenissero (anche) da un secondo canale di comunicazione indipendente, avrei a disposizione una certa informazione *estrinseca* (cioè esterna al primo decoder) riguardo ai possibili valori dei bit 0 o 1, e non sarebbe più corretto considerarli equiprobabili. Avrei cioè una informazione a-priori da sfruttare nella prima decodifica, e il secondo termine in (6.53) non sarebbe più nullo - magari con quello potremmo migliorare l'affidabilità della decisione. Questa informazione estrinseca può essere fornita al decoder nella forma di un valore di log-rapporto di probabilità $L^e(b[k])$ dove l'apice ^e significa che questo valore è stato ricavato estrinseca nulteriore ingresso del decoder BCJR (e di questo parleremo più tardi), ma soprattuto dobbiamo capire come riuscire a procurarci questo "secondo parere" sul bit di informazione!

La risposta alla seconda domanda viene dalla Fig. 6.18 che descrive la struttura del classico *codificatore turbo* a tasso 1/3 costituito dalla *concatenazione parallela di codificatori convoluzionali* (Parallel Concatenation of Convolutional Codes, PCCC). Vengono utilizzati in particolare due codificatori *costituenti* identici di tipo ricorsivo sistematico (Recursive Systematic Convolutional, RSC) con tasso 1/2, come ad esempio quello di Fig. 6.9. Il codificatore RSC1 riceve un blocco **b** di *K* bit di informazione b[0], b[1], ..., b[K-1] nell'ordine naturale, mentre il codificatore RCS2 riceve il blocco **b**_{π} ottenuto dal **b** di partenza attraverso un *permutatore* (interleaver) II. Il blocco **b**_{π} contiene i medesimi bit presenti in **b**, ma permutati in modo *pseudo-casuale*. Indicando con II la permutazione sugli interi, allora

$$\pi = [b[\Pi(0)], b[\Pi(1)], ..., b[\Pi(K-1)]]$$

(6.54)

Il permutatore è di fatto il componente che determina la lunghezza del blocco di informazione, che è usualmente abbastanza grande - da qualche centinaio a qualche migliaio: ecco realizzato in qualche modo un codice con blocchi *grandi* e scelti con un criterio che ha un elemento di *casualità*, proprio come i codici casuali del teorema della codifica di canale di Shannon !

b

Lo scopo del permutatore è di far sì che i due blocchi di bit in ingresso e uscita appaiano sostanzialmente *incorrelati*, cosicché i due blocchi di bit di *parità* (ridondanza) $\mathbf{p} \in \mathbf{p}_{\pi}$ prodotti dai due codificatori siano pure incorrelati (perchè ottenuti da blocchi di bit di informazione apparentemente differenti). La permutazione Π non è in realtà puramente casuale, me segue una legge deterministica di tipo *pseudo-random* nota al decodificatore.

Come si nota, l'uscita sistematica **p** dell'RSC1 viene conservata e diventa l'uscita sistematica del codificatore concatenato, mentre l'uscita sistematica \mathbf{b}_{π} dell'RSC2 viene scartata perchè facilmente ricostruibile da **p** attraverso permutazione inversa - lo stesso non può dirsi di \mathbf{p}_{π} rispetto a **p** perché i codificatori RSC hanno memoria e non sono facilmente invertibili. I tre blocchi **b**, **p**, e \mathbf{p}_{π} vengono interlacciati (multiplati) bit a bit creando l'uscita del codificatore sistematico turbo

$$\mathbf{a} = [b[0], p[0], p_{\pi}[0], b[1], p[1], p_{\pi}[1], ..., b[K-1], p[K-1], p_{\pi}[K-1]]$$
(6.55)

216 CODIFICA DI VANALE CODIFICA YI CANALE CODIFICA DI CANALE



Figura 6.18 PCCC Turbo encoder



Il blocco di codice ha lunghezza N = 3K e il tasso di codifica è quindi r = 1/3. La Fig. 6.19 rappresenta lo schema del codificatore standard 3gpp (www.3gpp.org) usato delle reti cellulari 3G UMTS e 4G LTE, che segue esattamente la costruzione appena descritta (con le diverse notazioni dello standard).

Come sempre, i simboli codificati vengono ri-mappati su $a_{\pm}[n] = 2a[n] - 1$ e inviati su canale AWGN, per cui il blocco dei 3K = N campioni soft ricevuti è, con notazione autoesplicativa, $\mathbf{r} = \mathbf{a}_{\pm} + \mathbf{w}$

6.3.2 Decodifica iterativa del Turbo Code

I componenti principali del turbo-decoder rappresentato in Fig. 6.20 sono i due decodificatori Soft-Input, Soft-Output BCJR relativi ai due RSC costituenti. Il blocco soft ricevuto **r** viene demultiplato nelle tre componenti soft **r**_b, **r**_p, **r**^{π}_p che rappresentano le versioni rumorose rispettivamente di **b**, **p**, e **p**_{π} (dopo ri-mappaggio). Poi, **r**_b e **r**_p vengono utilizzati dal primo decodificatore SISO DEC1 per ricavare i LAPPRs $L^{(1)}(b[k])$ seguendo l'algoritmo BCJR, e anche per ricavare il valore $L^{(1)}_{out}(b[k])$ che viene indirizzato al DEC2. Quest'ultimo riceve il vettore permutato localmente **r**_b (che chiamiamo **r**^{π}_b) e i corrispondenti campioni relativi alla parità **r**^{π}_p. La questione chiave è che il DEC2 riceve la quantità $L^{(1)}_{out}(b[k])$ che DEC1 ha calcolato usando simboli di parità (rumorosi) **r**_p che sono *estrinseci* (cioè esterni)



DECODIFICA ITERATIVA (TURBO) DI CODICI CONVOLUZIONALI CONCATENATI 217



al processo di codifica di RSC2: ecco il "secondo parere" che cercavamo! Ciò che DEC1 ha prodotto in uscita, ed è quindi per DEC1 è a-posteriori, DEC2 riceve in *ingresso* come informazione per se' a-priori $L_{in}^{(2)}(b[k])$, e che usa insieme con gli altri ingressi di canale per ricavare la sua propria versione a-posteriori dell'informazione d'uscita $L_{out}^{(2)}(b[k])$ che può essere rinviata indietro come *estrinseca* verso il DEC1 (calcolata infatti attraverso $\mathbf{r}_{\mathbf{p}}^{\pi}$, indisponibile a DEC1). Allora ha senso pensare di "rifinire" ulteriormente $L^{(1)}(b[k])$ usando $L_{out}^{(2)}(b[k])$ e interpretandolo come nuova informazione a priori d'ingresso $L_{in}^{(1)}(b[k])$, con la quale fornire una ulteriormente rifinita versione di $L_{out}^{(1)}(b[k])$ verso DEC2 e così via...

Abbiamo descritto la *decodifica iterativa a due stadi* con propagazione in feedback dell'informazione chiamata *decodifica turbo*. La circolazione dell'informazione estrinseca tra i due decoder ricorda la rotazione velocissima di una turbina nei motori turbo: in entrambi i casi si ottiene un incremento notevole di prestazioni proprio a causa dell'effetto *turbo*. Iterazione dopo iterazione, si vede che alcune LAPPR cambiano segno (cioè vengono corretti errore di canale), finché si arriva ad uno stato "stabile" in cui non vi sono più sostanziali cambiamenti, e le iterazioni possono terminare. La decisione finale su b[k] viene poi presa sulla base del segno di L_1 . L'Appendice C dettaglia la differenza tra la parte estrinseca delle LAPPR, la parte a comune che non viene propagata, e il calcolo della L(b[k]) totale sulla quale svolgere la decisione finale.

L'algoritmo di decodifica turbo è stato introdotto empiricamente e non ha alcuna pretesa di ottimalità. Potremmo quindi chiederci come implementare un decoder ottimale e quanto diverse sarebbero le sue prestazioni. Per quanto riguarda la prima domanda, il turbo encoder di Fig. 6.18 non è altro che una macchina a stati finiti, la cui evoluzione può essere rappresentata da un traliccio analogo a quello di Fig. 6.11 (a), e la cui stringa di input può essere decodificata in modo ottimo osservandone l'uscita rumorosa attraverso un opportuno DV. Quanti stati ha la macchina, e quindi, qual è la complessità del DV? Osserviamo che la *memoria* dell'encoder è sostanzialmente concentrata nel permutatore, che può essere considerato come una memoria ad accesso casuale che viene scritta in un certo ordine (naturale), e viene poi letta secondo una diversa sequenza pseudo-casuale. Quindi, la dimensione della memoria è K e il numero di stati è $N_s = 2^K$. Sappiamo già che K può essere grande fino a 10^4 , e quindi che N_s diventa un numero tanto grande da



Figura 6.21 Codificatore PCCC a tasso r=1/2 con perforazione (a) e schema di perforazione (b)

essere privo di significato - il decodificatore ottimale è una chimera. Riguardo la seconda questione, vedremo tra poco che in realtà c'è ben poco spazio per migliorare le prestazioni del decodificatore turbo, dal momento che, benchè sub-ottimali, sono già notevolmente vicine al limite di Shannon.

Possiamo anche chiederci come progettare un codificatore turbo con tasso diverso dal "nativo" 1/3 della Fig. ref PCCCencoder (a), per arrivare ad esempio a r = 1/2. La soluzione più semplice e più pratica è mostrata in Fig. 6.21 (a). L'unica differenza con lo schema originale è l'inserzione di un *perforatore* (puncturer) che modifica $\mathbf{p} \in \mathbf{p}_{m}$. Il blocco di parità di uscita \mathbf{p}_i (dove ovviamente il pedice sta per *perforato*) è ottenuto come rappresentato in Fig. 6.21 (b), cioè intercalando gli elementi pari di p con gli elementi dispari di \mathbf{p}_{π} e *perforando* (cioè eliminando) gli altri. Il blocco risultante \mathbf{p}_{n} ha dimensione K, in modo che il tasso complessivo sia il valore desiderato 1/2. Nel decoder, gli elementi mancanti (perforati) di $\mathbf{p} \in \mathbf{p}_{\pi}$ vengono ripristinati come valori 0, in modo che il decodificare abbia comunque a disposizione due blocchi di parità di dimensione K. Nonostante la perforazione, che provoca di fatto la cancellazione di alcuni bit di ridondanza, il decodificatore continua ad avere prestazioni sufficientemente buone, inferiori ovviamente al codice-madre non perforato, dato l'aumento del tasso.

6.3.3 BER performance

L'incremento di prestazioni ottenuto con la decodifica turbo è notevole. La Fig. 6.22 (a) mostra le curve di BER ottenute mediante simulazione di un codice turbo con tasso 1/3 e con un permutatore/interleaver pseudo-casuale con dimensioni K = 1500 bit e senza nessun tipo di ottimizzazione. Si nota che un grande miglioramento della BER viene già ottenuto con le prime iterazioni, in seguito l'ulteriore miglioramento tende a diminuire. È pure impressionante il confronto con la curva della BER non codificata (Fig. 6.22 (b)), che rivela un guadagno di codifica @ 10^{-5} di quasi 9 dB. Per riferimento, abbiamo anche riportato la curva di BER limite derivante dal teorema inverso della capacità di Shannon (4.56) su BIAWGN (4.92). Vediamo che questo codice "off-the-shelf" con un permutatore relativamente piccolo si trova a meno di 1 dB dal limite di Shannon.

Quando parliamo di iterazioni di decodifica, dovrebbe essere chiaro dalla nostra descri-



(b)

Figura 6.22 Curve di BER di un codice turbo al crescere del numero di iterazioni di decodifica (a) e confrontate con la BER non protetta (b)

220 CODIFICA DI VANALE CODIFICA YI CANALE CODIFICA DI CANALE

zione dell'algoritmo che tutte le iterazioni vengono eseguite *in tempo virtuale* "riutilizzando" nel decodificatore gli stessi dati ricevuti $\mathbf{r_b}$, $\mathbf{r_p}$, $\mathbf{r_p}^{\pi}$ nelle varie iterazioni per un dato blocco di codice di K bit, fino a raggiungere la convergenza. Prendendo come numero tipico di iterazioni il valore $N_{it} = 10$, possiamo valutare la complessità della decodifica turbo rispetto alla decodifica convenzionale di Viterbi del codice RSC costituente considerando che: i) la complessità di ciascun decodificatore SISO BCJR è circa il doppio di quelle di un DV; ii) ogni iterazione di decodifica richiede l'esecuzione di *due* BCJR; iii) facciamo 10 iterazioni. Quindi, la complessità di un decodificatore turbo è approssimativamente pari a $2 \cdot 2 \cdot 10 = 40$ volte quella del DV per i codici costituenti.

6.4 Codici Low-Density Parity-Check e Decodifica Iterativa a Scambio di Messaggi

6.4.1 Che cos'è un codice LDPC

Il nuovo concetto di decodifica iterativa probabilistica basata su algoritmi SISO generò un'ondata di nuove idee e di riconsiderazione di vecchi problemi e vecchi codici. Negli anni '60 Gallager aveva introdotto una classe di codici di tipo parity-check (analoghi in principio ai codici di Hamming della sez. 6.1.1) del tutto peculiari per l'epoca perché non seguivano il paradigma dei codici algebrici, tuttavia con buone proprietà metriche per blocchi lunghi. Gallager, ispirandosi alla nozione di codici casuali di Shannon, prese in considerazione una certa classe di matrici di controllo di parità generate casualmente, dimostrando che la d_{min} del codice implicato dalla matrice cresce (indefinitamente) al crescere delle dimensioni della matrice stessa, e che quindi il codice relativo è asintoticamente "buono" con probabilità unitaria. Gallager abbozzò anche un algoritmo per effettuare ciò che oggi chiamiamo "decodifica probabilistica" (probabilistic decoding), algoritmo che rimase sulla carta perché ritenuto di nessuna utilità. Negli anni '90, MacKay riscoprì i codici di Gallager, ne estese la costruzione e riformulò l'algoritmo iterativo di decodifica, trovando prestazioni molto vicine al limite di Shannon.

Un codice LDPC alla Gallager non è nient'altro che un codice a controllo di parità la cui matrice binaria di parity-check \mathbf{H} è i) grande, ii) *sparsa*, iii) generata casualmente. "Sparsa" significa che \mathbf{H} è principalmente costituita da elementi 0, e che gli 1 sono molto rari - low

density! H si presenta come l'esempio seguente per un codice (50,25) con r = 1/2:

Si intuisce il perché una matrice LDPC dia luogo a un "buon" codice: se le righe (colonne) della matrice sono costituite da quasi tutti 0, il numero minimo di righe (colonne) che è necessario sommare per dare un vettore $\mathbf{0}$, e cioè come sappiamo la d_{min} del codice, è alta: gli elementi 1 sono sporadici e quindi per "cancellarli" dal vettore somma, bisogna trovare un'altra riga (colonna) con un 1 nella stessa posizione - un evento a probabilità molto bassa che richiede di considerare *molte* righe (colonne) - si realizza quindi una grande distanza minima.

Vedremo che i migliori LDPC sono caratterizzati da matrici molto grandi (ad esempio il codice (64800, 32400) della televisione satellitare DVB-S2, cosicché una caratteristica peculiare di questi codec è che, oltre all'usuale necessità di trovare algoritmi di decodifica efficienti, qui è necessario anche trovare algoritmi di *codifica* altrettanto efficienti. In generale, l'operazione di codifica **a=bG** di un codice a blocco ha complessità quadratica con K, che per grandi dimensioni diventa improponibile. Il fatto che **H** sia low-density aiuta molto nell'operazione di decodifica, come vedremo successivamente, la cui complessità è praticamente lineare in N. Per la codifica, il codice deve per motivi pratici risultare come già sappiamo sistematico; se però **H** è totalmente destrutturata perché generata casualmente, si vede facilmente che la matrice generatrice **G** in forma sistematica (dopo eventuale eliminazione Gaussiana) *non* è low-density, rendendo la complessità della codifica di nuovo



Figura 6.23 Grafo di Tanner della matrice di parity-check (6.56)

alta.

Nella costruzione randomica della matrice, l'unico vincolo che si impone è la densità (media o esatta) degli elementi pari a 1. Un codice LDPC è *regolare* se il numero w_r di 1 in ogni riga di H (così come il numero w_c nelle colonne) è sempre lo stesso (i codici studiati da Gallager), altrimenti é *irregolare*. In generale, i codici irregolari sono lievemente migliori dei regolari, che però possono essere progettati più facilmente e danno luogo a codificatori in generale più efficienti.

Negli anni '80, sempre allo scopo di progettare codici efficienti, Tanner introdusse una rappresentazione di un codice parity-check che è particolarmente utile per visualizzare le proprietà dei (buoni) codici LDPC e per progettare decodificatori efficienti. Sappiamo che ogni riga della matrice di controllo della parità costituisce un'equazione di controllo che somma a 0 per ogni parola di codice a. Rappresentiamo questi vincoli con il cosiddetto grafo di Tanner associato alla matrice H, la cui costruzione andiamo ad esemplificare con una matrice un po' più semplice della (6.4.1), in particolare quella del consueto codice (N = 7, K = 4) di Hamming:

 $\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$ (6.56)

A partire da questa, costruiamo un grafo bipartito, nel quale cioè si trovano nodi di classi diverse, connessi da rami (archi). Una classe è quella degli N nodi di bit (chiamati anche nodi delle variabili a[n], n = 0, ..., N - 1) che rappresentiamo graficamente in Fig. 6.23 come circoletti disposti su di una riga e che ovviamente rappresentano i valori dei simboli della parola di codice. L'altra classe è quella degli N - K nodi di *controllo* c_i , i = 1, ..., N - K che indicano le N - K equazioni di parity check e che rappresentiamo graficamente come quadratini in un riga al di sotto della riga dei nodi di bit. La particolare matrice **H** viene rappresentata connettendo con un ramo il nodo di bit n al nodo di controllo i se l'elemento $h_{i,n}$ di **H** è uguale a 1: si ottiene il grafo di Fig. 6.23.

Vedremo nella prossima sezione come questo grafo (in cui i rami sono evidentemente low-density...) possa essere utile per capire l'algoritmo di decodifica. Diciamo solo che il grafo di un buon LDPC è caratterizzato da avere *cicli lunghi*. Un ciclo nel grafico è un percorso che inizia in un nodo e termina nello stesso nodo dopo un certo numero di passaggi su rami e nodi (distinti) intermedi. La minima lunghezza (minimo numero di rami) dei cicli di un certo grafo si chiama *girth* del grafo; maggiore è questo parametro (rispetto al minimo possibile che è 4 vista la bipartizione del grafo), migliore è in generale il codice.

Non insisteremo più di tanto sulla progettazione dei codici che, come per i codici turbo, è molto complicata, ma passiamo direttamente alla spiegazione della decodifica iterativa. Osserviamo che anche in questo caso, come per i turbo, i codici LDPC vanno incontro al concetto di *codice casuale con blocchi grandi* introdotto da Shannon nel teorema di capacità. Naturalmente, decodificare un codice veramente casuale con blocchi lunghi è praticamente impossibile anche con il meglio della tecnologia odierna: l'unico modo è una valutazione esaustiva delle distanze per implementare il decodificatore a distanza minima. Con il codice Low-Density però si può utilizzare un algoritmo iterativo sub-ottimo le cui prestazioni vanno vicinissime al limite di Shannon - la stessa situazione incontrata con i codici turbo...

6.4.2 Decodifica iterativa dei codici LDPC - I: Hard-Input

Il grafo di Tanner si rivela fondamentale per sviluppare e capire l'algoritmo iterativo di decodifica degli LDPC. Supponiamo di avere a disposizione come ingresso del decodificatore le uscite binarie $\mathbf{d} = [d[0], d[1], ..., d[N - 1]]$ di un BSC che riceve in ingresso la parola di codice **a**. Questo blocco rappresenta il punto di partenza (iterazione 0) della decodifica (cioè delle stima della parola trasmessa **a**), che verrà poi rifinita con un procedimento iterativo: $\mathbf{a}^{(0)} \stackrel{\triangle}{=} \mathbf{d}$.

Nel grafo di Tanner, i vari nodi devono essere pensati come processori locali che ricevono informazione d'ingresso dai rami cui sono connessi, effettuano calcoli, e restituiscono uscite in verso opposto sui medesimi rami. La stima iniziale di cui sopra rappresenta il primo set di messaggi che i nodi di bit inviano verso il basso ai nodi di controllo (prima semi-iterazione). I nodi di controllo ricevono i messaggi ed effettuano il calcolo della rispettiva equazione di parity-check, proprio a partire dei messaggi (cioè dei valori dei bit) appena ricevuti (in pratica sommano i valori di tutti i messaggi ricevuti). Se tutti i risultati del calcolo dei nodi di check sono pari a 0, tutte le equazioni di parità sono verificate, e quindi la parola attualmente "scritta" sui nodi di bit è la parola di codice decodificata \hat{a} - fine delle operazioni. Se invece qualche risultato è pari a 1 (come in Fig. 6.24 (a)), allora si deve ripartire con una ulteriore semi-iterazione di decodifica: il risultato della verifica (check) (1 o 0) è il nuovo messaggio di ritorno che il nodo di controllo spedisce verso l'alto ai vari nodi di bit (seconda semi-iterazione). La Fig. 6.24 rappresenta queste due prime semi-iterazioni relativamente al codice (6.56), supponendo che la parola di codice trasmessa sia $\mathbf{a} = [1011001]$, e che il canale abbia introdotto un errore sul secondo simbolo trasmesso, cosicché $\mathbf{d} = [1\mathbf{1}1100\mathbf{1}]$. La situazione dopo la prima semi-iterazione è quella di Fig. 6.24 (a), dove vengono indicati i valori calcolati dai nodi di controllo.

Adesso è necessario che a loro volta i nodi di bit effettuino il loro proprio calcolo dopo aver ricevuto i messaggi inviati verso l'alto dai nodi di controllo. In particolare, i nodi di bit devono decidere se restare al valore attuale, presumibilmente giusto, o cambiare il proprio valore perché presumibilmente errato, "migliorando" cosìla decodifica. Il criterio dei nodi di bit è il seguente: se un certo nodo riceve "molti" messaggi pari a 0 e "pochi" pari a 1, allora il valore corrente è ritenuto *corretto* e non viene modificato. Se invece vengono ricevuti molti messaggi pari a 1, è presumilbile che il valore corrente sia *errato* e debba essere invertito - da questo punto di vista, il nodo di bit interpreta il messaggio come un "flag" di probabile bit errato. Il calcolo effettuato dal nodo di bit è dunque il seguente: *invertire il valore corrente se e solo se il numero di messaggi pari a 1 è > t*, dove *t* è un valore di soglia che deve essere aggiustato empiricamente (è di fatto un parametro di progetto del decoder) che dipende dal numero di rami confluenti nel nodo di bit (il cosiddetto *grado* del nodo). Se nel nostro esempio poniamo t = 1, otteniamo la configurazione di Fig. 6.24 (b) dopo il calcolo dei nodi di bit, che indicano adesso la stima aggiornata della parola di codice **a**⁽¹⁾.





Si procede poi alla seconda semi-iterazione, scoprendo (non mostrato in figura) che ancora alcuni nodi di controllo sono pari a 1, cioè che la parola $\mathbf{a}^{(1)}$ non è la stima finale, e così via.

La decodifica termina se viene trovata una parola di codice durante le iterazioni, come già accennato, oppure se si raggiunge un numero massimo di iterazioni N_{it} prefissato, $\hat{\mathbf{a}} = \mathbf{a}^{(N_{it})}$. La parola di sorgente si ricava poi banalmente dalla parola di codice decodificata perché il codice è sistematico: $\hat{b}[k] = \hat{a}[k], k = 0, \dots, K - 1$. Non è possibile dimostrare analiticamente che la decodifica iterativa converga su di un punto fisso e/o che trovi una parola di codice. Dal punto di vista pratico, il decoder raggiunge comunque ottime prestazioni in termini di BER nei casi pratici.

6.4.3 Decodifica iterativa dei codici LDPC - II: Soft-Input

L'algoritmo di decodifica cambia radicalmente se abbiamo a disposizione l'ingresso soft $\mathbf{r} = \mathbf{a}_{\pm} + \mathbf{w}$ proveniente da canale AWGN come nel caso dei codici Turbo. Dobbiamo sviluppare un nuovo algoritmo *message-passing* (a scambio di messaggi) sul grafo di Tanner con messaggi probabilistici di tipo *soft*, in analogia a quanto visto per i turbo-codici.

Lo scopo del nuovo decodificatore SISO, oltre a fornire i simboli decodificati $\hat{a}[n]$, è quello di calcolare le probabilità a posteriori

$$\Pr\left\{a[n] = x | \mathbf{r}, \{c_i = 0\}_{i \in C_{i,n}}\right\} \quad x = 0, 1$$
(6.57)

ovvero i LAPPR

$$L(a[n]) \stackrel{\triangle}{=} \ln \left(\frac{\Pr\{a[n] = 1 | \mathbf{r}, \{c_i = 0\}_{i \in C_{i,n}}\}}{\Pr\{a[n] = 0 | \mathbf{r}, \{c_i = 0\}_{i \in C_{i,n}}\}} \right)$$
(6.58)

dove $C_{i,n} \subset \{0, 1, ..., N - K\}$ indica l'insieme degli indici *i* delle equazioni di parità che riguardano il bit a[n] (in pratica, il condizionamento significa che tutte la equazioni di parità che contengono a[n] devono essere soddisfatte).



(a)

(b)

Figura 6.25 Sottografo per a[0] di Fig. 6.23 (a) con i messaggi verso il basso (b)

La derivazione dettagliata dell'algoritmo SISO di message-passing è molto pesante e viene riportata in Appendice D. Ci limitiamo qui di seguito a illustrarne l'andamento generale e le relazioni principali. In generale, l'articolazione dell'algoritmo SISO in due semi-iterazioni con scambio di messaggi tra i nodi segue da vicino quanto già visto nella sezione precedente, ma i messaggi sono ovviamente diversi. In primis, sono di carattere soft, e poi seguono il principio della estrinsecità già discusso per il turbo-decoder, cioè devono fornire al nodo cui sono destinati una certa quantità di informazione esterna che il nodo stesso non può (e non deve) avere disponibile a-priori.

Torniamo dunque al nostro codice di Hamming (7,4), e isoliamo il sottografo relativo al primo nodo di bit a[0] (la prima colonna della matrice di parity-check) come in Fig. 6.25 (a). Abbiamo aggiunto al grafo un ulteriore ramo intendendo che il nodo, oltre a rappresentare il valore binario del simbolo di codice a[0], riceve anche informazione direttamente dal canale nella forma del valore soft r[0]. Prendiamo adesso in considerazione i messaggi soft che scorrono sui rami del grafo e indichiamo con $q_{n,i}$ il messaggio inviato (verso il basso) dal nodo di bit n al nodo di controllo i, e con $\rho_{i,n}$ il messaggio (verso l'alto) inviato dal nodo di controllo i al nodo di bit n, n = 0, ..., N - 1, i = 0, ..., N - K - 1. Prendiamo inoltre come esempio particolare il messaggio dal nodo di bit 0 al nodo di check 2, cioè $q_{0,2}$ come illustrato in Fig.6.25 (b). Nella figura, le freccine indicano chiaramente il verso dei messaggi sui rami, suggeriscono il tipo di calcolo che il nodo esegue, e risolvono la questione della estrinsecità: il nodo c_2 riceve da a[0] un messaggio calcolato attraverso i relativi input, e cioè r[0] e i messaggi di controllo ricevuti verso l'alto da a[0] stesso nella precedente *iterazione*, e cioè i $\rho_{i,0}$. Tra questi ultimi *non* compare $\rho_{2,0}$ perché l'informazione che apporterebbe non è estrinseca ad a[0] perché è stata calcolata con un valore di a[0] della iterazione precedente, che il nodo a[0] conosce già.

Una situazione similare si ha considerando i messaggi inviati dai nodi di controllo ai nodi di bit, esemplificata in Fig. 6.26 per $\rho_{0,4}$, dove il nodo di controllo non riceve $q_{4,0}$

225

226 CODIFICA DI VANALE CODIFICA YI CANALE CODIFICA DI CANALE



perché non estrinseco. Nella notazione delle figure, i messaggi appaiono dipendenti da una variabile x: in teoria, su di ogni ramo non scorre *un solo* valore, ma *entrambe le probabilità* di certi eventi, probabilità dipendenti da presumere il valore del bit a[n] = x, x = 0, 1. In particolare,

$$q_{n,i}(x) \stackrel{\triangle}{=} \Pr\left\{a[n] = x \left| r[n], \{\rho_{i',n}\}_{i' \neq i}\right\} \ x = 0, 1$$
 (6.59)

mentre

$$\rho_{i,n}(x) \stackrel{\Delta}{=} \Pr\left\{c_i = 0 \ \middle| \ a[n] = x, \{q_{n,i'}\}_{i' \neq i}\right\} \quad x = 0, 1$$
(6.60)

Abbiamo detto poc'anzi "in teoria" riguardo i messaggi, perché in realtà ciò che l'algoritmo effettivamente calcola e scambia tra i nodi come messaggio è il LAPPR dei messaggi probabilistici di cui sopra:

$$L(q_{n,i}) \stackrel{\triangle}{=} \ln \frac{q_{n,i}(1)}{q_{n,i}(0)} , \quad L(\rho_{i,n}) \stackrel{\triangle}{=} \ln \frac{\rho_{i,n}(1)}{\rho_{i,n}(0)}$$

$$(6.61)$$

Conosciamo già bene il concetto di LAPPR L(b[k]) relativo al valore di un bit di informazione. Nell'algoritmo *message-passing*, il messaggio $L(q_{n,i})$ ha un ruolo similare: è la "quantità di certezza" o l'*opinione* (in inglese, *belief*) che ciascun nodo di bit può dare al nodo di controllo relativamente al valore del bit stesso per consentire l' ulteriore calcolo del nodo di controllo (nella versione *bit-flipping*, per calcolare la parità) - è chiaro che il *belief* rappresenta qualcosa di più del solo valore binario. Similmente, il messaggio $L(\rho_{i,n})$ è la " quantità di certezza" che il nodo di check, soddisfacendo la propria equazione di parità, può dare al nodo di bit riguardo al valore del bit stesso, e così eventualmente indurre una modifica nella stima del bit (nella versione *bit-flipping*, eventualmente invertire il bit). Questo è il motivo per cui l'algoritmo *message-passing* è anche chiamato *belief propagation*: i messaggi che vengono scambiati, piuttosto che semplici decisioni binarie sono valori soft di *belief* riguardo le medesime quantità.

La caratteristica del decoder LDPC message-passing è la possbilità di essere facilmente implementato su di un componente HW VLSI seguendo l'idea dettata del grafo di Tanner,

che implica una computazione già estremamente *parallelizzata*. Ogni nodo diventa in hardware un processore elementare che rispetta la caratteristica di *località* della computazione tipica del calcolo parallelo. I processori di una certa classe (bit,controllo) ricevono ad ogni semi-iterazione un set di ingressi soft e calcolano *in parallelo* i rispettivi valori di uscita soft da inviare all'altra classe di processori, e così via. La struttura hardware risultante è molto regolare, efficiente, e di semplice progettazione, al contrario del decodificatore BCJR dei codici turbo che risulta di implementazione più complessa.

Le operazioni di calcolo specifiche per ciascuna classe di nodi sono ricavate in Appendice D, ma risultano veramente elementari. Per il calcolo dei messaggi dai nodi di bit ai nodi di controllo all'iterazione m + 1 abbiamo:

$$L^{(m+1)}(q_{n,i}) = r[n] + \sum_{\substack{i' \in C_{i',n}, i' \neq i}} L^{(m)}(\rho_{i',n})$$
(6.62)

Possiamo interpretare questa legge come una specie di "decisione soft a maggioranza" sul bit a[n]: il primo termine è l'uscita nuda del canale che porterebbe a una decodifica di a[n]senza usare le eventuali correzioni apportate del codice; il secondo termine è la somma di tutte le indicazioni (opinioni) sul bit provenienti dai nodi di check, ottenute quindi sfruttando la ridondanza del codice, che possono portare al cambiamento del segno complessivo del messaggio e quindi all'opinione complessiva sul bit. Più opinioni sono in accordo (messaggi dello stesso segno), più è probabile che l'opinione complessiva sul bit sia corretta.

L'espressione del messaggio dai nodi di controllo ai nodi di bit sembra formalmente più complicata, ma è in realtà altrettanto semplice da interpretare e da comprendere, perlomeno nella versione semplificata *normalized-min-sum* del decodificatore che qui presentiamo:

$$L^{(m+1)}(\rho_{i,n}) = -\left\{\prod_{n'\in R_{i\setminus n}} \operatorname{sgn}[-L^{(m)}(q_{n',i})]\right\} \cdot \min_{n'\in R_{i\setminus n}} |L^{(m)}(q_{n',i})|$$
(6.63)

La relazione (6.63) va interpretata in *modulo e segno*: il segno derivante dal calcolo è quello che fa verificare l'equazione di parity check del nodo, cioè fa verificare l'evento $c_i = 0$ in (6.60) - per convincersene, bisogna osservare che nel nostro mapping dei valori logici 0 e l rispettivamente in -1 e +1 di canale, l'operazione di somma logica $a \oplus b$ viene mappata in $-(-a_{\pm} \cdot -b_{\pm})$. Per quanto riguarda il modulo, questo rappresenta l'affidabilità del *belief* riguardo ad a[n] ed "eredita" per sicurezza la minima affidabilità dei messaggi che la compongono.

Ad ogni step di decodifica, e come nel bit-flipping, il decoder min-sum calcola la stima corrente $\mathbf{a}^{(m)}$ della parola di codice e verifica l'eventuale annullamento di tutti nodi di check per eventualmente terminare l'iterazione e dichiarare $\hat{\mathbf{a}}^{(m)} = \mathbf{a}^{(m)}$, altrimenti prosegue con le iterazioni fino al numero massimo prestabilito N_{it} . La decisione di tentativo sulla parola di codice per verificare i parity-check viene presa calcolando il LAPPR su a[n] (6.58) secondo quanto illustrato in Appendice D:

$$L^{(m)}(a[n]) = r[n] + \sum_{i \in C_{i,n}} L^{(m)}(\rho_{i,n})$$
(6.64)

Ancora una volta, il LAPPR è composto da un termine "di canale" indipendente dal codice (vedi 33), e da un termine ulteriore che riassume in se' l'esecuzione dell'algoritmo di message-passing. Ad ogni iterazione di decodifica, i LAPPR si modificano , e alcuni *cambiano di segno* in modo da correggere eventuali errori di canale. In questo modo la BER migliora in generale da iterazione a iterazione fino al termine della decodifica.

La complessità dell'algoritmo di decodifica dipende dal numero di messaggi che vengono scambiati (e che quindi devono essere calcolati) sul grafo. Se la matrice **H** fosse casuale con elementi binari equiprobabili, in ogni nodo di bit confluirebbero mediamente (N - K)/2 rami e la complessità della decodifica sarebbe O(N(N - K)) cioè quadratica con l'ordine del codice. Se invece la matrice è low-density, e supponiamo per semplicità di un codice regolare, in ogni nodo di bit confluiscono w_c rami, con $w_c \ll N$ (altrimenti il codice non è low density) e la complessità è $N \cdot w_c$ cioè *lineare* con l'ordine del codice.

Come esempio didattico, vediamo in Fig. 6.27 le curve di BER dopo decodifica per un codice LDPC regolare sistematico (8000,4000) non ottimizzato su canale AWGN. Il miglioramento della BER al crescere del numero di iterazioni di decodifica è evidente. Ciò che al contrario la Fig. 6.27 non mette in evidenza è un fenomeno che accade a probabilità di errore molto basse: la pendenza della curva BER all' aumentare di E_s/N_0 cambia, e in particolare diminuisce; le curve tendono a un andamento quasi "orizzontale" chiamato floor che diminuisce l'efficienza del codice. Nelle applicazioni che si attendono un bit rate molto basso, come il video digitale ad alta qualità che richiede almeno 10^{-10} , il fenomeno del floor può costituire un inconveniente importante. La contromisura è ben nota e già adottata da tempo proprio nel campo della TV digitale: aggiungere all'LDPC, usato come codice interno, un codice esterno che corregga gli errori residui del decodificatore e elimini il fenomeno del floor (codici concatenati). Questo è esattamente ciò che viene fatto nel codice di canale per la trasmissione video DVB-S2, dove un LDPC con blocchi lunghi (N = 64800) è concatenato con un codice algebrico esterno BCH (32400,32208) a tasso molto alto ($r_o \simeq 0.994$). La Fig. 6.28 illustra le prestazioni eccellenti di questo codice concatenato BCH/LDPC quando viene utilizzato con le diverse costellazioni QP-SK,8PSK,16APSK,32APSK dello standard televisivo satellitare DVB-S2 con N = 64800 e con diversi tassi complessivi $r = r_i r_o$.

6.5 Codici Polari con Decodifica a Cancellazioni Successive

6.5.1 C'è davvero necessità dell'(n+1)-esimo codice?

I successi spettacolari dei codici Turbo ed LDPC con decodifica iterativa hanno quasi archiviato la ricerca sui codici di canale. Questi codici, nelle implementazioni più efficienti (un esempio è quello già discusso del concatenato BCH/LDPC del DVB-S2), giungono a meno di 1 dB di efficienza del limite di Shannon. Tuttavia è vero che queste prestazioni si ottengono con lunghezze di blocco molto elevate, ad esempio N = 64800 per il DVB-S2. Blocchi di codice così lunghi sono proibitivi per comunicazioni che richiedono il tempo reale o comunque vincoli stringenti di *latenza* (voce o video bidirezionale, machine-to-machine communications ecc.).

Tecnologie per collegamenti ottimizzati con blocchi (pacchetti) corti erano ancora un problema aperto alla fine degli anni 2000, anche dopo l'avvento della decodifica iterativa. Prendiamo come esempio le curve di BER del codice 3gpp PCCC turbo di Fig. 6.19 mostrate in Fig. in Fig. 6.29 al variare della lunghezza del pacchetto (blocco di sorgente). Si nota che il guadagno di codifica è *fortemente* dipendente dalla lunghezza del blocco.

Questo fenomeno è in realtà ben noto: la curva della capacità di Shannon che abbiamo usato nella Fig. 4.23 rappresenta la cosiddetta capacità *unconstrained*, cioè senza alcun vincolo sulla lunghezza dei blocchi di codice, cha anzi si suppone arbitrariamente grande. Per blocchi finiti, la capacità si riduce, e ci si deve attendere una degradazione per *qualunque* codice. La Fig. 6.30 mostra appunto un limite inferiore (in rosso) e un limite superiore







Figura 6.28 BER del codice concatenato BCH/LDPC con le costellazioni del DVB-S2. Courtesy of WISER srl



Figura 6.30 Degradazione per codici di lunghezza finita rispetto alla capacità non vincolata di Shannon

(in blu, tratteggiato) della degradazione in termini di E_b/N_0 a parità di efficienza spettrale rispetto alla capacità non vincolata per una certa dimensione del blocco di informazione K. Si nota una degradazione di un paio di dB per blocchi di qualche centinaio di bit, e in queste condizioni trovare codici efficienti come turbo/LDPC potrebbe non essere semplice. L'esigenza era dunque: trovare codici ottimali a blocchi corti - l'oggetto della prossima sezione.



Figura 6.31 Canale vettoriale ottenuto come aggregazione di canali scalari

6.5.2 Teorema di polarizzazione e codici polari

I codici polari sono stati inventati da Arikan nel 2009, e sono basati sull'idea della *polarizzazione* di un insieme di canali paralleli di comunicazione. Il "parallelo" dei canali è una questione concettuale e viene dal considerare gli N usi consecutivi e indipendenti di un singolo canale (BSC, BEC o quant'altro) per inviare N simboli codificati come un *unico* canale *parallelo* di dimensione N la cui uscita è direttamente d (supponendo hard-output). La capacità totale del canale (parallelo) è $C_{vec} = NC$ dove C è la capacità del canale "scalare" originale in bit/c.u.. Se il canale originario è un BSC o un BEC, sappiamo che la capacità è pari alla mutua informazione quando l'ingresso è equiprobabile.

Questo canale vettoriale è chiaramente un artefatto concettuale, dato che gli usi del canale originario sono indipedenti tra di loro, e quindi aggregare i canali in un unico canale vettoriale (congiunto) non aggiunge niente al problema. Scatta però adesso l'idea fondamentale del polar coding, e cioé quella di "ridistribuire" la capacitá creando un set di canali *differenti* dagli originari attraverso una certa pre-codifica dei bit di informazione. La capacità totale del nuovo set di canali sarà comunque uguale a C_{vec} perchè la pre-codifica sarà una trasformazione deterministica invertibile, ma le nuove capacità singole dei nuovi canali saranno (molto) diverse dalle originali, con alcuni canali nel nuovo insieme decisamente "buoni" cioè con capacità vicino a 0 - in altre parole, realizzando la *polarizzazione* dei canali di di comunicazione.

Cerchiamo di illustrare questo principio generale con un esempio semplice 2×2 . La Fig. 6.32 rappresenta una semplice pre-codifica di due bit di ingresso c[0] e c[1] per ottenere due bit codificati a[0] e a[1] da inviarsi in due canali paralleli identici (ad esempio due BSC indipendenti), le cui uscite sono rispettivamente d[0] d[1]. I due nuovi canali di comunicazione non sono semplicemente analoghi agli originali ad ingresso singolo e uscita singola (cioè da c[0] a d[0] da c[1] a d[1] separatamente): il nuovo canale #0 deve intendersi da c[0] all'*intero vettore ricevuto* $\mathbf{d} = [d[0], d[1]]$, mentre il canale #1 é da $c[1] a (\mathbf{d}, c[0])$ dove concatenare d con c[0] significa decodificare c[1] *utilizzando anche la decodifica già effettuata di* c[0]. La capacità di questo nuovo canale vettoriale dato dall'unione di #0 e #1 è ancora $C_0 + C_1 = 2C$, però è chiaro che $C_1 \ge C$: il nuovo canale #1 si avvale di ulteriori osservazioni di quantità (d[0] = c[0]) che il vecchio canale 1 da c[1] a d[1] non aveva - la mutua informazione è sicuramente non inferiore. Di conseguenza, $C_0 \le C$, cioè complessivamente $C_0 \le C \le C_1$: i due nuovi canali sono differenti, uno migliore dell'altro, cioè sono *polarizzati*. Ad esempio, se i due canali originali sono due BEC(p_{CE})



Figura 6.32 Pre-Codifica polare 2×2

indipendenti, entrambi con capacità $C = 1 - p_{CE}$, allora

$$\mathcal{C}_0 = (1 - p_{CE})^2 < 1 - p_{CE} < 1 - p_{CE}^2 = \mathcal{C}_1 \tag{6.65}$$

Il nostro "esempio giocattolo" 2×2 suggerisce già uno schema intuitivo di codifica (N = 2, K = 1) a tasso r = 1/2: basta *non* utilizzare il nuovo canale "cattivo" #0, e usare soltanto il nuovo canale "buono" #1; la codifica si realizza facilmente *congelando* [0] a un valore noto e concordato, per esempio 0, e utilizzando come bit di informazione (soltanto) l'altro bit d'ingresso c[1] (ponendo cioè c[1] = b[0]). Il decodificatore sa che c[0] è pari 0 e usa quest'informazione per decodificare im modo ottimo c[1] sulla base del blocco ricevuto $\mathbf{d} = [d[0], d[1]]$.

I veri codici polari saranno naturalmente più complicati. In primis, dobbiamo estendere l'idea della polarizzazione parecchio più in là: all'interno di un blocco di N simboli **c** dobbiamo creare K canali di bit la cui capacità sia molto vicina ad 1, con comunicazione totalmente affidabile, accompagnati da N - K canali aggiuntivi la cui capacità sarà vicina a 0 (canali inutili) che useremo soltanto per "trasmettere" bit congelati. Il tasso del codice sarà dunque il solito r = K/N, e potremo avvalerci del *teorema di polarizzazione*:

Dati N canali scalari indipendenti con capacità C ciascuno, se N è sufficientemente grande, per ogni ε (arbitrariamente piccolo), il numero di canali polarizzati ottenuti dagli N canali originali e la cui capacità è maggiore di $1-\varepsilon$ è pari a NC, e il numero di canali la cui capacità è più piccola di ε è N - NC

Il teorema garantisce che per un N sufficientemente grande (ancora blocchi lunghi...), il tasso del codice polare r = K/N = (NC)/N è asintoticamente uguale alla capacità C del canale scalare originale, e dimostra quindi il raggiungimento del limite di Shannon da parte del codice - un fatto unico nel panorama dei codici fino ad oggi (2020) utilizzati.

6.5.3 Costruzione e decodifica dei codici polari

Il concetto di polarizzazione si può estendere rispetto all'esempio 2×2 per creare codici polari con blocchi lunghi e cercare di raggiungere la capacità. Consideriamo la Fig. 6.33,che mostra l'*i*-esimo canale di bit polarizzato avente come ingresso il bit c[i] e come uscita l'intero vettore ricevuto d *e anche* gli *i* bit precedentemente decodificati c[0], ..., c[i-1]. Qual è la pre-codifica da utilizzare per realizzare la polarizzazione? La risposta non è troppo sorprendente: se N è sufficientemente grande, potremmo usare un *polarizzatore casuale* - una semplice permutazione casuale dei bit dà luogo con alta probabilità a una buona polarizzazione, come si vede in Fig. 6.34 che rappresenta la capacità dei vari canali di bit polarizzati in un blocco di N = 32 bit di ingresso con polarizzazione casuale su di un vettore di 32 BEC(1/2) (capacità C = 1/2) indipendenti. Si nota chiaramente la presenza di

CODICI POLARI CON DECODIFICA A CANCELLAZIONI SUCCESSIVE



Figura 6.34 Polarizzazione di N = 32 BEC(1/2) con polarizzatore casuale

canali "buoni" e "cattivi" in una percentuale circa 50-50, cosicché il tasso del codice che congela i bit sui canali cattivi e trasmette solo quelli sui canali buoni è r = 1/2, proprio come la capacità dei canali originali e come predetto dal teorema di polarizzazione.

Come ci si può attendere però, un polarizzatore completamente random e destrutturato non porta ad algoritmi semplici ed efficienti di de/codifica. Però la costruzione "giocattolo" 2×2 del polarizzatore che riprendiamo in Fig. 6.35 può essere generalizzata ricorsivamente e conduce a buoni risultati con complessità adeguata. Infatti, si può costruire un pre-coder (chiamiamolo d'ora in poi genericamente *codificatore*) di ordine N = 4 a partire da quello di ordine 2 se lo raddoppiamo verticalmente e aggiungiamo un secondo strato di 4/2 = 2

233



Figura 6.35Codificatore polare di ordine 2



(b)

Figura 6.36 Costruzione ricorsiva del codificatore polare di ordine 4

codificatori in cascata ai due precedenti, con la differenza che gli ingressi dei due nuovi codificatori sono presi con un offset reciproco pari a 4/2 = 2 linee di ingresso, ottenendo così lo schema di Fig. 6.36. A questo punto, abbiamo già capito come estendere all'ordine N = 8: partiamo con due encoder di ordine 4 uno sopra l'altro e aggiungiamo un secondo strato di codifica fatto da una schiera di 8/2 = 4 encoder di ordine 2 con ingressi offsettati di 8/2 = 4 linee di ingresso, ottenendo il codificatore di Fig. 6.37. In generale, il codificatore di ordine 2N si ottiene impilando due codificatori di ordine N e aggiungendo una schiera di N codificatori di ordine 2, ciascuno con ingressi offsettati di N linee.

La formalizzazione matematica di questa costruzione è semplice. La parola di codice a in ingresso al "canale parallelo" è ottenuta all'uscita del codificatore che riceve il blocco c (contenente bit di informazione e congelati), come in Fig. 6.37. La relazione tra c e a si generalizza ad ogni ordine $N = 2^m$ come il semplice prodotto vettore(-riga)-matrice $\mathbf{a} = \mathbf{c}\mathbf{G}^{(m)}$, dove la matrice di pre-codifica quadrata $2^m \times 2^m \mathbf{G}^{(m)}$ per $m \ge 1$ si ricava ricorsivamente come segue:

$$\mathbf{G}^{(m+1)} = \mathbf{G}^{(m)} \otimes \mathbf{G}^{(1)} , \ \mathbf{G}^{(1)} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$
(6.66)

dove \otimes indica il prodotto di Kronecker binario tra matrici. Ad esempio,

$$\mathbf{G}^{(2)} = \mathbf{G}^{(1)} \otimes \mathbf{G}^{(1)} = \mathbf{G}^{(1)} \otimes \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{G}^{(1)} & \mathbf{0}_{2 \times 2} \\ \mathbf{G}^{(1)} & \mathbf{G}^{(1)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$
(6.67)

CODICI POLARI CON DECODIFICA A CANCELLAZIONI SUCCESSIVE



igura 0.57 Councatore polare e canale di ordine

corrispondente all' encoder di Fig. 6.36.

La costruzione ricorsiva per ordini potenza di 2 ricorda molto da vicino la costruzione altrettanto ricorsiva della matrice binaria ortogonale **H** di *Walsh-Hadamard* (2.137), e/o la matrice, altrettanto ortogonale complessa **W** dell'algoritmo di DFT. Questa constatazione suggerisce che la complessità algoritmica della codifica polare è del tipo $N \log N$, come per gli algoritmi di trasformata di Fourier o di Walsh veloce, e quindi ben gestibile in tempo reale anche per grandi lunghezze di blocco.

Riprendiamo adesso l'esempio di ordine 8 con canale BEC(1/2) (capacità C = 1/2 bit/c.u.) di Fig. 6.37. Dopo qualche calcolo, si trovano le capacità dei canali polarizzati secondo questa pre-codifica, che sono indicate alla sinistra dello schema nella figura. Osservando questi valori, notiamo che si può individuare uno schema di codifica immaginando di *congelare* i simboli di ingresso # 0,1,2, e 4 (canali peggiori), e usare solo i simboli #3,5,6,7 per inviare i bit di informazione. Il codice polare (8, 4) risultante sarà dunque

$$\mathbf{a} = [\phi[0], \phi[1], \phi[2], b[0], \phi[3], b[1], b[2], b[3]] \mathbf{G}^{(3)}$$
(6.68)

dove ϕ indica il generico bit congelato, il cui valore deve essere concordato tra codificatore e decodificatore (ad esempio, tutti 0) e fa parte integrante dello schema di codifica. L'individuazione dei bit da congelare e da trasmettere dipende evidentemente dalle particolari proprietà del canale originario. In pratica, si trovano delle regole generali abbastanza universali da applicarsi a qualunque canale (ad esempio, BSC o AWGN), specialmente per blocchi relativamente lunghi.

235

236 CODIFICA DI VANALE CODIFICA YI CANALE CODIFICA DI CANALE

6.5.4 Decodifica dei codici polari con esempi

La strategia di decodifica dei codici polari è intrinseca alla definizione del canale polarizzato con uscita vettoriale e conoscenza dei simboli già decodificati mostrata in Fig. 6.33. In quest'ottica, l'algoritmo di decodifica si basa sulla *cancellazione successiva* dei bit già decodificati, nonché ovviamente sulla conoscenza *a-priori* dei bit congelati.

Ripartendo dall'esempio di Fig. 6.37, la procedura inizia ponendo $\hat{c}[0] = 0$ perchè si tratta di un bit congelato, così come $\hat{c}[1] = 0$ e $\hat{c}[2] = 0$. La prima vera operazione di decodifica si applica a c[3], ed è basata sul LAPPR

$$L(c[3]) = \log \frac{\Pr\left\{c[3] = 1 \mid \mathbf{d}, \hat{c}[0], \hat{c}[1], \hat{c}[2]\right\}}{\Pr\left\{c[3] = 0 \mid \mathbf{d}, \hat{c}[0], \hat{c}[1], \hat{c}[2]\right\}}$$
(6.69)

ovvero, poichè i bit di informazione si considerano equiprobabili, sul LLR

$$L(c[3]) = \log \frac{\Pr\left\{\mathbf{d}, \hat{c}[0], \hat{c}[1], \hat{c}[2] \mid c[3] = 1\right\}}{\Pr\left\{\mathbf{d}, \hat{c}[0], \hat{c}[1], \hat{c}[2] \mid c[3] = 0\right\}}$$
(6.70)

Dal segno di L(c[3]) troviamo come sempre $\hat{c}[3] = \hat{b}[0]$, e usiamo poi tale stima di c[3], insieme con il valore congelato di c[4], per decodificare c[5] di nuovo attraverso lo LLR:

$$L(c[5]) = \log \frac{\Pr\left\{\mathbf{d}, \hat{c}[0], \hat{c}[1], \hat{c}[2], \hat{c}[3], \hat{c}[4] \mid c[5] = 1\right\}}{\Pr\left\{\mathbf{d}, \hat{c}[0], \hat{c}[1], \hat{c}[2], \hat{c}[3], \hat{c}[4] \mid c[5] = 0\right\}}$$
(6.71)

e così via. In generale

$$L(c[n]) = \log \frac{\Pr\left\{\mathbf{d}, \hat{c}[0], ..., \hat{c}[n-1] \mid c[n] = 1\right\}}{\Pr\left\{\mathbf{d}, \hat{c}[0], ..., \hat{c}[n-1] \mid c[n] = 0\right\}}$$

dove le stime precedenti $\hat{c}[i]$ sono un misto tra vere stime e valori congelati.

Il calcolo dei vari LLR dipende dal tipo di canale (BSC, BEC, etc.). Per il canale AWGN, il blocco osservato diventa il consueto vettore soft r (invece del vettore digitale d) che abbiamo già incontrato nella decodifica dei codici turbo ed LDPC. Il calcolo degli LLR e l'algoritmo a cancellazioni successive è riportato per completezza in Appendice E. Tale decodificatore "indotto" dalla struttura del pre-codificatore si rivela però sub-ottimo, e rende i codici polari inferiori agli LDPC/Turbo. Ci sono due motivi principali che rendono insoddisfacente il *Successive-Cancelation Decoder* (SCD): in primis, la decodifica non è *veramente* iterativa, nel senso che una volta che un bit di informazione è stato deciso, non si fa nessuna ulteriore iterazione di decodifica per migliorare la decisione sulla base di informazioni successive; inoltre, la stima di un certo bit c[i] non sfrutta la conoscenza a-priori dei bit congelati nei $c[\ell], \ell > i$.

L'idea attuale per migliorare relativamente a entrambi i due punti deboli lo SCD è quella di "rinviare" la decisione il più possibile attraverso un algoritmo *a lista* chiamato Successive-Cancelation List Decoder (SCLD): invece di eseguire una decisione definitiva ad ogni step di decodifica, e usarla poi per cancellare il bit nei passi successivi, l'algoritmo mantiene lo LLR, ed esegue due passi successivi di decodifica per *entrambi* i possibili valori decisi 0 e 1, mantenendo di nuovo gli LLR e così via. È chiaro che questo procedimento genera un "albero" di decodifica che cresce in modo esponenziale e può diventa rapidamente ingestibile. Il trucco sta nel mantenere soltanto, dopo i vari step, una *lista* raccorciata degli L migliori candidati per la stima finale, cioè le sub-sequenze che accumulano il massimo LLR totale. La complessità dell'algoritmo diventa $L \cdot N \log N$, pari ad avere L SCDs



(b)

Figura 6.38 Curve di BER di un codice polare N=2048, r=1/2 con SCD (rosso) e con SCLD (altri colori, al variare di L)

standard che lavorano in parallelo (vedi sotto). Vediamo in Fig. 6.5.4 il miglioramento delle curve di BER del decodificatore a lista per diversi valori di L rispetto al semplice SCD (curva rossa).

Non insistiamo ulteriormente sugli algoritmi di decodifica e le relative prestazioni per i codici polari; al momento (2020), il decoder SCLD è, ancorché non ottimale, è considerato lo standard. La ricerca è ancora in corso, ed è probabile che il migliore algoritmo di decodifica ad oggi disponibile venga in un futuro anche vicino soppiantato da altri ancora non noti e migliori. In ogni modo, riportiamo in Appendice E la derivazione-base degli LLR sul canale AWGN e il risultante algoritmo-base a cancellazioni successive. La complessità del decodificatore è $N \log N$ poiché segue la struttura ricorsiva per potenze di 2 del codificatore - confrontabile con la complessità lineare del decodificatore LDPC.

Il "trucco" finale per rendere il codice polare competitivo, come per gli altri codici "moderni", è quello di concatenarlo con un codice algebrico *esterno* ad alto tasso (vedi Fig. 6.17). Per il codice polare, la concatenzione è in realtà più profonda di quella applicata agli LDPC, per esempio. Il decodificatore infatti non é semplicemente la cascata dei due decodificatori separati per i due codici, ma è un algoritmo specifico per la concatenazione. L'approccio dello standard 5G New Radio per un codice polare standard è il seguente: si utilizza un Cyclic Redundancy Code (CRC) per aggiungere P bit di ridondanza ai K bit di informazione, ancora prima di aggiungere gli ulteriori N - K - P bit congelati per costruire il blocco c (quindi facendo una concatenazione convenzionale). Durante la decodifica SCLD, si mantengono in lista i candidati che hanno la più alta metrica *e* verificano il CRC (*expurgated* list). Il miglioramento di prestazioni è illustrato dalla Fig. 6.39, il cui codice comprende solo P = 16 bit di parità aggiuntivi.



Figura 6.39 Curve di BER di un codice polare N=2048, r=1/2 con SCD (rosso) e concatenato CRC/polare con SCLD (altri colori, al variare di L)

6.6 Codifica Link-Layer con codici "Rateless"

Abbiamo spesso citato esempi provenienti da un contesto Internet-like di comunicazione pacchettizzata, a cominciare dal canale BEC di Fig. 4.3 (b) che modella eventi di perdita di pacchetti. Abbiamo appena introdotto i codici polari, sviluppati inizialmente proprio per canali BEC, e abbiamo anche visto precedentemente come correggere cancellazioni (anche) con codici tradizionali: dopotutto nella decodifica di un codice perforato come in Fig. 6.21 si è costretti a presentare al decoder (per il codice originale) una versione del segnale ricevuto dove si introducono esplicitamente delle cancellazioni - dunque recuperare una cancellazione non è troppo diverso da correggere un errore. In generale però aggiungere (molta) ridondanza per cautelarsi contro possibili cancellazioni (in particolare perdita di pacchetti) può non rivelarsi la migliore strategia se si ha a disposizione una informazione di feedback dal destinatario al mittente. In questo caso, se il destinatario si accorge di una cancellazione (ad esempio, un CRC di pacchetto errato), può rinviare al trasmettitore un messaggio di mancata ricezione (NACK, Negative Acknowledgement of Recepit) e richiedere la ritrasmissione del pacchetto. La richiesta di ritrasmissione è tipicamente gestita all'interno del layer (livello) Data Link dello stack OSI, dopo che il Physical layer ha fatto del suo meglio, inclusa la codifica di canale, per consegnare bit corretti. Se si verifica una perdita di pacchetto, è appunto compito del Link Layer di gestire questo evento a più alto livello.

Ripetere il pacchetto perduto (anche più di una volta se la probabilità di perdita è alta) può essere considerata un'ulteriore aggiunta di ridondanza che viene applicata ai bit di informazione proprio dal Link Layer, e che può essere classificata come una forma di *Link-Layer Coding* (LLC). Già sappiamo che *ripetere* costituisce la forma più elementare di codifica, e ci chiediamo quindi: esistono schemi più efficienti di LLC? La risposta è stata data per primo da Luby che ha introdotto nel 2002 il concetto di *codice a fontana* (Fountain Code).
6.6.1 Rateless/Fountain Codes

L'idea di base dei codici a fontana (FC) è in realtà semplice: supponiamo che il mittente invii un messaggio (ad esempio, un file di musica .mp3) costituito da K pacchetti di lunghezza fissa ℓ , per un totale di $K\ell$ bit. Il codificatore a fontana ideale è una sorgente (teoricamente infinita) di blocchi di bit codificati di lunghezza ℓ (la fontana, appunto) ottenuti in qualche modo dal messaggio originale, con la proprietà che non appena il destinatario riceve K' pacchetti codificati, con K' (appena) più grande di K, è in grado di ricostruire il messaggio originale, indipendentemente da quali pacchetti vengano ricevuti e in che ordine. La definizione allude al fatto che durante l'invio possono essere persi (cancellati) alcuni pacchetti, ma che il messaggio viene recuperato non appena ne vengano correttamente ricevuti K' > K - il contesto è quindi quello di un canale con erasure a *blocco*.

La definizione giustifica anche il perché gli FC sono chiamati *senza tasso (rateless)*: l'effettivo tasso di codifica non è definito perché dipende da caso a caso - se il canale è buono vengono persi pochi pacchetti e la decodifica richiede l'emissione di pochi pacchetti oltre K (poca ridondanza), viceversa se il canale è cattivo è necessario trasmettere più pacchetti codificati e la ridondanza sarà maggiore.

I codici a fontana sono adatti sia per le comunicazioni punto-punto individuale con feedback, sia per un contesto multicast/broadcast: nel primo caso, non appena il messaggio viene recuperato, il destinatario rinvia al mittente un suo messaggio di conferma positivo (ACK) e la "fontana" viene interrotta. Nel secondo caso, si fissa una massima lunghezza del messaggio codificato della fontana, sperando che tutti gli utenti non abbiano mai bisogno di ulteriori pacchetti; ogni singolo utente è comunque in grado di decodificare il messaggio dopo un (suo proprio) numero adeguato di pacchetti ricevuti, numero sperabilmente inferiore a quello massimo.

Questa tecnologia si utilizza anche per l'archiviazione distribuita di una grande mole di dati (lunghezza $K\ell$ molto grande) su supporti fisici differenti (pensiamo per semplificare a un hard-disk in configurazione RAID). Se il file da memorizzare è grande, è molto probabile che alcuni blocchi del file vengano danneggiati durante la scrittura/lettura sui/dai diversi supporti fisici. Quindi, invece di scrivere semplici copie del messaggio originale, scriviamo una fontana di blocchi (pacchetti) del messaggio (file) per renderlo robusto contro errori di archiviazione.

I FC sono anche utili per lo scaricamento di *file* molto grandi (un film, l'aggiornamento del sistema operativo ecc.) da una rete di server distribuiti. I server, ricevuto il comando di download, possono ciascuno far partire una (diversa) fontana ottenuta dallo stesso file, e il client ricostruirà tutto il messaggio (il file) non appena riceverà un numero di pacchetti sufficienti, anche derivanti da server differenti, minimizzando il tempo di download e senza richiedere coordinamento tra i sever stessi.

6.6.2 Il codice a fontana lineare casuale

Supponiamo dunque di avere un messaggio $\mathbf{B} = [\mathbf{b}[0], \mathbf{b}[1], \dots, \mathbf{b}[K-1]]$ composto da K pacchetti (blocchi) di lunghezza ℓ bit ciascuno. Il messaggio viene memorizzato interamente (in modo da poter usare ogni pacchetto per la codifica) dopodiché la Fontana inizia a produrre pacchetti codificati $\mathbf{a}[i]$ della medesima lunghezza ℓ , $i = 0, 1, \dots$ calcolando una combinazione lineare di tutti i pacchetti del messaggio:

$$\mathbf{a}[i] = \bigoplus_{k=0}^{K-1} g_{i,k} \mathbf{b}[k] =$$
(6.73)



Figura 6.40 Codice a fontana con perdita di pacchetti

I coefficienti binari $g_{i,k}$ della combinazione lineare sono *bit casuali indipendenti* generati dalla fontana prima della codifica di un nuovo pacchetto a (ancora una codifica casuale !). In pratica, i coefficienti seguono una sequenza pseudo-casuale preordinata.

Immaginiamo ora che il destinatario riceva (i primi) K' pacchetti non cancellati $\bar{\mathbf{a}}[0] \stackrel{\triangle}{=} \mathbf{a}[i_0], \bar{\mathbf{a}}[1] \stackrel{\triangle}{=} \mathbf{a}[i_1], \dots, \bar{\mathbf{a}}[K'-1] \stackrel{\triangle}{=} \mathbf{a}[i_{(K'-1)}]$ e voglia decodificare il messaggio⁴, come mostrato nella Fig. 6.40. Dato che i coefficienti di codifica sono casuali, si capisce che qualunque pacchetto è equivalente a qualunque altro nella decodifica. Raccogliendo i K' pacchetti, otteniamo un vettore $1 \times K'$ di pacchetti codificati legato al messaggio $1 \times K$ come segue:

$$[\bar{\mathbf{a}}[0], \bar{\mathbf{a}}[1], \dots, \bar{\mathbf{a}}[K'-1]] = [\mathbf{b}[0], \mathbf{b}[1], \dots, \mathbf{b}[K-1]] \mathbf{G}$$
 (6.74)

ovvero $\mathbf{A} = \mathbf{B}\mathbf{G}$, dove $\mathbf{G} = \{g_{i,k}\}$ è la *matrice generatrice* di dimensione $K \times K'$ di questo codice casuale. Proprio perché \mathbf{G} è casuale, non possiamo essere certi al 100% di riuscire a decodificare con successo. La decodifica va a buon fine infatti solo se \mathbf{G} ha *rango pieno*, cioè possiamo trovare al'interno di \mathbf{G} (almeno) K colonne linearmente indipendenti, e con quelle costruire una sottomatrice \mathcal{G} quadrata invertibile $K \times K$ in modo che $\mathbf{B} = \mathbf{A}\mathcal{G}^{-1}$.

Se invece la matrice non è a rango pieno, la decodifica con K' pacchetti fallisce chiamiamo la probabilità di questo evento p_{DF} (Decoding Failure), cosicché la probabilità di decodifica sará $1 - p_{DF}$. Il risultato fondamentale della codifica con trasformazione casuale (Random Transform Coding) è che

$$p_{DF} \le 2^{-(K'-K)} \tag{6.75}$$

e il limite è molto stretto (accurato) per K grande. Volendo assicurarsi una certa (bassa) probabilità di mancata decodifica p_{DF} , dobbiamo accettare un *overhead* ovvero una ridondanza

$$K' - K \ge \log_2\left(\frac{1}{p_{DF}}\right) \tag{6.76}$$

⁴Non sappiamo quanti pacchetti devono essere inviati dalla fontana prima di riceverne K' a causa delle perdite (cancellazioni) sul canale; non sappiamo neanche qual è il numero d'ordine di invio *i* di tali pacchetti per lo stesso motivo. Ecco perchè dobbiamo utilizzare la notazione (complicata) $i_{k'}, k' = 0, 1, \ldots, K' - 1$

cosicchè il tasso di codifica al destinatario è

$$r \le \frac{1}{1 - \log_2\left(p_{DF}\right)/K} \tag{6.77}$$

mentre quello al trasmettitore, tenendo conto di una certa probabilità di perdita dei pacchetti p_{CE} sarà

$$r \le \frac{1 - p_{CE}}{1 - \log_2\left(p_{DF}\right)/K} \tag{6.78}$$

La buona notizia del codice con trasformazione casuale è che la probabilità di mancata decodifica può essere resa arbitrariamente piccola aumentando K' oltre K, e che inoltre la ridondanza necessaria imponendo una probabilità di mancata decodifica "ragionevolmente piccola" e messaggi "ragionevolmente lunghi" è molto piccola; la cattiva è che la complessità della codifica è K^2 , e della decodifica è K^3 (derivante dall'inversione di \mathcal{G}) - poco attraente per applicare in pratica questa tecnica.

6.6.3 FC basati sulla trasformata di Luby

Un buon codice a fontana dovrebbe mantenere le ottime prestazioni del codice casuale, migliorando però la complessità della codifica e decodifica. L'idea è quella di continuare a costruire pacchetti codificati come combinazione lineare di quelli del messaggio, scegliendo i coefficienti in modo ancora aleatorio, ma con una distribuzione diversa dalla equiprobabile/indipendente, in modo che la matrice G diventi *sparsa*, ovvero low-density, e quindi riducendo a lineare la complessità come per gli LDPC!

Seguendo il criterio della Trasformata di Luby, per ottenere il pacchetto codificato a[i], si fissa per prima cosa il numero D di coefficienti pari a 1 nell'insieme dei $\{g_{i,k}\}_{k=0}^{K-1}$ (il grado del pacchetto codificato), cioè si decide quanti dei pacchetti b[k] verranno sommati per ottenere a[i]; poi, i particolari pacchetti che devono essere utilizzati vengono scelti a caso con equiprobabilità tra i b[k], k = 0, ..., K - 1. Il numero D è una variabile aleatoria che segue una certa legge di probabilità $p(d) \triangleq \Pr\{D = d\}$ che dovrà presentare alti valori per piccoli numeri d, in modo che si sommino pochi pacchetti e quindi la matrice generatrice **G** risulti low-density. La complessità della codifica si riduce a $K' \in \{D\}$, cioè quasi lineare in K se $E\{D\} \ll K$, requisito fondamentale che la p(d) deve soddisfare.

La decodifica avviene per cancellazioni successive. Associamo alla matrice G un grafo di Tanner, dove i nodi di bit rappresentano i K pacchetti $\mathbf{b}[k]$ da decodificare, e i nodi di check sono i K' pacchetti codificati a[i] - vediamo un esempio in Fig. 6.41 dove i pacchetti sono composti soltanto da $\ell = 1$ bit, K = 3, K' = 4 e i pacchetti ricevuti a[i]sono i valori 1011. In primis, per poter decodificare i K' pacchetti osservati, è necessario accertarsi che non vi siano nodi di bit "isolati", cioè bisogna accertarsi che nella scelta casuale dei K pacchetti di sorgente che compongono il set dei K' codificati, i pacchetti di sorgente siano stati (con alta probabilità) usati tutti, altrimenti la decodifica fallisce. Questo ulteriore vincolo dovrà essere tenuto in dovuta considerazione quando fisseremo la legge di probabilità p(d), insieme con il requisito di densità media dei rami del grafo di Tanner già menzionata. Inoltre, la decodifica si puà avviare solo se esiste (almeno) un pacchetto ricevuto che ha grado 1 (e questo è noto al decoder che conosce i coefficienti $g_{i,k}$), come in Fig. 6.41 (a). Se questa seconda condizione non si verifica, è necessario attendere di ricevere ulteriori pacchetti, oppure si dichiara la decodifica fallita se si ha un vincolo temporale e non si possono attendere altri pacchetti. Chiamiamo questo pacchetto iniziale $a[i_0]$. Il relativo nodo di controllo esegue la decodifica $\mathbf{b}[i_0] = \mathbf{a}[i_0]$ e invia il messaggio $\mathbf{b}[i_0]$ verso l'unico



Figura 6.41 Decodifica del codice a fontana con trasformata di Luby (da [10])

nodo di bit cui è connesso (Fig. 6.41 (b)). Il nodo di bit memorizza questo valore (finale) e lo re-invia verso tutti i nodi di controllo cui è connesso, i quali lo aggiungono al loro valore corrente, cancellando il bit decodificato. Il nodo di bit $\mathbf{b}[i_0]$ e tutti i rami connessi vengono eliminati dal grafo (sono ormai inutili) ottenendo il sottografo di Fig. 6.41 (c), sul quale si riparte da capo con un'altro step di decodifica: si cerca un nodo di check con grado 1, si decodifica $\mathbf{b}[i_1]$... fino alla decodifica completa.

La decodifica può chiaramente fallire ad ogni passo, e quindi il progetto della legge di probabilità dei gradi è cruciale: da una parte vogliamo bassi gradi per mantenere la codifica low-density e avere almeno un pacchetto codificato a peso 1; dall'altra, abbiamo bisogno ogni tanto di pacchetti codificati ad alto grado per essere sicuro di non creare pacchetti di sorgente "isolati". Luby dimostra che il minimo grado medio per "coprire" tutti i pacchetti di sorgente senza lasciarne isolati è $E\{D\} = \ln(K)$, che garantisce anche una sufficientemente bassa densità del grafo, pari a $E\{D\} \cdot K = K \ln(K)$. La legge p(d) che ha con buona approssimazione questo valore medio è la cosiddetta *a solitone*:

$$p_S(d) = \begin{cases} \frac{1}{K} & d = 1\\ \frac{1}{d(d-1)} & 2 \le d \le K \end{cases}$$
(6.79)

Con queste probabilità, pacchetti a grado 1 sono molto probabili, ma la probabilità abbastanza alta (anche) degli altri valori causa mancate decodifiche abbastanza frequentemente. Per evitare questo, si introduce la legge di probabilità del *solitone robusto* come segue:

$$p_{RS}(d) = \frac{p_S(d) + p_\tau(d)}{P}$$
(6.80)

dove $P = 1 + \sum_{d=1}^{K} p_{\tau}(d)$ è una costante di normalizzazione, e dove la legge addizionale

CODIFICA LINK-LAYER CON CODICI "RATELESS" 243



Figura 6.42 Confronto tra le leggi di probabilità a solitone e a solitone robusto

di probabilità $p_{\tau}(d)$ dipende da due parametri:

$$p_{\tau}(d) = \begin{cases} \frac{\Delta}{K} \cdot \frac{1}{d} & d = 1, 2, \dots, K/\Delta - 1\\ \frac{\Delta}{K} \cdot \ln \frac{\Delta}{\delta} & d = K/\Delta\\ 0 & d = K/\Delta + 1, \dots, K \end{cases}$$
(6.81)

La nuova componente $p_{\tau}(d)$ nella legge di probabilità del grado D introduce una "punta" sul valore (relativamente grande) $d = K/\Delta$ (dove Δ è un parametro variabile di progetto) per accertarsi che pacchetti con alto grado siano abbastanza probabili. Il secondo parametro δ regola la p_{DF} : si dimostra che con la distribuzione del solitone robusto (6.81), il numero medio di pacchetti che si devono considerare nella decodifica per avere $p_{DF} < \delta$ è pari a $K' = P \cdot K$. Le due leggi (6.79) e (6.81) sono confrontate in Fig. 6.42.

Esempio 6.34

Imagine we need to save a 100 MB-file segmented into K = 100,000 blocks (packets) with length $\ell = 1024 \times 8$ bits each.

6.6.4 Codici "Raptor"

Abbiamo raggiunto una complessità ragionevole $K \ln(K)$ per un codice a fontana basato su di una trasformazione lineare aleatoria ottimizzando la legge di probabilità del grado dei pacchetti codificati. Il fattore $\ln(K)$ risulta infatti dal grado medio derivante dalla legge a solitone robusto, la quale garantisce che non vi siano pacchetti isolati e che la decodifica abbia successo con grande probabilità $1 - p_{DF}$.

La complessità può essere ancora ridotta seguendo l'impostazione di Shokrollahi che rinuncia ad ottimizzare la trasformata di Luby, usando invece una trasformazione con grado medio pari circa a 3, e quindi complessità 3K molto bassa, accettando un numero

244 CODIFICA DI VANALE CODIFICA YI CANALE CODIFICA DI CANALE



controllato di mancate decodifiche dopo i consueti K' pacchetti codificati. Ne segue che la procedura di decodifica genera essa stessa un certo numero (controllato) di *cancellazioni* end-to-end a *link layer* dopo il decoding a fontana (chiamiamole *cancellazioni esterne*) - niente a che vedere con le cancellazioni introdotte a *physical layer* dalla trasmissione dei pacchetti (che possiamo chiamare cancellazioni *interne*). L'ulteriore idea per ovviare alle mancate decodifiche è quella di concatenare al codice a fontana un ulteriore codice *esterno* operante su interi pacchetti invece che singoli bit, con lo scopo di abbattere le cancellazioni *esterne* provocate dalla fontana! Shokrollahi ha proposto come codice esterno un codice LDPC irregolare nel quale, come del resto nella fontana stessa, i nodi di bit e nodi di check operano su *interi pacchetti* invece che su singoli bit, e ha chiamato questa struttura "Rapid Tornado (RapTor)" code.

Il progetto e l'ottimizzaizone dei codici Raptor è molto complicata, ma il risultato è molto efficace, tant'è che sono stati standardizzati nelle reti cellulari 3G e 4G per multicast di contenuti multimediali. Un esempio delle prestazioni del cosiddetto codice Raptor R10 è riportato in Fig. 6.43 che illustra la probabilità di mancata decodifica p_{DF} in funzione dell'overhead $(1 - p_{CE})(K' - K)/K$ espresso come percentuale, per K = 1000. Le prestazioni sono vicinissime a quelle di un codice teorico lineare casuale anche con p_{CE} alta.

CAPITOLO 7

ANYWHERE, ANYTIME - ELEMENTI DI PROPAGAZIONE RADIO



"There is no distance on Earth that radio communications cannot conquer."

-Guglielmo Marconi, 1901

I tempi pionieristici di Marconi per i radiocollegamenti ormai appartengono a più di un secolo fa. Ma ancora oggi realizzare un collegamento digitale affidabile e ad alta capacità

Comunicazioni Digitali, I Edizione. di M. Luise Copyright © 2023 University of Pisa



Figura 7.1 Trasmissione satellitare con antenna parabolica direttiva

(Gbit/s...) richiede di conoscere approfonditamente le caratteristiche fondamentali della propagazione radio - il tema di questo capitolo.

7.1 Il canale wireless per comunicazioni digitali

Il semplice canale AWGN che abbiamo molto spesso considerato finora è tipico delle comunicazioni con un satellite geostazionario¹ tramite un'antenna direttiva puntata verso il satellite stesso - il caso per antonomasia è la televisione satellitare ricevuta direttamente dall'utente domestico come in Fig. 7.1, per la quale quale il modello del segnale all'antenna è relativamente semplice:

$$x(t) = a \exp(j\phi) \cdot s(t-\tau) + w(t) \tag{7.1}$$

dove il segnale s(t) (equivalente in banda base) è a potenza unitaria, $E\{|s(t)|^2\} = 1$.

In quest'espressione, $a = \sqrt{2P_R}$ è l'*ampiezza* del segnale ricevuto, così ché la potenza (radio) ricevuta è pari a P_R , ϕ è la *fase* della portante in uscita dal demodulatore I/Q di Fig. 2.18, τ è il *ritardo di gruppo*, cioè il ritardo accumulato dall'inviluppo complesso del segnale trasmesso durante la propagazione, e w(t) è il consueto termine di AWGN complesso con componenti I/Q indipendenti, entrambe con densità spettrale di potenza N_0 . I parametri a, ϕ, τ sono in generale ignoti al ricevitore, e devono essere stimati e/o compensati prima di procedere alla demodulazione dati, in modo che la costellazione digitale che deve essere demodulata sia perfettamente ripristinata in ampiezza/fase e che il segnale in banda base sia campionato agli istanti di clock ottimali. Queste funzioni vengono svolte da sottosistemi (algoritmi) dedicati e incorporati nel ricevitore e sono evidentemente crucial per il buon funzionamento del modem in un collegamento digitale.

Viceversa, in un collegamento radio a terra, il modello del segnale è in generale più complicato perchè le antenne che vengono utilizzate, spesso su terminali mobili, sono *omnidirezionali* cioè ricevono/trasmettono da/verso una qualunque direzione sul piano orizzontale proprio per la necessità di mantenere il collegamento anche in movimento (maggiori dettagli su questo punto nella sezione 10.4.1). La ricezione diventa più complicata e necessita di un modello più accurato del semplice AWGN (7.1).

Nelle sezioni successive introdurremo un modello semplice per descrivere il canale radio in un collegamento wireless a terra che ci sarà utile per descrivere la (stra)grande maggioranza dei casi pratica di comunicazione digitale, senza però dilungarci eccessivamente

¹cioè visto dalla superficie terrestre come *fisso* (stazionario) in un punto del cielo, vedi Fig. 10.13



Figura 7.2 Descrizione del fenomeno dei cammini multipli (*multipath*) radio

sull'analisi generale dei canali variabili nel tempo e nella frequenza che è molto complicata e raramente si rivela davvero necessaria.

7.1.1 Il canale radio a cammini multipli (multipath)

Il modello per il canale AWGN (7.1) descrive la propagazione in *visibilità diretta* tra trasmettitore e ricevitore e prevede che il segnale ricevuto, a parte la componente di rumore, sia una versione scalata in ampiezza (a seconda dell'attenuazione dovuta alla propagazione radio) e traslata nel tempo del segnale originale, con in più uno sfasamento incognito dell'oscillazione portante dovuto alla propagazione e alla conversione in banda base da parte del demodulatore I/Q. La componente di segnale è *unica* perchè l'antenna ricevente a terra è direttiva, cioè raccoglie potenza radio sollo lungo la direzione principale di "puntamento" - maggiori dettagli sulla direttività delle antenne si trovano nel paragrafo 10.4.1.

Per collegamenti a terra abbiamo invece antenne *omnidirezionali* di trasmissione/ricezione che rispettivamente inviano/raccolgono il segnale radio verso/da qualunque direzione sul piano orizzontale, per essere comunque in grado di comunicare reciprocamente in condizioni di mobilità. Questa situazione si può analizzare attraverso la semplice teoria *geometrica* della propagazione radio, secondo la quale l'onda viaggia lungo una serie di *raggi*, gli stessi dell'ottica geometrica della fisica elementare; questa descrizione è valida fin quando gli oggetti contenuti nello scenario nel quale l'onda si propaga hanno dimensioni fisiche molto maggiori della lunghezza d'onda λ_0 . Per un'onda radio con frequenza $f_0 \ge 1$ GHz sappiamo che $\lambda_0 = c/f_0 \le 30$ cm, cosicché l'ipotesi di validità dell'ottica geometrica per descrivere la propagazione, ad esempio, in ambiente urbano, è ben verificata.

In uno scenario di propagazione che potrebbe essere considerato come tipico di una rete radio cellulare, l'ambiente circostante le antenne comprende edifici, colline, alberi, che agiscono sui "raggi" descriventi la propagazione come *riflettori* delle onde radio - non perfetti, ma comunque in grado di generare onde riflesse che si aggiungono a quelle direttamente generate dal trasmettitore. Tale situazione è rappresentata in Fig. 7.2 dove abbiamo delineato i vari *cammini (path)* di propagazione dell'onda radio dal trasmettitore al ricevitore, incluse le varie riflessioni.

La conseguenza immediata di questa descrizione della propagazione radio è che il segnale

Ω



Figura 7.3 Esempio di scenario virtuale con ray-tracing elettromagnetico (da www.remcom.com)

ricevuto non presenta una *sola* componente, ovvero una sola replica di s(t) come nella (7.1). Al contrario, s(t) "viaggia" secondo un certo numero N_p di diversi *cammini* corrispondenti ai diversi raggi diretti o riflessi della propagazione geometrica. Questi raggi, (anche se) provenienti da direzioni le più varie vengono comunque raccolti dall'antenna ricevente (omnidirezionale) e generano il seguente segnale risultante dalla *propagazione per cammini multipli* (multipath propagation):

$$e(t) = \sum_{i=1}^{N_p} a_i \exp(j\phi_i) \cdot s(t - \tau_i) = \sum_{i=1}^{N_p} h_i \cdot s(t - \tau_i)$$
(7.2)

dove $h_i \stackrel{\triangle}{=} a_i e^{j\phi_i}$ - una generalizzazione di (7.1), a parte il termine di rumore w(t) che qui non consideriamo (ancora).

Questo modello, benché dichiaratamente semplificato, è in grado di descrivere il principale effetto della propagazione in un ambiente reale: i cammini multipli. Il segnale raccolto dall'antenna RX x(t) è composto da N_p "copie" del segnale trasmesso, dove N_p è appunto il numero dei modi di propagazione (cammini) che possono essere identificati (in gergo tecnico, *risolti*). Ogni cammino è caratterizzato da tre parametri: l'*ampiezza* $a_i > 0$ con la quale il segnale viene ricevuto sul cammino #i, lo sfasamento ϕ_i della portante radio lungo lo stesso cammino, e infine il ritardo τ_i della componente in banda base del segnale (il cosiddetto *ritardo di gruppo*, cioè dell'inviluppo). Questi parametri possono essere fisicamente misurati da strumentazione adeguata o valutati attraverso programmi di simulazione del campo elettromagnetico applicati a un modello cartografico digitale, che ricostruiscono e "seguono" i vari raggi mentre si propagano nell'ambiente virtuale, come rappresentato in Fig. **??** (algoritmi di *ray-tracing*, abbastanza simili a quelli usati nella grafica computerizzata per i giochi).

Che effetto hanno i cammini multipli sulla ricezione del segnale digitale? Continuando a trascurare la componente di rumore w(t), immaginiamo di riuscire, con il nostro demodulatore digitale, di stimare esattamente i parametri (ampiezza,fase,ritardo) della replica ricevuta dal cammino #1 e di riuscire a compensarli esattamente, "agganciando" (sincronizzando) in pratica il ricevitore alla replica proveniente dal cammino #1, intesa come nostro riferimento. Questa condizione significa in pratica che nell'espressione del segnale ricevuto

(7.2) possiamo supporre

$$a_1 = 1$$
 , $\phi_1 = 0$, $\tau_1 = 0$ (7.3)

e quindi

$$x(t) = s(t) + \sum_{i=2}^{N_p} h_i \cdot s(t - \tau_i) = s(t) + \sum_{i=2}^{N_p} a_i e^{j\phi_i} \cdot s(t - \tau_i)s(t) + x_{MP}(t)$$
(7.4)

dove normalmente si intende $\tau_2 < \tau_3 < ... < \tau_{N_p}$. In questa versione del multipath, di fatto i parametri di ogni raggio (a_i, ϕ_i, τ_i) devono intendersi come *relativi* al primo. Pensando di utilizzare un ricevitore digitale convenzionale (ad esempio a filtro adattato) secondo il formato di s(t), notiamo la presenza del termine di (auto-)interferenza $x_{MP}(t)$ causato dai cammini multipli e che in generale disturberà la ricezione di s(t) attraverso un termine di Inter-Symbol Interference (ISI). Dobbiamo attenderci una degradazione della BER dovuta a questo termine, e dobbiamo anche valutarne l'effettiva rilevanza.

Esempio 7.35

Procediamo con la strategia dei "toy example" e immaginiamo di effettuare una comunicazione digitale con formato NRZ (impulso di simbolo rettangolare a durata piena T_s) su di un canale con multipath minimalista, avente $N_p = 2$. Per semplicità, immaginiamo di avere a disposizione il ricevitore "intelligente" che aggancia il primo raggio (cammino), e che i parametri del secondo cammino siano $a_2 = 1/2$, $\phi_2 = \pi$, $\tau_2 = T_s/2$ dove T_s è l'intervallo di simbolo del segnale digitale. Il segnale trasmesso è il consueto $s(t) = \sum_k s[k]p(t - kT_s)$ dove p(t) è l'impulso NRZ di Fig. 7.4 (a) e s[k] èil k-esimo simbolo trasmesso, appartenente a una certa costellazione I/Q.

Secondo il nostro modello, il segnale ricevuto dal canale multipath è

$$x(t) = s(t) - \frac{1}{2}s\left(t - \frac{T_s}{2}\right) = \sum_k s[k]p(t - kT_s) - \frac{1}{2}\sum_k s[k]p\left(t - kT_s - \frac{T_s}{2}\right)$$
$$= \sum_k s[k]q(t - kT_s)$$
(7.5)

dove q(t) è l'impulso ricevuto attraverso i cammini multipli:

$$q(t) = p(t) - \frac{1}{2}p\left(t - \frac{T_s}{2}\right)$$

che è raffigurato nella Fig. 7.4 (b). Si nota benissimo l'effetto dei cammini multipli: l'impulso q(t) è una versione (fortemente) distorta di p(t); in particolare, la durata dell'impulso è adesso pari a $3T_s/2$, cosicchè il simbolo k-esimo causerà (forte) ISI al (k + 1)-esimo. Altrettanto chiaro è che il grado di distorsione dipende in modo cruciale dal valore dei parametri dei cammini multipli. Che succede ad esempio se $\tau_2 \ll T_s$?

La relazione (7.2) che definisce il canale a cammini multipli è quella di un sistema lineare stazionario (SLS) la cui risposta impulsiva è evidentemente

$$h_C(t) = \sum_{i=1}^{N_p} a_i \exp(j\phi_i) \cdot \delta(t - \tau_i)$$
(7.6)

50 ANYWHERE, ANYTIME - ELEMENTI DI PROPAGAZIONE RADIO



Figura 7.4 Impulso all'ingresso (a) e all'uscita (b) del canale con cammini multipli



Figura 7.5 Risposta impulsiva (a) e Profilo Potenza-Ritardo (b) di un canale multipath a 5 cammini

nella quale ogni funzione impulsiva "marca" la presenza di una "eco" di segnale. Una rappresentazione visiva (seppur incompleta) di $h_C(t)$ è data in Fig. 7.5 (a), dove possiamo facilmente apprezzare i ritardi degli echi, le loro ampiezze (indicate impropriamente come diverse ampiezze delle frecce che rappresentano le funzioni δ) e le fasi.

La maniera più comune per caratterizzare il multipath di un canale radio è il *profilo potenza-ritardo* (power-delay profile) mostrato nella Fig. 7.5 (b), che indica l'andamento (profilo) delle potenze dei vari raggi (cammini) al variare del ritardo degli stessi. L'altezza di ogni barra rappresenta la potenza di segnale che il ricevitore raccoglie dal raggio con il dato ritardo, ed è normalmente espressa in dB rispetto a una qualche scala (spessissimo rispetto al massimo del profilo stesso). Non é indicata alcuna informazione sulle fasi dei vari cammini, per un motivo che vedremo in seguito.

Il modello della riposta impulsiva del canale multipath (7.6) non è completamente realistico, perché prevede la presenza di un numero finiti di echi molto ben identificabili e isolabili (risolvibili). Nella realtà, non sempre si riesce ad effettuare una misurazione degli echi così accurata. Invece di avere pochi echi "concentrati" come nel modello di Fig. 7.5 (a), abbiamo invece un certo numero di echi che si accavallano l'uno sull'altro creando una risposta impulsiva con un spettro potenza-ritardo *continuo* rispetto a τ , e dando quindi luogo a una riposta impulsiva del canale *continua* $h_C(t)$ come quella di un qualunque SLS fisico. La caratterizzazione del multipath ad echi discreti (7.2) "degenera" nel consueto

250



Figura 7.6 Misura del profilo potenza-ritardo di canali multipath. (a) Ambiente urbano, (b) In interni.

integrale di convoluzione tra s(t) e $h_C(t)$:

$$x(t) = \sum_{i=1}^{N_p} h_i \cdot s(t - \tau_i) \rightarrow \int h_C(\tau) s(t - \tau) d\tau$$

Saremo più precisi su questo punto nella sez. 7.2; per fare un paragone sonoro, il modello (7.2) corrisponde all'eco che ascoltiamo in alta montagna quando gridiamo per chiamare i nostri compagni di passeggiata che si sono attardati indietro: possiamo ascoltare echi isolati e ben identificabili riflessi dai monti vicini della nostra voce. La riposta impulsiva continua invece ricorda il *riverbero* indistinto dei suoni che si ha in una grande cattedrale o in un palazzetto dell sport, derivante dalle moltissime riflessioni prodotte dei muri, dal soffitto e dal pavimento degli edifici.

Riportiamo in Fig. 7.6 (a) un esempio di profilo potenza-ritardo *misurato* in un ambiente di propagazione all'aperto (outdoor) urbano con pochi picchi derivanti dai pochi echi isolati come nel modello idealizzato della Fig. 7.5. La misurazione di Fig. 7.6 (b) è invece relativa ad un ambiente interno (indoor) con molti echi ravvicinati, e ricorda di più la situazione del riverbero sonoro con la riposta impulsiva continua nel tempo.

7.1.2 Canali radio selettivi nella frequenza e/o nel tempo

Nell'esempio 35, abbiamo visto che il multipath è potenzialmente dannoso per la qualità dal collegamento digitale, ma che la rilevanza del "danno" può dipendere in modo sostanziale dei parametri del canale cioè da "quanto" multipath è presente. Supponiamo che la velocità di simbolo dell'esempio sia $R_s = 1$ kbaud, e quindi $T_s = 1$ ms. In una rete cellulare in ambiente urbano, il valore tipico dei ritardi dei cammini è dell'ordine di 1 μ s (corrispondente a una diversità nella lunghezza dei percorsi da un cammino all'altro di $\Delta d = c\tau = 300$ m), supponiamo dunque $\tau_2 = 1\mu$ s. Ciò significa che (come anticipato nell'esempio...) $\tau_2 \ll T_s$ e l'impulso q(t) in Fig. 7.4, benché mutato di ampiezza e fase, *non* è sostanzialmente distorto e i cammini multipli non sono stati così dannosi come ci si aspettava - tutto dipende dai parametri del *canale* e del *segnale*.

Generalizzando, se *tutti* i ritardi nella (7.2) sono $\ll T_s$, allora gli echi del canale vengono ricevuti perfettamente "sovrapposti" sulla scala temporale dell'intgervallo di simbolo T_s , e i vari cammini collassano in uno solo con ampiezza e fase date dalla somma (vettoriale)

252 ANYWHERE, ANYTIME - ELEMENTI DI PROPAGAZIONE RADIO

delle ampiezze e fasi dei vari cammini ma senza distorsione dell'impulso di simbolo:

$$x(t) \cong s(t) \sum_{i=1}^{N_p} a_i \exp(j\phi_i) = A \exp(j\Phi) \cdot s(t)$$
(7.7)

dove

$$A \exp(j\Phi) \stackrel{\triangle}{=} \sum_{i=1}^{N_p} a_i \exp(j\phi_i) \tag{7.8}$$

è la consueta costante complessa che descrive le modifiche in ampiezza e fase introdotte dal canale. Ma il canale non introduce distorsione e non vi è alcun termine di ISI - un ricevitore a filtro adattato per canale Gaussiano (con l'implicita compensazione preventiva (7.3) di ampiezza, fase e ritardo) va benissimo!

Se invece $\tau_i \approx T_s$ per qualche *i*, dobbiamo analizzare ulteriormente l'influenza dei cammini multipli, ripartendo dalla relazione (7.6) che definisce la risposta impulsiva del canale come SLS, trovandone la *risposta in frequenza*:

$$H_C(f) \stackrel{\triangle}{=} \mathcal{F}[h_C(t)] = \sum_{i=1}^{N_p} a_i e^{j\phi_i} \exp\{-j2\pi f\tau_i\}$$
(7.9)

L'espressione generale che abbiamo trovato è un po' criptica - possiamo però semplificarla per il solito caso $N_p = 2$ e con la normalizzazione (7.3). In questo caso, otteniamo il cosiddetto modello di canale a *due raggi*,

$$x(t) = s(t) + a_2 \exp(j\phi_2)s(t - \tau_2)$$

usato esattamente in questa forma per valutare le prestazioni dei ponti radio a microonde con antenna parabolica (tipicamente utilizzati per collegare la BTS di una rete cellulare e la più vicina centrale interconnessa ad Internet). Usando la notazione standard utilizzato in questo contesto possiamo scrivere

$$x(t) = s(t) + b \exp(j\phi)s(t-\tau)$$
(7.10)

cioè $\tau = \tau_2, \phi = \phi$ e $b = a_2 \ge 0$. La risposta in frequenza del canale a due raggi è

$$H_2(f) \stackrel{\triangle}{=} X(f)/S(f) = 1 + a_2 \exp\{-j2\pi f\tau_2 + j\phi_2\} = 1 - b \exp\{-j2\pi (f - f_N)\tau\}$$
(7.11)

dove abbiamo introdotto un nuovo parametro, la cosiddetta frequenza di notch

$$f_N \stackrel{\triangle}{=} \frac{1}{2\tau} + \frac{\phi_2}{2\pi\tau} \tag{7.12}$$

in modo che appaia il segno "meno" tra i due termini della risposta in frequenza. Dalla (7.11) ricaviamo la risposta in *ampiezza* del canale:

$$|H_2(f)| = \sqrt{1 + b^2 - 2b\cos[2\pi(f - f_N)\tau]}$$
(7.13)

che rappresentiamo in Fig. 7.7 per alcuni valori di b. La caratteristica principale di $|H_2(f)|$ è la *periodicità* rispetto alla frequenza con un periodo $1/\tau$. Nell'ambito di questo periodo, l'andamento è oscillatorio tra |1 + b| e |1 - b|, e il minimo viene raggiunto proprio per $f = f_N$. Quando b = 1, cioè l'ampiezza del raggio secondario ("riflesso") ritardato di



Figura 7.7 Risposta in ampiezza del canale a due raggi con $f_N = 1/(2\tau)$.

 τ è uguale all'ampiezza del raggio principale ("diretto"), allora il minimo alla frequenza di notch e proprio pari a 0 creando una "tacca" (notch) nell'andamento della risposta in ampiezza.

L'andamento di $|H_2(f)|$ non è confortante. Qualunque segnale a "larga banda" vedrà le proprie componenti spettrali modificate dal canale a due raggi in modo *selettivo*, visto che $X(f) = S(f) \cdot H_C(f)$. Che cosa significa però l'accezione "larga banda"? Osservando la Fig. 7.8 (a), ci accorgiamo che quando la banda B_s del segnale s(t) è *molto più piccola* della periodicità frequenziale $1/\tau$ della risposta del canale, allora lo spettro S(f) non viene trattato selettivamente, perché all'interno di tale banda l'andamento della risposta del canale è sostanzialmente *piatto*. Se dunque $B_s \ll 1/\tau$, ovvero, per il segnale digitale con velocità di simbolo $1/T_s$ (e quindi banda $B \approx 1/T_s$), $\tau \ll T_s$, il segnale *non* viene distorto perché la risposta in frequenza del canale è costante (piatta). Il segnale subisce solo una modifica di ampiezza/fase:

$$X(f) = H_C(0)S(f) \implies x(t) = H_C(0)s(t)$$

cioè la stessa condizione cha abbiamo già ricavato ragionando nel dominio del tempo nella (7.7), ove $A \exp(j\Phi) = H_C(f_0)$. Se invece la condizione $B_s \ll 1/\tau$ nonè verificata, il segnale è a "banda larga" (ecco qual è l'accezione giusta) e la risposta del canale, non essendo piatta, tratta in modo diverso, cioè *selettivo*, le varie componenti frequenziali dello spettro, introducendo distorsione.

La condizione di *selettività in frequenza* viene meglio caratterizzata introducendo il concetto di *banda di coerenza* B_{co} del canale con cammini multipli, che definiamo come l'ampiezza del massimo intervallo frequenziale sul quale la riposta in ampiezza del canale può considerarsi sostanzialmente costante (piatta). Nel nostro esempio del canale a due raggi la si può valutare facilmente come una (adeguatamente) piccola frazione della periodicità frequenziale della risposta del canale $1/\tau$:

$$B_{co} \stackrel{\triangle}{=} \frac{1}{10 \div 100} \frac{1}{\tau} \tag{7.14}$$



Figura 7.8 Condizione di canale piatto (a) e selettivo in frequenza (b)

Il coefficiente $1/10 \div 1/100$ è empirico e dipende dall'accuratezza con la quale si intende considerare la riposta piatta.

Con questa definizione, se il segnale radio s(t) è tale per cui $B_s \approx 1/T_s \leq B_{co}$, sappiamo che tutte le componenti dello spettro del segnale vengono "trattate" nella stessa maniera dal canale, non vi sarà distorsione di segnale, e il canale è *piatto in frequenza*. Se al contrario $B_s > B_{co}$, abbiamo una situazione simile a quella della Fig. 7.8 (b), il comportamento del canale è *selettivo in frequenza* e il segnale risulta *distorto* (in misura maggiore o minore, a seconda di quanto la B_s è maggiore della B_{co}). In questo caso, il termine di multipath nella (7.4) può diventare anche molto rilevante, e si devono adottare contromisure per controbatterlo, come vedremo in seguito.

Esempio 7.36

Riprendiamo il canale a due raggi "giocattolo" dell'Esempio 35, e valutiamo la degradazione della BER del ricevitore a filtro adattato che riceve il segnale y(t) = x(t) + w(t)con formato NRZ-BPSK e ampiezza ricevuta unitaria, supponendo che il ricevitore possa sincronizzarsi perfettamente sul raggio diretto.

Tralasciando i dettagli del calcolo (elementare), la variabile di decisione soft all'uscita del filtro adattato all'istante di campionamento ottimo $(k + 1)T_s$ è

r[k] = s[k] - 0.25s[k] - 0.25s[k-1] + n[k] = 0.75s[k] - 0.25s[k-1] + n[k](7.15)

dove $s[k] \in \pm 1$ è il k-esimo simbolo BPSK trasmesso, e n[k] è AWGN a media nulla e con varianza $\sigma^2 = (2E_s/N_0)^{-1}$ (2.111). Il parametro di SNR E_s/N_0 è quello che avremmo su canale AWGN senza il raggio riflesso. Si nota chiaramente l'effetto dei cammini multipli, e cioè i) il termine di ISI derivante dal simbolo precedente, e ii) la diminuzione di ampiezza sul simbolo attuale. La BER si trova con il metodo esaustivo:

$$P(E) = \frac{1}{2}P(E|s[k-1] = -1) + \frac{1}{2}P(E|s[k-1] = +1)$$



Figura 7.9 BER del canale a due raggi dell'Esempio 36

$$P(E|s[k-1] = -1) = P(E|s[k-1] = +1) = \frac{1}{2}Q\left(0.5\sqrt{\frac{2E_s}{N_0}}\right) + \frac{1}{2}Q\left(\sqrt{\frac{2E_s}{N_0}}\right)$$

e quindi
$$P(E) = \frac{1}{2}Q\left(0.5\sqrt{\frac{2E_s}{N_0}}\right) + \frac{1}{2}Q\left(\sqrt{\frac{2E_s}{N_0}}\right)$$
(7.16)

La degradazione della BER è chiara, e viene messa in evidenza nella Fig. 7.9

Quando il multipath è"ricco", cioè vi sono molte più componenti nel segnale ricevuto delle due che abbiamo finora considerato, la risposta in ampiezza del canale multipath è (molto) più "frastagliata" di quella del canale a due raggi (vedremo più avanti un esempio), con molti picchi e molti notch, ma la nozione di banda di coerenza è ovviamente la stessa dal punto di vista concettuale, e può essere calcolata come segue

$$B_{co} \stackrel{\triangle}{=} \frac{1}{10 \div 100} \frac{1}{\tau_{N_n} - \tau_1} \tag{7.17}$$

avendo ordinato come già accennato i ritardi del modello (7.4) in modo che $\tau_1 < \tau_2 < \cdots < \tau_{N_p}$.

7.1.3 Canali Radio con Fading (Evanescenza)

Nella caratterizzazione del multipath, abbiamo fin qui implicitamente considerato una situazione *stazionaria* nel tempo, come risulterebbe dalla comunicazione tra due modem radio installati in punti fissi. Spesso e volentieri però i collegamenti wireless vengono utilizzati proprio per la capacitaà di mantenere vivo un collegamento tra mezzi mobili. La

descrizione del multipath (7.2) diventa allora una "fotografia" che congela ad un certo istante ciò che in realtà evolve quando i modem radio risultano in movimento reciproco.

In questo caso più generale è bene immaginare che il tempo scorra a due diverse velocità: da un lato abbiamo la consueta scala temporale "veloce" dei simboli digitali, con una cadenza (almeno) di centinaia di migliaia al secondo; dall'altro abbiamo una scala temporale molto più lenta (per motivi fisici che esamineremo più avanti) con la quale vediamo mutare lo scenario di propagazione radio a causa del movimento dei terminali radio. Nella nostra "fotografia" il tempo compare ancora, ma è il tempo veloce del segnale digitale, mentre il tempo "lento" dell'ambiente circostante è stato congelato a un certo istante di ricezione, ad esempio, di un certo pacchetto IP. Per ricomprendere la variabilità dello scenario dobbiamo descrivere i cambiamenti al "rallentatore" (rispetto al symbol rate) delle *ampiezze*, *fasi*, e *ritardi* delle componenti dei cammini multipli:

$$x(t) = \sum_{i=1}^{N_p(t)} a_i(t) \exp(j\phi_i(t)) \cdot s(t - \tau_i(t)) = \sum_{i=1}^{N_p(t)} h_i(t) \cdot s(t - \tau_i(t))$$
(7.18)

Dobbiamo adesso caratterizzare adesso la *variabilità* di questi parametri, che sarà ovviamente collegata alla *velocità* con la quale i terminali si muovono, senza confondere le due scale temporali.

Possiamo predire la forma d'onda di, ad esempio, $a_i(t)$? Possiamo sicuramente registrarla attraverso un'opportuna strumentazione che digitalizza il segnale radio e lo salva su di un hard-disk mentre un veicolo opportunamente attrezzato con la relativa strumentazione si muove in città. Oppure possiamo costruire il modello cartografico digitale di una città di riferimento e, attravero i già citati algoritmi di *ray-tracing*, seguire il movimento del veicolo virtuale nelle strade e ricostruire l'espressione del segnale ricevuto punto per punto lungo la traiettoria percorsa all'interno di questo "mega-Google Maps" dal mezzo mobile. Entrambe le opzioni sono piuttosto complicate - un approccio più ragionevole è quello di costruire un *modello statistico* che, sulla base delle molte misurazioni svolte dal veicolo attrezzato e registrate in un grande data-base, estrae gli andamenti caratteristici del segnale ricevuto in determinati ambienti-tipo (cittadino, rurale, in interni, autostrada ecc.) e per una certa velocità di movimento. Tale modello non ricostruisce *esattamente* nessun caso reale, ma si adatta *mediamente* ai molti diversi casi particolari della stessa tipologia che si incontrano in pratica.

Un modello molto utilizzato e molto semplice parte proprio dall'espressione "parametrica" (7.18) e analizza il tipo di variabilità statistica dei parametri del canale (ampiezze, fasi, ritardi, numero di cammini). Dalle misure effettuate, si nota che, su di un tempo dell'ordine di un pacchetto o una trama dati, la variazione nel tempo del numero di cammini e del loro ritardo è trascurabile, e in prima approssimazione può essere ritenuta costante: $N_p(t) \equiv N_p$, $\tau_i(t) \equiv \tau_i$, cosicché il modello del canale diventa

$$x(t) = \sum_{i=1}^{N_p} a(t) \exp(j\phi_i(t)) \cdot s(t-\tau_i) = \sum_{i=1}^{N_p} [h_{i,I}(t) + jh_{i,Q}(t)] \cdot s(t-\tau_i)$$
(7.19)

Viceversa, la variazione dell'ampiezza/fase della portante alla frequenza f_0 associata all'inviluppo complesso del segnale digitale (e di conseguenza anche delle componenti I/Q $h_{i,I}(t)$ e $h_{i,Q}(t)$) è relativamente veloce e deve essere presa in considerazione.

Per capire meglio le proprietà di variabilità *temporale* del canale, (ri)partiamo dal caso (più) semplice del canale piatto in frequenza (7.7)-(7.8) $x(t) = A \exp(j\Phi)s(t)$.

Considerando la variabilità temporale dello scenario abbiamo

$$x(t) = A(t) \exp(j\Phi(t))s(t) = \psi(t) \cdot s(t)$$
(7.20)

dove $\psi(t) = \psi_I(t) + \mathcal{y}\psi_Q(t) = A(t) \exp(\mathcal{y}\Phi(t))$ è un *processo aleatorio* complesso che non distorce il segnale ricevuto (la costellazione dei punti) e non introduce ISI ma, istante per istante, ne varia la fase e l'ampiezza complessiva. L'effetto dello scenario sulla propagazione non è esattamente predicibile, e quindi $\psi(t)$ viene modellato come un processo aleatorio, che chiamiamo processo di *fading* (evanescenza), e del quale dovremo ricavare le proprietà statistiche.

Dalla (7.8) sappiamo che $\psi(t)$ risulta dalla sovrapposizione di molte componenti derivanti dai diversi cammini di propagazione che si combinano con ritardi molto piccoli per creare un unico valore complesso. Con buona approssimazione, le statistiche di questa componente sono Gaussiane per il teorema-limite centrale. Per maggiore precisione, $\psi_I(t) \in \psi_O(t)$ sono processi Gaussiani stazionari mutuamente indipendenti, con caratteristiche che dobbiamo ricavare. In ambiente urbano, le due componenti I/Q risultano a media nulla e hanno la stessa varianza σ_{ψ}^2 , cosicché la potenza media del processo di fading è $P_{\psi} = 2\sigma_{\psi}^2$. Per poter facilmente confortare questa situazione con quella del semplice canale AWGN, usualmente normalizziamo $\psi(t)$ in modo che $\sigma_{\psi}^2 = 1/2$. La Fig. 7.10 rappresenta una realizzazione tipica del fading $\psi(t)$ al variare del tempo, scomposto nelle componenti di ampiezza $A(t) = |\psi(t)|$ e fase $\Phi(t) = \angle \psi(t)$. Si nota che l'ampiezza di quando in quando precipita verso lo 0 (e in tali condizioni tipicamente la fase varia moltissimo da -180 a 180 gradi), per cui il segnale radio "svanisce" (fades out) e c'è il concreto rischio di interruzione del collegamento. Ecco perchè abbiamo chiamato "fading" il processo $\psi(t)$, e perchè le condizioni di propagazione variabile nel tempo che abbiamo appena analizzato vengono indicate brevemente con il nome di canale con "flat fading" o fading piatto (in frequenza).

Dunque in ambiente urbano il processo di fading ha media nulla, e quindi l'ampiezza del fading $A(t) = |\psi(t)| = [\psi_I^2(t) + \psi_Q^2(t)]^{1/2}$ ha istante per istante una densità di probabilità di *Rayleigh*:

$$f_A(a) = \frac{2a}{P_{\psi}} \cdot \exp\left(-\frac{a^2}{P_{\psi}}\right) \quad , \quad a > 0 \tag{7.21}$$

mentre la fase $\Phi(t)$ è *uniformemente distribuita* in $[-\pi, \pi)$ - per questo il fading in ambiente urbano si chiama anche *Rayleigh fading*. In ambiente sub-urbano o rurale, accade spesso che esista un cammino di propagazione *diretto* o in *visibilità ottica* (LOS, Line Of Sight) che viene ricevuto deterministicamente con ampiezza/fase pressoché fissa, accompagnato dalla stessa componente variabile dell'ambiente urbano:

$$x(t) = \psi(t)s(t) = (\psi_0 + \psi'(t))s(t)$$
(7.22)

dove $\psi_0 \stackrel{\triangle}{=} E\{\psi(t)\} \ge 0$ è l'ampiezza della componente LOS, mentre $\psi'(t)$ ha media nulla e varianza σ_{ψ}^2 per componente. La potenza totale del fading è allora $P_{\psi} = |\psi_0|^2 + 2\sigma_{\psi}^2$ (che per normalizzazione normalmente viene fissata a 1), e l'ampiezza segue la densità di probabilità di *Rice*

$$f_A(a) = \frac{2(K_r + 1)}{P_{\psi}} a \exp\left[-\frac{(K_r + 1)a^2 + K_r}{P_{\psi}}\right]$$
(7.23)
 $\cdot I_0\left(2\sqrt{\frac{K_r(K_r + 1)}{P_{\psi}}}a\right) , a > 0 , K_r \stackrel{\triangle}{=} \frac{|\psi_0|^2}{2\sigma_{\psi}^2}$



Figura 7.10 Esempio di variazione del processo di fading in ampiezza e fase

dove il *fattore di Rice* K_r è il rapporto (normalmente espresso in dB) tra la potenza $|\psi_0|^2$ ricevuta attraverso il cammino in visibilità e la potenza $2\sigma_{\psi}^2$ della componente evanescente, e dove $I_0(\cdot)$ è la funzione di Bessel modificata del primo tipo di ordine 0. Per il fading di Rice, la pdf della fase del segnale ricevuto è un po' complicata:

$$f_{\Phi}(\phi) = \frac{e^{-K_r}}{2\pi} \left[1 + \sqrt{2K_r} \cos(\phi) e^{-K_r \cos^2(\phi)} Q\left(-\sqrt{2K_r} \cos(\phi)\right) \right]$$
(7.24)

e vale ovviamente per $\phi \in [-\pi, \pi)$. Il modello di fading piatto alla Rice è usato talvolta anche per il caso di comunicazioni mobili con un satellite, in cui il collegamento in visibilità è (deve essere) sempre presente, e il multipath è debole, con un $K_r \approx 10 dB$.

Esempio 7.37

Riprendiamo il nostro cavallo di battaglia del BPSK con impulsi NRZ, e calcoliamo la BER del ricevitore a filtro adattato quando il segnale si propaga attraverso un canale piatto di Rayleigh con AWGN. Aggiungiamo anche l'ipotesi di fading (molto) lento rispetto alla scala temporale dell'intervallo di simbolo T_s , cosicchè il segnale ricevuto nell'intervallo di simbolo $0 \le t < T_s$ è

$$x(t) = \psi(t)s(t) \approx \psi(0) \cdot s(t) \quad \Rightarrow \quad r(t) = \psi(0) \cdot s(t) + w(t) \tag{7.25}$$

dove w(t) è il consueto AWGN e $\psi(t)$ è normalizzato in modo che $\mathbb{E}\{|\psi(t)|^2\} = 1$. Supponiamo di riuscire a estrarre perfettamente il sincronismo di tempo, e anche di riuscire a compensare altrettanto perfettamente la *fase* del fading $\Phi(0) = \angle \psi(0)$ in modo da poter recuperare in modo coerente il simbolo BPSK. Il modello di segnale allora diventa

$$z(t) = A \cdot s(t) + w(t) \tag{7.26}$$



Figura 7.11 Confronto tra la BER su canale AWGN con e senza fading piatto di Rayleigh

dove $A = |\psi(0)|$ è una *variabile aleatoria di Rayleigh* a potenza unitaria. Dalla (7.26) possiamo immediatamente calcolare la BER del collegamento immaginando che il valore di *A* sia *fisso* e non aleatorio - in altre parole, calcoliamo la BER *condizionata* al valore di A = a, che è quella della rivelazione a filtro adattato:

$$P(E|A=a) = Q\left(\sqrt{\frac{2a^2 E_b}{N_0}}\right) = \frac{1}{\sqrt{2\pi}} \int_{a\sqrt{2E_b/N_0}}^{\infty} \exp(-\beta^2/2)d\beta \qquad (7.27)$$

dove E_b/N_0 si riferisce al caso di canale AWGN senza fading, cioè con $A \equiv 1$, ma si può anche intendere come valore *medio* ottenuto sul canale con fading, visto che $E\{A^2\} = E\{|\psi(t)|^2\} = 1$. La BER finale si ottiene *mediando* la BER condizionata (7.27) sulla statistica di Rayleigh di A:

$$BER = \mathbb{E}_{A} \left\{ Q \left(A \sqrt{\frac{2E_{b}}{N_{0}}} \right) \right\} =$$
$$BER = \int_{0}^{\infty} 2\alpha \exp(-\alpha^{2}) \cdot Q \left(\alpha \sqrt{\frac{2E_{b}}{N_{0}}} \right) d\alpha$$
(7.28)

L'integrale si risolve per parti prendendo la BER condizionata come fattor finito, ottenendo:

$$BER = \frac{1}{2} \left(1 - \sqrt{\frac{E_b/N_0}{1 + E_b/N_0}} \right)$$
(7.29)

La differenza tra la curva di BER sui canali AWGN e con Rayleigh fading è abbastanza impressionante, come vediamo nella Fig. 7.11. Purtroppo i periodi in cui l'ampiezza del segnale svanisce (vedi Fig. 7.10) comportano una BER "locale" molto cattiva che risulta dominante nella media a lungo termine rispetto a quelli in cui invece l'ampiezza è buona,

260 ANYWHERE, ANYTIME - ELEMENTI DI PROPAGAZIONE RADIO



Figura 7.12 Calcolo dello spostamento Doppler Δf

e porta il valore mediato a essere molto peggiore di quello su canale AWGN a parità di E_b/N_0 . Torneremo su quest'argomento in un Esempio successivo.

Dobbiamo ancora risolvere una questione lasciata aperta relativamente al grado di variabilità temporale del fading, cioè cercare di determinare la "velocità" con la quale i processi di fading $\psi_I(t) \in \psi_Q(t)$ variano nel tempo - abbiamo esordito dicendo che presumibilmente la loro scala temporale sarà più lenta del symbol rate e dipendente dalla velocità del movimento dei modem, ma questa congettura deve essere verificata.

Partiamo come sempre da un caso semplice: sappiamo dalla fisica elementare che la fase istantanea di un'onda monocromatica (per noi la portante del segnale radio) generata da un trasmettitore in movimento varia al ricevitore producendo un cambiamento della frequenza ricevuta che va sotto il nome di *effetto Doppler*. In particolare, se il trasmettitore si muove di moto rettilineo a velocità costante v, la frequenza ricevuta è *spostata* rispetto alla trasmessa di una quantità Δf (lo *spostamento Doppler*, appunto) che si ricava facilmente, come mostrato in Fig. 7.12:

$$\Delta f = \frac{v}{c} f_0 \cdot \cos(\gamma) \tag{7.30}$$

dove c è la velocità di propagazione dell'onda e γ è l'angolo tra la direzione di propagazione dell'onda e la traiettoria del mezzo mobile. L'effetto è reciproco, nel senso che accade ugualmente nel caso in cui sia il ricevitore a muoversi anziché il trasmettitore, e vale anche se entrambi sono in movimento: quel che conta è il valore della velocità reciproca, cioè la differenza vettoriale tra le due velocità, proiettata lungo il vettore congiungente, nella 7.30 la quantità $v \cos(\gamma)$. Lo spostamento Doppler varia in generale nel tempo anche se v è costante, perché varia l'angolo γ a seconda della geometria della traiettoria e della posizione di TX e RX, come ci insegna "l'effetto ambulanza" cioè proprio il cambiamento istante per istante della frequenza della sirena di un'autoambulanza mentre si avvicina o si allontana dalla posizione dell'ascoltatore.

Se abbiamo a che fare con un segnale digitale modulato a banda stretta (come nella nostra ipotesi di canale piatto in frequenza), l'effetto Doppler aggiunge uno spostamento della frequenza della portante, riportato in banda base:

$$x(t) = As(t)e^{j(2\pi\Delta f t + \Phi)}$$
(7.31)

dove $A \in \Phi$ sono come sempre l'ampiezza e la fase della portante. Se abbiamo uno scenario di propagazione complesso, ad esempio urbano con molte riflessioni, avremo "molti" raggi che raggiungono il ricevitore provenienti da diversi angoli γ_i . Avremo quindi "molti" diversi spostamenti Doppler $\Delta f_i = (vf_0/c) \cdot cos(\gamma_i)$ derivanti da diversi modi di propagazione generati dagli oggetti circostanti che fungono da "diffusori (scatterer)" dell'onda radio, e il segnale ricevuto sarà

$$x(t) = s(t) \left[A_1 e^{j(2\pi\Delta f_1 t + \Phi_1)} + A_2 e^{j(2\pi\Delta f_2 t + \Phi_2)} + \dots \right] = s(t) \cdot \psi(t)$$
(7.32)

Le varie componenti ricevute dall'ambiente circostante generano diversi spostamenti Doppler che si distribuiscono attorno alla frequenza centrale f_0 (0 in banda base). In generale tali componenti sono molte, e con spostamenti Doppler molto vicini tra loro: con ottima approssimazione, possiamo immaginare che il termine moltiplicativo $\psi(t)$ nella (7.32) risulti composto da un'*infinità* di contributi spettrali distribuiti *con continuità* all'interno dell'intervallo frequenziale $[-f_D, f_D]$ dove f_D è il massimo spostamento Doppler possibile, quello cioè che si ottiene con $\gamma = 0$:

$$f_D = \frac{v}{c} f_0 \tag{7.33}$$

Questo parametro prende il nome di *Doppler spread* (dispersione Doppler) perché appunto indica quanto "disperse" sono le componenti frequenziali causate dal Doppler attorno alla portante. Di fatto, la Doppler spread rappresenta (anche) la banda del processo $\psi(t)$, e quindi è pienamente rappresentativa della velocità di variazione temporale del fading - proprio il parametro che stavamo cercando!

Esempio 7.38

Valutiamo la Doppler spread per tre casi di movimento: un pedone, un'automobile in ambiente urbano, un treno ad alta velocità, e facciamo riferimento a un collegamento di tipo 4G LTE con frequenza portante $f_0 = 2.1$ GHz e un collegamento di tipo 5G NR con $f_0 = 28$ GHz. I risultati sono riassunti nella Tab. 7.1 e mostrano un ampio range di

Mobilità	Velocità (Km/h)	Doppler Spread	Doppler Spread
		LTE (@2.1 GHz) (Hz)	NR (@28 GHz) (Hz)
Pedone	5	9.72	130
Automobile in città	40	77.8	1037
Treno Frecciarossa	300	583	7778

 Tabella 7.1
 Valori della dispersione Doppler per vari casi di mobilità e frequenza portante

variabilità. Ovviamente, il caso più delicato è quello del 5G per il treno ad alta valocità, in cui il Doppler spread è pari a 7.8 kHz. Questo significa che il processo di fading può variare significativamente su periodi pari a $1/f_D=130 \ \mu$ s, piuttosto breve. Un aspetto che dovremo affrontare sistematicamente.

L'effettivo andamento dello spettro del processo di fading (o, per essere più precisi, della densità spettrale di potenza $S_{\psi}(f)$ del processo) all'interno di questa larghezza di

262 ANYWHERE, ANYTIME - ELEMENTI DI PROPAGAZIONE RADIO

banda è difficile da individuare, poiché dipende fortemente dalla particolare distribuzione dei "diffusori" nell'ambiente circostante (edifici, alberi, ecc.) nonché dalla particolare traiettoria del mezzo mobile (velocità, accelerazione, ecc.). In assenza di tali informazioni, una ragionevole supposizione è che $S_{\psi}(f)$ sia sostanzialmente piatta all'interno della larghezza di banda $B = f_D$, con una potenza totale $P_{\psi} = 1$. Alcuni modelli che vengono normalmente citati (di Clarke/Jakes) sono in qualche modo arbitrari perchè dipendono da ipotesi molto (troppo) particolari di movimento e ambiente. Il parametro che davvero conta è lo spread f_D che ci dice con che velocità massima varia il fading, e che ci permette di dire se esso è "lento" o "veloce". In particolare, f_D deve essere confrontato con la velocità di simbolo R_s della trasmissione digitale. In linea di massima, se $f_D \leq R_s/10$, allora possiamo ritenere il fading *lento*, così come abbiamo ipotizzato nell'Esempio 37.

Questa considerazione sulla "velocità" del fading ci porta alla definizione del *tempo di* coerenza del canale T_{co} , che fa il paio (in senso di dualità) con la definizione di *banda* di coerenza (7.14). Dunque T_{co} è il massimo intervallo temporale sul quale possiamo ritenere il processo di fading $\psi(t)$ costante e ,come si intuisce, è legato al Doppler spread f_D . Se questo tempo di coerenza è non inferiore al tempo di simbolo T_s della comunicazione digitale, cioè se $T_s \leq T_{co}$, allora potremo dire che il canale non è selettivo nel tempo, il fading è lento, e la costellazione dati non viene alterata se non con i consueti fattori di ampiezza e fase, come ipotizzato nell'Esempio 37. Se viceversa $T_s > T_{co}$, allora il processo di fading (il canale) varia (più o meno) apprezzabilmente all'interno dell'intervallo di simbolo, e la costellazione sarà alterata da questa variazione temporale - il fading è veloce.

Il tempo di coerenza si definisce facilmente utilizzando proprio il parametro di Doppler spread che descrive la "banda" del processo di fading. Da questo punto di vista,

$$T_{co} \stackrel{ riangle}{=} rac{1}{10 \div 100} rac{1}{f_D} = rac{1}{10 \div 100} rac{c}{v \cdot f_0} \, \, .$$

in analogia con la definzione gemella di B_{co} (7.14). Il fattore $\frac{1}{10\div100}$ è empirico e dipende da quanto "rigorosa" vuol essere la definizione.

Esempio 7.39

Riprendiamo gli scenari dell'Esempio 38 e valutiamo stavolta il *tempo di coerenza* del canale, per capire se una certa comunicazione soffra o meno di selettività nel tempo. Nel calcolo di T_{co} utilizzeremo un fattore 50, intermedio tra 10 e 100. I risultati sono riassunti

Mobilità	Velocità (Km/h)	\mathbf{T}_{co} (μ s)	\mathbf{T}_{co} (μ s)
		LTE (@2.1 GHz)	NR (@28 GHz)
Pedone	5	2060	154
Automobile in città	40	257	19
Treno Frecciarossa	300	34	2.6

 Tabella 7.2
 Valori del tempo di coerenza del canale per vari casi di mobilità e frequenza portante

nella Tab. 7.2 e continuano a mostrare un ampio range di variabilità, con valori però sempre abbondantemente superiori al μ secondo. La conclusione è che non appena la velocià di simbolo è superiore a qualche centinaio di kbaud, il canale non risulta selettivo nel tempo, neanche nel caso più sfavorevole di treno ad alta velocità. Vero è però che su periodi di tempo pari a un pacchetto o una trama dati, il canale varia comunque nel tempo,

e quindi il ricevitore dovrà provvedere ad *inseguire* il canale, cioé realizzare uno stimatore dell'ampizza/fase della costellazione che varia anch'esso nel tempo.

Esempio 7.40

Abbiamo già interpretato il risultato (7.29) della BER sul canale con fading di Rayleigh come la media a lungo termine di un BER a breve termine per un certo stato del canale (7.27), intendendo un concetto di BER "variabile nel tempo". In certi istanti la BER è soddisfacente, in altri no, a seconda del valore che l'ampiezza del canale ha in quel certo periodo. Una volta definito un certo livello di BER "soddisfacente", piuttosto che il valore della BER media a lungo termine, potremmo essere interessati a calcolare la *frazione di tempo* entro la quale la BER è appunto soddisfacente, cioè nella quale BER \leq BER₀, dove BER₀ è la specifica di *Qualita del Servizio* (QoS) che dipende dall'applicazione (ad esempio, BER₀ = 10⁻³ per la telefonia, BER₀ = 10⁻¹⁰ per il video di qualità). Supponendo che BER₀ coincida con il valore che avremmo all'uscita del canale AWGN senz'alcun fading (A = 1), troviamo immediatamente

$$\Pr\{\text{BER} \le \text{BER}_0\} = \Pr\{A \ge 1\} = \int_1^\infty 2\alpha \exp(-\alpha^2) d\alpha = 1/e$$
(7.35)

Questa quantità è la probabilità della disponibilità del collegamento, e il suo complemento 1 - 1/e è invece la probabilità di fuori servizio P_{fs} .

Il valore di P_{fs} appena ricavato, pari circa al 63%, non è per niente soddisfacente. Per ovviare a ciò, è necessario introdurre un *margine di potenza*, cioè aumentare la potenza *media* trasmessa in modo che la probabilità di avere una potenza così piccola da mandare il collegamento in fuori servizio diminuisca adeguatamente. Spenderemo quindi un fattore $10 \log M$ (dB) (il margine) di potenza trasmessa in più in modo che lo E_s/N_0 medio all'uscita

A = 1. In tali condizioni la P_{fs} diventa

$$P_{fs} \stackrel{\triangle}{=} \Pr \left\{ \text{BER} \ge \text{BER}_0 \right\} = \Pr \left\{ A\sqrt{M} \le 1 \right\}$$
$$= \int_0^{1/\sqrt{M}} 2\alpha \exp(-\alpha^2) d\alpha = 1 - \exp(-1/M) \tag{7.36}$$

che è mostrata in Fig. 7.13 in funzione di M. È chiaro che la P_{fs} può essere resa arbitrariamente piccola a patto di introdurre un margine di potenza spesa maggiore.

Chiarita la questione della variabilità temporale del canale piatto in frequenza, torniamo al modello generale di fading selettivo (7.19). Ribadito che, in prima approssimazione, su di una sessione di comunicazione pari a un pacchetto o una trama si possono ritenere costanti il numero di raggi N_p e i valori dei ritardi τ_i , $i = 1, ..., N_p$, resta la variabilità delle componenti $h_{i,I}(t)$ e $h_{i,Q}(t)$ dei vari cammini.

Ragionando in modo simile a quanto fatto finora, quando i ritardi del multipath sono risolvibli, e quindi il fading è selettivo in frequenza, si trova che l'analisi fin qui fatta per l'unica componente di fading può/deve essere ripetuta per *ciascuno* dei cammini di propagazione in (7.19), che risultano anche statisticamente indipendenti. Abbiamo così N_p



Figura 7.13 Probabilità di fuori servizio sul canale con fading piatto di Rayleigh

componenti di fading indipendenti di Rayleigh $h_i(t)$, ciascuna con la medesima descrizione statistica di quella relativa a $\psi(t)$ in (7.20), in particolare accomunate dalla medesima Doppler spread, ma con le diverse potenze relative descritte dal profilo potenza-ritardo di un particolare scenario. La *banda* e il *tempo* di coerenza del canale vengono quindi ricavati come già visto.

Come regola generale, la condizione di selettività in frequenza si verifica quasi *sempre* nelle comunicazioni wireless a larga banda, mentre quella di selettività nel tempo non si verifica quasi *mai*. In casi limite, il canale può risultare *doppiamente selettivo*. Questa condizione si verifica raramente (perché?) e può essere evitata con un progetto accurato del formato del segnale digitale rispetto alle esigenze di mobilità (cioè per le particolari velocità massime di spostamento) del collegamento.

7.2 Modellistica ed emulazione del canale radio

La descrizione del canale multipath sulla base del profilo potenza-ritardo di Fig. 7.6 si riferisce ad un canale variabile nel tempo piuttosto che a un canale statico - cerchiamo di capirne il perché, supponendo di voler sviluppare un *simulatore SW* di un canale radio (ad esempio in linguaggio C++ o MatLab), o anche un *emulatore*, cioè una implementazione (digitale) in tempo reale del modello del canale da applicare a un segnale radio generato da un modem che si intende collaudare in condizioni controllate di riferimento (un certo modello di canale avente un dato profilo) ma verosimili.

Immaginiamo di voler costruire un simulatore, partiremo da un set di campioni I/Q in banda base $s[n] = s(nT_c)$ del segnale modulato (T_c =intervallo di campionamento) per produrre i campioni $x[n] = x(nT_c)$ all'uscita del canale radio prescelto, secondo il modello (7.19). Dobbiamo per prima cosa ricavare la *Doppler spread* f_D (7.33) dai valori della frequenza portante e della velocità del mobile, e poi dobbiamo simulare gli N_p processi $h_i[n] = h_i(nT_c)$ di ogni cammino con la spread data. Per fare ciò, partiamo da N_p sequenze pseudo-casuali Gaussiane mutuamente indipendenti che ogni linguaggio (compreso il C++) mette a disposizione, e che nella simulazione rappresentano N_p processi Gaussiani bianchi $w_i[n]$. Se la varianza dell'*i*-esima sequenza pseudo-casuale $w_i[n]$ così generata è σ_i^2 , la densità spettrale di potenza del processo bianco che viene simulato è $S_i(f) = T_c \cdot \sigma_i^2$. Questi processi vengono poi filtrati con N_p filtri passa-basso di banda f_D che conferiscono



Figura 7.14 Procedura di simulazione/emulazione del canale multipath

al processo il giusto spettro $\bar{S}_{\psi}(f)$, (ad esempio piatto nella banda f_D). Deve poi essere regolata la potenza dell'uscita del filtro in modo che risulti pari al valore P_{h_i} dato dal profilo - l'uscita deve essere scalata del fattore (calcolato per uno spettro del fading piatto)

$$lpha_i \stackrel{ riangle}{=} \sqrt{rac{P_{h_i}}{2 f_D \, T_c \, \sigma_i^2}}$$

Poichè la banda f_D del filtro che fornisce lo spettro dei vari processi di fading $h_i[n]$ è normalmente *molto* più piccola della banda di simulazione $1/(2T_c)$, la generazione e il filtraggio dei processi bianchi $w_i[n]$ possono essere per semplicità effettuati a una frequenza di campionamento molto più piccola di $1/T_c$, e i valori così ottenuti possono essere mantenuti costanti per un certo numero di campioni di segnale s[n], realizzando di fatto un'elaborazione *multi-rate*.

A questo punto dobbiamo realizzare i vari ritardi dei cammini τ_i prescritti dal modello. Per fare questo, vi sono due opzioni: i) approssimiamo i valori dei ritardi del profilo con multipli di T_c , implementando quindi un ritardo $\bar{\tau}_i = T_c \operatorname{int}(\tau_i/T_c)$ (e quindi ritardando semplicemente s[n] di un numero intero di campioni $n_i = \operatorname{int}(\tau_i/T_c)$) - questa opzione è molto efficiente ma introduce una certa imprecisione dovuta alla "quantizzazione" dei ritardi; oppure ii) implementiamo qualche forma di *interpolazione* per ricostruire i valori ritardati tra un campione e l'altro con maggiore precisione. Infine, costruiamo la combinazione lineare di valori di s[n] ritardati e pesati con gli $h_i[n]$ per ottenere x[n] come da (7.19).

La procedura di simulazione/emulazione è riassunta nella Fig. 7.14, nella quale si indica anche esplicitamente l'opportunità di utilizzare due diverse frequenze di simulazione: $f_c = 1/T_c$ come frequenza principale di simulazione del segnale, e $f_{cf} = f_c/P$ come frequenza ridotta del fattore di sottocampionamento P, da scegliersi sulla base del particolare valore della Doppler spread f_D da emulare. La generica "Delay Line" produce i valori ritardati del segnale d'ingresso con o senza interpolazione, come sopra accennato.

Il profilo potenza-ritardo caratterizza i vari tipi di ambiente che si intendono simulare/emulare, come abbiamo già evidenziato nella Fig. 7.6, distinguendo tra ambiente in esterni, con pochi cammini aventi ritardi tra 1 e 10 μ s, e ambiente in interni con molti raggi aventi ritardi tra 10 e 100 ns. La Tab. 7.4 riporta i profili di alcuni modelli di canale standardizzati dal comitato 3gpp www.3gpp.org e che si applicano alle reti cellulari 3G/4G/5G a 2 GHz. Nell'ambiente urbano, si nota che a causa del multipath ricco il cammino dominante non è il più breve (che può subire attenuazioni anche maggiori a seconda

1	ρί	τ _i [μs]	θ _i [rad]
1	0,057 662	1,003 019	4,855 121
2	0,176 809	5,422 091	3,419 109
3	0,407 163	0,518 650	5,864 470
4	0,303 585	2,751 772	2,215 894
5	0,258 782	0,602 895	3,758 058
6	0,061 831	1,016 585	5,430 202
7	0,150 340	0,143 556	3,952 093
8	0,051 534	0,153 832	1,093 586
9	0,185 074	3,324 866	5,775 198
10	0,400 967	1,935 570	0,154 459
11	0,295 723	0,429 948	5,928 383
12	0,350 825	3,228 872	3,053 023
13	0,262 909	0,848 831	0,628 578
14	0,225 894	0,073 883	2,128 544
15	0,170 996	0,203 952	1,099 463
16	0,149 723	0,194 207	3,462 951
17	0,240 140	0,924 450	3,664 773
18	0,116 587	1,381 320	2,833 799
19	0,221 155	0,640 512	3,334 290
20	0,259 730	1,368 671	0.393 889



Figura 7.15 Risposta in ampiezza del canale multipath di Tab. 7.3

del percorso), e in generale vi sono molti cammini significativi. Nel modello pedonale, i ritardi sono piccoli perchè rappresentano i molti cammini con percorsi simili di un pedone nel cosiddetto "canyon urbano", cioè in una strada circondata da edifici alti.

Dunque ciò che serve per costruire il modello statistico variabile nel tempo del canale multipath è il profilo potenza-ritardo e la dispersione Doppler - non serve una descrizione della *fase* dei cammini, come già più volte menzionato. La descrizione della fase serve invece se vogliamo costruire un modello *deterministico* di un canale multipath *statico* tra punti fissi. Lo standard per la televisione digitale terrestre DVB-T descrive un canale radio 'benchmark'', cioè di riferimento, costituito da 20 raggi con le ampiezze/fasi/ritardi riportati in Tab. 7.3. La risposta in ampiezza di questo canale è mostrata nella Fig. 7.15 e risulta molto più complicata di quella del canale a due raggi della Fig. 7.7, a causa delle molte componenti multipath che si combinano in modo costruttivo o distruttivo. In ogni caso, la banda di coerenza si calcola riferendosi alla differenza tra il massimo e minimo ritardo, come già indicato nella (7.17).

Concludiamo con un approccio alternativo per la simulazione/emulazione del canale

MODELLISTICA ED EMULAZIONE DEL CANALE RADIO 267



 Tabella 7.4
 Modelli statistici con fading di Rayleigh del 3gpp (4G/5G)



Figura 7.16 Confronto tra risposta impulsiva del canale multipath con modello parametrico (a) e con modello Tapped-Delay-Line a banda limitata (b)

rispetto a quello parametrico (7.2) che abbiamo appena discusso. Già precedentemente abbiamo osservato che il modello (7.6) della riposta impulsiva $h_C(t)$ del canale con cammini multipli è puramente teorico - le funzioni δ hanno banda infinita e non sono osservabili. Quando facciamo una misura di $h_C(t)$ ciò che effettivamente osserviamo è una versione di (7.6) a banda limitata $h_{BL}(t)$ che peraltro, applicata a s(t) per ottenere x(t) è perfettamente equivalente, visto che anche il segnale radio avrà la stessa banda limitata. La $h_{BL}(t)$ misurata ha l'aspetto dei grafici in Fig. 7.6, dove i picchi estremi delle funzioni δ si "sfocano" in una riposta più dolce, limitata in banda. Con questo, ripetiamo, non si introduce alcuna approssimazione - basta che la banda di misura della $h_{BL}(t)$ sia maggiore della banda del segnale radio. Detto questo, possiamo calcolare la convoluzione tra il segnale trasmesso e la riposta del canale in modo digitale (discreto nel tempo), ottenendo un risultato equivalente proprio perché s ed h sono limitate in banda, e supponendo ovviamente che la condizione di Nyquist sia soddisfatta: :

$$x(t) = s(t) \otimes h_C(t) = s(t) \otimes h_{BL}(t) \to x[n] = \sum_{i=0}^{N_{tap}-1} h_{BL}[i]s[n-i]$$
(7.37)

dove $h_{BL}[i] = h_{BL}(iT_c)$ è la riposta impulsiva a banda limitata discretizzata (campionata).

Questa relazione assomiglia alla (7.2) ma è concettualmente diversa: anzichè considerare N_p ritardi τ_i derivanti dalla modellazione fisica dello scenario, qui si considerano N_{tap} campioni equispaziati di T_c della riposta impulsiva continua $h_{BL}(t)$ e si esegue la convoluzione numerica nella forma di filtro FIR. Per questo il modello si chiama *tapped delay-line*. I ritardi non corrispondono (più) ai ritardi fisici dei raggi, e gli N_{tap} valori degli h_i che devono essere considerati possono essere, a parità di accuratezza, molti di più degli N_p "fisici", specie quando il canale ha ritardi relativamente grandi: la Fig. 7.16 confronta in un caso particolare ma significativo questi due approcci, evidenziando il modello parametrico (a) composto da (soli) 4 raggi, e quindi molto efficiente, confrontato con la $h_{BL}[n]$ limitata in banda da utilizzare nella (7.37) composta invece da molti coefficienti.

Dalla discussione molto approfondita sul canale di propagazione radio e la sua modellistica di questa sezione, il messaggio risultante principale è il seguente: in generale, un segnale digitale avrà sempre a che fare con problemi di selettività in frequenza. Dobbiamo quindi specificamente occuparci di quest'aspetto e del come risolverlo efficacemente - nel prossimo Capitolo.

CAPITOLO 8

UN ARCOBALENO DI DATI - LE TECNOLOGIE MULTIPORTANTE



"Somewhere over the rainbow/Skies are blue/And the dreams that you dare to dream/Really do come true"

—1939, dalla colonna sonora del film "Il Mago di Oz", cantata da Judy Garland

Che cosa sognano i progettisti di comunicazioni wireless? Di trovare una tecnologia

Comunicazioni Digitali, I Edizione. di M. Luise Copyright © 2023 University of Pisa



Figura 8.1 Rappresentazione della selettività in frequenza

"definitiva" per realizzare collegamenti robusti e affidabili ad alto bit rate e larga banda su di una certa frequenza radio. Sufficientemente robusti da resistere alle enormi distorsioni che, come sappiamo, i segnali a larga banda subiscono quando si propagano via radio, al contrario di quelli a banda stretta - ci vorrebbe una tecnologia per la quale il segnale si comporta allo stesso tempo a *banda stretta* e a *banda larga*... La soluzione esiste, e si chiama *modulazione multiportante*: sistemare molti segnali digitali a banda stretta fianco a fianco su molte (sotto)portanti adiacenti che coprono la banda (larga) di trasmissione, in modo che nessuno di questi subisca forte distorsione: un *arcobaleno* di dati di colori diversi che fanno avverare il nostro sogno.

8.1 Le Comunicazioni Multiportante

8.1.1 Cammini Multipli e Distorsione

Se la banda B_x di un segnale digitale x(t) inviato su di un canale radio è pià larga della banda di coerenza B_{co} di quel canale, il segnale ricevuto è distorto a causa della selettivià frequenziale, e tanto più quanto risulterà $B_x > B_{co}$. Il comportamento selettivo del canale è ben rappresentato in Fig. 8.1, nella quale si notano i (molti) "poggi e buche" della riposta in ampiezza $|H_C(f)|$ del canale nell'ambito della banda del segnale. Dagli esempi numerici che abbiamo già visto, sappiamo che la banda di coerenza del canale radio per comunicazioni in ambienti esterni non eccede qualche decina di kHz, e quindi *qualunque* segnale a banda larga è sempre martoriato dalla selettività frequenziale. Anche in ambienti interni (indoor), dove ci aspetteremmo una minore selettività, B_{co} è comunque limitata a qualche centinaio di kHz, e i segnali digitali come Wi-Fi (IEEE 802.11 da g in su) che occupano (almeno) 20 MHz di banda rischiano forti distorsioni.

Dunque la selettività in frequenza è *il* problema da risolvere per garantire comunicazioni radio affidabili - anche se il rapporto segnale-rumore è molto buono, come accade normalmente nei collegamenti Wi-FI, quando abbiamo forte selettività la qualità del collegamento non è sufficiente. La contromisura tradizionale alla questione risale ai primi anni '60 e fu



Figura 8.2 Compensazione della distorsione di canale tramite un *equalizzatore*

introdotta per migliorare le comunicazioni digitali su linea telefonica: dotare il ricevitore di un *equalizzatore*, cioè un (pre-)filtro che, attraverso una ben progettata risposta in frequenza E(f) compensi le distorsioni di canale, come mostrato nella Fig. 8.2. L'equalizzatore deve, appunto, *equalizzare* il più possibile la selettività del canale, in modo che la risposta in frequenza complessiva di canale+equalizzatore $H(f) \cdot E(f)$ risulti il più possibile piatta in ampiezza e lineare in fase sulla banda del segnale. L'equalizzatore deve anche essere in grado di adattarsi alle condizioni di propagazione in generale variabili nel tempo, deve cioè essere *adattativo*. Per far ciò, si realizza in generale un filtro digitale FIR come in Fig. 2.21 nel quale i coefficienti della riposta impulsiva vengono adattati (variati) sulla base di un qualche criterio di ottimalità. Quando la selettività è molto forte, però, l'equalizzatore adattativo a filtro FIR diventa di implementazione problematica.

Esempio 8.41

Consideriamo il collegamento dati a velocità $R_b = 1$ Mbit/s di un aereo senza pilota (drone) verso la propria stazione base. Il formato (robusto) è QPSK con codice LDPC a tasso 1/2 e presenta quindi una velocità di simbolo $R_s = 1$ Mbaud. L'ambiente in cui il drone opera è rurale, e può essere efficacemente modellato da un canale a due raggi con $\tau = 10 \ \mu s = 10T_s$. L'impulso di simbolo trasmesso è lo standard SRFRC (2.97) con rolloff $\beta = 0.35$, simile a quello del DVB-S, e il demodulatore dati nella stazione base viene realizzato con una architettura zero-IF (I/Q) con conversione A/D in banda base come in Fig. 2.34, utilizzando una frequenza di conversione (campionamento) $f_{sa} = 1/T_{sa} = 2R_s$. Con questo valore si ottengono *due campioni per simbolo* e viene rispettata la condizione di Nyquist per il campionamento corretto di un segnale limitato in banda, visto che la banda delle componenti I/Q del segnale QPSK è $B = (1 + \beta)R_s/2 = 0.6125R_s$. In generale, la frequenza di campioni per simbolo - il cosiddetto *fattore di sovracampionamento* riferito alla velocità di simbolo.

Cerchiamo adesso di progettare un equalizzatore che compensi la distorsione introdotta dal multipath. Ripartendo da (7.11), la risposta in frequenza ideale dell'equalizzatore E(f) è

$$E(f) = \frac{1}{H_2(f)} = \frac{1}{1 - b \exp\{-j2\pi(f - f_N)\tau\}}$$
(8.1)

che compensa perfettamente la risposta del canale. Per ragioni storiche, questo equalizzatore è chiamato *zero-forcing* perchè riduce (forza) a 0 la componente di ISI evidenziata della (7.4). In particolare, dovremo progettare un filtro digitale con risposta impulsiva e[n]che implementi l'equalizzatore attraverso DSP. Dalla (8.1), supponendo |b| < 1, abbiamo anche

$$E(f) = \sum_{\ell=0}^{\infty} \left(b \exp\{-j2\pi (f - f_N)\tau\} \right)^{\ell} = \sum_{\ell=0}^{\infty} b^{\ell} \exp\{-j2\pi (f - f_N)\ell\tau\}$$
(8.2)

La risposta $H_2(f)$ del canale a due raggi ha banda infinita, e quindi anche l'equalizzatore (8.2) ha in generale banda infinita. Poichè però il segnale QPSK è limitato in banda, possiamo immaginare di utilizzare un equalizzatore anch'esso limitato in banda, e in particolare nella banda $f_{sa}/2 = R_s$ in modo da facilitarne la successiva implementazione digitale. Questo equalizzatore limitato in banda ha risposta in frequenza

$$E_B(f) = \operatorname{rect}\left(\frac{f}{2R_s}\right) \sum_{\ell=0}^{\infty} b^{\ell} \exp\{-j2\pi(f-f_N)\ell\tau\}$$
(8.3)

ovvero, antitrasformando, ha risposta impulsiva

$$e_B(t) = 2R_s \sum_{\ell=0}^{\infty} (b \exp\{j 2\pi f_N \tau\})^{\ell} \operatorname{sinc}(2R_s(t - \ell\tau))$$
(8.4)

La risposta impulsiva e[n] dell'equalizzatore digitale si ottiene con il metodo dell'*invarianza impulsiva*, cioè imponendo $e[n] = e_B(n/f_{sa})/f_{sa}$:

$$e[n] = \sum_{\ell=0}^{\infty} (b \exp\{j2\pi f_N \tau\})^{\ell} \operatorname{sinc}(n - 2R_s \ell \tau)) =$$
$$= \sum_{\ell=0}^{\infty} \Psi^{\ell} \operatorname{sinc}(n - 20\ell) = \sum_{\ell=0}^{\infty} \Psi^{\ell} \delta[n - 20\ell]$$
(8)

.5

ove $\Psi \triangleq b \exp\{j2\pi f_N \tau\}$ e dove abbiamo utilizzato la proprietà della funzione sinc sinc $(n) = \delta[n]$ (vedi Fig.2.6) e (2.77).

Il nostro equalizzatore risulta a risposta impulsiva infinita (IIR); e[n] è costituito infatti da una serie teoricamente illimitata di valori "isolati" spaziati di 20 campioni (20 è il valore del ritardo del multipath τ misurato in intervalli di campionamento T_{sa}) - possiamo classificarlo come segnale *sparso*.

Per ottenere un filtro FIR, possiamo *troncare* questa riposta. Osserviamo infatti che i valori di e[n] diversi da 0 decadono esponenzialmente col tempo poiché $|\Psi| = |b| < 1$. Se il multipath non è troppo forte (cioè |b| abbastanza più piccolo di 1) possiamo conservare una ventina di valori significativi di e[n] scartando i successivi, ottenendo un equalizzatore FIR con 20*20+1=401 coefficienti - parecchi!

È sì vero che la maggior parte dei 401 coefficienti sono pari a zero, ma è anche vero che questo è un caso molto, *molto* speciale che si è prodotto per il valore particolare del ritardo del secondo cammino ($\tau = 10T_s = 20T_{sa}$), pari a un multiplo esatto dell'intervallo di campionamento. Quando questo non si verifica (e non si verifica quasi mai), allora *tutti* i 401 campioni dell'equalizzatore sono significativi e diversi da 0, e il filtro diventa complicato da progettare e realizzare. Inoltre, per adattarsi alla particolare situazione di multipath che il drone incontra a un certo istante l'equalizzatore deve in generale *adattarsi*, cioè deve essere *addestrato* su di un preambolo di trasmissione per trovare i coefficienti richiede in generale un tempo *molto* lungo e risulta incompatibile con i requisiti di instaurazione veloce del collegamento con il drone (o qualunque alta applicazione in generale).

L'Esempio 41 ci ha insegnato che l'equalizzatore di canale ha in generale bisogno di un numero di coefficienti (taps) dell'ordine di alcune centinaia. Deve inoltre essere *adattativo* - come esempio banale, quando facciamo *zapping* tra i canali TV, l'equalizzatore deve (ri)adattare i propri 400+ coefficienti ogni volta che vogliamo vedere un nuovo programma (che vede in generale una configurazione di cammini multipli diversa perché è trasmesso su di un canale a frequenza diversa), e ciò richiede un tempo intollerabilmente lungo.

Per risolvere la questione, l'idea è quella di cambiare l'approccio per controbattere la selettività in frequenza: invece di usare un formato digitale convenzionale, per poi correggere la distorsione del canale con un equalizzatore nel ricevitore, perché non *prevenire* l'insorgenza della distorsione già nel trasmettitore usando un formato digitale *piú robusto* rispetto alla selettività in frequenza? In particolare, perché non cercare di progettare un segnale digitale che, come suggerisce il prologo di questo Capitolo, risulta allo stesso tempo a banda stretta (per non subire la selettività del canale) e a banda larga (per trasmettere un bit rate elevato)? La riposta a questa domanda è: dobbiamo usare un formato *multiportante* (multicarrier).

Nella modulazione multiportante, suddividiamo il singolo flusso di dati ad alta velocità di segnalazione (e quindi banda larga) $R_s = 1/T_s$ nel "parallelo" di molti sotto-flussi ottenuti per demultiplazione (conversione serie-parallelo) del flusso originale, e quindi modulando con questi molti flussi un ugual numero di *sottoportanti* adiacenti. Se indichiamo con N il numero di questi sotto-flussi, la velocità di segnalazione su ciascuna sottoportante $R_M = 1/T_M$ (dove $_M$ sta per *Multiportante*) sarà *più lenta* di un fattore N rispetto alla velocità originale, $R_M = R_s/N$, e l'intervallo di simbolo su ciascuna sottoportante sarà $T_M = NT_s$, vale a dire N volte *più lungo*. La banda occupata dal generico sotto-flusso sarà quindi pari a R_M , e cioè N volte inferiore alla banda $B_x = 1/T_s$ del segnale digitale originario.

In questo modo, aumentando opportunamente N, possiamo ottenere la condizione in cui ogni sotto-canale ha una banda molto *stretta*, in particolare *più stretta della banda di coerenza del canale* B_{co} , in modo che il sotto-canale "veda" una porzione sostanzialmente *piatta* della risposta in frequenza globale del canale radio $H_C(f)$. Questa situazione è illustrata nella Fig. 8.3 dove gli spettri colorati rappresentano i sotto-flussi modulati sulle varie sottoportanti adiacenti. Ad esempio, nelle comunicazioni cellulari 4G LTE il numero di sottoportanti è N = 2048, e nello standard DVB-T2 per la televisione digitale terrestre possiamo avere fino a N = 32768 sottoportanti!

Quando considereremo un po' più a fondo la questione, scopriremo che l'effetto del canale selettivo su di un segnale multiportante è quello di introdurre soltanto una certa attenuazione e uno sfasamento sulle varie costellazioni dati dei vari sotto-canali, *senza* introdurre interferenza intersimbolica o distorsione. Tali fattori di ampiezza/fase cambiano però, come si vede dalla Fig. 8.3, da sottoportante a sottoportante e devono essere stima-ti/compensati individualmente - una specie di equalizzazione frequenza-per-frequenza del segnale. Saremo più precisi su questi aspetti nella sez. 8.4.

8.2 Dal Generico Formato Multicarrier all'OFDM

8.2.1 II modulatore multiportante

Nella precedente discussione di carattere euristico non abbiamo affrontato alcuni aspetti fondamentali del formato multiportante, come la spaziatura in frequenza delle sottoportanti,





e la maniera per costruire in modo efficiente un formato di segnale così complicato.

La Fig. 8.4 rappresenta lo schema di massima del modulatore multiportante il cui ingresso è rappresentato da un flusso di simboli c[n] appartenenti a una costellazione I/Q convenzionale (tipicamente M-PSK o M-QAM) con velocità di segnalazione R_s. Questo flusso ad alta velocità viene parallelizzato (demultiplato) in N sotto-flussi che definiscono N sotto-canali caratterizzati da una velocità di segnalazione "più lenta" di u fattore N, cioè la velocità di simbolo multiportante $R_M = R_s/N$. Questa parallelizzazione induce anche una segmentazione del flusso d'ingresso in blocchi di N simboli di costellazione, e la durata di ogni blocco viene indicata con il nome di tempo di simbolo multiportante $T_M = NT_s$. Nella Fig. 8.4 si nota che ogni sottoflusso viene poi modulato su di una particolare sottoportante per renderlo successivamente distinguibile da ogni altro. Per procedere con la descrizione del modulatore, in particolare descrivere analiticamente la particolare forma del segnale digitale multiportante, introduciamo una notazione a due indici: con k = 0, 1, ..., N - 1 indichiamo il numero identificativo del generico sottocanale, e quindi della generica frequenza sottoportante $f_k \stackrel{\triangle}{=} k \cdot f_{sc}$, dove f_{sc} è il quanto di spaziatura frequenziale (regolare) tra le sottoportanti. Introduciamo poi la variabile mche identifica tutto il blocco di N simboli di ingresso che occupano l'intervallo di tempo $mT_M \leq t < (m+1)T_M$ (cioè l'*m*-esimo intervallo di simbolo multiportante).

Dallo schema di Fig. 8.4 vediamo che il segnale digitale multicarrier $x_{MC}(t)$ è ottenuto dalla somma delle N componenti $y_k(t)$, dove ogni componente è ottenuta dalla modulazione del k-esimo sotto-flusso in banda base $x_k(t)$ attorno alla k-esima sottoportante avente frequenza $f_k = k \cdot f_{sc}$:

$$x_{MC}(t) = \sum_{k=0}^{N-1} y_k(t) = \sum_{k=0}^{N-1} x_k(t) e^{j2\pi k f_{sc} t}$$
(8.6)


Figura 8.4 Schema generale del modulatore multiportante

D'altro canto, il k-esimo sottoflusso $x_k(t)$ è

$$c_k(t) = \sum_{m=-\infty}^{+\infty} c^{(k)}[m]p(t - mT_M)$$
(8.7)

dove $c^{(k)}[m]$ è il simbolo di costellazione, derivante dal flusso originario c[n], che nell'ambito del blocco *m*-esimo (e cioè durante l'intervallo $mT_M \leq t < (m+1)T_M$) viene modulato sulla sottoportante #k, e dove $p(t - mT_M)$ è un impulso (rettangolare) NRZ che "delimita" la durata dell'*m*-esimo simbolo multiportante, in quanto p(t) per definizione si estende con ampiezza unitaria da t = 0 a $t = T_M$ ed è nullo al di fuori.

La relazione tra i simboli di ingresso c[n] con un clock pari a R_s e i simboli "rimappati" sulle sottoportanti $c^{(k)}[m]$ con un clock $R_M = R_s/N$ è semplice: considerando che le sottoportanti hanno un identificatore $0 \le k \le N - 1$, abbiamo

$$c^{(k)}[m] \stackrel{\triangle}{=} c[mN+k]$$

Questa relazione è rappresentata graficamente nella Fig. 8.5 per N = 8 ed evidenzia la relazione tra i due clock (velocità di simbolo). Detto questo, la relazione finale e completa che descrive il segnale digitale multiportante risulta

$$x_{MC}(t) = \sum_{k=0}^{N-1} \sum_{m=-\infty}^{+\infty} c^{(k)}[m] p(t - mT_M) e^{j2\pi k f_{sc} t}$$
(8.8)



Figura 8.5 Relazione tra gli indici di simbolo $n, k, m \operatorname{con} N = 8$



Figura 8.6 Ricevitore multiportante ottimo per la k-esima sottoportante

8.2.2 Demodulatore Multiportante e OFDM

L'idea fondamentale dietro al formato multicarrier (8.8) dovrebbe a questo punto essere abbastanza chiara. Dobbiamo adesso mostrare come demodulare i dati $c[n] = c[mN + k] = c^{(k)}[m]$, discutendo in particolare un aspetto cruciale: modulare simboli diversi su diverse sottoportanti senza particolari accortezze relativamente alla spaziatura tra queste sottoportanti può causare fenomeni di *inter-carrier interference* (ICI) quando esse risultano troppo "vicine" per poter essere separate.

Supponiamo allora che la sottoportante # k sia la *sola* che viene trasmessa, senz'alcun altra sottoportante attiva che la accompagni. Immaginando di ricevere questo segnale accompagnato da AWGN, il segnale ricevuto sarà

$$r(t) = y_k(t) + w(t) = c^{(k)}[0]e^{j2\pi f_{sc}t} , \ 0 \le t < T_M$$
(8.9)

e la struttura del ricevitore ottimo per rivelare i dati $c^{(k)}[m]$ su questa sottoportante (Fig. 8.6) è ben nota - un ricevitore *a correlazione* (2.109) con la forma d'onda che accompagna il simbolo nella (8.9), ovvero un demodulatore che converte il segnale ricevuto di nuovo in banda base eliminando la sottoportante, seguito da un filtro di ricezione adattato all'impulso del simbolo multiportante p(t), realizzato nella forma di un filtro integratore sul tempo di simbolo *m*-esimo. Nella Fig. 8.6 abbiamo ipotizzato m = 0 per semplicità di notazione, ma l'estensione all'intervallo $mT_M \leq t < (m+1)T_M$ è immediata. La variabile di decisione soft $z^{(k)}[0]$ prodotta dal ricevitore e relativa al simbolo $c_0^{(k)}$ è

$$z^{(k)}[0] = \frac{1}{T_M} \int_{0}^{T_M} y_k(t) e^{-j2\pi k f_{sc} t} dt$$
(8.10)

La lettrice può controllare che in assenza di rumore $z^{(k)}[0] \equiv c^{(k)}[0]$.

Che succede se adesso attiviamo una seconda sottoportante del *pettine* delle N frequenze, ad esempio la # i ? La variabile in uscita al ricevitore di Fig. 8.6 avrà un termine ulteriore rispetto alla $z^{(k)}[0]$ (8.10) - indichiamolo con $I^{(i,k)}[0]$ intendendo che esso rappresenta (per il simbolo multiportante m = 0) l'interferenza generata dalla sottoportante i sull'uscita

del ricevitore per la sottoportante k. Considerando che, analogamente alla (8.9), $y_i(t) = c^{(i)}[0]e^{j2\pi i f_{sc}t}, 0 \le t \le T_M$, otteniamo:

$$I^{(i,k)}[0] = \frac{1}{T_M} \int_{0}^{T_M} c^{(i)}[0] e^{j2\pi i f_{sc}t} \cdot e^{-j2\pi k f_{sc}t} dt = c^{(i)}[0] \frac{e^{j2\pi (i-k)f_{sc}T_M} - 1}{j2\pi (i-k)f_{sc}T_M}$$
(8.11)

In generale, questo termine è diverso da 0, ed è rilevante quando f_{sc} è piccolo rispetto a $1/T_M$ e *i* è vicino a *k* - quando cioè le sottoportanti sono "troppo vicine".

Una condizione ottimale sarebbe il caso in cui $I^{(i,k)}[0] = 0$ indipendentemente dai valori di $i \in k$ ($i \neq k$). Ciò accade quando $f_{sc} >> 1/T_M$ (grande spaziatura tra le sottoportanti) o quando $f_{sc} = q/T_M$, q intero. Qual è la condizione più favorevole? Valori grandi di q portano a grande spaziatura tra le sottoportanti; è intuitivo che aumentare la spaziatura significa aumentare la banda totale occupata dal segnale multiportante (i dettagli successivamente), diminuendo l'efficienza spettrale del collegamento. Il valore "giusto" di q è quindi il *minimo* che garantisca assenza di ICI, ovvero, in termini matematici, *ortogonalità* tra le sottoportanti: $q = 1 \Rightarrow I^{(i,k)}[0] = 0 \forall i \neq k$.

Se q=1, $f_{sc} = 1/T_M$, cioè la spaziatura tra le sottoportanti è esattamente uguale alla velocità di simbolo multiportante $R_M = 1/T_M$ sulle varie sottoportanti, e il formato multiportante viene chiamato OFDM (Orthogonal Frequency-Division Multiplexing):

$$x_{OFDM}(t) = \sum_{k=0}^{N-1} \sum_{m=-\infty}^{+\infty} c^{(k)}[m] p(t - mT_M) e^{j2\pi kt/T_M}$$
(8.12)

Grazie alla condizione di ortogonalità, il ricevitore ottimo per *tutto* il segnale OFDM mostrato nella Fig. 8.7 è una batteria di N ricevitori ottimi come quello di Fig. 8.6, ciascuno "sintonizzato" sulla portante $f_k = k/T_M$, k = 0, 1, ..., N - 1, e totalmente privo di ICI. Il formato OFDM è diventato trasversale a innumerevoli applicazioni (diciamo pure che rappresenta ormai una tecnologia ubiqua): reti cellulari 4G/5G, Wi-FI, televisione e radio digitali DVB/DAB eccetera.

8.2.3 Efficienza spettrale dell' OFDM

Una volta ricavata la condizione di ortogonalità applicando il minimo valore di q, dobbiamo valutare la bontà del segnale OFDM $x_{OFDM}(t)$ dal punto di vista dell'efficienza spettrale - dobbiamo cioè ricavarne la densità spettrale di potenza (dsp) $S_{OFDM}(f)$.

Riformuliamo la (8.6) nel caso particolare di OFDM:

$$x_{OFDM}(t) = \sum_{k=0}^{N-1} y_k(t) = \sum_{k=0}^{N-1} x_k(t) e^{j2\pi kt/T_M}$$
(8.13)

dove $x_k(t)$ è (ancora) come in (8.7). Tenendo conto di quest'ultima, vediamo facilmente che $x_k(t)$ è statisticamente indipendente da $x_i(t)$, $i \neq k$, perchè i due segnali contengono due sottoflussi di simboli *diversi* estratti da c[n]; da questo discende anche l'indipendenza di $y_k(t)$ da $y_i(t)$, che facilita molto il calcolo, visto che la dsp della somma di segnali indipendenti è pari alla somma delle relative dsp:

$$S_{OFDM}(f) = \sum_{k=0}^{N-1} S_{y_k}(f)$$
(8.14)



Figura 8.7 Demodulatore ottimo per OFDM (m = 0)

da cui, con il teorema della modulazione (2.12)

$$S_{OFDM}(f) = \sum_{k=0}^{N-1} S_{y_k}(f) = \sum_{k=0}^{N-1} S_{x_k}\left(f - \frac{k}{T_M}\right)$$
(8.15)

D'altronde, la dsp del segnale dati in banda base $x_k(t)$ è (2.95)

$$S_{x_k}(f) = \frac{C_2}{T_M} \cdot |P(f)|^2 = C_2 T_M \operatorname{sinc}^2(fT_M)$$

dove $C_2 \stackrel{\triangle}{=} E\{|c[n]|^2\}, P(f)$ è la trasformata di Fourier p(t), e dove vediamo che tale spettro è lo stesso per tutti i sottocanali. In conclusione,

$$S_{OFDM}(f) = C_2 T_M \sum_{k=0}^{N-1} \operatorname{sinc}^2\left(\left(f - \frac{k}{T_M}\right) T_M\right)$$
(8.16)

la quale conferma l'intuizione originaria della Fig. 8.3 secondo la quale lo spettro complessivo di OFDM è la sovrapposizione di N spettri identici derivanti dagli N sotto-flussi, ciascuno centrato sulla sua propria frequenza sottoportante.

La dsp (8.16) è rappresentatata nella Fig. 8.8. Lo spettro *non* è limitato in banda e, a parità di velocità di simbolo originaria $1/T_s$, decade con la frequenza più velocemente all'aumentare del numero di sottoportanti. La miglior tecnologia preesistente riguardo l'efficienza spettrale è il formato limitato in banda Square-Root Frequency Raised-Cosine (SRFRC), la cui banda (in banda radio) per una velocità di segnalazione R_s è pari a $B_{SRFRC} = (1 + \beta)R_s$ ($0 \le \beta \le 1$ è il fattore di roll-off). Lo spettro di OFDM non



Figura 8.8 Densità spettrale di potenza del segnale OFDM

è rigorosamente limitato in banda alternativa, come la banda a -3 dB o -20 dB. Usando una qualsiasi di queste definizioni otteniamo comunque il risultato $B_{OFDM} \simeq 1/T_s$, equivalente a quella dello spettro SRFRC con $\beta = 0$, quindi pari alla cosiddetta *banda di* Nyquist, la massima efficienza spettrale (minima banda) che si può adottare senza causare ISI - un altro fattore positivo di OFDM¹,

8.3 Implementazione DSP del Modem OFDM

8.3.1 Modulatore OFDM Digitale

Come già accennato, OFDM è diventata una tecnologia *pervasiva*, declinata in molte varianti, in particolare con un numero di sottoportanti variabile da N = 64 per IEEE 820.11 (Wi-Fi) a N = 32k (DVB-T2). È chiaro che realizzare un trasmettitore con 32k modulatori su sottoportante e un ricevitore con 32k demdulatori/filtri costituisce una sfida tecnologica non indifferente, che viene vinta realizzando questi apparati attraverso componenti DSP.

Consideriamo il segnale OFDM nell'intervallo $0 \le t < T_M$, cioè per il solo simbolo OFDM m = 0. L'espressione generale (8.12) diventa (ricordando che $p(t) \equiv 1$ in questo intervallo temporale)

$$x_{OFDM}(t) = \sum_{k=0}^{N-1} c^{(k)}[0]p(t) e^{j2\pi kt/T_M} = \sum_{k=0}^{N-1} c^{(k)}[0] e^{j2\pi kt/T_M} , \quad 0 \le t < T_M$$
(8.17)

In un'implementazione DSP, dobbiamo generare i campioni di questo segnale con una certa

¹È chiaro che il valore della banda occupata cambia leggermente utilizzando definizioni di banda empirica differenti; tutte quante però forniscono un valore leggermente maggiore di $1/T_s$, come se, nel paragone con lo spettro SRFRC, ammettessimo un valore molto piccolo di β .



Figura 8.9 Spettro OFDM simmetrizzato

frequenza di campionamento $f_{sa} = 1/T_{sa}$, a partire dal blocco di simboli di costellazione:

$$x[n] \stackrel{\triangle}{=} x_{OFDM}(nT_{sa}) = \sum_{k=0}^{N-1} c^{(k)}[0] e^{j2\pi k nT_{sa}/T_M}$$
(8.18)

Come dobbiamo scegliere f_{sa} ? Per uno spettro simmetrico in banda base limitato nella banda B, sappiamo che deve essere $f_{sa} \ge 2B$. Nel nostro caso, lo spettro di Fig. 8.8 non è simmetrico (e infatti è relativo a un segnale *complesso* e non reale). Lo spettro OFDM, nel quale tutte le sottoportanti sono su valori positivi della frequenza, ha già in sé una traslazione frequenziale alla frequenza $1/(2T_s)$ - infatti nel momento in cui si dovrà creare un segnale radio su di una banda simmetrica attorno alla frequenza portante f_{RF} , il modulatore I/Q dovrà usare un oscillatore locale alla frequenza $f_{RF} - 1/(2T_s)$. Immaginando di rimuovere questa traslazione intrinseca di $1/(2T_s)$, otteniamo lo spettro simmetrico in banda base di Fig. 8.9, per il quale la banda da usare nella condizione di Nyquist è $B_{sym} = 1/(2T_s)$ e la conseguente (minima) frequenza di campionamento è $f_{sa} = 2B_{sym} = 1/T_s$. Usando questa frequenza di campionamento nella 8.18, e ricordando che $T_M = NT_s$, otteniamo

$$x[n] \stackrel{\triangle}{=} x_{OFDM}(nT_{sa}) = \sum_{k=0}^{N-1} c^{(k)}[0] e^{j2\pi k nT_{sa}/T_M} =$$

$$\sum_{k=0}^{N-1} c^{(k)}[0] e^{j2\pi nk/N} \quad n = 0, 1, ..., N-1$$
(8.19)

Dunque l'algoritmo che il processore DSP deve applicare al blocco di N simboli di costellazione $c^{(k)}[0]$ per ottenere il blocco di N campioni I/Q di segnale OFDM x[n] è una *antitrasformata discreta di Fourier* (IDFT) che può essere implementata in HW o in SW attraverso il ben noto algoritmo *veloce* di Fast Fourier Transform (FFT) - un'ulteriore caratteristica positiva di OFDM. A questo algoritmo digitale seguirà poi la conversione D/A e la modulazione analogica I/Q per costruire l'effettivo segnale radio OFDM.

8.3.2 Demodulatore OFDM Digitale

Procediamo adesso all'implementazione digitale del *demodulatore* OFDM di Fig. 8.7, partendo da un segnale I/Q già campionato $r[n] \stackrel{\triangle}{=} r(nT_s)$, dove r(t) è il segnale analogico ricevuto, e dove abbiamo di nuovo utilizzato la frequenza di campionamento $f_{sa} = 1/T_s$. Nel ricevitore ottimo, la variabile di decisione $z^{(k)}[0]$ è ricavata come nella relazione (8.10); se immaginiamo che r(t) sia limitato in banda, possiamo calcolare il risultato dell'integrale che appare nella (8.10) attraverso la somma dei campioni di segnale raccolti nell'intervallo temporale di integrazione, scalata del fattore T_s che gioca il ruolo del differenziale dt nel calcolo dell'integrale:

$$z^{(k)}[0] = \frac{1}{T_M} \sum_{n=0}^{N-1} r[n] e^{-j2\pi k n T_s/T_M} T_s = \frac{1}{N} \sum_{n=0}^{N-1} r[n] e^{-j2\pi n k/N}$$
(8.20)

Otteniamo stavolta un algoritmo di trasformata di Fourier discreta (diretta), che inverte l'antitrasformata del modulatore. L'interpretazione è immediata: nel modulatore, i simboli di costellazione, uno per ogni frequenza, vengono trasformati in campioni OFDM nel tempo attraverso una trasformata *inversa* (da frequenza a tempo); nel demodulatore, i campioni nel tempo del segnale ricevuto vengono riportati nel dominio della frequenza da una trasformata *diretta* (da tempo a frequenza) per riottenere i simboli di costellazione. Ancora una volta abbiamo a che fare con una algoritmo veloce di efficiente implementazione - il cuore del modem OFDM è un processore FFT.

8.3.3 Sottoportanti Virtuali

C'è in realtà un aspetto nell'implementazione digitale del modem OFDM che abbiamo (intenzionalmente) trascurato, e riguarda la correttezza della frequenza di campionamento scelta per elaborare un segnale che *non* è limitato in banda. Sappiamo che lo spettro $\bar{S}_{OFDM}(f)$ del segnale campionato si ottiene da quello del segnale analogico aggiungendo infinite repliche centrate sui multipli della frequenza di campionamento (vedi (2.79)):

$$\bar{S}_{OFDM}(f) = \frac{1}{T_s} \sum_{\ell} S_{OFDM} \left(f - \frac{\ell}{T_s} \right)$$

Rappresentiamo questa situazione nella Fig. 8.10 (a), dove lo spettro OFDM originario è stato copiato-incollato attorno ai multipli di $1/T_s$. Si nota una certa quantità di aliasing proveniente dalle "code" dello spettro OFDM non limitato in banda. Per ridurre l'errore di aliasing potremmo aumentare la frequenza di campionamento, allontanando le repliche dello spettro. Così facendo modificheremmo però la forma degli algoritmi di modulazione o di demodulazione del segnale rispettivamente (8.19) e (8.20), che diverrebbero diversi dalla (I)DFT, perdendo l'efficienza dell'implementazione attraverso un algoritmo veloce. L'alternativa è quella di lasciare inalterata $f_{sa} = 1/T_s$ e di *diminuire la banda* del segnale.

Fortunatamente, la riduzione di banda si può ottenere in modo facilissimo: lasciando inalterato l'algoritmo di modulazione (8.19), basta comporre il blocco di simboli cui applicare la IDFT in modo diverso: anziché riempire tutto il blocco di simboli di costellazione e poi eseguire la trasformata, si prendono soltanto $N_a = N - N_v$ simboli di costellazione, aggiungendo in coda N_v simboli pari a *zero* per comporre il blocco completo di N valori. Il modulatore riceve questo blocco ed esegue la IDFT di ordine N secondo (8.19), con il risultato di costruire un segnale OFDM in cui le "ultime" N_v sottoportanti sono virtuali



Figura 8.10 Aliasing e sottoportanti virtuali OFDM

cioè ridotte per costruzione a 0. Di conseguenza, lo spettro ha banda più stretta perché le sottoportanti *attive* sono solo $N - N_v = N_a$ e si estendono (solo) fino alla frequenza N_a/T_s , come rappresentiamo in in Fig. 8.10 (b):

$$\bar{S}_{OFDM}(f) = \sum_{k=0}^{N_a - 1} S_{OFDM}\left(f - \frac{k}{T_s}\right)$$

Con una banda ridotta come in Fig. 8.10 (c), l'aliasing diventa trascurabile anche per la frequenza di campionamento $1/T_s$.

È chiaro però che usare le portanti virtuali, a parità di frequenza di campionamento, comporta sì la riduzione di banda desiderata, ma proprio per questo comporta anche un abbassamento della velocità di simbolo di ingresso del fattore N_a/N . In realtà, poiché la velocità di simbolo di ingresso deve restare inalterata e pari a $1/T_s$, per utilizzare le portanti virtuali si deve portare il tempo di simbolo OFDM al valore $T_{M,v} = N_a T_s < T_M$, la spaziatura tra le sottoportanti conseguentemente al valore $f_{sc,v} = 1/T_{M,v} = R_s/N_a > R_s/N = f_{sc}$ e la frequenza di campionamento a $f_{sa,v} = N/T_{M,v} = R_s N/N_a > f_{sa} = R_s$. In questo modo, la banda occupata sarà pari a $N_a \cdot f_{sc,v} = R_s = 1/T_s$ senz'alcuna perdita di efficienza.

Il trucco delle portanti virtuali viene sempre usato nei vari formati OFDM. Nella televisione digitale terrestre europea DVB-T, i due modi fondamentali comprendono N=2048(modo 2k) o N=8192 (modo 8k) portanti nominali. Nel primo caso $N_v = 343$, e nel secondo $N_v = 1375$ con una riduzione percentuale di banda occupata che è praticamente la stessa nei due casi: $(N - N_v)/N=0.83$. Nel Wi-FI, N = 64 e $N_v = 16$ con banda percentuale 0.75.

8.4 Equalizzazione di OFDM nel Dominio della Frequenza

Il formato OFDM è stato creato per ottenere un segnale a banda larga e bit rate elevato con intrinseca robustezza contro la selettività in frequenza. Possiamo davvero confermare che l'intuizione porta a risultati corretti se inviamo tale segnale su di un canale multipath? Sarà forse necessario modificare l'architettura del ricevitore basato su FFT che dopotutto abbiamo ricavato da uno schema ottimo su canale AWGN, senza considerare per niente eventuali distorsioni?

8.4.1 Tempo di guardia e prefisso ciclico

La forma d'onda OFDM subisce le consuete modifiche dovute alla propagazione per cammini multipli e descritte dalla risposta impulsiva del canale h(t) nella forma (7.4). Supponiamo di isolare la trasmissione del "primo" simbolo OFDM (cioè quello per m = 0) $x_0(t)$ come nella (8.17), limitato all'intervallo di tempo $0 \le t < T_M$. L'effetto macroscopico del multipath è quello di "allungare" la forma d'onda al ricevitore y(t), che si estenderà su di un intervallo di tempo più ampio, in particolare da t = 0 fino a $t = T_M + \tau_{max}$, dove $\tau_{max} = \tau_{N_p}$ nella7.4) è il massimo ritardo dei cammini di propagazione - rappresentiamo questa situazione nella Fig. 8.11 (a).

L'allungamento della forma d'onda del simbolo OFDM a causa del multipath crea la cosiddetta *Inter-Block Interference* (IBI) tra un simbolo OFDM e i successivo. Per costruzione del segnale, abbiamo stabilito una condizione di ortogonalità che previene la ISI entro un simbolo OFDM; adesso, con i cammini multipli, abbiamo una nuovo tipo di



Figura 8.11 IBI e tempo di guardia nell'OFDM

auto-interferenza, la IBI appunto, che può essere ugualmente dannosa per la qualità del collegamento.

Una possibile soluzione radicale per eliminare la IBI è quella di Fig. 8.11 (b) (un uovo di Colombo): invece di inviare il simbolo OFDM successivo, cioè nel nostro caso quello per m = 1, immediatamente dopo la fine del tempo del simbolo m = 0, e cioè all'istante $t = T_M$, attendiamo un certo periodo di tempo di "guardia" T_G , e facciamo partire il simbolo m = 1 all'istante $t = T_M + T_G$, in modo che in ricezione la "coda" del simbolo precedente causata dal multipath non collida con il simbolo successivo. Questo tempo di inattività nella trasmissione viene chiamato tempo di guardia e deve essere commisurato al multipath - naturalmente, per evitare IBI dovrà essere $T_G \ge \tau_{max}$.

È chiaro che il tempo di guardia introduce un fattore di inefficienza nel collegamento: volendo mantenere costante la velocità di simbolo inziale $R_s = 1/T_s$ e quindi la durata totale di un blocco dati $N_a T_s$, introducendo il tempo T_G si deve ridurre il tempo netto di trasmissione del simbolo OFDM $T'_{M,v}$ rispetto al precedente $T_{M,v}$ in modo che $T_{OFDM} \stackrel{\triangle}{=} T'_{M,v} + T_G = N_a T_s$. In questo modo avremo un aumento di banda occupata a parità di velocità di simbolo netta d'ingresso - torneremo su questo argomento più avanti - e l'aumento sarà tanto maggiore quanto T_G sarà lungo nei confronti di T_{OFDM} . In pratica, il trasmettitore OFDM, nel consueto tempo di simbolo $T_{OFDM} = N_a T_s$ deve eseguire la IDFT del blocco di simboli di ingresso (N_a attivi + N_v virtuali pari a 0) ottenendo N campioni I/Q del segnale OFDM. A questi N campioni nel tempo si deve aggiungere un numero N_G di campioni uguali a zero per creare l'intervallo di guardia, e quindi nel consueto tempo $N_a T_s$ il segnale è adesso composto non più di N campioni, ma di $N + N_G$. Questo significa che il trasmettitore dovrà diminuire l'intervallo di campionamento al valore $T'_{sa} = N_a T_s/(N + N_G)$ ovvero aumentare la frequenza di campionamento a $f'_{sa} = (1/T_s)(N + N_G)/N_a > 1/T_s$. In questo modo, il tempo netto di simbolo e il tempo

di guardia diventano ovviamente

$$T'_{M,v} = NT'_{sa} = N_a T_s \frac{N}{N + N_G} , \quad T_G = N_G T'_{sa} = N_a T_s \frac{N_G}{N + N_G}$$
(8.21)

ma soprattutto la nuova spaziatura tra le sottoportanti diventa, per mantenere l'ortogonalità,

$$\Delta f \stackrel{\triangle}{=} \frac{1}{T'_{M,v}} = \frac{1}{T_s} \frac{1}{N_a} \left(1 + \frac{N_G}{N} \right) \tag{8.22}$$

La nuova banda occupata sarà dell'ordine di $N_a \Delta f$ e cioé

$$B'_{OFDM} = \frac{1}{T_s} \left(1 + \frac{N_G}{N} \right) \tag{8.23}$$

dove si nota banalmente l'aumento dovuto all'introduzione dell'intervallo di guardia che diminuisce come anticipato l'efficienza spettrale del segnale digitale.

Per fare un esempio, consideriamo di nuovo il modo 2k del DVB-T, nel quale T_{OFDM} = 224 μ s. Sappiamo che $T_G \ge \tau_{max}$, ma sappiamo anche che il ritardo dei cammini multipli è sempre inferiore a circa 10 μ s, e quindi fissando T_G pari a questo valore la perdita di efficienza relativa all'introduzione dell'intervallo di guardia è modesta². Da questo ragionamento si capisce anche che la tecnica dell'intervallo di guardia *non* può essere banalmente usata nei sistemi convenzionali a singola portante per cercar di evitare l'ISI su canale multipath: in quel caso infatti T_G diventa ben maggiore della durata del simbolo di costellazione T_s (che è inferiore a T_{OFDM} di un fattore circa N), con un'inefficienza del tutto intollerabile.

Oltre alla perdita di efficienza, l'introduzione del prefisso ciclico porta un ulteriore inconveniente: il simbolo OFDM ricevuto, compreso il tempo di guardia è di lunghezza $T'_{M,v} + T_G$ e deve essere campionato alla stessa frequenza usata in trasmissione, cioè f'_{sa} , ottenendo $N + N_G$ campioni. Di questi campioni, ne devono essere selezionati N per effettuare la DFT e ricostruire i simboli di costellazione. Ma in presenza di multipath, quali (dove) sono questi campioni? Con che criterio devono essere identificati? Qual è l'effetto del canale distorcente sul processing del ricevitore? A tutte queste domande riusciamo a dare una valida risposta se modifichiamo ulteriormente il formato del segnale trasmesso. L'idea è la seguente: per analizzare il fenomeno dell'IBI abbiamo immaginato di inviare il simbolo OFDM *isolato* m = 0. Supponiamo al contrario che la forma d'onda $x_0(t)$

²Questa caratteristica di essere in grado di gestire facilmente canali con " echi " avneti grandi ritardi, consente di risolvere un problema importante in una rete di radiodiffusione radiofonica o televisiva. La robustezza agli echi " naturali " causati del multipath attraverso l'introduzione del tempo di guardia funziona infatti anche nel caso in cui l'eco provenga da un ripetitore di una rete di radiodiffusione diverso da quello principale che serve una certa area, ma che che opera sulla stessa frequenza e che, se pur con ampiezza minore, viene comunque percepito dal ricevitore (possiamo chiamarlo " eco o multipath artificiale "). Nelle reti convenzionali, questo segnale secondario viene considerato come interferenza ed è dannosissimo, tant'è che il problema viene risolto facendo operare ripetitori vicini su diversi canali frequenziali, e quindi costringendo i ricevitori a cambiare frequenza di ricezione quando si cambia la zona di copertura (ad esempio durante un viaggio). Nelle reti basate su OFDM, l'interferenza da un ripetitore secondario viene considerata come una componente di multipath utile alla ricezione, e questo semplifica notevolmente la progettazione e gestione della rete che diventa a singola frequenza (Single-Frequency Network, SFN). In una rete SFN basata su OFDM, ad ogni canale radio o televisivo viene assegnata un'unica frequenza valevole su di un territorio arbitrariamente grande (nazionale), senza bisogno di complicate assegnazioni multiple di canali su base geografica e/o di risintonizzazioni del ricevitore. L'unica differenza è che i ritardi del multipath " artificiale " in una SFN sono generalmente molto maggiori dei ritardi " naturali ", e quindi il requisito dell'intervallo di guardia deve essere adattato alla bisogna - per il modo DVB-T 2k T_G arriva fino a 1/4 di tutto T_{OFDM}

del simbolo m = 0 anziché essere trasmessa una sola volta sia *ripetuta* infinite volte per ottenere un segnale *periodico* $x_p(t)$ di periodo T_M (ignoriamo in questa discussione le portanti virtuali): $x_p(t) = \sum_m x_0(t - mT_M) = x_p(t + T_M)$. Se così fosse, anche il segnale in uscita al canale con cammini multipli $y_p(t)$ sarebbe periodico con lo stesso periodo come in Fig. 8.12 (a), e il ricevitore potrebbe osservare un qualunque periodo di lunghezza T_M (cioè N campioni digitali) senza temere di perdere alcuna informazione.

Prima di discutere la rilevanza di questa ipotesi di periodicità relativamente al segnale OFDM che tutto è meno che periodico, terminiamone l'analisi. Se $x_p(t)$ è periodico, possiamo svilupparlo in serie di Fourier (2.3)

$$x_p(t) = \sum_k X_k \, e^{j2\pi kt/T_M}$$
(8.24)

dove X_k è il coefficiente (2.4)

$$X_k \stackrel{\triangle}{=} \frac{1}{T_M} \int\limits_0^{T_M} x_p(t) e^{-j2\pi kt/T_M} dt \qquad (8.25)$$

Confrontando la generica (8.24) con l'espressione (8.17) di $x_0(t)$, notiamo che non c'è alcun bisogno di calcolare X_k : per ispezione, vediamo che $X_k = c^{(k)}[0], k = 0, 1, ..., N - 1$ (e 0 per k fuori da quest'intervallo: la serie degenera in una somma finita).

Detto questo, se inviamo $x_p(t)$ sul canale multipath con risposta h(t), l'uscita (periodica) sarà $y_p(t) = x_p(t) \otimes h(t)$, e il coefficiente di Fourier Y_k di $y_p(t)$ si trova immediatamente dalla relazione $Y_k = X_k \cdot H(k/T_M)$ dove H(f) è la risposta in frequenza del canale, e k/T_M è la k-esima frequenza armonica del segnale periodico. Incidentalmente, notiamo che le frequenze armoniche di $x_p(t)$ coincidono con le sottoportanti del segnale OFDM. Qual è però Y_k e il nostro problema? Ricordiamo che l'uscita del demodulatore OFDM (8.6) per la portante k è la variabile di decisione

$$z^{(k)}[0] = \frac{1}{T_M} \int_0^{T_M} y_p(t) e^{-j2\pi kt/T_M} dt$$
(8.26)

e vediamo che l'uscita del demodulatore ottimo OFDM è... esattamente il coefficiente di Fourier Y_k .

La conclusione è che implementando un demodulatore OFDM (anche) in forma digitale tramite DFT che osserva su di un periodo (N campioni) un simbolo periodicizzato, ciò che otteniamo è

$$z^{(k)}[0] = Y_k = X_k \cdot H(k/T_M) = c^{(k)}[0] \cdot H\left(\frac{k}{T_M}\right) \quad k = 0, 1, ..., N - 1$$
(8.27)

La variabile di decisione per la sottoportante k contiene il k-esimo simbolo di costellazione senz'alcuna come previsto (l'ortogonalità è preservata), e possiamo inoltre vedere chiaramente l'effetto del canale selettivo in frequenza: il simbolo di costellazione k-esimo viene modificato in ampiezza/fase dal valore in ragione della risposta in frequenza del canale alla frequenza della sottoportante su cui è stato inviato - una piena conferma della intuizione originaria, e la prova finale che l'invenzione del formato multicarrier funziona davvero.

Proprio vero? Manca in realtà l'ultimo dettaglio, cioè come mettere in relazione il segnale periodicizzato della nostra analisi (8.24)-(8.27) con l'effettivo segnale OFDM non



(b)



periodico. L'idea è la seguente: far *sembrare* il segnale OFDM il più possibile periodico. Per far questo, si può usare il tempo di guardia per trasmettere, anzichè... niente, una periodicizzazione della forma d'onda $x_0(t)$. La Fig. 8.12 (b) mostra come fare ciò: l'ultima sezione, di durata T_G (ovvero N_G campioni) del simbolo OFDM viene copiata-incollata all'inizio del simbolo stesso, creando il cosiddetto *prefisso ciclico*. Il simbolo completo di durata $N + N_G$ campioni ha quindi due sezioni identiche lunghe N_G in testa e in coda: invece di aggiungere N_G zeri per creare il tempo di guardia, il modulatore *premette* agli Ncampioni prodotti dalla IDFT gli ultim N_G campioni in uscita alla IDFT stessa.

Questa pseudo-periodicità creata dal modulatore si ritrova anche nel segnale ricevuto all'uscita del canale, come viene illustrato ancora in Fig. 8.12 (b), evidenziando anche la differenza con l'uscita nel caso (irrealistico ma che ben conosciamo) di ingresso veramente periodico. La cosa fondamentale è che, in una certo intervallo temporale, l'uscita del canale con la vera OFDM dotata di prefisso ciclico è *identica* a quella fittizia periodica $y_p(t)$. Se quindi il processore DFT del ricevitore osserva il segnale in questo intervallo, la sua uscita sarà identica alla (8.27) che abbiamo calcolato nel caso di segnale periodico, e che dimostra assenza di ICI come desideravamo!

Il prefisso ciclico è un'altra caratteristica di OFDM che viene sempre utilizzato, in tutti i vari formati standard. Come anticipato, il prefisso ciclico non viene gratis, perchè, a parità di velocità di simbolo di partenza, allarga la banda del segnale (il ragionamento è analogo a quello anticipato per l'intervallo di guardia). Con un calcolo un po' complicato si trova che la dsp del segnale OFDM con prefisso ciclico è

$$S_{OFDM}(f) = C_2 T_{OFDM} \sum_{k=0}^{N_a - 1} \operatorname{sinc}^2 \left(\left(f - \frac{k}{T'_M} \right) T_{OFDM} \right)$$
(8.28)

con la seguente interpretazione: come già accennato, la spaziatura tra le sottoportanti è aumentata e vale $\Delta f = 1/T'_M$, aumentando la banda totale fino al nuovo valore (8.23), ma la banda di ogni sottoportante è invariata e pari a $1/T_{OFDM}$ visto che comunque $T_{OFDM} = N_a T_S$. Visivamente, lo spettro ha una piccola "ondulazione" nella banda passante.

Esempio 8.42

Per concettualizzare tutti i vari parametri che abbiamo introdotto per il segnale OFDM, facciamo l'esempio del formato IEEE 802.11g, la base delle molte versioni di Wi-Fi in commercio alla data odierna (2020). Il formato fondamentale è OFDM con N=64, con 48 portanti dati e 4 portanti pilota fisse (discuteremo queste ultime nella prossima sezione), cioè $N_a=52$ e $N_v=12$. Al simbolo base vengono premessi $N_G=16$ campioni di prefisso ciclico, per un tempo totale di simbolo OFDM $T_{OFDM} = T'_M + T_G=4 \mu$ s, da cui ovviamente $T_G=0.8$ ns e $T'_M=3.2 \mu$ s. Di conseguenza, $f'_{sc} = 1/T'_M=312$ kHz. La banda nominale è $B'_{OFDM} = 52 * f'_{sc}=16.25$ MHz. Come formato MOD/COD di ingresso, il modulatore OFDM accetta costellazioni B/QPSK, 16-QAM, 64-QAM accoppiate a codici convoluzionali con r pari a 1/2, 3/4 or (soltanto per 64-QAM) 2/3.

Con l'introduzione del prefisso ciclico, risolviamo anche una ulteriore questione che abbiamo già accennato: dato che il simbolo OFDM con prefisso è costituito da $N + N_G$ campioni alla frequenza f'_{sa} , quali sono di questi gli N che dobbiamo selezionare come input del DFT processor? Ovviamente, quelli che iniziano dopo gli N_G iniziali, che sono



Figura 8.13 Sincronizzazione di tempo nell'OFDM

solo una copia dei finali. Il ricevitore dovrà *sincronizzare* il simbolo, cioè trovare l'istante temporale che indichiamo nella Fig. 8.13 con t_e . Che succede se per qualche motivo il sincronizzatore non è così accurato, introduce un errore di stima ε e quindi inizia a raccogliere i campioni da processare con la DFT all'istante $t_e - \varepsilon$ invece che t_e ? Se $\varepsilon \leq T_G - \tau_{max}$ notiamo dalla figura che la forma d'onda, ovvero gli N campioni tra $t_e - \varepsilon$ e $t_e - \varepsilon + T_M$ sono una *traslazione ciclica* di quelli (corretti) tra t_e and $t_e + T_M$, e dunque rappresentano comunque *tutti* i campioni del simbolo OFDM, anche se "ruotati". Qual è l'effetto di questo ritardo *ciclico* sull'uscita del ricevitore, cioè sull'espressione (8.27) della variabile di decisione $z^{(k)}[0]$? Ri-calcolando di fatto il coefficiente di Fourier del segnale (pseudo-)periodico ritardato ciclicamente, si trova che il ritardo ε introduce un ulteriore fattore di fase $\exp\{-j2\pi k\varepsilon/T_M\}$ nell'espressione di $Y_k = z^{(k)}[0]$. Questo fattore può essere inglobato nella risposta in frequenza del canale, definendo una risposta "estesa" che contiene questo ulteriore sfasamento, ovviamente ignoto al ricevitore tanto quanto la riposta del canale multipath vera e propria:

$$z^{(k)}[0] = c^{(k)}[0] \cdot H_k \quad , \quad H_k \stackrel{\Delta}{=} H(k/T_M) e^{-j2\pi k\varepsilon/T_M}$$
(8.29)

In conclusione, osserviamo che l'effetto del fattore H_k sulla demodulazione dati non può essere trascurata. Se $z^{(k)}[0]$ viene inviata a un decisore con le zone di decisione nominali della M-QAM usata nel collegamento, le decisioni sono *catastrofiche* perchè i simboli di costellazione sono alterati da un fattore (sconosciuto) di ampiezza $|H_k|$ e sono ruotati in fase di un fattore (altrettanto sconosciuto) $\angle H_k$. In realtà, questo accade anche nella trasmissione su di un canale piatto in frequenza, con la differenza che nel nostro caso il fattore di fase/ampiezza H_k è variabile da sottoportante a sottoportante, cosicché il demodulatore, per poter compensare la costellazione in fase/ampiezza, deve stimare N_a valori di H_k nel simbolo OFDM, e poi usare le stime per effettuare la *compensazione*, come suggerito dalla Fig. 8.14.

8.4.2 Stima e compensazione della risposta in frequenza del canale

A questo punto riprendiamo in considerazione la questione principale della ricezione dati, e riconsideriamo l'effetto del disturbo AWGN nel ricevitore. Il segnale ricevuto è quindi $y(t) = x_{OFDM}(t) \otimes h(t)$ all'uscita del canale selettivo, accompagnato da rumore: r(t) = y(t) + w(t) dove le componenti indipendenti a media nulla I/Q di w(t) hanno come sempre entrambe dsp $S_w(f) = N_0$. La variabile di decisione all'uscita del demodulatore



Equalizzazione nel dominio della frequenza per OFDM Figura 8.14

8.6 sarà costituita dalla componente di segnale (8.29) e da una componente di rumore:

$${}^{(k)}[0] = c^{(k)}[0] \cdot H_k + w(k)[0]$$
(8.30)

dove w(k)[0] è una variabile aleatoria Gaussiana complessa a media nulla (perchè derivante da trasformazione lineare di un processo Gaussiano) avente componenti I/Q indipendenti, entrambe con varianza $\sigma_w^2 = N_0/T_M$. Purtroppo non è possibile stimare direttamente H_k dall'osservazione di $z^{(k)}[0]$ perchè sia $c^{(k)}[0]$ che w(k)[0] non sono note al ricevitore. Se per assurdo i simboli $c^{(k)}[0]$ fossero noti al ricevitore, allora una stima semplice sarebbe

$$\hat{H}_k \simeq z^{(k)}[0]/c^{(k)}[0]$$
(8.31)

con un'accuratezza dipendente dall'ampiezza del rumore, cioè in parole povere dallo SNR al ricevitore.

Da quest'osservazione scaturisce allora una possibile strategia di stima di canale che ad esempio è utilizzata da Wi-Fi: il trasmettitore, invece di inviare dati, trasmette con una certa cadenza un preambolo OFDM, cioè un intero simbolo OFDM nel quale i simboli sono fissi e noti al ricevitore (stabiliti dallo standard). Se la risposta del canale è costante su di un certo numero di simboli OFDM successivi (come accade in caso di basso grado di mobilità), il ricevitore applica i valori \hat{H}_k ricavati nel preambolo per la demodulazione dei dati nei simboli seguenti, fino alla ricezione del preambolo successivo.

Stimata la risposta del canale H_k , il ricevitore deve *compensarla*. Un approccio elementare ma efficace è quello di "eliminare" l'effetto del canale dalla variabile di decisione $z^{(k)}[0]$ semplicemente implementando la divisione tra numeri complessi: $\xi_0^{(k)} \stackrel{\triangle}{=} z^{(k)}[0]/\hat{H}_k$. Infatti,

$$\xi^{(k)}[0] = \frac{z^{(k)}[0]}{H_k} \simeq c^{(k)}[0] + \frac{w(k)[0]}{H_k} = c^{(k)}[0] + \nu^{(k)}[0]$$
(8.32)

dove l'ipotesi è che la stima sia sufficientemente accurata da poter porre $\hat{H}_k \simeq H_k$. La *ratio* di questa relazione (che come già accennato viene indicata con il nome di equalizzazione zero-forcing (ZF)) è chiara: il simbolo $c^{(k)}[0]$ è ripristinato con i corretti valori di ampiezza e fase, e questo perchè, come il vecchio equalizzatore adattativo E(f) della sezione 8.1.1, il criterio ZF intende *invertire* la riposta del canale multipath per evitare la distorsione.

La relazione (8.32) ci mostra anche che il simbolo di costellazione è disturbato da un termine di rumore $\nu^{(k)}[0]$ la cui varianza è diversa da quella di w(k)[0]:

$$\nu^{(k)}[0] = \frac{w(k)[0]}{H_k} = \frac{w(k)[0]e^{-j \angle H_k}}{|H_k|} \quad \Rightarrow \quad \sigma_\nu^2 = \frac{\sigma_w^2}{|H_k|^2}$$

290



Figura 8.15 Pilota interlacciate per la stima di canale OFDM

Questa varianza varia da sottoportante a sottoportante e in particolare è grande quando $|H_k|$ è piccolo: possiamo avere un fenomeno di *incremento del rumore* che dovremo considerare attentamente nel calcolo della BER del ricevitore.

Un metodo alternativo per eseguire la stima del canale senza dedicare un intero simbolo OFDM a tale funzione, oppure per aggiornare la stima tra un preambolo e l'altro, si basa sull'introduzione di sottoportanti pilota (vedi Esempio 42). Una sottoportante pilota, invece di trasportare un simbolo di costellazione valido, contiene un valore noto (il simbolo pilota, appunto) dettato dallo standard, ad esempio (1 + j0). I simboli pilota sono interlacciati regolarmente in frequenza con le sottoportanti dati, ad esempio una pilota ogni 4 sottoportanti dati dati come in Fig. 8.15. Il ricevitore esegue la stima di canale secondo (8.31) in ogni simbolo OFDM, ma solo sulle sottoportanti contenenti le pilota. A prima vista, questi valori stimati sembrano essere inutili: non c'è alcuno bisogno di stimare/rimuovere l'ampiezza/fase del canale su sottoportanti che non contengono dati! In realtà, una volta ricavata H_k su tutte le sottoportanti pilota, possiamo usare questi valori come nodi (cioè punti-base) di una formula di interpolazione per ricostruire i valori mancanti H_k (anche) delle sottoportanti dati. Vengono in pratica utilizzate formule di interpolazione polinomiale o spline, o anche basate su approcci statistici più complessi come l'interpolazione di Wiener, che però richiedono modellizzazione del canale pià complicate e sulle quali non insistiamo ulteriormente.

Inserire un preambolo o le sottoportanti pilota nel formato per eseguire la stima di canale ovviamente introduce un certo *overhead*, cioè una certa perdita di efficienza del collegamento: è il prezzo da pagare per far funzionare in modo ottimale il ricevitore.

Esempio 8.43

Consideriamo un esempio nel quale le pilota sono inserite regolarmente con un formato 1-su-L, cioè due pilota sono separate da L - 1 sottoportanti dati. L'overhead dovuta alle pilota è pari a 1/L e non è del tutto trascurabile, visto che valori pratici per L sono nell'ordine di 4-5.

L'approccio più semplice ma abbastanza efficace per ricostruire i valori mancanti di H_k è l' *interpolatore lineare*. Dopo aver ricavato due stime consecutive della risposta del canale sulle sottoportanti pilota, chiamiamole $h_1 \stackrel{\triangle}{=} \hat{H}_{\ell L}$ e $h_2 \stackrel{\triangle}{=} \hat{H}_{(\ell+1)L}$, secondo (8.31), i valori intermedi $\hat{H}_{\ell L+1}, ..., \hat{H}_{\ell L+L-1}$ vengono ricavati come suggerito nella Fig. 8.16:

$$\hat{H}_k = h_1 + \frac{h_2 - h_1}{L} (k - \ell L) , \ \ell L < k < \ell L + L$$
(8.33)



Figura 8.16 Interpolazione lineare per la stima di canale

L'accuratezza dell'interpolazione dipende *in primis* dal valore di SNR nel ricevitore, che influenza direttamente la precisione delle stime "iniziali" sui nodi (sulle pilota) $\hat{H}_{\ell L}$ e $\hat{H}_{(\ell+1)L}$; dipende poi dalla *densità* delle pilota: più piccola è *L*, migliore sarà l'interpolazione perché minore sarà la variazione della risposta del canale tra una pilota e l'altra (ma aumenterà l'overhead). Infine, l'accuratezza dipende anche dalla bontà e complessità dell'algoritmo di interpolazione scelto - tutti questi fattori devono essere attentamente valutati nel fissare lo standard (e quindi *L*) e nel progettare il ricevitore (l'algoritmo di interpolazione, che *non* è fissato nello standard).

Dopo la *stima*, il ricevitore deve effettuare la *compensazione* della risposta in frequenza del canale, applicando un criterio di *equalizzazione nel dominio della frequenza* come lo ZF della (8.32). Per generalizzare quest'approccio, possiamo dire che un algoritmo di equalizzazione nel dominio della frequenza (cioè sottoportante per sottoportante nell'ambito dello spettro del segnale...) si svolge calcolando $\xi^{(k)}[0] \stackrel{\triangle}{=} z^{(k)}[0] \cdot \gamma_k$, dove γ_k è un opportuno coefficiente complesso che dipende dal criterio adottato - per l'equalizzatore ZF, $\gamma_k = 1/\hat{H}_k$. Se la costellazione utilizzata è una M-PSK, è possibile compensare soltanto la *fase* e porre $\gamma_k = \hat{H}_k^*$ - un criterio che assomiglia al semplice "filtraggio adattato". Si può anche adottare un criterio di equalizzazione di tipo MMSE, che minimizza cioè $E \{|\xi^{(k)}[0] - c^{(k)}[0]|^2\}$ e che tiene conto sia della risposta del canale, sia della presenza del rumore w(k)[0]. In tal caso si ottiene

$$\gamma_k = \frac{\hat{H}_k^*}{|\hat{H}_k|^2 + 1/(2E_s/N_0)}$$

dove E_s/N_0 è il consueto parametro di SNR calcolato al ricevitore. Se l'SNR è molto alto, l'equalizzatore MMSE si comporta come lo ZF (infatti, $1/(2E_s/N_0) \rightarrow 0$ e $\gamma_k \rightarrow \hat{H}_k^*/|\hat{H}_k|^2 = 1/\hat{H}_k$); se al contrario l'SNR è molto basso, $\gamma_k \rightarrow \hat{H}_k^* \cdot (2E_s/N_0)$ e l'MMSE coincide con il filtro adattato. Nei casi intermedi, l'equalizzatore ha comunque lo scopo di prevenire l'incremento di rumore già evidenziato per lo ZF, introducendo un termine di "regolarizzazione" $1/(2E_s/N_0)$ al denominatore di γ_k che non tende mai all'infinito, neanche quando H_k è molto piccolo.

Al termine dell'esame di tutti gli accorgimenti per ottimizzare il funzionamento di un collegamento OFDM, riportiamo nella Fig. 8.17 uno schema architetturale di massima del



EQUALIZZAZIONE DI OFDM NEL DOMINIO DELLA FREQUENZA 293

Figura 8.17 Architettura di massima del modem OFDM

collegamento (o meglio, del modem) OFDM con i vari blocchi componenti, indicando la successione delle varie funzioni e il tipo di elaborazione "parallela" che è necessaria.

8.4.3 BER del ricevitore OFDM con equalizzazione ZF

Per ricavare la BER del ricevitore OFDM, dobbiamo preliminarmente "calibrare" le potenza di segnale e rumore in gioco. Riprendiamo l'espressione (8.8) del segnale OFDM:

$$x_{OFDM}(t) = \sum_{k=0}^{N-1} y_k(t) = \sum_{k=0}^{N-1} x_k(t) e^{j2\pi kt/T_M}$$

$$= \sum_{k=0}^{N-1} \sum_{m=-\infty}^{+\infty} c^{(k)}[m] p(t-mT_M) e^{j2\pi kt/T_M}$$
(8.34)

dove per semplicità non consideriamo portanti virtuali e/o prefisso ciclico. La potenza media a radiofrequenza del segnale è (considerando che i segnali sulle sottoportanti sono indipendenti, e osservando che $|e^{j2\pi kt/T_M}|^2 = 1$)

$$P_{OFDM} = \frac{1}{2} \sum_{k=0}^{N-1} P_{x_k} = \frac{N}{2} \cdot P_{x_k} = \frac{N}{2} \cdot P_x$$

(il fattore 1/2 tiene conto del passaggio da banda base a radiofrequenza), dove P_x è la potenza del segnale su di una qualunque sottoportante (la stessa $\forall k$). Considerando poi il formato (8.7) del k-esimo segnale dati di sottoportante (e in particolare il fatto che l'energia dell'impulso NRZ p(t) è pari a T_M) vediamo che $P_x = C_2$ e quindi $P_{OFDM} = NC_2/2$. Volendo imporre una certa potenza radio P_{RF} , l'espressione "calibrata" del segnale OFDM è

$$x_{OFDM}(t) = \sqrt{\frac{2P_{RF}}{NC_2}} \sum_{k=0}^{N-1} \sum_{m=-\infty}^{+\infty} c^{(k)}[m]p(t-mT_M) e^{j2\pi kt/T_M}$$
(8.35)

Con questa calibrazione possiamo calcolare in modo non ambiguo i rapporti SNR in qualunque punto della catena TX/RX. Per prima cosa, dobbiamo includere il fattore di scala del segnale nella (8.30):

$$z^{(k)}[0] = \sqrt{\frac{2P_{RF}}{NC_2}}c^{(k)}[0] \cdot H_k + w(k)[0] = c^{(k)}[0] \cdot H'_k + w^{(k)}[0]$$
(8.36)

dove H'_k adesso include il fattore di scala del segnale. Applicando l'equalizzazione ZF con stima *perfetta* del canale ($\hat{H}_k \equiv H_k$) otteniamo

$$\xi^{(k)}[0] = c^{(k)}[0] + \nu^{(k)}[0]$$
(8.37)

come in (8.32), però la varianza del rumore $\nu^{(k)}[0]$ è adesso

$$\sigma_{\nu}^{2} = \sigma_{w}^{2} / |H_{k}|^{2} \cdot \frac{NC_{2}}{2P_{RF}} = \frac{C_{2} \cdot N_{0}}{2P_{RF}|H_{k}|^{2} \cdot T_{M}/N} = \frac{C_{2} \cdot N_{0}}{2P_{r}^{(k)}T_{s}} = \frac{C_{2}}{2E_{s}^{(k)}/N_{0}}$$
(8.38)

dove $P_r^{(k)} \stackrel{\triangle}{=} P_{RF} |H_k|^2$ è la potenza ricevuta sulla k-esima sottoportante, e $E_s^{(k)} = P_r^{(k)} T_s$ è l'energia-per-simbolo ricevuta su ogni sottoportante, variabile in generale al variare di k.

È chiaro che ogni sottoportante si trova ad operare in una condizione di SNR $E_s^{(k)}/N_0$ diversa, a seconda del valore $|H_k|$ della riposta in ampiezza del canale - come già sapevamo. Illustriamo questa situazione in Fig. 8.18 che rappresenta i livelli della dsp di segnale e rumore in dB - la "distanza" in verticale tra le due curve è suggestiva del valore di $E_s^{(k)}/N_0$ (dB) su ogni sottoportante, che puó essere parecchio variabile.

Il calcolo della BER del collegamento è semplice per una costellazione 4-QAM (QPSK) con simboli appartenenti all'alfabeto $\{\pm 1 \pm j\}$. In tal caso, $C_2 = 2$ e la BER si trova considerando una tra le componenti I o Q di $\xi^{(k)}[0]$:

$$\xi_I^{(k)}[0] = c_I^{(k)}[0] + \nu_I^{(k)}[0] = \pm 1 + \nu_I^{(k)}[0]$$
(8.39)



Figura 8.18 Variabilità dello SNR da sottoportante a sottoportante

dove, dalla (8.38), la varianza della componente di rumore è $\sigma_{\nu}^2 = (E_s^{(k)}/N_0)^{-1}$. La BER si trova immediatamente:

$$BER_k = \mathcal{Q}\left(\sqrt{\frac{E_s^{(k)}}{N_0}}\right) = \mathcal{Q}\left(\sqrt{\frac{2E_b^{(k)}}{N_0}}\right)$$
(8.40)

e, come evidenziato nella notazione, dipende dal numero d'ordine della sottoportante k attraverso il valore del k-esimo SNR $E_s^{(k)}/N_0$. Poichè i bit di informazione associati ai simboli di costellazione vengono smistati equamente su tutte le varie sottoportanti, la BER complessiva (media) del collegamento si ricava come la media delle varie BER_k di sottoportante:

$$BER = \frac{1}{N} \sum_{k=0}^{N-1} BER_k = \frac{1}{N} \sum_{k=0}^{N-1} \mathcal{Q}\left(\sqrt{\frac{2E_b^{(k)}}{N_0}}\right)$$
(8.4)

Interpretiamo questo risultato con un esempio.

Esempio 8.44

Riprendiamo l'esempio della tecnologia Wi-Fi IEEE 802.11g con $N_a = 48$ sottoportanti dati e DFT di ordine N=64. Supponiamo di avere un canale wireless quasi-ovunque perfetto, a meno di una frequenza di notch, e che quest'ultima coincida con una sottoportante, come in Fig. 8.19. In parole povere, $H_k = 1$ salvo per $k = \bar{k}$ dove $H_{\bar{k}} = 0$. Per semplificare, immaginiamo anche che lo $E_s^{(k)}/N_0$ sulle sottoportanti "buone" ($k \neq \bar{k}$) sia così alto che praticamente su tali sottoportanti $BER_k \simeq 0$. Viceversa, sulla sottoportante "cattiva" ($k = \bar{k}$) il segnale è completamente annullato, e quindi $BER_{\bar{k}} = 1/2$. Quanto vale le BER media del collegamento? Visto che il canale è quasi-ovunque buono dobbiamo aspettarci una buona prestazione complessiva? Dalla (8.41) abbiamo (considerando solo le sottoportanti dati)

$$BER = \frac{1}{48} \sum_{k=0}^{47} BER_k = \sum_{k \neq \bar{k}} BER_k + BER_{\bar{k}} \simeq 0 + \frac{1}{48} \cdot \frac{1}{2} \simeq 10^{-2}$$
(8.42)

Il risultato è in realtà assai deludente: la BER complessiva è uno striminzito 10^{-2} , al limite del fuori servizio per qualunque applicazione, nonostante la presenza di una sola frequenza di notch su tutta la banda.

3.41)



Figura 8.19 Canale wireless quasi-ovunque ideale

Quest'esempio ci dimostra che la presenza di frequenze di notch (profonde) nella risposta del canale crea un *pavimento* (floor) di BER anche per valori di SNR mediamente molto alti. La soluzione a questo problema sta nelle molte tecniche di codifica dei canale che abbiamo già studiato nel capitolo 6, che diventano parte integrante ed essenziale della tecnologia - così vero che il formato OFDM è spesso indicato come COFDM (Coded OFDM). Con il codice di canale, il pavimento di BER può essere ridotto convenientemente a livelli più accettabili. Nelle sezioni successive approfondiremo la questione, ricavando in particolare il valore della capacità di Shannon per un collegamento in tecnologia multiportante.

8.4.4 Modellistica ed emulazione del canale wireless nel dominio della freguenza

Per un segnale multicarrier, così come abbiamo effettuato la *stima di canale* e l'*equalizzazione* in ambito frequenziale, è anche possible implementare efficientemente l'*emulazione del canale wireless*, affrontata in generale nel paragrafo 7.2, operando nell'ambito della frequenza.

Nel caso del multicarrier è conveniente introdurre una semplificazione nel modello generale di canale multipath (7.19): molto spesso il valore della Doppler spread f_D è molto più piccolo del symbol rate multicarrier $1/T_M$. Di conseguenza, possiamo adottare un modello quasi-stazionario del multipath, secondo il quale i valori dei processi di fading $h_i(t)$ nella (7.19) possono ritenersi costanti nell'ambito del generico intervallo di simbolo multicarrier $mT_M \leq t < (m+1)T_M$:

$$x(t) = \sum_{i=1}^{N_p} h_i[m] \, s(t - \tau_i) \ , \ mT_M \le t < (m+1)T_M$$
(8.43)

dove $h_i[m] \stackrel{\triangle}{=} h_i(mT_M)$ è una versione discretizzata al tempo di simbolo multicarrier T_M del processo di fading *i*-esimo - di fatto $h_i(t)$ viene approssimato con una sua versione *costante a tratti* (sample & hold). Nella simulazione, questa sequenza di valori $h_i[m]$ può essere facilmente modellata attraverso il filtraggio di un processo Gaussiano bianco a tempo discreto, come già descritto nel paragrafo 7.2.

Detto questo, è immediato ricavare i valori dei coefficienti di canale in ambito frequenziale $H_k = H(k/T_M)$ da utilizzare per simulare/emulare la relazione (8.27) che fornisce il segnale multicarrier ricevuto, sottoportante per sottoportante, comprensivo di distorsione del canale multipath. Infatti, dalla (8.43) otteniamo, sempre nell'ambito $mT_M \le t < (m+1)T_M,$

$$X(f) = \sum_{i=1}^{N_p} h_i[m] S(f) e^{-j2\pi f\tau_i} \Rightarrow H(f) = \sum_{i=1}^{N_p} h_i[m] e^{-j2\pi f\tau_i}$$
(8.44)

e quindi

$$H_k[m] = \sum_{i=1}^{N_p} h_i[m] e^{-j2\pi k\tau_i/T_M}$$
(8.45)

dove la notazione $H_k[m]$ indica che la risposta in frequenza è "locale" al simbolo multicarrier *m*-esimo, e deve essere variata da simbolo a simbolo secondo la prescritta Doppler spread.

Fissato il profilo del canale multipath da emulare/simulare (ad esempio uno di quelli mostrati in Tab. 7.4), e quindi fissati i ritardi che restano costanti durante la simulazione, i valori $e^{-j2\pi k\tau_i/T_M}$, per $i = 1, ..., N_p$ e k = 0, ..., N - 1 possono essere pre-calcolati e memorizzati in una tabella all'inizio della simulazione; in ogni intervallo di simbolo multicarrier, i valori $H_k[m]$ devono essere poi (ri)calcolati durante la simulazione secondo la (8.45). Dal punto di vista delle proprietà statistiche osserviamo che i coefficienti $H_k[m]$ sono un insieme di valori aleatori congiuntamente Gaussiani perchè ottenuti attraverso combinazione lineare di valori $h_i[m]$ Gaussiani indipendenti (e quindi a loro volta congiuntamente Gaussiani).

Esempio 8.45

Fissato un certo simbolo multicarrier #*m* (il cui indice ometteremo nella notazione), ricaviamo la funzione di autocorrelazione $R_H[k, k + \mu]$ dei coefficienti H_k del canale multipath in ambito frequenziale:

$$R_{H} [k, k + \mu] \stackrel{\triangle}{=} \mathbb{E} \left\{ H_{k} \cdot H_{k+\mu}^{*} \right\}$$

$$= \mathbb{E} \left\{ \sum_{i=1}^{N_{p}} h_{i} e^{-j2\pi k\tau_{i}/T_{M}} \sum_{\ell=1}^{N_{p}} h_{\ell}^{*} e^{j2\pi (k+\mu)\tau_{\ell}/T_{M}} \right\}$$

$$= \sum_{i=1}^{N_{p}} \sum_{\ell=1}^{N_{p}} \mathbb{E} \left\{ h_{i} h_{\ell}^{*} \right\} e^{j2\pi (k\tau_{\ell} - k\tau_{i} + \mu\tau_{\ell})/T_{M}}$$

$$= \sum_{i=1}^{N_{p}} P_{h_{i}} e^{j2\pi \mu\tau_{i}/T_{M}} = R_{H} [\mu]$$
(8.46)

Fisato il simbolo m, i coefficienti $H_k[m]$ rappresentano un processo Gaussiano che, nell'ambito della frequenza k, è *stazionario*, e le cui caratteristiche (in particolare la sequenza di autocorrelazione $R_H[\mu]$) dipendono (solamente) dal profilo di canale attraverso le potenze P_{h_i} e i ritardi τ_i come indicato chiaramente dalla (8.46).

Con questo metodo si riesce ad emulare esattamente ed efficientemente la ricezione del segnale multicarrier quando il ritardo massimo del canale multipath è inferiore alla lunghezza del prefisso ciclico e non vi sono errori di sincronismo nell'identificare l'inizio del simbolo e rimuovere il prefisso stesso. Un eventuale errore di sincronismo può essere facilmente incorporato nella simulazione secondo quanto descritto nel paragrafo 8.4.1.



Figura 8.20 Allocazione delle sottoportanti a utenti distinti in OFDMA

8.5 Vantaggi, svantaggi, e varianti della tecnologia OFDM

8.5.1 Multiplazione e accesso multiplo flessibili

8.5.1.1 OFDM Multiutente (aka OFDMA) Contrariamente al suo stesso nome, piuttosto che per effettuare una multiplazione OFDM è stato introdotto e sviluppato solo per contrastare in un collegamento a singolo utente il problema della selettività in frequenza. Questo vale sicuramente per i numerosi esempi che abbiamo visto nelle pagine precedenti relativi a Wi-Fi o DVB-T, ma quando si parla di 4G LTE o 5G NR, OFDM è usato in un modo leggermente differente, che possiamo chiamare "OFDM multiutente". Nel downlink di LTE/NR, le varie sottoportanti disponibili per i dati sono *ripartite* tra i diversi utenti nella cella, quindi ogni sottoportante è assegnata esclusivamente a un singolo utente. Questa tecnica ricorda molto FDM, ma con una differenza: in LTE, le sottoportanti appartenenti a un determinato utente non necessariamente sono *adiacenti* in frequenza, cioè non appartengono ad un'unica banda contigua di frequenze (se così fosse avremmo un banale FDMA implementato digitalmente ...). Viceversa, sono in generale *distribuite* su tutta larghezza di banda prevista dallo standard, come si vede in Fig. 8.20.

Questa tecnica, chiamata Orthogonal Frequency-Division Multiple Access (OFDMA) anche se si riferisce alla multiplazione di downlink, presenta grande flessibilità nell'allocazione della capacità disponibile, poiché a diversi utenti può essere assegnato un numero diverso di sottoportanti, allocando quindi una capacità maggiore o minore (vedi di nuovo Fig. 8.20). La distribuzione delle sottoportanti di ogni utente sull'intera banda comporta anche un'assegnazione equa delle risorse di canale, nel senso della qualità delle sottoportanti: nel caso di allocazione contigua, un determinato utente può avere tutte o quasi le sue proprie sottoportanti in una regione sfavorevole della spettro, come quella ombreggiata nella Fig. 8.20. Distribuendo le portanti, ogni utente avrà viceversa un mix equo di sottoportanti "buone" e "cattive". Il ricevitore nel terminale mobile demodula l'intero pettine di N sottoportanti attraverso un demodulatore OFDM standard, ma considera solo quelle di propria pertinenza, secondo l'allocazione effettuata dal controllo di rete.

8.5.1.2 OFDMA in senso stretto La tecnica OFDMA vera e propria viene applicata più o meno nella stessa maniera nell'uplink della rete, con la differenza che l'accesso avviene da punti fisicamente diversi (il vero e proprio *accesso multiplo*) e quindi il segnale OFDMA che giunge alla stazione radio base viene costruito in modo *distribuito*. Ogni utente crea il proprio segnale OFDM di ordine N ma solo con un sottoinsieme di portanti attive, quelle che sono state assegnate dal controllo di rete a quel certo utente, e tratta tutte le altre sottoportanti come virtuali. Per costruire questo segnale OFDMA, i terminali di rete

4G LTE (e anche 5G NR opzionalmente) utilizzano una variante di OFDM, la SC-FDMA, che esamineremo nelle prossime pagine.

8.5.2 Svantaggi delle tecnologie OFDM

Fino ad ora, tutte le funzionalità della nuova tecnologia sono state descritte come innovazioni positive, trascurando alcuni aspetti negativi che devono invece essere presi in considerazione.

8.5.2.1 Selettività nel tempo Creare un segnale multicarrier comporta allungare il tempo di simbolo dal valore originario T_s al valore del simbolo OFDM $T_M = T_s \cdot N$ (trascuriamo ancora per semplicità portanti virtuali e prefisso ciclico), e in modo molto rilevante perché N è spesso molto grande per contrastare la selettività in frequenza. Nel nostro modello di canale abbiamo sempre data per scontata la stazionarietà, cioè l'assenza di variazione del canale nel tempo del simbolo OFDM - dobbiamo adesso verificare questa condizione. Un esempio illuminante è costituito dal formato 8k del DVB-T per il quale T_M = 896 μ e $f_0 \approx 800$ MHz³. Se non vogliamo problemi di selettività temporale, dobbiamo assicurarci che il tempo di coerenza T_c del canale sia superiore a T_M . Prendendo un fattore intermedio pari a 20 nella definizione del tempo di coerenza (ref cohtime), la condizione di *assenza* di selettività temporale risulta

$$T_c > T_M \Rightarrow v \le \frac{c}{20f_0 T_M} = 20.9 \text{ m/s} = 75.3 \text{ km/h}$$
 (8.47)

implicando quindi un livello di mobilità piuttosto basso (pena la selettività frequenziale *e* temporale), che per una rete di broadcasting a ricevitori fissi *non* è un problema. Il caso è diverso per la rete 4G LTE con banda 20-MHz e @ $f_0 \simeq 2.1GHz$ con f'_{sc} =15 kHz, cosicché $T'_M = 1/f'_{sc}$ =66.7 µs. In questo caso, la condizione di non selettività è $v \leq 112$ m/s=405 km/h, senza nessuna limitazione di mobilità, neanche alle velocità dei treni ad alta velocità.

8.5.2.2 Sensibilità all'instabilità di frequenza Qualunque demodulatore I/Q è soggetto a una certa dissintonia di frequenza: le componenti I/Q della portante locale non oscillano esattamente alla frequenza della portante del segnale ricevuto, o a causa di uno spostamento Doppler (come in sez. 7.1.3), oppure più banalmente per imperfezioni e instabilità dell'HW (specie se a basso costo). In ogni caso, il segnale ricevuto y(t) porta con se' un termine oscillante residuo: $y(t) = x(t) \exp\{j2\pi\Delta\nu t\}$ dove $\Delta\nu$ è appunto la dissintonia di frequenza (frequency offset). Per comprendere l'effetto della dissintonia sul segnale OFDM, immaginiamo di attivare soltanto la sottoportante #k, in modo che

$$u(t) = u_k(t) = c^{(k)}[0]e^{j2\pi(\Delta\nu + k/T_M)t}$$

e consideriamo il ricevitore ottimo per tale sottoportante, come nella Fig. 8.6, ottenendo

$$z^{(k)}[0] = \frac{1}{T_M} \int_0^{T_M} y_k(t) e^{-j2\pi kt/T_M} dt = \frac{1}{T_M} c^{(k)}[0] \int_0^{T_M} e^{-j2\pi\Delta\nu t} dt$$
$$= c^{(k)}[0] \frac{e^{j2\pi\Delta\nu T_M} - 1}{j2\pi\Delta\nu T_M} = c^{(k)}[0] e^{j\pi\Delta\nu T_M} \operatorname{sinc}(\Delta\nu T_M)$$
(8.48)

³In realtà per il DVB-T è $T_{OFDM} = 896 \mu s$, ma in prima approssimazione trascuriamo la durate del prefisso ciclico.

Se adesso attiviamo anche la sottoportante # i, come già fatto nella (8.11), troviamo, nonostante la condizione di ortogonalità, la rinascita di un termine di interferenza della portante *i* sulla *k*:

$$I^{(i,k)}[0] = \frac{c^{(i)}[0]}{T_M} \int_0^{T_M} e^{j2\pi[(i-k)/T_M + \Delta\nu]t} dt = c^{(i)}[0] \frac{e^{j2\pi[(i-k)f_{sc}T_M + \Delta\nu]} - 1}{j2\pi[(i-k)f_{sc}T_M + \Delta\nu]}$$
$$= c^{(i)}[0] e^{j\pi(i-k+\Delta\nu T_M)} \operatorname{sinc}(i-k+\Delta\nu T_M) , \quad i \neq k$$
(8.49)

Osservando le due equazioni, possiamo notare due effetti. In primis, abbiamo un termine di ampiezza/fase che modifica il simbolo utile $c_0^{(k)}$ e che verrà gestito dal successivo equalizzatore nel dominio della frequenza (possiamo immaginare questo termine di nuovo come parte della H_k complessiva). Il termine di fase verrà compensato e non ha alcun effetto sulla ricezione; il termine di ampiezza dato dalla funzione sinc è < 1 e provoca un degrado (diminuzione) dell'SNR sulla variabile di decisione. Secondariamente, vediamo che il degrado è ulteriormente incrementato dal termine di *interferenza* (8.49) causato dalla dissintonia che distrugge l'ortogonalità: - la sottoportante #i (e tutte le altre, non esplicitamente incluse nel calcolo per semplicità, ma ugualmente presenti e che si sommano per i = 0, 1, ..., N - 1 $i \neq k$) interferiscono sulla #k, con un'interferenza che cresce al crescere di $\Delta \nu$.

Per rendere il degrado dovuto alla dissintonia accettabile, dobbiamo assicurarci che $\Delta \nu \ll 1/T_M$, in modo che la funzione sinc nella (8.48) sia praticamente uguale a 1, e la (le) funzione(i) sinc nella (8.49) è (siano) praticamente uguale(i) a 0. Quindi l'accuratezza necessaria nella compensazione della possibile dissintonia di frequenza deve essere una piccola frazione della spaziatura tra le sottoportanti. In un demodulatore dati convenzionale, con ragionamenti analoghi si trova invece che la degradazione dovuta alla dissintonia deve di nuovo essere molto più piccola dell'inverso del tempo di simbolo T_s (anziché del tempo di simbolo T_M): la stabilità di frequenza e/o la compensazione della dissintonia deve essere N volte migliore nel ricevitore OFDM rispetto a un ricevitore convenzionale, a parità di velocità di simbolo $1/T_s$. Pertanto, la funzione di stima/correzione della frequenza portante nel demodulatore OFDM risulta essere *cruciale*.

8.5.2.3 II PAPR e il formato SC-FDMA (OFDM precodificata) Un inconveniente finale della tencologia OFDM deriva dalle proprietà del segnale multicarrier. Per sua stessa natura, il segnale trasmesso è una combinazione lineare di un numero elevato (N) di componenti indipendenti, cioè i vari segnali dati modulati in banda base $y_k(t)$. Per questo motivo, le statistiche delle componenti I/Q del segnale trasmesso $x_{OFDM}(t)$ tendono ad essere *Gaussiane* in virtù del teorema limite centrale - l'ampiezza del segnale I/Q tende quindi a una pdf di Rayleigh. La Fig. 8.21 mostra il diagramma e le forme d'onda I/Q di un segnale OFDM con le specifiche indicate nella didascalia, dove non si riscontra la presenza di alcuna costellazione dati, anzi, il segnale sembra "analogico" con una certa banda.

Se la potenza media del segnale è P_{RF} , la pdf dell'*ampiezza* $a(t) = |x_{OFDM}(t)|$ è appunto di Rayleigh,

$$f_A(a) = \frac{a}{P_{RF}} e^{-\frac{a^2}{2P_{RF}}} , \ a \ge 0$$
(8.50)

e quella della potenza $p(t) = |x_{OFDM}(t)|^2/2$ è esponenziale:

$$f_P(p) = \frac{1}{P_{RF}} e^{-p/P_{RF}} , \ p \ge 0$$
(8.51)



Figura 8.21 Forme d'onda I/Q (a) e diagramma I/Q di un segnale OFDM con N = 2048

Anche se la potenza media è P_{RF} , poichè il segnale è Gaussiano (ampiezza di Rayleigh), la sua ampiezza di picco (cioè massima) è in teoria infinita. Ciò non può essere vero nella realtà - si tratta infatti di un paradosso introdotto dal teorema-limite centrale, e che è vero solo asintoticamente.

Esempio 8.46

Possiamo calcolare *esattamente* la potenza di picco per il segnale OFDM (8.35). Il valore di picco del segnale OFDM, per una costellazione M-QAM, si ottiene infatti per t = 0 (quando tutte le oscillazioni delle sottoportanti valgono 1 + j0) e quando tutti i simboli di costellazione sono casualmente pari a $c^{(k)}[0] = (\sqrt{M} - 1) + j(\sqrt{M} - 1) \forall k$ (o il suo opposto). In tal caso, tenendo conto che $C_2 = 2(M - 1)/3$, la potenza teorica di picco è pari a

$$P_{peak} = |x_{OFDM}(0)|^2 / 2 = 3NP_{RF} \frac{(\sqrt{M} - 1)^2}{M - 1} \simeq 3N \cdot P_{RF}$$
(8.52)

e quindi il il cosiddetto PAPR (*peak to average power ratio*) P_{peak}/P_{RF} è pari circa a 3N (per grandi costellazioni). Se N = 2048, PAPR=38 dB (elevatissimo). La relazione (8.52)

si può anche esprimere come

$$PAPR_{ofdm} = N \cdot 3 \frac{(\sqrt{M} - 1)^2}{M - 1} = N \cdot PAPR_{qam}$$
(8.53)

Il PAPR non è infinito come nell'approssimazione Gaussiana, ma è comunque molto elevato e poco utile: la probabilità che tutti i simboli di costellazione sulle sottoportanti siano identici e pari al simbolo di massimo modulo M-QAM è infinitesimamente bassa, quindi nella pratica il valore di picco teorico non si presenterà praticamente *mai*.

Una migliore definizione di potenza di picco (perlomeno più utile e realistica), e quindi di PAPR, si trova identificando quel valore P_{peak} la cui probabilità teorica di essere ecceduto è sufficientemente bassa, diciamo 10^{-5} - in pratica questo valore diventa un livello che il segnale non raggiunge pressoché mai, e si ricava in funzione della potenza media:

$$\int_{P_{peak}}^{\infty} \frac{1}{P_{RF}} e^{-p/P_{RF}} dp = 10^{-5} \Rightarrow P_{peak} = 5 \ln 10 P_{RF} = 11.5 P_{RF}$$
(8.54)

In questo caso PAPR=11.5 o 10.6 dB. Questo rapporto relativamente alto rispetto a formati convenzionali tipo QAM (per i quali vale 3-4 dB) rende inefficiente l'amplificatore radio del trasmettitore, un problema particolarmente sentito per terminali portatili nei quali la durata della batteria deve essere massimizzata, e quindi ogni inefficienza va eliminata.

In effetti, se il PAPR del segnale è alto, l'amplificatore di potenza radio nel trasmettitore deve consentire un "margine" nell'ampiezza del segnale in modo che la dinamica di ingresso/uscita sia sufficiente ad accogliere un segnale che ha sì una certa potenza media P_{RF} , ma che occasionalmente si spinge fino a P_{RF} · PAPR, cioè più di 10 dB oltre la media. Nella Fig. 8.22 vediamo la caratteristica di ingresso/uscita dell'amplificatore radio, e mostriamo chiaramente la differenza tra il punto di funzionamento medio a P_{RF} e il valore di potenza di picco necessario P_{peak}, che per OFDM è molto grande: il rischio è che l'amplificatore si trovi a lavorare in zona nonlineare e introduca forti distorsioni - una situazione che invece, per segnali convenzionali a dinamica ridotta non si produce, come mostrato ancora in Figura. Dunque, per evitare distorsioni nonlineari che si manifestano come saturazione (taglio) dei livelli alti di segnale in uscita, è necessario usare un amplificatore di potenza con una dinamica molto estesa rispetto al punto di lavoro medio, e che necessita di un livello di alimentazione altrettanto alto "sprecando" potenza di alimentazione anche quando la potenza istantanea del segnale è bassa. In tale condizioni, l'efficienza dell'amplificatore in termini di rapporto tra il la potenza RF effettivamente trasmessa e potenza di alimentazione diventa bassa; questa inefficienza non costituisce un particolare problema per il trasmettitore di downlink nella stazione radio base (connessa alla rete elettrica), é invece deleteria per il trasmettitore uplink in un terminale portatile, riducendo di molto la vita della batteria senza un effettivo riscontro di potenza radio utile.

Sono state studiate molte tecniche per ridurre il PAPR. La soluzione applicata nelle reti cellulari 4G LTE e 5G NR per la trasmissione uplink è il cosiddetto formato SC-FDMA (Single Carrier FDMA), una opportuna variante di OFDM. Nell'SC-FDMA, i simboli di costellaizone, *prima* di subire la IDFT (8.19) vengono *precodificati* con una legge opportuna che porterà ad segnale OFDM a PAPR ridotto. I valori complessi ottenuti dalla precodifica vengono poi mappati sulle varie sottoportanti assegnate a un certo utente in modalità OFDMA, ma non sono più valori di costellazione (e questo è il motivo per cui il PAPR è



Figura 8.22 PAPR e amplificatore di potenza radio

ridotto). Nel ricevitore, viene eseguita l'elaborazione inversa, facendo seguire la precodifica inversa alla demodulazione FFT.

Qual è la particolare precodifica usata da 4G/5G (e che viene chiamata transform precoding)? I simboli di costellazione vengono raggruppati a sottoblocchi di lunghezza M (ad eesmpio, M = 64) e come precodifica viene applicato un semplice algoritmo di DFT di ordine pari a M. La spiegazione intuitiva è semplice: poiché stiamo calcolando uno dietro l'altro un algoritmo prima di DFT e poi di IDFT, possiamo aspettarci di ritrovare i simboli di costellazione nel tempo, come in una modulazione tradizionale, e quindi di tornare ad avere il PAPR (basso) di tale formato, senza perdere la struttura multiportante del segnale trasmesso sul canale con la relativa facilità di stima del canale/equalizzazione in frequenza.

Le cose sono in realtà un po' più complicate perché i due algoritmi sono di ordine diverso, e quindi non sono l'uno inverso dell'altro (nel qual caso tutto quanto sarebbe... inutile), e le portanti allocate ad un utente non sono in generale contigue. Tralasciando però questa seconda eventualità, supponiamo di inviare M simboli di costellazione $c^{(n)}[0]$, n = 0, 1, ..., M - 1 (estratti dagli N di tutto il blocco OFDMA) nel tempo di simbolo OFDM m = 0 - d'ora in poi, ometteremo questa indicazione di m = 0 per semplicità di notazione. I simboli precodificati saranno

$$C^{(k)} = \frac{1}{M} \sum_{n=0}^{M-1} c^{(n)} e^{-j2\pi nk/M} , \quad k = 0, 1, ..., M-1$$

e quindi, supponendo di allocare le prime M sottoportanti delle N disponibili nel pettine OFDM, il segnale SC-FDMA sarà, per $0 \le t < T_M$,

$$x_{SF}(t) = \sum_{k=0}^{M-1} C^{(k)} e^{j2\pi kt/T_M} = \frac{1}{M} \sum_{k=0}^{M-1} \sum_{n=0}^{M-1} c^{(n)} e^{-j2\pi nk/M} e^{j2\pi kt/T_M}$$
$$= \frac{1}{M} \sum_{n=0}^{M-1} c^{(n)} \sum_{k=0}^{M-1} e^{j\frac{2\pi k}{T_M} \left(t - n\frac{T_M}{M}\right)} = \sum_{n=0}^{M-1} c^{(n)} \frac{1 - e^{j\frac{2\pi M}{T_M} \left(t - n\frac{T_M}{M}\right)}}{M \left(1 - e^{j2\pi \left(t - n\frac{T_M}{M}\right)/T_M}\right)}$$



Figura 8.23 Kernel di Dirichlet

$$=\sum_{n=0}^{M-1} c^{(n)} e^{j\frac{2\pi(M-1)}{2T_M} \left(t-n\frac{T_M}{M}\right)} \frac{\operatorname{sinc}\left(M\left(t-n\frac{T_M}{M}\right)/T_M\right)}{\operatorname{sinc}\left(\left(t-n\frac{T_M}{M}\right)/T_M\right)}$$
$$\simeq e^{j\frac{2\pi(M-1)t}{2T_M}} \sum_{n=0}^{M-1} (-1)^n c^{(n)} \operatorname{Dir}\left(\frac{1}{T_M} \left(t-n\frac{T_M}{M}\right), M\right)$$
(8.5)

dove Dir(x, M) è il cosiddetto kernel di Dirichlet

$$\operatorname{Dir}(x,M) \stackrel{\triangle}{=} \frac{\sin(\pi M x)}{M \sin(\pi x)} = \frac{\operatorname{sinc}(M x)}{\operatorname{sinc}(x)}$$
(8.56)

che rappresentiamo in Fig. 8.23 - praticamente la ripetizione a segni alterni di un impulso sinc pulse. Nella (8.56), essendo $0 \le t < T_M$, interessa soltanto la parte "principale" dell'impulso.

Dopo questo lungo calcolo, possiamo giustificare il nome SC-FDMA: dalla (8.55), notiamo che il segnale inviato "in forma di OFDM" è in realtà costituito dagli M simboli di costellazione originali $c^{(n)}$, ciascuno associato ad un impulso Dir nel tempo, e ciascuno trasmesso al tempo nT_M/M : in pratica, il tempo di simbolo OFDM viene ripartito in Mslot, ciascuno dei quali viene utilizzato per trasmettere uno degli M simboli di costellazione, uno dopo l'altro nel tempo. Questo pacchetto temporale di dati viene poi "piazzato" a cavallo di un'unica (sotto)portante alla frequenza $(M - 1)/(2T_M)$, a metà strada cioè tra le frequenze 0 e $(M - 1)/T_M$ allocate alle sottoportanti OFDM di quell'utente. La portante di questo utente è quindi una, giustificando la qualificazione di single-carrier del formato. Gli altri utenti della cella utilizzeranno altre sottobande, in modo da implementare sostanzialmente un access multiplo di tipo FDMA.

Il motivo per cui SC-FDMA riduce il PAPR è chiaro: il formato è fondamentalmente una modulazione convenzionale a *singola portante*, e quindi con basso PAPR, generata attraverso l'utilizzo di un modulatore OFDM che consente comunque di eseguire un'ottima equalizzazione di canale . La differenza rispetto all'OFDM convenzionale si può anche immaginare come segue: nella OFDM, i dati vengono trasmessi "in parallelo" entro il



Figura 8.24 Confronto tra OFDM e SC-FDMA (M=4)

(lungo) tempo di simbolo OFDM sulle varie sottoportanti attive. Dopo la modulazione sulle sottoportanti, i simboli modulati si combinano in modo distruttivo o o costruttivo in base al loro particolare valore di costellazione, generando un segnale finale di ampiezza rispettivamente da molto bassa a molto alta. Lo SC-FDMA presenta invece i simboli dell'utente uno dietro l'altro nel tempo, tutti quanto con la stessa ampiezza - rappresentiamo questa differenza nella Fig. 8.24.

8.6 Multicarrier meets Information Theory: Le tecnologie DSL e la capacità del canale Gaussiano colorato

La tecnologia attualmente più utilizzata per la connessione ad Internet "a larga banda" (cioè ad alta capacità) all'abbonato residenziale, coprendo efficientemente e a basso costo l'*ultimo chilometro* della rete, è costituita dalla famiglia degli standard digitali per la rete d'abbonato xDSL, ove DSL sta per Digital Subscriber Line, e x può assumere vari valori, caratteristici di varie tecnologie con varie capacità: ADSL, VDSL in ordine di disponibilità commerciale. Nei prossimi paragrafi verranno descritte le caratteristiche principali dei sistemi xDSL ed in particolare verrà presentato l'aspetto information-theoretic legato al formato DMT (Discrete Multi Tone) per la massimizzazione della capacità del mezzo trasmissivo.

8.6.1 Architettura di un sistema xDSL per l'ultimo miglio

Le tecnologie xDSL per la rete d'accesso hanno lo scopo di fornire connettività Internet all'utente finale residenziale (o commerciale con modesti requisiti) attraverso il doppino telefonico della preesistente rete telefonica. La connessione copre come già accennato l'ultimo chilometro - la distanza media tra la postazione fissa dell'abbonato e il più vicino puto di presenza della rete di trasporto di un operatore: un ufficio centrale o un armadio di commutazione. In particolare, il requisito è quello di fornire un collegamento ad una velocità fino al centinaio di Mbit/s mantenendo contemporaneamente inalterato il servizio telefonico tradizionale (POTS, Plain Old Telephone Service).

La versione più popolare e più a basso costo della famiglia xDSL è la ADSL, cioè Asymmetric DSL, per la quale la capacità downstream è maggiore di quella upstream. L'architettura generale del sistema xDSL è quella di Fig. 8.25. L'abbonato ha una connessione



Figura 8.25 Architettura di un collegamento xDSL

digitale cui è collegato un terminale d'utente, tipicamente un router/access point WiFi, e una connessione analogica per un telefono tradizionale. Analizzando il flusso in upstream, le due connessioni sono accoppiate attraverso uno Splitter, composto da un filtro passa-basso che seleziona la parte inferiore dello spettro per la connessione analogica ed un filtro passaalto che seleziona la parte superiore per la connessione digitale. Quest'ultima si avvale di una unità di trasmissione ADSL remota (ATU-R, ADSL Transmission Unit-Remote side, in parole povere il *modem*) che genera il segnale analogico modulato da accoppiare col segnale analogico telefonico. I due segnali vengono inviati su doppino in centrale, ove il segnale telefonico viene disaccoppiato (tramite uno Splitter lato centrale) e inviato a un commutatore di rete telefonica tradizionale(PSTN, Public Switched Telephone Network), mentre il segnale modulato digitale viene demodulato tramite un'unità di trasmissione ADSL in centrale (ATU-C, il modem di centrale) e inviato alla rete di trasporto digitale a larga banda. In particolare, i flussi provenienti dai diversi ATU-C vengono multiplati dal DSLAM (Digital Subscriber Line Access Multiplexer) che riceve i flussi e li trasmette sulla rete di trasporto mediante fibra ottica. Dunque, oltre al DSLAM di centrale, l'unica modifica all'infrastruttura esistente consiste negli apparati ATU-R e ATU-C da installare rispettivamente sul sito d'abbonato e in centrale (o in un armadio intermedio se la centrale è troppo lontana).

8.6.2 Duplexing e Modellistica di Canale per ADSL

La trasmissione di un flusso ad alta capacità su un mezzo trasmissivo come il doppino pone alcune questioni tecniche di difficile risoluzione. In primo luogo si deve risolvere la coesistenza tra linea digitale e analogica. Come già accennato, la questione è risolta con un multiplexing a divisione di frequenza (FDM) schematizzato in Fig. 8.26: il servizio telefonico usa la sua propria banda base, mentre la connessione digitale viene modulata

306



Figura 8.26 Piano frequenziale dell'ADSL

su di una banda passante, distanziata dalla banda telefonica da una banda "di guardia" (guard band). Inoltre, il duplexing della connessione digitale avviene ancora a divisione di frequenza (FDD), con la banda (più stretta) inferiore dedicata all'upstream e la banda (più larga) superiore dedicata al downstream.

Chiariti questi dettagli, devono ancora essere presi in considerazione due problemi inerenti alla struttura fisica del doppino: i) la risposta in frequenza del doppino telefonico e ii) i disturbi cui è soggetto. Per quanto riguarda il calcolo della risposta in frequenza, il doppino può essere modellato come un linea di trasmissione a parametri distribuiti le cui caratteristiche ingresso-uscita dipendono quindi dalla lunghezza (nonché dal tipo di doppino, ad esempio dalla sezione del cavetto). Su di una lunghezza tipica di un chilometro, la risposta in ampiezza è comunque assai variabile, e si pone quindi un problema di selettività in frequenza del canale.

Un modello approssimato della risposta in ampiezza del doppino in funzione della lunghezza L del collegamento è

$$|H(f;L)|_{dB} = -\alpha \cdot L \cdot \sqrt{f} \tag{8.57}$$

che, oltre alla consueta attenuazione esponenziale (lineare in dB) con la lunghezza del cavo, mostra un rapido decadimento con la frequenza, e ove α è una opportuna costante che tiene conto delle caratteristiche fisiche del cavetto, ad es. la sezione del doppino stesso. Il canale risulta *selettivo* sulla banda di interesse (la selettività si ha a partire da qualche kHz) ed il segnale risulta quindi distorto, diciamo pure *fortemente* distorto quando si considera una banda di 1-, 2 o 12 MHz (!) dei servizi rispettivamente ADSL, ADSL2+, VDSL.

Il doppino, come ogni mezzo fisico, è anche soggetto a disturbi; piuttosto che il rumore AWGN del ricevitore, ciò che in definitiva limita le prestazioni del collegamento ADSL è una particolare forma di interferenza: la *diafonia (crosstalk)*. La diafonia è causata dal'accoppiamento tra i diversi cavetti telefonici raggruppati in un unico cavo multiconduttore (*binder*) che creano mutua interferenza.

La diafonia è dovuta alla presenza di più coppie ritorte non schermate (doppini) raggruppate nel cavo sotterraneo multiconduttore che collega le utenze con l'armadio o la centrale, il cosiddetto *binder* rappresentato in Fig. 8.27. Il doppino viene "intrecciato" (ritorto) su se stesso per evitare il più possibile tale interferenza, ma non è schermato dalle interferenze come ad esempio un cavo coassiale. Nella figura mettiamo in evidenza due diversi meccanismi di diafonia: la far-end crosstalk o FEXT causata da trasmettitori *lontani* dal



Figura 8.27 NEXT e FEXT sul binder xDSL

ricevitore, e la near-end crosstalk o NEXT, causata invece da trasmettitori vicini al ricevitore. La prima è causata dai segnali che viaggiano nello stesso verso di quello del doppino di riferimento, mentre la NEXT nasce a causa dei segnali che viaggiano in verso opposto. Per la caratterizzazione della diafonia, osserviamo che i vari doppini di un singolo cavo (qualche decina) generano interferenze statisticamente indipendenti (poiché provenienti da comunicazioni indipendenti); la somma dei vari singoli disturbi produce la NEXT e la FEXT totali che (in virtù del teorema-limite centrale) possono essere modellate come processi di rumore Gaussiano colorato cioè con densità spettrale di potenza non piatta. La "colorazione" del disturbo è dovuta alla selettività del fenomeno dell'accoppiamento tra conduttori, e comunque alle caratteristiche spettrali non uniformi dei segnali interferenti. In particolare, il fenomeno di accoppiamento che genera interferenza è di tipo capacitivo, e quindi le componenti a frequenza più alta dello spettro interferiscono maggiormente perché vedono una impedenza d'accoppiamento minore. Analisi empiriche del fenomeno consigliano di modellare la densità spettrale di potenza della NEXT N(t) come segue:

$$S_N(f) \cong \alpha_N \cdot f^{3/2} S_x(f) \tag{8.58}$$

ove α_N è una costante dipendente dal numero di doppini nel cavo, e $S_x(f)$ è lo spettro di potenza del segnale ADSL interferente. Un modello simile è anche adottato per la FEXT:

$$S_F(f) \cong \alpha_F \cdot f^2 \cdot L \cdot |H(f;L)|^2 S_x(f) \tag{8.59}$$

ove α_F è un coefficiente di proporzionalità. La presenza di H(f; L) (8.57), contenente il termine di attenuazione già discusso, rende appunto la FEXT di norma meno limitativa rispetto alla NEXT. Nella FEXT infatti il segnale interferito (chiamato vittima nel gergo ADSL) è su tutto il cavo dello stesso ordine di grandezza del segnale interferente, mentre nella NEXT il segnale vittima è debole (nelle vicinanze del ricevitore, essendo stato attenuato dal doppino) laddove il segnale interferente è forte (cioè nelle vicinanze del trasmettitore interferente).

Dalla discussione sulle diafonie si capisce che il flusso upstream è molto soggetto alla NEXT: in centrale, i doppini vengono raccolti nel binder immediatamente all'uscita degli ATU-C, e quindi nella prima parte del binder la diafonia dei (molti) segnali di downstream ad alta potenza sui segnali upstream a bassa potenza è rilevante. Per mitigare questo fenomeno, il downstream viene trasmesso sulla banda inferiore del duplex a divisione di frequenza

308

(vedi Fig. 8.26), dove l'attenuazione del doppino è minore e dove la NEXT (8.58) è pure minore, perché minore è l'accoppiamento capacitivo tra i doppini. Nel downstream, la NEXT è ridotta perché nelle prime decine di metri del collegamento i doppini di utente sono *separati* e non danno luogo a diafonia - quando vengono inseriti nel binder la NEXT che si produce è minore. Comprendiamo anche perché la banda dell'upstream resta comunque minore di quella del downstream: allargandola, si includerebbero componenti frequenziali tropo troppo soggette a NEXT che non porterebbero un apprezzabile incremento di capacità, come vedremo più dettagliatamente in seguito, diminuendo viceversa apprezzabilmente e inutilmente la capacità del donwstream. È questo il motivo per cui il servizio risulta giocoforza *asimmetrico*.

Oltre alla diafonia, altri disturbi tipici del doppino sono il rumore impulsivo e le interferenze radio (doppino non schermato), nonché i disadattamenti dovuti a trasformatori d'accoppiamento, e infine il rumore del ricevitore. Tutti questi fenomeni sono comunque di entità inferiore alla diafonia. Dal punto di vista del link, abbiamo dunque un problema di *canale distorcente* e di *rumore additivo gaussiano colorato*, in un contesto in cui è essenziale ottenere un'alta efficienza nell'utilizzazione del mezzo fisico. Il doppino, nato e ottimizzato per trasportare segnali telefonici analogici di 4 kHz di banda, viene improvvisamente chiamato a trasportare segnali digitali di banda ben più ampia per i quali non è intrinsecamente adatto - una vera *sfida tecnologica*. È interessante affrontare la questione dal punto di vista della teoria dell'informazione per ricavare la capacità di canale, e possibilmente trarre indicazione su come implementare una tecnologia che avvicini tale capacità - il percorso intrapreso da John Cioffi, pioniere dell'ADSL, nei primissimi anni '90.

8.7 Capacità del canale Gaussiano "colorato"

Nella sezione 4.5.2 abbiamo ricavato la formula di Shannon sulla capacità del canale Gaussiano con densità spettrale di rumore uniforme su di una banda limitata *B*:

$$\mathcal{C} = \frac{1}{2T_s} \cdot \log_2\left(1 + \frac{\sigma_X^2}{\sigma_W^2}\right) = B \cdot \log_2\left(1 + \text{SNR}\right) = B \cdot \log_2\left(1 + \frac{P}{N_0 B}\right) \text{ [bit/s]} (8.60)$$

dove SNR è il rapporto segnale-rumore all'uscita del canale, $P = \sigma_X^2$ denota la potenza (varianza) del segnale trasmesso e $S_w(f) = N_0/2$ è la densità spettrale di potenza del rumore Gaussiano additivo bianco, per cui $\sigma_W^2 = N_0/2 \cdot 2B = N_0 \cdot B$ è la potenza (varianza) del rumore di canale misurata sulla banda passante B. Evidentemente questo risultato non si applica al problema dll'ADSL in cui i) il segnale d'ingresso viene distorto dal canale di comunicazione, e ii) il disturbo (la NEXT N(t))è colorato (ACGN, Additive Colored Gaussian Noise), cioè ha densità spettrale di potenza (d.s.p.) $S_N(f)$ variabile nella banda del segnale. Dobbiamo cercare di estendere la relazione (8.60) a quest'ultimo caso.

La prima questione è facilmente risolta. Infatti la situazione appena descritta, rappresentata in Fig. 8.28 (a), è equivalente a quella di Fig. 8.28 (b), in cui il rumore N'(t) ha densità spettrale di potenza

$$S'_{N}(f) = \frac{S_{N}(f)}{|H(f)|^{2}}$$
(8.61)

e naturalmente conserva statistiche gaussiane. Se il filtro è invertibile (come accade in generale per i sistemi fisicamente realizzabili), la capacità di canale calcolata all'uscita del filtro, è esattamente la stessa di quella calcolata all'*ingresso* del filtro: una trasformazione invertibile non aggiunge né toglie informazione. Dal punto di vista pratico, se si suppone di



Figura 8.28 Canale ACGN e suo equivalente

Figura 8.29 Suddivisione in sottobande della banda B

conoscere la risposta in frequenza del canale H(f), può essere inserito un filtro invertente, cioè un equalizzatore che compensa perfettamente la distorsione di canale senza alterare la capacità. Ci riconduciamo allora alla situazione di Fig. 8.28 (c), in cui il canale non è distorcente, ma il rumore è colorato con densità spettrale di potenza $S'_N(f) = S_N(f)/|H(f)|^2$. In questo modo, la distorsione di canale è stata "inglobata" nella dsp del rumore colorato equivalente $S'_N(f)$.

Resta un secondo problema da risolvere, cioè quello del rumore (fortemente) colorato, in particolare molto più intenso alle alte frequenze a causa i) dello spettro crescente di $S_N(f)$ (8.58), e ii) della risposta in frequenza fortemente passa-basso H(f) (8.57). Possiamo cercare di calcolare la capacità di Shannon del canale ACGN cercando di ricondursi al caso già noto di rumore con densità spettrale di potenza piatta. Immaginiamo di suddividere l'intera banda B a disposizione del collegamento in un numero N di sottobande adiacenti equispaziate, ciascuna di ampiezza $\Delta f = B/K$ (Fig. 8.29). Possiamo allora suddividere la nostra trasmissione di un singolo flusso informativo alla velocità R_b in K sotto-flussi trasmessi in parallelo e indipendentemente l'uno dall'altro, usando per ciascuno un segnale $x_k(t)$ con modulazione passa-banda a banda rigorosamente limitata nella k-esima sottobanda, $[k\Delta f, (k+1)\Delta f), k = 0, ..., K - 1$, e ricevuti previo filtraggio (ideale) nella stessa banda. Se K è sufficientemente grande, in ciascuna di queste sottobande il segnale trasmesso $x_k(t)$ viene disturbato in ricezione da un rumore Gaussiano $N'_k(t)$ con densità spettrale di potenza sostanzialmente piatta e pari a

$$S'_{N_{k}}(f) \cong S'_{N}(k\Delta f) = S_{N}(k\Delta f)/|H(k\Delta f)|^{2}$$
(8.62)

I disturbi sui vari sottocanali $N'_k(t)$ sono processi Gaussiani passa-banda bianchi con d.s.p. diverse da zero su bande non sovrapposte. Questi processi sono dunque incorrelati e, poiché Gaussiani, anche indipendenti. Questo garantisce che la capacità totale è data dalla somma delle capacità dei singoli sottocanali. La capacità del sottocanale k-esimo è

$$\mathcal{C}_{k} = \Delta f \cdot \log\left(1 + \mathrm{SNR}_{k}\right) = \Delta f \cdot \log\left(1 + \frac{P_{k}}{\sigma_{N_{k}'}^{2}}\right) =$$
$$= \Delta f \cdot \log\left(1 + \frac{P_{k}}{2 \cdot \Delta f \cdot S_{N}'(k\Delta f)}\right) = \Delta f \cdot \log\left(1 + \frac{2 \cdot \Delta f \cdot S_{N}(k\Delta f)}{2 \cdot \Delta f \cdot S_{N}'(k\Delta f)}\right)$$
$$= \Delta f \cdot \log\left(1 + \frac{S_{X}(k\Delta f)|H(k\Delta f)|^{2}}{S_{N}(k\Delta f)}\right) \text{ [bit/s]} \tag{8.63}$$

310
La capacità totale sarà dunque

$$\mathcal{C} = \sum_{k=0}^{K-1} \mathcal{C}_k = \sum_{k=0}^{K-1} \Delta f \cdot \log\left(1 + \frac{P_k |H(k\Delta f)|^2}{2S_N(k\Delta f)\Delta f}\right)$$
$$= \sum_{k=0}^{K-1} \Delta f \cdot \log\left(1 + \frac{S_X(k\Delta f) |H(k\Delta f)|^2}{S_N(k\Delta f)}\right) \text{ [bit/s]}$$
(8.64)

dove $S_X(f)$ è la d.s.p. del segnale inviato X(t). Se il numero delle sottobande è molto elevato, Δf tende a 0, e la capacità tende quindi a

$$C_{ACGN} = \int_{0}^{B} \log_2\left(1 + \frac{S_x(f)|H(f)|^2}{S_N(f)}\right) df$$
(8.65)

Questo risultato teorico generalizza la formula della capacità con rumore bianco, ed è importante di per sé. È ancora più importante perché suggerisce la tecnologia, di tipo *multicarrier*, che permette di realizzare una connessione con bit-rate molto vicino alla capacità. Il procedimento di suddividere un unico flusso digitale in molti flussi ortogonali (indipendenti) su sottobande diverse è quello che ha condotto alla definizione del formato OFDM (8.12) che qui verrà ripreso con le dovute differenze.

Cechiamo di semplificare la (8.65), o meglio la sua versione "finita" multiportante (8.63):

$$C = \sum_{k=0}^{K-1} C_k = \Delta f \sum_{k=0}^{K-1} \log (1 + \text{SNR}_k) = \Delta f \log \prod_{k=0}^{K-1} (1 + \text{SNR}_k) \quad [\text{bit/s}] \quad (8.66)$$

Se i rapporti segnale-rumore sulle sottobande sono abbastanza alti, $1 + SNR_k \cong SNR_k$ e quindi

$$\mathcal{C} = \Delta f \log \prod_{k=0}^{K-1} \text{SNR}_k = \frac{B}{K} \log \prod_{k=0}^{K-1} \text{SNR}_k = B \log \text{SNR}_{geo} \cong B \log(1 + \text{SNR}_{geo})$$
(8.67)

dove SNR_{geo} è la *media geometrica* dei rapporti segnale-rumore sulle varie sottoportanti. La capacità equivalente del collegamento multiportante ACGN è allora quella di un collegamento fittizio (monoportante) AWGN sulla stessa banda totale e con un rapporto-segnale rumore pari a SNR_{geo} . Esprimendo gli SNR in dB, la media geometrica espressa in dB equivale alla media aritmetica dei valori espressi in dB, per cui

$$\mathcal{C} \cong \frac{B \cdot \left(\frac{1}{K} \sum_{k=0}^{K-1} \mathrm{SNR}_k |_{dB}\right)}{3} [\mathrm{bit/s}]$$
(8.68)

8.7.1 Massimizzazione della capacità del canale Gaussiano "colorato"

La formula (8.65) della capacità del canale Gaussiano con rumore colorato è valida per un certo spettro di potenza del segnale $S_x(f)$. Mentre lo spettro del rumore $S_N(f)$ è imposto dallo stato del canale e non è modificabile, lo spettro del segnale può essere liberamente

scelto in fase di progetto o addirittura adattato in tempo reale dal modem ADSL. Dunque, come possiamo distribuire in modo ottimale (o più precisamente *allocare*) la potenza totale di segnale P_{tot} di X(t) sulla banda B? Ragionando dal punto di vista pratico, e cioè sulla versione "finita" multiportante della tecnologia, come possiamo determinare i vari P_k , in modo che la capacità totale C del collegamento risulti massima?

Indichiamo per semplificare la notazione $\sigma_k^2 \stackrel{\triangle}{=} \sigma_{N'_k}^2$. Il problema da risolvere è allora: Trovare

$$P_k$$
, $k = 0, ..., K - 1$

tali che

$$\mathcal{C} = \max_{P_0, P_1, \dots, P_{K-1}} \Delta f \sum_{k=0}^{K-1} \log\left(1 + \frac{P_k}{\sigma_k^2}\right)$$

con il vincolo

$$\sum_{k=0}^{K-1} P_k = P_{tot}$$

Senza il vincolo della potenza totale massima, il problema non ha senso perché la capacità potrebbe crescere indefinitamente facendo crescere indefinitamente le potenze P_k . Per risolvere il problema di massimo vincolato, costruiamo come sempre la Lagrangiana da massimizzare

$$\mathcal{L}(P_0, P_1, \dots, P_{K-1}; \lambda) \stackrel{\triangle}{=} \Delta f \sum_{k=0}^{K-1} \log_2 \left(1 + \frac{P_k}{\sigma_k^2} \right) + \lambda \left(\sum_{k=0}^{K-1} P_k - P_{tot} \right)$$
(8.69)

Derivando rispetto a P_k e uguagliando a 0 si ottiene

$$\frac{\Delta f \log_2 e}{1 + \frac{P_k}{\sigma_k^2}} \frac{1}{\sigma_k^2} + \lambda = 0 \quad , \quad k = 0, 1, \dots, K - 1$$
(8.70)

ovvero

$$P_k + \sigma_k^2 = -\frac{\Delta f \log_2 e}{\lambda} , \quad k = 0, 1, \dots, K - 1$$
 (8.71)

Sommando su k tutte le precedenti equazioni si ricava anche

$$P_{tot} + \sigma_{tot}^2 = -K \frac{\Delta f \log_2 e}{\lambda} \quad \to \quad -\frac{\Delta f \log_2 e}{\lambda} = \frac{P_{tot} + \sigma_{tot}^2}{K}$$
(8.72)

dove abbiamo sfruttato il vincolo $\sum P_k = P_{tot}$ e dove $\sigma_{tot}^2 \stackrel{\triangle}{=} \sum \sigma_k^2$ è la potenza totale di rumore su tutte le sottobande. Utilizzando (8.72) nella (8.71) si conclude che

$$P_k + \sigma_k^2 = \bar{P} \stackrel{\triangle}{=} \frac{P_{tot} + \sigma_{tot}^2}{K}$$
(8.73)

In versione "continua" (esercizio per il lettore/lettrice) otteniamo

$$S_x(f) + S'_N(f) = \bar{S} \ , \ \bar{S} \stackrel{\triangle}{=} \frac{1}{B} \int_0^B \left(S_x(f) + S'_N(f) \right) df = \frac{P_{tot} + \sigma_{tot}^2}{B}$$
(8.74)



Figura 8.30 Interpretazione del criterio a riempimento d'acqua

che, dato $S'_N(f)$, individua la forma dello spettro di potenza $S_x(f)$ del segnale trasmesso che massimizza la capacità di canale con rumore ACGN, con il vincolo sulla potenza P_{tot} di segnale fissata.

8.7.2 Criterio di water-filling: interpretazione della massimizzazione della capacità del canale Gaussiano "colorato"

Interpretiamo per prima cosa il risultato riguardo la particolare forma dello spettro di potenza del segnale trasmesso trovata dalla massimizzazione della capacità del canale ACGN:

$$\begin{cases} S_x(f) = \bar{S} - S'_N(f) = \bar{S} - \frac{S_N(f)}{|H(f)|^2} & S'_N(f) < \bar{S} \\ S_x(f) = 0 & S_{W'}(f) \ge \bar{S} \end{cases}$$
(8.75)

che corrisponde a una densità spettrale *totale*, cioè di segnale + rumore, al ricevitore pari a $S_{tot}(f) = S_x(f) + S'_N(f) = \overline{S}$ cioè uno spettro totale piatto sulla banda del canale, quando il rumore non eccede il valore \overline{S} . La situazione è quella rappresentata in Fig. 8.30, e la particolare forma del segnale trasmesso è quella corrispondente al problema del "riempimento d'acqua" (water-filling) di un "acquario" che ha un fondale di sabbia con altezza variabile e pari a quella dello spettro del rumore modificato N'(t). Il criterio di massimizzazione della capacità prescrive quindi di utilizzare un segnale la cui d.s.p. crei una d.s.p. totale sulla banda di interesse, piatta cioè come il livello dell'acqua nell'acquario. La potenza di segnale trasmessa è rappresentata dall'area grigia della figura 8.30, cioè dalla "quantità d'acqua" versata nell'acquario, mentre la potenza di rumore è rappresentata dalla quantità di sabbia sul fondale (area bianca nella figura).

La costante S (o P in versione finita) dipende dalle potenza di rumore $P_{W'}$ e dalla potenza di segnale disponibile P_{tot} . Nei casi in cui il disturbo è particolarmente intenso, si può verificare una situazione sfavorevole in certe zone dello spettro (nell'xDSL questo accade tipicamente sulle alte frequenze, quando il binder ha doppini con utenze tutte attive e/o il doppino vittima è molto lungo). Può accadere che alcune bande di frequenza restino "scoperte", cioè che la quantità d'acqua non riesca a "coprire" il fondale dell'acquario, come in Fig. 8.31. La zona scoperta è anche quella nella quale o lo spettro del rumore originario è molto forte, oppure la risposta in ampiezza del canale è molto piccola (e quindi il segnale trasmesso viene molto attenuato). In queste condizioni la prescrizione del teorema del riempimento d'acqua è chiara: *non* si devono usare le sottobande disturbate o attenuate (in rosso), evitando del tutto di trasmettervi componenti di segnale.



Figura 8.31 Portanti rumorose nel criterio a riempimento d'acqua

8.8 La modulazione DMT (Discrete MultiTone)

Ri-consideriamo adesso il problema della massimizzazione della capacità in un canale ACGN, con l'attenzione rivolta all'implementazione pratica dei concetti appena visti. La capacità è stata calcolata dividendo la banda B in un numero K di sottobande separate, realizzando quindi un sistema di trasmissione multiportante. Abbiamo poi calcolato le potenze P_k da associare a ciascun sottocanale per massimizzare la capacità in un tale sistema. A questo proposito, osserviamo che il criterio di water filling per l'allocazione della potenza sulle sottobande è controintuitivo: la (8.73) dice che $P_k = \bar{P} - \sigma_k^2$, cioè piuttosto che cercare di compensare le bande rumorose con più potenza, si dà maggior potenza alla sottobande *meno* rumorose. Dopo l'allocazione di potenza quindi, ogni sottobanda opera a un particolare valore di SNR, e questi valori saranno molto diversi tra loro, dando luogo a capacità molto diverse tra loro (tipicamente molto alte sulle basse frequenze e molto basse sulle frequenze più alte).

Più in dettaglio, la capacità su ogni sottobanda è

$$C_k = \Delta f \log_2 \left(1 + \frac{P_k}{\sigma_k^2} \right) \quad \text{[bit/s]}$$
(8.76)

Data una certa R_b che il modem intende trasmettere sul link, dobbiamo partizionare tale bit-rate sulle varie sottobande inviando sulla k-esima un certo bit-rate $R_b^{(k)}$ in modo da rispettare il limite di Shannon e arrivare al totale R_b :

$$R_b^{(k)} \le C_k$$
, $\sum_{k=0}^{K-1} R_b^{(k)} = R_b$ (8.77)

In altre parole, deve essere effettuata, successivamente alla *power allocation*, una *bit allocation*, cioè una ripartizione del bit-rate totale tra le varie sottobande.

Riassumendo, la tecnologia multiportante usata per i formati xDSL, che prende il nome di DMT (Discrete MultiTone), è in grado di massimizzare la capacità di sistema effettuando in pratica due operazioni in trasmissione: i) *allocazione della potenza* secondo il principio del riempimento d'acqua ii) *allocazione del bit-rate* secondo la capacità dei vari canali. Sappiamo già come realizzare un sistema multiportante efficiente dal punto di vista dell'architettura del modem: dobbiamo implementare algoritmi di FFT in trasmissione e ricezione. Questo è ciò che viene fatto anche per l'ADSL, inclusa la caratteristica del

LA MODULAZIONE DMT (DISCRETE MULTITONE) 315



prefisso ciclico già esaminata per OFDM. La differenza tra OFDM e DMT sta in un fattore fondamentale: nell'ADSL è disponibile un *canale di ritorno* attraverso il quale è possibile rendere la modulazione *adattativa* (Rate-Adaptive DMT), nel senso che il bit-rate è ottimizzato nei confronti della capacità disponibile: il ricevitore puòò misurare il livello di rumore σ_k^2 su di ogni sottobanda e comunicarlo tramite il canale di ritorno al trasmettitore, il quale può conseguentemente effettuare le operazioni di power allocation e bit allocation già accennate.

Tornando alla bit allocation, per ottenere l'ortogonalità dei vari sottocanali, sappiamo che il symbol-rate su di ogni sottoportante deve essere lo stesso, e deve essere uguale alla spaziatura tra le sottoportanti: $R_M = \Delta f$. L'unica maniera per variare il bit-rate su di ogni sottoportante è allora quella di cambiare su ogni sottoportante il MOD/COD (cioè la modulazione e codifica, vedi paragrafo 4.5.2) in modo che $R_b^{(k)} = \Delta f \cdot N_b^{(k)}$. Nell'ADSL classico, Δf =4.3125 kHz con un numero massimo di sottoportati K = 256(vedi anche la Fig. 8.26) e il tasso di codifica r, viene mantenuto per semplicità *identico* su tutte le sottoportanti - il bit-rate viene variato cambiando il numero dei punti $M^{(k)}$ della costellazione: $R_b^{(k)} = \Delta f \cdot r \log_2(M^{(k)})$. Come già anticipato, l'adattatività del DMT necessita della conoscenza del livello di

Come già anticipato, l'adattatività del DMT necessita della conoscenza del livello di rumore sulle sottoportanti. Al momento dell'instaurazione della sessione, il modem invia un preambolo costituito da *potenza uniformemente allocata* su tutti i canali, in modo che il ricevitore è in grado di misurare il rapporto segnale-rumore su ogni sottoportante SNR_k , e può comunicare questi valori al trasmettitore attraverso il canale di ritorno. In

316 UN ARCOBALENO DI DATI - LE TECNOLOGIE MULTIPORTANTE

questo modo il trasmettitore può allocare il giusto numero di bit su ogni sottoportante variando adattativamente la costellazione di punti della modulazione da usare su ogni sottoportante. In pratica, le portanti più "disturbate", cioè quelle con SNR_k basso useranno modulazioni semplici con pochi bit/simbolo (BPSK, QPSK) o addirittura verranno annullate in accordo al principio del riempimento d'acqua. Viceversa, le portanti con un alto SNR_k useranno costellazioni a molti bit/simbolo. La modulazione DMT per ADSL prevede un massimo di 32 sottoportanti con un prefisso ciclico di 5 simboli in upstream e 256 sottoportanti con un prefisso ciclico di 32 simboli in downstream. Le costellazioni possono allocare fino a un massimo di 15 bit per simbolo, con codifica a traliccio. Viene inoltre utilizzato un codice di Reed-Solomon come codice esterno contro errori a pacchetti previo interleaving/deinterleaving come nello schema di codifica concatenata del DVB-T (Fig. 6.17).

Lo schema di massima della catena di trasmissione ADSL è simile a quello di Fig. 8.17 relativo ad OFDM, ma con un'importante differenza. Al contrario di quanto accade per la OFDM cioè, in cui il blocco funzionale di mapping è unico (e può quindi essere posizionato prima del convertitore S/P), il blocco funzionale "constellation encoder" (o mapper) in DMT mappa i dati paralleli in ingresso in varie costellazioni QAM secondo l'algoritmo di bit allocation. Inoltre, i coefficienti della IDFT (che trasforma i flussi paralleli di dati nel dominio frequenziale in flussi paralleli di dati nel dominio temporale), saranno diversi per ciascuna sottobanda, secondo l'operazione di power allocation. In linea di principio, le differenze fra l'implementazione di DMT e OFDM sono rappresentate nello schema di Fig. 8.32. Si noti che per le modulazioni OFDM di tipo broadcasting l'operazione di power allocation non avrebbe comunque significato: per ottimizzare la capacità occorrerebbe un trasmettitore per ciascun ricevitore (come avviene in ADSL in cui c'è un ATU-C per ogni ATU-R), ma questo andrebbe contro la definizione stessa di trasmissione broadcasting.

Esempio 8.47

Dunque la capacità del canale ACGN è

$$\mathcal{C}_{ACGN} = \int_{0}^{B} \log_2\left(1 + \frac{S_x(f)}{S'_N(f)}\right) df \tag{8.78}$$

Riusciamo da questa espressione più generale ritrovare quella più semplice del canale AWGN?

Possiamo cominciare con l'inserire $S'N'(f) = N_0/2$ nella (8.78), ma naturalmente non possiamo procedere oltre se non specifichiamo l'andamento di $S_x(f)$; ciò può essere fatto allocando la potenza di X(t) secondo il criterio di water-filling:

$$S_x(f) = \bar{S} - N_0/2 \tag{8.79}$$

per cui

$$\mathcal{C}_{ACGN} = \int_{0}^{B} \log_2\left(\frac{2\bar{S}}{N_0}\right) df = B \log_2\left(\frac{2\bar{S}}{N_0}\right)$$
(8.80)

D'altronde sappiamo che

$$\bar{S} = \frac{P_x + \sigma_{N'}^2}{2B} = \frac{P_x + N_0 B}{2B}$$
(8.81)

LA MODULAZIONE DMT (DISCRETE MULTITONE) 317

e quindi

$$\mathcal{C}_{ACGN} = B \log_2\left(\frac{2(P_x + N_0 B)}{2BN_0}\right) = B \log_2\left(1 + \frac{P_x}{N_0 B}\right)$$
(8.82)

come già sapevamo.





CAPITOLO 9

QUANDO IL DISTURBO È PIÙ FORTE DEL SEGNALE - COMUNICAZIONI SPREAD-SPECTRUM



"La Torre di Babele (1563), Kunsthistorisches Museum Wien, Vienna, Austria"

—Pieter Bruegel il Vecchio (1525?-1569)

In quelle settimane concitate durante la II guerra mondiale nelle quali il musicista George Antheil e l'attrice Hedy Lamarr inventarono la tecnica *spread spectrum* per comunicare in segretezza, nessuno avrebbe immaginato che i principi di quell'invenzione sarebbero diventati la base delle tecnologie di comunicazione sicura dei successivi ottant'anni, e che avrebbero ispirato la realizzazione della terza generazione delle reti cellulari a copertura mondiale (UMTS), quella per la quale, per la prima volta, il traffico di connessione a Internet superò quello delle telefonate. In questo capitolo, quindi, esamineremo le basi e alcuni dettagli avanzati per aiutare il lettore e la lettrice a capire ciò che Andrew Viterbi chiamò in un cult-paper degli anni '70 "*The Myths and Realities of Spread-Spectrum Communications*".

9.1 Comunicare con un segnale a spettro espanso

Più di una volta abbiamo sottolineato nei capitoli precedenti che, di regola, la banda occupata da un segnale digitale è pari o comunque direttamente collegata al symbol rate $R_s = 1/T_s$ o, nel caso di segnalazione binaria, al bit-rate $R_b = 1/T_b$. Spesso le regole hanno eccezioni, e tutto questo capitolo riguarda appunto una di queste - la possibilità/opportunità/necessità di creare un segnale digitale per il quale intenzionalmente la banda occupata è molto (ma molto) maggiore del bi-rate trasportato. Tra gli esempi più noti in questo senso possiamo citare il segnale trasmesso verso terra dai satelliti del GPS (Global Positioning System), il primo tra i vari Global Navigation Satellite Systems (GNSS) oggi disponibili (uno per tutti il sistema Europeo GALILEO) e usato da tutti gli smartphone e le autovetture per effettuare la localizzazione del ricevitore. Ciascuno dei 24 satelliti GPS rappresentati in Fig. 9.1 che orbitano ad una altitudine di circa 27,000 Km invia verso terra un segnale digitale con un bit-rate $R_b=50$ bit/s che, tra la altre cose, contiene le coordinate spaziali x - y - z del satellite stesso¹. Per motivi che vedremo in seguito, la banda occupata dal segnale digitale (in un banale formato NRZ/BPSK) anziché 50 Hz come suggerirebbe l'esperienza pregressa, è pari a 1.023 MHz, cioè 20460 volte maggiore di quanto la suddetta regoletta della banda suggerirebbe !

Si capisce che nel caso del GPS, ma anche nel caso, per citare un'altra tecnologia molto popolare, dei cellulari 3G UMTS degli anni 2000 o delle reti di prossimità *Bluetooth*, lo spettro (*spectrum*) del segnale digital è stato espanso(*spread*) (molto) oltre quanto strettamente necessario con modalità e per motivi che dovremo scoprire nei prossimi paragrafi.

9.1.1 Banda stretta e a banda larga: Direct-Sequence Spread Spectrum

9.1.1.1 Costruzione del segnale DS/SS Ripartiamo dal semplicissimo segnale NRZ/BPSK (equivalente in banda base) del paragrafo 2.7.1:

$$x(t) = \sqrt{2P_x} \sum_{n=-\infty}^{+\infty} a[n]p(t - nT_s)$$
(9.1)

dove P_x è la potenza del segnale, $T_s = T_b$ è l'intervallo di simbolo (bit), $a[n] \in \{-1, +1\}$ sono i dati binari (equiprobabili e indipendenti) e dove p(t) è l'impulso (rettangolare) NRZ

¹il ricevitore GPS riceve contemporaneamente (almeno) 4 di questi messaggi da 4 satelliti in punti diversi del cielo, misura le 4 distanze r_1 , r_2 , r_3 , r_4 da se stesso ai satelliti attraverso la misura del tempo di propagazione (quattro osservazioni) e con queste misure può calcolare la sua propria posizione (tre incognite x - y - z) e lo scarto del suo clock di ricezione Δt rispetto al clock dei satelliti (quarta incognita)



di Fig. 9.2 (a). Lo spettro (densità spettrale di potenza) del segnale dati è quello già ricavato in (2.95) e riportato in Fig. 2.24:

$$S_x(f) = 2P_x \frac{|P(f)|^2}{T_s} = 2P_x T_s \operatorname{sinc}^2(fT_s)$$
(9.2)

La banda del segnale, valutata come sempre al primo nullo visto che il formato NRZ è teoricamente a banda illimitata, è pari a $B_x = 1/T_s$, cioè pari al clock di trasmissione dati, come già sappiamo dalla Fig. 2.24.

Se vogliamo creare un formato a banda molto più larga di B_x , dobbiamo ricorrere a un qualche artificio. In particolare, per allargare (*to spread*) lo spettro useremo un *codice di spreading* c(t), cioè un segnale digitale NRZ con un clock intrinseco (molto) più veloce del clock dati $1/T_s$, nella forma che segue:

$$c(t) = \sum_{\ell=-\infty}^{+\infty} c[\ell]q(t-\ell T_c)$$
(9.3)

dove $c[\ell] \in \{-1, +1\}, q(t)$ è l'impulso NRZ, e il clock di segnalazione è $R_c = 1/T_c \gg R_s = 1/T_s$, scandito dall'indice ℓ . Normalmente, per comodità di progetto HW, si sceglie

un rapporto intero e grande tra $R_c e R_s$, cioè $R_c/R_s = M$ intero. A causa del fatto che il clock è più veloce, l'impulso NRZ q(t) del codice di spreading non è identico all'impulso dati p(t) - è infatti M volte più breve, come chiaramente visibile nel confronto tra le Figg. 9.2 (a) e (b), e viene chiamato *impulso di chip*.

Resta da capire che cosa siano i valori binari simmetrici $c[\ell]$ nel codice di spreading c(t), che non hanno niente a che fare con i simboli (bit) a[n] del segnale dati x(t), e che vengono chiamati *chip*. Tali valori sono quelli di una sequenza pseudo-casuale, cioè che apparentemente non segue una legge ben chiara o ripetitiva e che appaiono come estratti casualmente, ma che in realtà segue un algoritmo ben preciso e che quindi può essere (ri)generata in qualunque momento. Per questo c(t) si chiama codice: il fatto che *sembri* casuale allude a un suo uso come "chiave" di cifratura, la cui logica interna non può essere ricostruita facilmente - lo scopo iniziale per cui lo *spread spectrum* fu creato.

Il segnale *spread spectrum* $x_{ss}(t)$ si crea nel momento in cui applichiamo il codice al segnale dati in una forma peraltro banale:

$$x_{ss}(t) = x(t) \cdot c(t) \tag{9.4}$$

Con l'aiuto della Fig. 9.3 ci convinciamo facilmente che il segnale spread-spectrum "eredita" il clock veloce del codice c(t) (il cosiddetto *chip-rate* R_c) e si può esprimere facilmente come segue:

$$x_{ss}(t) = \sqrt{2P_x} \sum_{\ell=-\infty}^{+\infty} \gamma[\ell]q(t-\ell T_c)$$
(9.5)

dove $\gamma[\ell]$ è un valore digitale che "corre" a tempo di chip rate, ed è ottenuto dalla combinazione dei dati digitali con i chip binari del codice:

$$\gamma[\ell] \stackrel{\triangle}{=} c[\ell] \cdot a[\lfloor \ell/M \rfloor] \tag{9.6}$$

Il significato della notazione $a[\lfloor \ell/M \rfloor]$ è chiaro (vedi ancora Fig. 9.3): i dati digitali corrono a un clock più *lento* di quello dei chip di un fattore M, e quindi l'indice $n = \lfloor \ell/M \rfloor$ dei dati resta *costante* esattamente per M colpi del chip rate, cioè per M valori consecutivi dell'indice di chip ℓ . I chip hanno questo nome perchè l'applicazione del codice al segnale dati (ancora la Fig. 9.3) ha l'effetto di "tagliare" il simbolo dati in M fette, cioè *chips* (come le patatine del celebre piatto anglosassone *fish and chips*). Notiamo esplicitamente che l'applicazione del codice di spreading *non altera la potenza* del segnale dati (cioé $P_{ss} = P_r$), visto che il codice è binario NRZ, e quindi $|x_{ss}(t)|^2 = |x(t) \cdot c(t)|^2 = |x(t)|^2 \cdot |c(t)|^2 =$ $|x(t)|^2$.

Il formato che abbiamo ottenuto si chiama Direct-Sequence Spread-Spectrum (DS/SS) perchè la sequenza dei chip (il codice) di spreading viene *direttamente* applicata al segnale dati per realizzare l'espansione spettrale. Dobbiamo però confermare che tale espansione sussista veramente.

9.1.1.2 Spettro del segnale DS/SS È relativamente semplice valutare lo spettro $S_{ss}(f)$ di $x_{ss}(t)$ se adottiamo una approssimazione: se supponiamo cioè che la sequenza dei chip $c[\ell]$ sia veramente *casuale* piuttosto che pseudocasuale, e se in particolare i chip vengono considerati equiprobabili e indipendenti. Nella realtà, quest'ipotesi è spesso ben verificata, e quindi i risultati del caso reale sono con buona approssimazione quelli che stiamo per ricavare sotto le nostre approssimazioni.

Con i chip equiprobabili e indipendenti, è immediato convincersi che anche i simboli $\gamma[\ell]$ risultano... equiprobabili e indipendenti (nonché binari); la conclusione immediata



A CARLEY AND AN AND AN A DECEMPTOR

è che la forma dello spettro di $x_{ss}(t)$ è quella consueta di un segnale digitale NRZ con simboli indipendenti, cioè del tutto analoga alla (9.2), con le dovute modifiche:

$$S_x(f) = 2P_x \frac{|Q(f)|^2}{T_c} = 2P_x T_c \operatorname{sinc}^2(fT_c)$$
(9.7)

In particolare, la banda al primo nullo è stavolta $B_{ss} = 1/T_c = M/T_s = MB_x$ esattamente come ci si attendeva: la banda occupata dal segnale è stata espansa (spread) di un fattore esattamente pari ad M - ecco perchè questo parametro prende il nome di *spreading factor*.

La Fig. 9.4 mette a confronto gli spettri del segnale NRZ con e senza spectral spreading. Oltre alla banda espansa del fattore M indicato, si nota anche l'abbassamento del livello della densità spettrale per f = 0 esattamente dello stesso fattore M (si confronti la (9.2) con la (9.7)): anziché "alto e stretto" lo spettro espanso si presenta "basso e largo" poiché la potenza totale del segnale, che è la medesima P_r nei due casi, viene (ri)distribuita in modo diverso sull'asse delle frequenze. Il segnale GPS segue alla lettera il formato (9.5)-(9.6) con R_b =50 bit/s e R_c =1.023 Mchip/s (mantenuto esattamente uguale al valore nominale da clock atomici a bordo dei satelliti, con compensazione di effetti relativistici), per un fattore di spreading M = 1023000/50 = 20460 come avevamo anticipato, e con uno spettro identico a quello di Fig. 9.4 con gli adattamenti del caso.

9.1.1.3 Ricostruire il segnale dati Espandere lo spettro porterà sicuramente qualche vantaggio, che dovremo esaminare nei paragrafi successivi. In ogni caso, deve essere possible ricostruire i dati informativi a partire dal segnale DS/SS che abbiamo costruito. L'operazione che abbiamo genericamente chiamato di "applicazione" del codice c(t) al segnale dati x(t) è riassunta nello schema di Fig. 9.5 (a) e si chiama spreading. L'operazione duale che permette di ricostruire il segnale dati (a banda stretta) a partire dal DS/SS si chiama despreading, ed è indicata in Fig. 9.5 (b) - è di fatto analoga allo spreading, e consiste nel riapplicare il codice di spreading stesso per rimuoverne l'effetto e ricompattare (de-spreading) lo spettro nella sua banda originaria.



Figura 9.4 Densità spettrale di potenza del segnale a banda stretta e DS/SS: (a) scala lineare con M = 8 - (b) scala in dB con M = 16



Figura 9.5 Operazioni duali di *spreading* (a) e *despreading* (b)

Come è chiaro, l'operazione di spreading è effettuata dal trasmettitore DS/SS (il satellite GPS), mentre il despreading deve essere effettuato dal ricevitore (il navigatore satellitare) per ricostruire il messaggio digitale. Quindi in sede di ricezione deve essere possibile ricostruire e riapplicare il codice c(t): per questo motivo c(t) non può essere costituito da valori estratti randomicamente in tempo reale al trasmettitore (tipo lancio della moneta), bensì tali valori devono essere individuati attraverso algoritmi ben precisi di generazione, noti al trasmettitore e al ricevitore. Torneremo su quest'argomento nel paragrafo 9.2.3.

9.1.2 Ricezione del segnale DS/SS su canale AWGN

9.1.2.1 Struttura del Ricevitore e BER Evidentemente, applicare il codice di spreading altera le caratteristiche del segnale dati. Ci si deve quindi chiedere se e come è possibile comunicare efficientemente con tali segnali quando la trasmissione è disturbata da rumore di ricezione, distorsione, interferenza ecc. Cominciamo ad esaminare il caso più importante e semplice: l'effetto del rumore Gaussiano bianco additivo (AWGN).

Immaginando di effettuare una trasmissione su canale AWGN con ricezione coerente (offset di frequenza e fase sulla portante perfettamente compensati), il segnale ricevuto sarà del tipo

$$r(t) = x_{ss}(t) + w(t) = x(t)c(t) + w(t)$$
(9.8)

dove $w(t) = w_I(t) + \jmath w_Q(t)$ è AWGN con componenti indipendenti aventi entrambe densità spettrale di potenza $S_W(f) = N_0$. Piuttosto che affrontare ancora una volta la questione di ricavare la strategia ottima di ricezione dei dati attraverso la teoria della stima, prendiamo una strada più empirica. Dal paragrafo precedente, sappiamo che per recuperare il segnale dati il ricevitore deve effettuare il despreading - analizziamo quindi una struttura di processing di questo tipo, cioè analizziamo le caratteristiche dell'uscita del despreading di r(t) (comprensivo di AWGN):

$$z(t) \stackrel{\Delta}{=} r(t)c(t) = x_{ss}(t)c(t) + w(t)c(t) = x(t)c^{2}(t) + n(t) = x(t) + n(t)$$
(9.9)

Otteniamo come ci si aspettava il segnale dati originario x(t), accompagnato da una componente di disturbo n(t) = w(t)c(t) le cui statistiche dobbiamo studiare.

Cominciamo col dire che n(t) è Gaussiano perchè si ottiene da una trasformazione lineare di un processo Gaussiano w(t) (la moltiplicazione istante per istante per una forma d'onda *nota al ricevitore* e quindi deterministica c(t)). Poichè però c(t) è variabile nel tempo, non è detto che n(t) risulti stazionario - dobbiamo allora ricavarne funzione valor medio e autocorrelazione in senso generale. Per il valor medio abbiamo

$$\eta_n(t) \stackrel{\scale}{=} \mathbf{E}\{n(t)\} = \mathbf{E}\{w(t)c(t)\} = c(t) \cdot \mathbf{E}\{w(t)\} = 0 \tag{9.10}$$

L'ultimo passaggio segue dal fatto che w(t) è bianco e quindi a valor medio nullo, e il penultimo dal fatto che c(t) è un valore deterministico e quindi rispetto all'operatore valore atteso *costante*. Per l'autocorrelazione abbiamo inoltre

$$R_n(t;t-\tau) \stackrel{\bigtriangleup}{=} \mathbb{E}\left\{n(t)n^*(t-\tau)\right\}$$
$$= c(t)c(t-\tau)\mathbb{E}\left\{w(t)w^*(t-\tau)\right\} = c(t)c(t-\tau) \cdot R_w(\tau)$$
(9.11)

Sembra restare confermato che il processo non è stazionario. In realtà sappiamo che $R_w(\tau) = 2N_0\delta(\tau)$, dove $\delta(\tau)$ è la funzione di Dirac (2.17) che, per così dire, è diversa da 0 solo per $\tau = 0$. Con questa proprietà,

$$R_n(t;t-\tau) = c(t)c(t-\tau)2N_0\delta(\tau) = 2N_0c(t)c(t)\delta(\tau) = 2N_0\delta(\tau)$$
(9.12)



Figura 9.6 Ricevitore a despreading-matched filter per DS/SS (ricevitore a correlazione)

quindi in realtà n(t) è esso stesso AWGN con le medesime statistiche di w(t).

A questo punto la relazione (9.9) ci dice che, dopo despreading, abbiamo un *canale AWGN sul segnale a banda stretta* orginario, che potrà essere ricevuto in modo ottimale con un semplice ricevitore a filtro adattato all'impulso dati (rettangolare) NRZ di x(t), (anche) realizzato nella forma di un banale filtro integratore (2.58). La struttura di ricevitore (ottimo!) che risulta comprende un despreader e un filtro adattato ed è mostrata in Fig. 9.6. Di fatto, il ricevitore della Fig. 9.6 calcola la correlazione sul tempo di simbolo del segnale ricevuto con il codice di spreading c(t), ricavando la variabile di decisione *soft* z[n]relativa al simbolo dati a[n] - per questo viene anche indicato con il nome di *ricevitore a correlazione* (cfr. anche la Fig. 2.26).

9.1.2.2 Sincronizzazione di Codice Lo schema della Fig. 9.6 evidenzia una caratteristica del ricevitore DS/SS che abbiamo finora tralasciato, ma che riveste particolare importanza, e cioè la *sincronizzazione di codice*.

Il modello completo del segnale DS/SS su canale AWGN (così come viene ricevuto, sempre per restare sul consueto esempio, da un satellite GPS) segue la relazione generale (7.1):

$$r(t) = x(t-\tau)c(t-\tau)e^{j\theta} + w(t)$$

= $\sqrt{2P_r} \sum_{n=-\infty}^{+\infty} a[n]p(t-nT_s-\tau)c(t-\tau)e^{j\theta} + w(t)$ (9.13)

dove abbiamo introdotto lo sfasamento θ , il ritardo τ , e l'ampiezza $A = \sqrt{2P_r}$ derivante dalla potenza ricevuta P_r . Non considerando per il momento lo sfasamento, che può essere stimato e recuperato *dopo* l'operazione di despreading-matched filter perchè si "propaga" senza modifiche fino a z[n] (esercizio per chi legge), consideriamo invece la presenza del ritardo τ e immaginiamo che il ricevitore, ignaro di questo ritardo, proceda con le operazioni di despreading-matched filter *usando il suo proprio riferimento temporale* per il codice, cioè usando c(t) nella forma nativa - sincronizzato rispetto all'istante iniziale t = 0, che nel gergo DS/SS si chiama "epoch" del codice. L'uscita del ricevitore, per la parte di segnale utile, sarà

$$z[0] = \frac{1}{T_s} \int_0^{T_s} x(t-\tau)c(t-\tau)c(t) dt$$
(9.14)



Figura 9.7 Il problema della sincronizzazione di codice per DS/SS

dove si evidenziano, con l'aiuto della Fig. 9.7, alcuni problemi:

- 1. Il codice di spreading del segnale ricevuto $c(t \tau)$ non è sincronizzato con il codice che stiamo applicando nel ricevitore per effettuare il desprading
- 2. Il tempo di integrazione $[0, T_s)$ non è allineato con i tempi di simbolo entro il segnale dati $x(t \tau)$
- 3. Di conseguenza, la variabile di decisione che pensavamo essere relativa al simbolo con n = 0 (che presupponiamo essere all'interno di $[0, T_s)$) contiene *altri* simboli dati.

I punti 2. e 3. sono in realtà tipici di qualunque ricevitore digitale, e si risolvono attraverso la (consueta) *sincronizzazione di simbolo*. Il punto 1. è invece caratteristico del ricevitore spread-spectrum e deve essere ulteriormente approfondito - peraltro, nel momento in cui riuscissimo ad avere nel ricevitore un codice "locale" $c(t-\hat{\tau})$ dove $\hat{\tau} \simeq \tau$, cioè praticamente allineato (sincronizzato) con quello del segnale ricevuto, anche i punti 2. e 3. sarebbero automaticamente risolti, e quindi ci concentriamo in quel che segue solo sul punto 1.

In particolare, ricalcoliamo z[0] utilizzando l'espressione del segnale DS/SS (9.5) e supponendo anche per brevità che $\sqrt{2P_x} = 1$:

$$z[0] = \frac{1}{T_s} \int_0^{T_s} \sum_{\ell=-\infty}^{+\infty} \gamma[\ell] q(t - \ell T_c - \tau) c(t) dt$$

$$= \frac{1}{T_s} \int_0^{T_s} \sum_{\ell=-\infty}^{+\infty} a[\lfloor \ell/M \rfloor] c[\ell] q(t - \ell T_c - \tau) \sum_{m=-\infty}^{+\infty} c[m] q(t - mT_c) dt$$
$$= \frac{1}{T_s} \sum_{m=-\infty}^{+\infty} \sum_{\ell=-\infty}^{+\infty} a[\lfloor \ell/M \rfloor] c[\ell] c[m] \int_0^{T_s} q(t - \ell T_c - \tau) q(t - mT_c) dt \qquad (9.15)$$

Per capire meglio la natura di quest'espressione (piuttosto complicata), supponiamo che i chip del codice c(t) siano binari equiprobabili e indipendenti, e calcoliamo allora la *media* di z[0] rispetto ai chip di codice, per liberarci dalla variabilità della scelta di un particolare codice - di fatto, se la sequenza è pseudo-casuale questa variabilità non sussiste perchè ogni sequenza è statisticamente equivalente a ogni altra, e la differenza tra il valore atteso e l'effettivo valore per una particolare sequenza (specie quando M è grande e l'integrazione



Figura 9.8 Impulso $q_{\tau}(t)$ (a) e Correlazione $\rho_c(\tau)$ (b)

comprende quindi molti chip) è trascurabile. Considerando la linearità dell'operatore valore atteso, troviamo $z[0] \simeq \mathrm{E}\{z[0]\} =$

$$E\left\{\frac{1}{T_s}\sum_{m=-\infty}^{+\infty}\sum_{\ell=-\infty}^{+\infty}a[\lfloor\ell/M\rfloor]c[\ell]c[m]\int_0^{T_s}q(t-\ell T_c-\tau)q(t-mT_c)\,dt\right\} \frac{1}{T_s}\sum_{m=-\infty}^{+\infty}\sum_{\ell=-\infty}^{+\infty}a[\lfloor\ell/M\rfloor]E\{c[\ell]c[m]\}\int_0^{T_s}q(t-\ell T_c-\tau)q(t-mT_c)\,dt = \frac{1}{T_s}\sum_{\ell=-\infty}^{+\infty}a[\lfloor\ell/M\rfloor]\int_0^{T_s}q(t-\ell T_c-\tau)q(t-\ell T_c)\,dt = \frac{1}{T_s}\sum_{\ell=-\infty}^{+\infty}a[\lfloor\ell/M\rfloor]\int_0^{T_s}q_\tau(t-\ell T_c)\,dt$$
(9.16)

dove abbiamo tenuto conto del fatto che $E\{c[\ell]c[m]\} = 1$ se $m = \ell$ o 0 altrimenti, e dove abbiamo posto

$$q_{\tau}(t) \stackrel{\Delta}{=} q(t-\tau)q(t)$$

Dalla definizione, si vede che, per $\tau > 0$, l'impulso $q_{\tau}(t)$ è diverso da 0 solo per T_c – $\tau \leq t < T_c$ (Fig. 9.8 (a)). Se in particolare $\tau > T_c$ l'impulso è identicamente nullo e quindi anche E $\{z[0]\}$ - similmente per $\tau < 0$. Questo significa che sicuramente per ottenere una uscita significativa del ricevitore, la precisione di sincronizzazione del codice deve essere *inferiore a un chip* - quindi ben maggiore (di un fattore M) della precisione normalmente necessaria (un simbolo) per la sincronizzazione del simbolo dati. Ecco perché stiamo insistendo molto su quest'aspetto, che risulta specifico per il formato DS/SS (e spread-spectrum in generale) e in certi casi critico.

Per procedere ulteriormente con lo sviluppo di (9.16) ammettiamo che il ricevitore abbia raggiunto una sincronizzazione non perfetta ma comunque grossolana (coarse), e che quindi il riferimento in ricezione si discosti da τ di meno di un chip, cioè $|\tau/T_c| < 1$. In questo caso, l'intervallo di integrazione [0, T) contiene effettivamente il simbolo a[0], che possiamo a questo punto assumere perfettamente sincronizzato a meno di errori inferiori al chip (e quindi sul simbolo del tutto trascurabili, considerando che normalmente $M \gg 1$ e quindi $T_c \ll T_s$). Questo significa che possiamo anche limitare la sommatoria su ℓ ai soli valori contenuti nell'intervallo di integrazione $0 \le \ell \le M - 1$, e che $a[\lfloor \ell/M \rfloor \equiv a[0]$. In

328

conclusione, possiamo semplificare la (9.16) per $T_c \ge \tau \ge 0$:

$$z[0] \simeq \frac{1}{T_s} a[0] \sum_{\ell=0}^{M-1} \int_0^{T_s} q_\tau (t - \ell T_c) \, dt = a[0] \frac{M}{T_s} (T_c - \tau) = a[0] \frac{T_c - \tau}{T_c} \tag{9.17}$$

Ripetendo il calcolo per $\tau < 0$ otteniamo infine

$$z[0] \simeq a[0] \left(1 - \frac{|\tau|}{T_c}\right) \tag{9.18}$$

da cui si può anche *quantificare* la precisione necessaria nella sincronizzazione fine (*fine*) di codice: l'uscita del ricevitore è nominale e pari ad a[0] se $\tau = 0$ (errore nullo), altrimenti decresce rispetto alla situazione nominale di un fattore proporzionale all'errore (in modulo) $1 - |\tau/T_c|$. Al di sopra del tempo di chip, l'uscita del ricevitore è *nulla* e la ricezione del dato è impossibile.

Il fattore mostrato nella Fig. 9.8 (b) di fatto rappresenta la misura della *correlazione* normalizzata $\rho_c(\tau)$ tra il codice in seno al segnale ricevuto $c(t - \tau)$ e il codice locale c(t)

$$\rho_c(\tau) \stackrel{\triangle}{=} \frac{1}{T_c} \int_{-T_c/2}^{T_c/2} \mathbb{E}\{c(t)c(t-\tau)\} dt = \left(1 - \frac{|\tau|}{T_c}\right) \operatorname{rect}\left(\frac{\tau}{T_c}\right)$$
(9.19)

ed è massima quando i due segnali sono perfettamente sincronizzati.

9.2 Frequency-Hopping Spread Spectrum e Altri Formati

Dopo la descrizione generale del principio dello spread-spectrum, vediamo in questo paragrafo alcuni dettagli in più riguardo ai particolari formati che si incontrano nella pratica e che differiscono anche di molto dal semlice DS/SS con impulsi di chip NRZ che abbiamo usato come cavallo di battaglia.

9.2.1 Frequency-Hopping Spread Spectrum

9.2.1.1 Modulatore FH/SS II formato DS/SS è semplice da comprendere ed altrettanto semplice da implementare. Quando però si vogliono realizzare fattori di spreading M veramente molto grandi per distribuire la potenza del segnale su bande molto larghe (1 GHz e più) l'implementazione diventa (più) problematica perchè il chip rate R_c deve necessariamente essere molto alto (1 Gchip/s e più) e il convertitore I/Q per portare alla freuenza radio (si veda il paragrafo 2.4.3) deve essere in grado di trattare segnali a banda *molto* larga.

In questi casi (e vedremo successivamente la ragione per la quale talvolta è opportuno avere bande molto grandi) si ricorre a un formato in cui lo spettro viene espanso come sempre attraverso l'uso di un codice, ma in una maniera indiretta che prende il nome di salto di frequenza (Frequency Hopping Spread Spectrum, FH/SS) e che è mostrata nella Fig. 9.9. Diversamente dallo schema di principio della Fig. 9.5 (a) che descrive il modulatore DS/SS in banda base (e che di fatto comprende soltanto lo spreading), la Fig. 9.9 indica come generare il segnale FH/SS direttamente a frequenza radio. La partenza è il consueto segnale dati x(t) con symbol rate $1/T_s$ (e quindi a banda stretta), il quale viene modulato ad una frequenza portante che non è costante al consueto valore f_0 , bensì varia (salta) con una certa cadenza temporale prefissata, diciamo ogni T_H secondi (il tempo di salto), all'interno



Figura 9.9 Architettura del Modulatore FH/SS

di un insieme di 2H + 1 valori prefissati $f_0 + f_i$, i = -H, -(H - 1), ..., 0, ..., H - 1, H, dove $f_i = i \cdot \Delta f$. La quantità Δf è un parametro di progetto che rappresenta il passo frequenziale di salto. Il criterio con cui si sceglie una frequenza f_i piuttosto che un'altra in un certo intervallo temporale è pseudocasuale, seguendo cioè una sequenza di salto apparentemente randomica, in realtà generata da un algoritmo ben preciso.

Esempio 9.48

Un esempio molto popolare di FH/SS è previsto dallo standard Bluetooth per la comunicazioni radio a corto raggio (PAN, Personal Area Network) sulla banda ISM attorno a 2.4 GHz usata anche da WI-Fi, che nella versione-base prevede un hopping step $\Delta f = 1MHz$ e un insieme di 79 frequenze di salto, da 2402 MHz a 2480 MHz, con una cadenza di salto $R_H = 1/T_H = 1600$ hop/s.

Dunque l'equivalente in banda base del segnale FH/SS è

$$x_{fh}(t) = x(t) \sum_{\ell} \exp\left(\jmath 2\pi \alpha[\ell] \Delta f t\right) q(t - \ell T_H)$$
(9.20)

dove stavolta l'impulso NRZ q(t) che marca il tempo di hop ha una durata pari a T_H , e dove $\alpha[\ell] \in \{-H, -(H-1), ..., H-1, H\}$ è l'indice della ℓ -esima frequenza su cui il segnale salta nell'intervallo $\ell T_h \leq t < (\ell + 1)T_H$. Nella (9.20), x(t) è il segnale a banda stretta con una certa modulazione dati a symbol rate T_s - spesso vengono utilizzate modulazioni di frequenza (FSK, Frequency-Shift Keying) che risultano particolarmente semplici da demodulare (è questo proprio il caso del Bluetooth). Possiamo anche scrivere, per cercare di trovare analogie con il DS/SS, $x_{fh}(t) = x(t)c_h(t)$ dove $c_h(t)$ è l'elemento che realizza l'espansione spettrale, cioè la serie di portanti (in banda base) a salto.

Il concetto del FH/SS può essere efficacemente rappresentato sul diagramma tempo/frequenza della Fig. 9.10 (una sorta di *spettrogramma*), nella quale si indica chiaramente che all'interno di un certo intervallo temporale di hop $\ell T_H \leq t < (\ell + 1)T_H$ il segnale occupa "localmente" la banda di frequenza individuata dalla portante a salto, cioè $\alpha[\ell]\Delta f$ un concetto di spettro del segnale *a corto termine*.

Mentre nel DS/SS sussiste la relazione tra chip rate e symbol rate $R_c = MR_s$, con M sempre e comunque $\gg 1$, la relazione tra T_s e T_H non è altrettanto a senso unico. Molto



Figura 9.10 Diagramma tempo/frequenza del segnale FH/SS

spesso T_H è maggiore di T_s (sempre di un fattore intero), ma in certi casi si ha anche $T_H = T_s$ o addirittura $T_s > T_H$ - questi ultimi formati caratterizzano il *Fast* FH/SS e sono tipici delle comunicazioni sicure o militari. Detto questo, dobbiamo capire se e come il formato (9.20) effettivamente realizzi l'espansione spettrale desiderata.

9.2.1.2 Occupazione spettrale Poichè il segnale dati e la sequenza di salto di frequenza sono statisticamente indipendenti, la densità spettrale di potenza $S_{fh}(f)$ del FH/SS si troverà come

$$S_{fh}(f) = S_x(f) \otimes S_c(f) \tag{9.21}$$

dove $S_c(f)$ è lo spettro di potenza di c(t). L'espressione generale di $S_c(f)$ e successivamente di $S_{fh}(f)$ è piuttosto complicata - per quel che serve qui, ricaviamola per un caso semplificato di *slow hopping* in cui $T_H \gg T_s$. Anche prima del calcolo dettagliato di $S_{fh}(f)$, è chiaro che maggiore è Δf (e quindi più sono distanziate le frequenze di salto), maggiore sarà la banda occupata dal segnale FH/SS, e quindi supporremo anche (vedi l'esempio del Bluetooth) $\Delta f \gg 1/T_H$ per avere un buon fattore di spreading.

Concentriamoci dunque sul calcolo di $S_c(f)$, osservando che l'espressione di c(t) nella (9.20) rispetta un ordinamento temporale dei vari intervalli di salto uno dopo l'altro. Possiamo riformulare c(t) pensando a un ordinamento *frequenziale*, cioè raggruppare gli intervalli temporali corrispondenti alle stesse frequenze, come se guardassimo il diagramma di Fig. 9.10 "attraverso l'asse Y":

$$c_h(t) = \sum_{i=-H}^{H} \exp\left(j2\pi i\Delta ft\right)c_i(t) \quad , \quad c_i(t) \stackrel{\triangle}{=} \sum_{\ell} c_i[\ell]p(t-\ell T_H) \tag{9.22}$$

dove i $c_i[\ell]$ che appaiono nei segnali digitali $c_i(t)$ (che marciano a tempo T_H) valgono 0 o 1. In particolare, nell' ℓ -esimo intervallo temporale $\ell T_H let < (\ell + 1)T_H$, vi sono esattamente 2H valori $c_i[\ell]$ uguali a 0 e *un solo* valore $c_i[\ell] = 1$, quello per cui $i = \alpha[\ell]$. Di fatto, ℓ e $i = \alpha[\ell]$ sono le coordinate dell'*unica* casellina occupata al tempo ℓT_H nel diagramma di Fig. 9.10. Dato che il clock dei vari $c_i(t)$ è $1/T_H$, la banda occupata singolarmente da ciascuno di essi è pari circa a (qualche multiplo di) $1/T_H$. Se $\Delta f \gg 1/T_H$

$$S_c(f) = \sum_{i=-H}^{H} S(f - i\Delta f)$$
(9.23)

332 QUANDO IL DISTURBO È PIÙ FORTE DEL SEGNALE - COMUNICAZIONI SPREAD-SPECTRUM

dove $S(f) = S_i(f) \forall i$ è lo spettro di potenza di ciascuno dei segnali $c_i(t)$. Il fatto che tutti i $c_i(t)$ condividano il medesimo spettro è immediato: considerando la sequenza di salto $\alpha[\ell]$ puramente casuale con valori indipendenti ed equiprobabili nell'insieme $\{-H, ..., H\}$, segue che i vari $c_i[\ell]$ sono, fissao i, (temporalmente) indipendenti e con probabilità $p_1 = \Pr\{c_i[\ell] = 1\} = 1/(2H + 1), p_0 = \Pr\{c_i[\ell] = 1\} = 1 - p_1$. I $c_i(t)$ non sono indipendenti rispetto ad i (uno e solo uno è diverso da 0 in ogni intervallo di salto), ma singolarmente hanno tutti lo stesso formato/proprietà, e quindi lo stesso spettro. Gli spettri dei vari $c_i(t)$ vengono spostati attorno alla frequenza $i\Delta f$ dall'oscillazione alla medesima frequenza nella (9.22), e nell'espressione dello spettro 9.23 non appaiono termini "incrociati" derivanti dall'interazione tra $c_i(t) \exp(j2\pi i\Delta f t)$ e $c_m(t) \exp(j2\pi m\Delta f t), i \neq m$, perchè la banda dei c_i è molto stretta nei confronti di Δf , e quindi le rispettive traslazioni in frequenza rendono i segnali incorrelati nel tempo ovvero non sovrapposti in frequenza.

Resta da ricavare il generico S(f), tenendo conto che $c_i(t)$ non è a media nulla, e che quindi vi saranno in generale componenti discrete (righe) nello spettro - il calcole è piuttosto complicato. Ricorriamo allora ad un'ulteriore semplificazione: se, come spesso accade, la banda del segnale dati è molto maggiore della frequenza di salto $1/T_H$, lo spettro S(f) è a banda molto stretta rispetto a $S_x(f)$, e praticamente tutte le righe "collassano" in una riga unica posta alla frequenza zero, e con potenza 1/(2H + 1), che è la potenza totale di ciascun $c_i(t)$. Di fatto, lo spettro di $c_h(t)$ diventa quello di un oscillatore multifrequenza:

$$S_c(f) \simeq \frac{1}{2H+1} \sum_{i=-H}^{H} \delta(f - i\Delta f)$$
(9.24)

e quindi, finalmente

$$S_{fh}(f) = S_x(f) \otimes S_c(f) \simeq \frac{1}{2H+1} \sum_{i=-H}^{H} S_x(f-i\Delta f)$$
 (9.25)

Adesso possiamo comprendere la funzione di espansione spettrale: il salto di frequenza crea 2H + 1 "copie" dello spettro dati che si distribuiscono su di un intervallo frequenziale esteso fino ad $H\Delta f + B_x \gg B_x$ non appena H e/o Δf sono grandi, come esemplifichiamo in Fig. 9.11. Questa espansione, anche molto rilevante, non richiede *clock veloci* - non più veloci del clock dei dati. L'unico requisito è quello di avere a disposizione, in trasmissione e in ricezione, un generatore della portante (oscillatore) sufficientemente *agile*, cioè che possa variare il valore della frequenza portante su di un intervallo (*span*) molto ampio. Come già per il DS/SS, contestualmente all'allargamento spettrale, vediamo anche una *riduzione del livello* dello spettro, in questo caso di un fattore 2H + 1, pari al numeri di frequenze su cui si salta.

9.2.1.3 Demodulatore FH/SS L'architettura del demodulatore FH/SS è conseguente a quella del modulatore, è mostrata nella Fig. 9.12, e non necessita di particolari commenti. Come è chiaro, il ricevitore deve sincronizzare la sequenza di salto con quella del ricevitore, una funzione simile alla sincronizzazione di codice del DS/SS, ma meno critica da realizzare perché non vi sono in gioco stavolta clock più veloci di quello dei dati. Gli schemi di modulatore e demodulatore FH/SS che abbiamo presentato nelle Figg. 9.9 e 9.12 sono pressochè identici a quelli che appaiono nel brevetto originale di Lamarr (con il vero nome Hicoler Markey)/Antheil del 1942 e che per pura civetteria tecno-storica riportiamo nella Fig. 9.13. La batteria di condensatori commutati che appaiono sia nel modulatore che nel demodulatore sono quelli che servono per cambiare la frequenza dell'oscillatore a salto.



Figura 9.11 Esempio di spettro di potenza di un segnale FH/SS con slow hopping (a) scala lineare (b) dB - R_s =500 Kbaud, 2H + 1=9, Δf =1 MHz



Figura 9.12 Architettura del Demodulatore FH/SS

In teoria, anche il FH/SS è trasparente rispetto alle prestazione in termi di BER del collegamento, nel momento in cui fissiamo il formato di x(t). L'unica osservazione è che, aldilà di questo schema di principio, quando l'oscillatore della portante salta in frequenza è difficile garantire che ad ogni salto mantenga una fase iniziale nota, cosicché molto spesso si evitano formati di modulazione digitale che richiedano il recupero della fase (PSK o QAM) - come già accennato, si preferisce usare modulazioni di frequenza del tipo FSK, anche multilivello e il modello più appropriato di segnale è il seguente:

$$x_{fh,nc}(t) = x(t) \sum_{\ell} \exp\left\{ j \left(2\pi\alpha[\ell] \Delta f t + \phi_{\ell} \right) \right\} q(t - \ell T_H)$$
(9.26)

dove le fasi ϕ_{ℓ} sono randomiche e indipendenti, e sono quelle che l'oscillatore di portante assume quando salta da una frequenza alla successiva (hopping *non coerente*). Se viceversa sussiste il vincolo (come abbiamo in realtà implicitamente assunto nella nostra modellistica iniziale (9.20)) che la fase delle varie oscillazioni alle varie frequenza viene mantenuta da intervallo di salto ad un altro, si parla di FH/SS *coerente*.

9.2.2 DS/SS a Banda Limitata

Se è vero che con il formato spread-spectrum si intende espandere la banda del segnale digitale, non è detto però che la si debba espandere senz'alcun vincolo. Specialmente per applicazioni commerciali, è necessario controllare strettamente la banda occupata dal segnale per rispettare limiti di canalizzazione nell'ambito della frequenza derivanti dall'assegnazione di una specifica banda, spesso oggetto di concessione da parte di un ente pubblico. Per questo motivo, piuttosto che utilizzare un impulso di chip q(t) di tipo NRZ come ipotizzato nella (9.5), si può optare per un impulso sagomato *a banda limitata*, ad esempio il classico SRFRC g(t) (2.97)-(2.98), mantenendo il resto inalterato:

$$x_{BLss}(t) = \sqrt{2P_x} \sum_{\ell=-\infty}^{+\infty} \gamma[\ell]g(t-\ell T_c)$$
(9.27)

Naturalmente, l'impulso SRFRC è relativo in questo caso ad una frequenza di segnalazione pari al chip rate $1/T_c$, e ha una banda (a frequenza radio) $(1 + \beta)/T_c$. Il formato (9.27) è



Figura 9.13 Dal brevetto originario dell'FH/SS di Lamarr/Antheil (1942)



Figura 9.14 Architettura del modulatore (a) e del demodulatore (b) DS/SS a banda limitata - ρ =4

esattamente quello adottato dalla rete 3G UMTS degli anni 2000, nella quale universalmente, per qualunque apparato di rete o telefonino, R_c =3.84 Mchip/s e β =0.22.

Un formato di questo tipo non può essere generato con il semplice "spreader" di Fig. 9.5 (a). Il segnale limitato in banda viene infatti generato in forma digitalizzata (a tempo discreto) come sequenza $x_{BLss}[n] = x_{BLssl}(nT_{sa})$ dove $f_{sa} = 1/T_{sa}$ è una frequenza di campionamento opportuna, normalmente pari a qualche campione/chip ($f_{sa} = \rho R_c$, ρ fattore di sovracampionamento) e comprende anche la sagomatura di chip, come indicato nella Fig. 9.14 (a). Il filtro di sagomatura provvede anche ad incrementare la frequenza di campionamento di un fattore ρ prima di procedere alla conversione digitale/analogioco. Il corrispondente schema di ricezione è indicato nella Fig. 9.14 (b), e comprende il filtro digitale *adattato* all'impulso di chip che contestualmente effettua la decimazione del medesimo fattore ρ . Il sommatore su M campioni effettua poi la funzione di filtro adattato all'impulso dei dati analogamente al filtro I & D nella Fig. 9.6.

9.2.3 Codici di Spreading

Le sequenze di spreading usate nello spread-spectrum non sono casuali, viceversa seguono una logica ben precisa di generazione. In particolare, la sequenza dei chip $c[\ell]$ è in generale periodica di un certo periodo L che varia molto da applicazione ad applicazione:

$$c[\ell] = c[\ell + L]$$

In molti casi, ad esempio nel downlink di UMTS, si ha che L = M - questo formato si chiama *codice corto* e prevede che la lunghezza del codice sia esattamente pari ad un simbolo dati. In alcuni casi L < M: nel GPS, L = 1023 in modo che la durata del codice sia esattamente pari a 1 ms, e come sappiamo M = 20460. In molti altri casi, $L \gg M$: nell'uplink di UMTS, L = 38400 e $4 \le M \le 256$ (a seconda del bit rate del collegamento). In tutto questo paragrafo, talvolta ci riferiremo implicitamente a chip binari che assumono valori *antipodali* (simmetrici) ± 1 , come abbiamo sempre fatto finora, ma anche a chip binari che assumono i livelli *logici* {0, 1}, e che vengono poi mappati in valori simmetrici nel momento in cui si costruisce il segnale fisico.

In generale, maggiore è la lunghezza di ripetizione del codice L, più la sequenza è assimilabile a una sequenza casuale: il caso limite è il codice di spreading del segnale GPS militare (il cosiddetto *P-code* non utilizzato dai navigatori convenzionali) per il quale $R_c = 10.23$ MChip/s e $L = 6.187104 \cdot 10^{12}$ pari a un tempo di ripetizione di 604,800



Esempio di LFSR con $P = 6 e G(x) = 1 + x + x^{2} + x^{4} + x^{6}$ Figura 9.15

secondi ovvero esattamente una settimana. Vediamo di seguito alcune categorie notevoli di codici di spreading.

9.2.3.1 Codici Memorizzati Ouando la seguenza del codice non è particolarmente lunga, viene direttamente memorizzata nel trasmettitore e nel ricevitore in una ROM - questo è ad esempio il caso delle sequenze di spreading con L = 4096 del sistema GNSS europeo GA-LILEO che sono state ricavate attraverso un'ottimizzazione numerica e non sono generabili attraverso un algoritmo semplice.

9.2.3.2 Codici a Massima Lunghezza Il modo più semplice ma efficace per generare una sequenza binaria pseudocasuale è basato su di un Linear Feedback Shift Register (LFSR), cioè un registro di memoria a P elementi la cui uscita viene richiusa in feedback sull'ingresso, previa somma di un certo numero di contenuti intermedi del registro stesso, come mostrato nell'esempio di Fig. 9.15. Il registro viene caricato in fase di reset con una parola di inizializzazione e poi viene lasciato libero di marciare a tempo di clock di chip R_c . La particolare sequenza che viene generata è determinata da quanti e quali ingressi intermedi vengono sommati; in particolare la periodicità della sequenza è legata alle caratteristiche di un particolare polinomio G(x) associato al generatore: x rappresenta l'elemento di ritardo unitario di ciascun flip-flop del registro, quindi x^k denota l'ingresso del registro ritardato di k colpi di clock (cioè il contenuto dell'elemento di memoria k-esimo), con k che va da 0(l'ingresso stesso) a P (l'uscita $c[\ell]$). Nell'esempio di Fig. 9.15, $G(x) = 1 + x + x^2 + x^4 + x^6$. È chiaro che la sequenza di valori in uscita $c[\ell]$ è determinata dalla sequenza degli stati "visitati" dal LFSR: quest'ultimo è infatti una macchina sequenziale a $N_s = 2^P$ stati, dati dai 2^{P} possibili contenuti distinti del registro ad un dato istante di clock. La massima lunghezza di ripetizione della sequenza si ottiene se la configurazione è tale per cui gli stati vengono visitati tutti senza ripetizioni e uno di seguito all'altro in una certa successione. Tutti, meno uno stato proibito che è assorbente, cioè se per caso viene in qualche modo raggiunto, non è possibile uscirne: lo stato 000...00. Dunque la perdiocità massima è $L = 2^{P} - 1$; quando la sequenza è di questo tipo viene chiamata a massima lunghezza. Dato un certo valore di P (ordine) del registro, le sequenze ML di quell'ordine sono relativamente poche, e i polinomi che le generano sono facilmente reperibili in rete.

Le sequenze ML, che vengono anche chiamate PRBS, Pseudo Random Binary Sequences o PN, Pseudo Noise, hanno particolari proprietà che le rendono appunto simili a sequenze casuali. Ad esempio, nel periodo $L = 2^{\bar{P}} - 1$, il numero di '0' logici e '1' logici sono rispettivamente L/2 - 1 e L/2 - poichè normalmente L è grande, ci sono in pratica tanti '1' quanti '0' e la sequenza è bilanciata. Inoltre (e soprattutto), si trova che la sequenza di

337



Figura 9.16 Autocorrelazione di una sequenza a Massima Lunghezza - L = 63

correlazione periodica $R_c[m] = R_c[m+L]$ di un codice ML è

$$R_{c}[m] \stackrel{\triangle}{=} \frac{1}{L} \sum_{\ell=0}^{L-1} c[\ell] \cdot c[\ell+m] = \begin{cases} 1 & m=0\\ -1/L & \text{altrimenti} \end{cases}, \quad 0 \le m \le L-1 \quad (9.28)$$

Quando L è grande, la forma di questa sequenza, vista nell'ambito del periodo di ripetizione, è molto simile a $\delta[m]$ (vedi Fig. 9.16) - la sequenza è approssimativamente delta-correlata e quindi il suoi spettro di potenza è *piatto*, come quello del segnale casuale per antonomasia, cioè il *rumore bianco*. Le sequenze ML vengono tipicamente usate come codici "lunghi" cioè con $L \gg M$.

9.2.3.3 Codici di Walsh-Hadamard Abbiamo già incontrato nel paragrafo 2.8.2.3 l'insieme di Walsh-Hadamard degli L codici ortogonali di lunghezza L potenza di 2 che vengono usati per effettuare il Code-Division Multiplexing. Non c'è niente di nuovo da osservare qui, salvo notare che i codici non sono in generale assimilabili a una sequenza PN (si veda la prima riga della matrice \mathbf{H}_8 (2.138)...), e che vengono usati in configurazione di "codice corto" con L = M.

9.2.3.4 Codici di Gold I codici di Gold sono una famiglia di L + 2 codici di lunghezza $L = 2^P - 1$ ottenuti a partire da due sequenze ML $c_1[\ell]$ e $c_2[\ell]$ diverse ma della stessa lunghezza L, e selezionate in modo particolare (*preferred pair*). Come rappresentato nella Fig. 9.17, il codice di Gold *i*-esimo $g_i[\ell]$, i = 0, ..., L - 1 si ottiene sommando la $c_1[\ell]$ con la $c_2[\ell]$ traslata ciclicamente nel tempo di *i* colpi di clock:

$$g_i[\ell] \stackrel{\Delta}{=} c_1[\ell] \oplus c_2[\ell+i] \tag{9.29}$$

Oltre alle L sequenze così ottenute, le ulteriori sequenze $g_L[\ell]$ e $g_{L+1}[\ell]$ sono le ML originarie $c_1[\ell]$ e $c_2[\ell]$. Questi codici sono ad esempio usati con L = 1023 nel segnale satellitare GPS, uno distinto per ogni satellite (il cosiddetto codice C/A). Sono utilizzati perchè sono quasi-ortogonali: la cross-correlazione tra due qualunque codici assume uno tra i 3 possibili valori -1/L, $-\mu(P)/L$, $(\mu(P) - 2)/L$ dove $\mu(P) = 1 + 2^{(P+1)/2}$ se P è dispari o $\mu(P) = 1 + 2^{(P+2/2)}$ se P è pari. Nel caso del GPS, P = 10, $\mu(P) = 65$ e i tre valori sono -1/1023, -65/1023, 63/1023, tutti molto piccoli rispetto a 1 - se in generale



Figura 9.17 Generatore dei Codici C/A con L = 1023 dei satelliti GPS

L è molto grande, i tre valori sono circa $\pm 2/\sqrt{L}$ e -1/L. Questi tre valori sono anche quelli assunti dall'*auto*correlazione $R_c[m]$ di uno qualunque dei codici (oltre naturalmente al valore 1 per *m*=0). Infatti, rispetto ai codici di WH (esattamente ortogonali e quindi migliori dal punto di vista della cross-correlazione), i codici di Gold hanno (molto) migliori proprietà di autocorrelazione, assomigliano cioè a una sequenza PN molto più dei WH, e questo facilita la sincronizzazione di codice del ricevitore.

9.2.3.5 Codici di Kasami Un'altra famiglia di codici quasi ortogonali (cioè a bassa cross-correlazione) è l'insieme di Kasami - in realtà di tratta di *due* insiemi, il *piccolo* e il *grande*.

La generazione dei codici del *piccolo insieme di Kasami* segue un criterio simile a quello dei codici di Gold: si parte stavolta da una unica sequenza ML con *P pari* $c_{ML}[\ell]$, e da questa si deriva una sequenza secondaria a[m] di lunghezza $L' = 2^{P/2} - 1$ mediante *decimazione*, cioè considerando soltanto i valori di $c_{ML}[\ell]$ intervallati di $2^{P/2} + 1$: $a[m] = c_{ML}[m(2^{P/2} + 1)]$, e che risulta ML *essa stessa*. Fatto questo, si crea una ulteriore sequenza $\alpha[\ell]$ di lunghezza L, in modo che risulti localmente periodica: $\alpha[\ell] = a[|\ell|_{L'}]$, $\ell = 0, 1, ..., L$. Infine, le sequenze dell'insieme piccolo di Kasami $\kappa_i[\ell]$, i = 0, ..., L' - 1si ottengono sommando la sequenza ML originaria con una tra tutte le sequenze ottenute dalla traslazione ciclica della nuova sequenza α decimata/periodicizzata, e poi integrando l'insieme con la ML originaria:

$$\begin{cases} \kappa_i[\ell] \stackrel{\Delta}{=} c_{ML}[\ell] \oplus \alpha[\ell+i] & i = 0, ..., L' - 1\\ \kappa_{-1}[\ell] = c_{ML}[\ell] & i = -1 \end{cases}$$

$$(9.30)$$

Quindi il piccolo insieme è costituito da $L' + 1 = 2^{P/2}$ sequenze di lunghezza $L = 2^P - 1$. Queste sequenze sono "poche" (circa la metà delle Gold di pari lunghezza) ma hanno eccellenti proprietà di auto- e cross-correlazione, le quali assumono al variare di m, come le Gold, tre possibili valori: -1/L, $-\mu(P)/L$, $(\mu(P) - 2)/L - 1/L$, dove stavolta $\mu(P) = 2^{P/2} + 1$. Se L è grande, gli ultimi due valori sono all'incirca uguali a $\pm 1/\sqrt{L}$, minori di quelli delle Gold.

Il grande insieme di Kasami contiene tutti i codici di Gold e quelli del piccolo insieme che abbiamo appena visto, più altri codici che si ottengono componendo tre diverse sequenze:

in primis, la consueta $c_{ML}[\ell]$ di lunghezza $L = 2^P - 1$, dove però P deve essere del tipo P = 4Q + 2 con Q intero (cioè $|P|_4 = 2$). Oltre a questa ML, definiamo $a[m] = c_{ML}[m(2^{P/2} + 1)]$, di lunghezza $L' = 2^{P/2} - 1$ ottenuta come per il piccolo insieme per decimazione del fattore $2^{P/2} + 1$, e che poi periodicizziamo fino a lunghezza L: $\alpha[\ell] = a[|\ell|_{L'}]$. Definiamo poi una ulteriore $b[m] = c_{ML}[m(2^{P/2+1} + 1)]$, di lunghezza $L'' = 2^{P/2-1} - 1$ ottenuta per decimazione del fattore $2^{P/2+1} + 1$ che periodicizziamo per ottenere $\beta[\ell] = b[|\ell|_{L''}]$. I codici $\bar{\kappa}_{i,j}[\ell]$ del grande insieme di Kasami si ottengono come segue:

$$\bar{\kappa}_{i,j}[\ell] \stackrel{\triangle}{=} \begin{cases} c_{ML}[\ell] & i = -2; j = -1\\ \beta[\ell] & i = -1; j = -1\\ c_{ML}[\ell] \oplus \beta[\ell+i] & i = 0, \dots, L-1; j = -1\\ c_{ML}[\ell] \oplus \alpha[\ell+i] & i = -2; j = 0, \dots, L'-1\\ \beta[\ell] \oplus \alpha[\ell+i] & i = -1; j = 0, \dots, L'-1\\ c_{ML}[\ell] \oplus \alpha[\ell+i] \oplus \beta[\ell+j] & i = 0, \dots, L-1; j = 0, \dots, L'-1 \end{cases}$$
(9.31)

In questo modo si ottiene un insieme di $2^{3P/2}$ sequenze (cioè *molte*, all'incirca \sqrt{L} volte le sequenze di Gold di pari lunghezza $L = 2^P - 1$), le cui auto-e cross-correlazioni assumono stavolta *cinque* possibili valori: -1/L, $(-1 \pm 2^{P/2})/L$, $(-1 \pm 2^{P/2+1})/L$ (queste ultime approssimativamente $\pm 1/\sqrt{L} e \pm 2/\sqrt{L}$). Dato che l'insieme è molto vasto, viene utilizzato per creare sequenze *uniche*, cioè identificative in modo unico di ogni terminale utente, per effettuare lo spreading in uplink nella rete 3G.

9.2.3.6 Codici Compositi Quando si usa il DS/SS per effettuare un multiplexing (cioè in versione CDM, vedi il paragrafo 2.8.2), i codici di prima scelta sono gli WH ortogonali, alcuni dei quali hanno proprietà di auto- e cross-correlazione del tutto insufficienti. Una maniera di mantenere l'ortogonalità e migliorare la correlazione è quella di combinare i WH con un *altro* codice ottenendo i codici cosiddetti *compositi*. Se indichiamo con $w_i[\ell]$ l'*i*-esimo codice WH, $i = 0, ..., L - 1, L = 2^P$, il set di codici compositi è

$$c_i^{(c)}[\ell] \stackrel{\Delta}{=} w_i[\ell] \otimes s^{(c)}[\ell] \ , \ i = 0, ..., L - 1$$
 (9.32)

dove $s^{(c)}[\ell]$ è il codice di *scrambling* (tipicamente un codice di Gold o ML se necessario accorciato) ed è *unico* per tutto il set (ad esempio, si applica a tutti gli utenti di downlink in una cella ed è di fatto identificativo della cella stessa). In fase di ricezione, per prima cosa si rimuove chip a chip il codice di scrambling $s^{c}[\ell]$, fatto questo si procede al despreading usuale con il codice WH per recuperare il particolare utente del multiplex. Da questo punto di vista, i codici WH vengono chiamati codici di *canalizzazione*.

In generale, la lunghezza dei codici di canalizzazione e di quelli di scrambling è differente: nel caso di quelli di canalizzazione, sappiamo che deve essere L = M in modo che il codice sia esattamente lungo quanto un simbolo dati (per avere ortogonalità); il codice di scrambling è del tutto indipendente da questo vincolo, e può avere una lunghezza anche (molto) maggiore di un simbolo (in UMTS downlink, il codice di scrambling è lungo 38400 chip e copre un intero burst di dati di 10 ms).

9.2.4 Spread-Spectrum e Capacità di Shannon

La proprietà-chiave dei segnali spread-spectrum di occupare una banda molto maggiore del bit-rate ha un'ulteriore conseguenza relativamente alla capacità di Shannon su AWGN

ALICE, BOB, EVE & MALLORY 341



Figura 9.18 I Personaggi nel Teatro delle Comunicazioni Sicure

intrinseca a tale formato. Ripartiamo dalla relazione generale "di buon pilotaggio" per AWGN a banda limitata (4.102)

$$\frac{E_b}{N_0} \ge \frac{2^{R_b/B} - 1}{R_b/B}$$
(9.33)

e particolarizziamola allo spread-spectrum. Senza entrare in dettagli riguardo la relazione tra bit rate, chip rate e symbol rate, in generale la definizione di segnale spread-spectrum richiede che $B \gg R_b$, e quindi comporta un'efficienza spettrale $R_b/B \ll 1$, che è ovviamente insoddisfacente per applicazioni commerciali, ma che può portare, come si vede dalla Fig. 4.23, un'altissima efficienza energetica: con un opportuno MOD/COD, *molto alta*, il formato spread-spectrum può raggiungere il limite del -1.7 dB.

Il motivo dell'alta efficienza energetica è stato spiegato in pratica nel paragrafo 2.8.5, e in particolare nella Fig. 2.50 in cui già si è accennato all'applicazione di codici a protezione d'errore per il CDMA: usando segnali intrinsecamente spread-spectrum, è possible combinare codifica e spreading in modo che la banda del segnale resti costantemente pari a $1/T_c$ anche con codici a tassi *r molto bassi* e quindi con eccellenti proprietà di guadagno di codifica. Torneremo su questa caratteristica nel paragrafo 9.5.2 che interpreterà ancora meglio i risultati della Fig. 2.51 sull'efficienza del CDMA.

9.3 Alice, Bob, Eve & Mallory

Il vero regno dello spread-spectrum è quello delle *comunicazioni sicure*, cioè resilienti ai vari tipi di attacco o di intercettazione o di comportamenti maliziosi fin dal *livello fisico*. I ben noti personaggi di questo teatro sono quelli del titolo del paragrafo (Fig. 9.18: Alice e Bob (Trasmettitore A e Ricevitore B): gli utenti legittimi ai due capi del collegamento digitale; Eve (*Eavesdropper*), colei che tenta di intercettare passivamente la comunicazione destinata a Bob; Mallory (*Malicious*), che tenta di bloccare la comunicazione o addirittura di sostituirsi ad Alice per ingannare Bob con comunicazioni non veritiere - e potremmo continuare con ulteriori personaggi che interpretano altre parti su questo palcoscenico.

9.3.1 Jamming

L'azione più semplice che Mallory può effettuare attivamente, è quella di cercare di *impedire* la comunicazione legittima tra Alice e Bob. In campo radio, è semplice realizzare il cosid-



Figura 9.19 Esempio di Jammer multi-banda (3G, 4G, 5G, Wi-Fi 2.4/5 GHz, GPS/GALILEO, UHF/VHF, Telecomandi



detto *jamming* della trasmissione piazzando nelle vicinanze di Bob (ricevitore) un apparato simile a quello di Fig. 9.19 che trasmette un disturbo destrutturato (tipicamente, rumore elettronico molto amplificato, cioè il *jam* (la marmellata)) sulla banda di comunicazione $B = 1/T_s$. Proprio perché il jammer è nelle vicinanze, la sua potenza radio P_J ricevuta da Bob è in genere (molto) più grande di quella P_R ricevuta da Alice. Il cosiddetto signal-tointerference ratio SIR = P_R/P_J è normalmente (molto) minore di 1, e la comunicazione è del tutto impedita.

9.3.1.1 Narrowband Jamming Immaginiamo però che, accortisi del jamming, A & B cambino il loro formato di trasmissione in DS/SS (9.5); possiamo modellare la situazione vista da Bob come in Fig. 9.20 dove J(t) è il segnale di jamming prodotto da Mallory. È interessante seguire i cambiamenti nello spettro del segnale utile e di jamming lungo la catena di elaborazione del ricevitore: nel punto (a) in cui $r(t) = x_{ss}(t) + J(t)$ vediamo lo spettro "lecito" di Alice in formato DS/SS e quindi su di una banda $1/T_c = M/T_s$, insieme con il *jammer a banda stretta* sulla banda $B = 1/T_s$ (Fig. 9.21 (a)). Nel punto (b), cioè dopo despreading, abbiamo z(t) = x(t) + J(t)c(t): lo spettro lecito è stavolta a banda stretta $1/T_s$, mentre dobbiamo capire che succede al jammer, che viene moltiplicato per lo spreading per $x_{ss}(t)$, e il jammer diventa quindi a *banda larga* (Fig. 9.21 (b)), ma il SIR, sia in (a) che in (b), è il medesimo. Se però osserviamo quel che succede in (c) (Fig. 9.21 (c)), cioè dopo il filtro adattato, notiamo che lo spettro del segnale è sostanzialmente inalterato (il filtro adattato ha per definizione la stessa banda del segnale), mentre lo spettro del jammer

viene modificato dal filtro, mantenendone solo le componenti nella banda passante. Nel punto \bigcirc quindi, il SIR è parecchio cambiato: ragionando come in nel paragrafo 2.3.3.2, troviamo che la potenza del segnale è nominale (per definizione di filtro adattato), cioè P_x resta invariata, mentre la potenza del jammer si calcola facilmente osservando che il suo spettro è sostanzialmente piatto nella banda del filtro adattato $1/T_s$, e può essere considerato come rumore bianco - la sua potenza è quindi

$$P_{J,c} = \frac{1}{2} \frac{2P_J}{1/T_c} \frac{1}{T_s} = \frac{P_J}{T_s/T_c} = \frac{P_J}{M}$$
(9.34)

dove il fattore 1/2 tiene conto del fatto che stiamo ragionando sugli equivalenti in banda base, $2P_JT_c$ é la d.s.p. del rumore bianco equivalente al jammer $2J_0$ e T_s è il tempo di integrazione del filtro adattato. Dunque,

$$SIR_c = \frac{P_x}{P_J/M} = M \cdot SIR \tag{9.35}$$

Usare un segnale spread-spectrum fornisce un guadagno intrinseco nei confronti di un jammer a banda stretta di un fattore M rispetto al caso di segnale pure a banda stretta. Per questo la quantità M si chiama anche *processing gain*. Se, anzichè considerare il rapporto segnale rumore, si considera il rapporto E_b/J_0 e formati di segnale che comprendono anche codici a protezione d'errore e/o costellazioni multilivello, si trova che il processing gain coincide invece con la quantità $G_p = T_b/T_c$.

9.3.1.2 Wideband Jamming Sapendo che A & B usano spread-spectrum, Mallory potrebbe decidere di usare anch'egli un jammer a banda larga, su di un banda pari a $1/T_c$ come quella del segnale legittimo. La situazione nel punto (a) sarebbe stavolta quella di Fig. 9.22, che sembra (più) problematica. Se però considero quel che succede nel punto (b) le cose non sono così drammatiche: quando parliamo di *jamming* con un segnale pseudo-rumore destrutturato, intendiamo che Mallory *non* conosce il codice di spreading di Alice/Bob. Può anche usare un codice suo con lo stesso chip-rate di A & B per occuparne la stessa banda, ma il punto è che quando effettuiamo il despreading con il codice legittimo, lo spettro del jammer *non* subisce despreading - al contrario, resta a banda larga con la medesima d.s.p. $2P_JT_c$. La situazione in (b) e (c) è quindi esattamente la stessa del caso a banda stretta: il *medesimo processing gain* anche per jammer a banda larga.

9.3.2 Eavesdropping/Low Probability of Intercept

Le cose cambiano se attori diversi da Alice e Bob conoscono o sono in grado di ricostruire il codice di spreading del collegamento. Allora Eve può tranquillamente e in modo passivo, cioè molto difficilmente rilevabile, ricevere la comunicazione destinata a Bob - svolgere cioè la funzione di *intercettazione* (eavesdropping). È chiaro che in questo contesto di comunicazioni sicure, la sequenza di spreading c(t), per garantire sicurezza, assume proprio la connotazione di *codice* e l'operazione di spreading assume le caratteristiche di una *cifratura* vera e propria svolta a livello fisico (*physical security*). Da questo punto di vista, i codici come WH, Gold, Kasami, che abbiamo introdotto precedentemente sono del tutto inadatti perchè, essendo prodotti da circuiti lineari, possono essere decifrati con attacchi semplicissimi a complessità altrettanto lineare. Il P-code del GPS militare (peraltro noto solo ad attori autorizzati) è generato con algoritmi non-lineari e, anche per la grande lunghezza, risulta ad oggi non decifrabile.

344 QUANDO IL DISTURBO È PIÙ FORTE DEL SEGNALE - COMUNICAZIONI SPREAD-SPECTRUM



Figura 9.21 Spettri di segnale e jammer a banda stretta lungo la catena di ricezione - M = 8, SIR=0 dB - Linea blu: jammer; linea rossa: segnale legittimo



Figura 9.22 Spettri di segnale e jammer a banda larga all'ingresso del ricevitore - M = 8, SIR=-3 dB

ALICE, BOB, EVE & MALLORY 345



Figura 9.23 Quando il disturbo è più forte del segnale...

In ogni caso, lo spread-spectrum ha un certo grado di difesa (anche) contro l'interecettazione. Infatti, la maniera standard per capire che Alice sta effettuando una qualche comunicazione radio con Bob è quella di osservare su di un analizzatore di spettro la banda di frequenza abitualmente usata e cercare di rivelare lo spettro della comunicazione. La Figura 9.23 rappresenta questa situazione: sul display dell'analizzatore di spettro sono ben visibili du spettri, uno (simil-LTE) a 1.9 GHz e l'altro (simil-Wi-Fi) a 2.4 GHz relativi a due trasmissioni a banda (relativamente) stretta: sono molto chiaramente identificabili al di sopra del livello -30 dB del rumore di ricezione. Viceversa, lo spettro in nero rappresenta un FH/SS su più di 1 GHz di banda il cui spettro, molto espanso, *resta al disotto del livello del rumore* (ricordate il titolo di questo capitolo?) e non può essere facilmente rivelato.

Esempio 9.49

Anche senza voler intenzionalmente "nascondersi", spesso il segnale spread-spectrum si trova *comunque* al di sotto del livello del rumore di ricezione: nel caso del GPS, il livello nominale² di ricezione del segnale è P_x =-125 dBm, mentre il rumore di ricezione (termico) totale nella banda di ricezione a radiofrequenza $B_x = 2 \times 1.023 = 2.046$ MHz è pari a -110 dBm, cioè 15 dB *maggiore* ! Quando effettuiamo il despreading però, esattamente come nel caso del jamming, abbiamo un guadagno pari al processing gain M = 20460 = 43 dB, e quindi il rapporto segnale rumore sul dato digitale trasmesso dal satellite è in realta -15+43=28 dB che permette anche un buon margine per ulteriori perdite, attenuazioni ecc.

²Il satellite GPS a 20,200 km di altitudine trasmette una potenza di 45 W alla frequenza cosiddetta L1 pari a $f_0 = 1575.43MHz$, con un guadagno d'antenna $G_T = 12dB$; se l'antenna di ricezione ha un guadagno $G_R = 4dB$, e assumendo perdite atmosferiche pari a 5 dB, usando la (10.100) otteniamo il valore citato

9.3.3 Spoofing

Nel momento in cui il codice di A & B è noto o può essere decifrato, Mallory ha l'opzione di usarlo per cercare di sostituirsi ad Alice e creare una *comunicazione ingannevole* (non legittima) che Bob riconosce come autentica. Questa tecnica (concettualmente simile al Trojan diffuso su Internet) prende il nome di *spoofing*, e può essere smascherata solo utilizzando appositi protocolli o sistemi di autenticazione il cui studio esula dallo scopo di queste note, e più propriamente appartiene al dominio della sicurezza informatica (cybersecurity). Al di là e oltre la sicurezza a livello fisico che lo spread-spectrum può conferire, esistono infatti (molte) altre tecniche di protezione del messaggio digitale e di riconoscimento degli attacchi informatici che si volgono ai vari livelli superiori della gerarchia OSI della rete.

9.4 DS/SS in ambiente Multipath

La forma peculiare del segnale spread-spectrum richiede di esaminare con attenzione le peculiarità e prestazioni di questa tecnologia quando viene utilizzata per effettuare un collegamento digitale in ambiente multipath (7.2). Vedremo che la banda espansa del segnale, non direttamente significativa dell'entità del bit-rate, porta dei *vantaggi* rispetto alle potenziali distorsioni del canale (nominalmente selettivo in frequenza vista la banda larga del segnale).

9.4.1 Cancellazione del Multipath

Per capire se e come il canale multipath incida sulla ricezione di un segnale spread spectrum, partiamo con un caso semplificato ma significativo: in primis, consideriamo il semplice canale a due raggi (7.10); inoltre, immaginiamo di considerare un formato DS/SS a codice corto, cioè per il quale la forma d'onda del codice si esaurisce e si ripete nell'ambito dell'intervallo di simbolo dati:

$$s(t) = \sum_{n} a[n]c(t - nT_s) \ , \ c(t) = \sum_{\ell=0}^{M-1} c_{\ell}q(t - \ell T_c)$$
(9.36)

All'uscita del canale radio, avremo

$$x(t) = s(t) + be^{j\phi}s(t-\tau) = \sum_{n} a[n] \left\{ c(t-nT_s) + be^{j\phi}c(t-nT_s-\tau) \right\}$$
(9.37)

che sembra caratterizzato in generale dalla consueta interferenza intersimbolica già analizzata nell'Esempio 35.

Vediamo però che cosa accade all'uscita del ricevitore a despreading-matched filter di Fig. 9.6, adottando il medesimo approccio utilizzato nel ricavare la (9.16), cioè calcolando il valore atteso rispetto ai chip del codice di spreading (operatore E_c):

$$z[0] = \frac{1}{T_s} \int_0^{T_s} \mathbf{E}_c \{x(t)c(t)\} dt =$$

= $\sum_n a[n] \frac{1}{T_s} \int_0^{T_s} \mathbf{E}_c \{c(t - nT_s)c(t) + be^{j\phi}c(t - nT_s - \tau)c(t)\}$
= $\sum_n a[n] \{\rho_c(nT_s) + be^{j\phi}\rho_c(\tau + nT_s)\}$ (9.38)


Figura 9.24 Funzione di autocorrelazione di un codice ML con impulsi di chip NRZ

dove $\rho_c(\tau)$ è la funzione di autocorrelazione media del codice di spreading (9.19). L'andamento di questa funzione indica che il *tempo di correlazione* del codice di spreading, proprio perchè la sequenza dei chip è di tipo pseudo-random, è molto piccolo, pari in particolare al tempo di chip: il codice $c(t - \tau)$ diventa *incorrelato* con c(t) (proprio perchè noise-like) non appena la traslazione temporale τ è maggiore (in modulo) di un tempo di chip T_c .

Consideriamo ora i due termini nella sommatoria della (9.38): proprio alla luce dell'andamento dell'autocorrelazione del codice, è chiaro che il primo termine è diverso da zero solo per n = 0, e quindi il risutato della somma è pari ad a[0], che corrisponde peraltro alla consueta componente "utile" dopo despreading derivante dal raggio LOS del multipath (la medesima che avremmo su canale ideale privo di multipath). Per il secondo termine, dobbiamo capire di che entità è il ritardo del multipath rispetto al tempo di chip (e tempo di correlazione del codice) T_c . Il caso più favorevole, che si verifica spesso in pratica, è che $T_c < \tau$ (ma anche $\tau < T_s$): in questo caso, la componente di secondo raggio del multipath risulta per quanto appena detto incorrelata con il codice di despreading, e quindi viene di fatto *cancellata* dall'uscita del ricevitore e riconduce le prestazioni del ricevitore a quelle di un canale AWGN.

Questa caratteristica di *cancellazione del multipath* è tipica dei segnali spread-spectrum, anche nella forma FH/SS. In questo caso infatti, la banda occupata dal segnale (pensiamo allo slow-hopping) è pari a $H\Delta f + B_x \simeq H\Delta f$ e quindi il tempo di correlazione del segnale è uguale a $1/H\Delta f$ - la cancellazione avviene se $\tau > 1/H\Delta f$ e quindi è tanto più probabile/efficace quanto maggiore è la banda espansa del segnale. Inoltre, la cancellazione avviene anche nel caso di multipath "ricco" (7.2) con molte componenti riflesse, purchè $\tau_i > T_c$ (ovvero $\tau_i > 1/H\Delta f$) $\forall i$. Questa condizione è quasi sempre verificata in ambienti outdoor, ma potrebbe non essere verificata per comunicazioni wireless in ambienti indoor in cui i ritardi del multipath sono molto piccoli.

Di fatto, la forma d'onda c(t) del codice non è perfettamente casuale, e quindi la componente secondaria del multipath non sarà mai perfettamente cancellata. Ad esempio, se la sequenza dei chip nella (9.36) è una ML, si trova facilmente che la funzione di autocorrelazione periodica è come in Fig. 9.24, e quindi il multipath non viene completamente eliminato - naturalmente, se M = L è sufficientemente grande, la cancellazione avviene approssimativamente e in pratica è efficace.

9.4.2 Ricevitore "Rake"

Dunque un ulteriore aspetto favorevole dei formati spread-spectrum è quello della sostanziale cancellazione del multipath, efficace quando la banda occupata è molto larga. È però vero che cancellare le componenti riflesse, se da una parte evita problemi di di distorsione del segnale e/o di interferenza intersimbolica, dall'altra rischia di non sfruttare la potenza delle componenti di multipath oltre la LOS che, anzichè essere considerate utili perchè comunque portano informazione relative al simbolo a[0], vengono cancellate e quindi non considerate per niente - quest'approccio potrebbe risultare sub-ottimo quando lo SNR non è favorevole. Esiste un approccio alternativo che risulti ottimo nei confronti del rumore di ricezione?

9.4.2.1 Derivazione del ricevitore ottimo Riprendiamo la (9.37) ed esprimiamola in una forma equivalente, aggiungendo anche la compente di disturbo AWGN:

$$r(t) = \sum_{n} a[n]\beta(t - nT_s) + w(t) \quad , \quad \beta(t) \stackrel{\Delta}{=} c(t) + be^{j\phi}c(t - \tau)$$
(9.39)

La forma d'onda a valori complessi $\beta(t)$ è in pratica il codice di spreading modificato dal canale multipath - dunque dalla 9.39 vediamo che il problema di ricezione ottima del segnale DS/SS si riconduce a quello della ricezione di un segnale digitale modulato con un impulso di simbolo particolare, e cioè proprio $\beta(t)$. Per semplificare la questione, immaginiamo una trasmissione *one-shot* del solo simbolo a[0]:

$$r(t) = a[0]\beta(t) + w(t)$$
(9.40)

Evidentemente, il ricevitore ottimo su AWGN è quello a filtro adattato o a correlazione; prendendo questa seconda versione, la variabile di decisione (normalizzata) è

$$z[0] = \frac{1}{T_s} \int_0^{T_s + \tau} r(t) \beta^*(t) dt$$
(9.41)

dove abbiamo tenuto conto che $\beta(t)$, vista la presenza del multipath, ha una durata pari a $T_s + \tau$ (vedi Esempio 35). Sviluppiamo il processing del ricevitore:

$$z[0] = \frac{1}{T_s} \int_0^{T_s + \tau} r(t) \left[c(t) + b e^{-j\phi} c(t - \tau) \right] dt$$
$$= \frac{1}{T_s} \int_0^{T_s} r(t) c(t) dt + b e^{-j\phi} \frac{1}{T_s} \int_{\tau}^{T_s + \tau} r(t) c(t - \tau) dt = z_1[0] + b e^{-j\phi} z_2[0] \quad (9.42)$$

dove gli estremi di integrazione considerano il fatto che $c(t) \neq 0$ solo se $0 \leq t < T_s$. Per costruire questa variabile di decisione, il ricevitore deve per prima cosa *stimare il canale multipath*, cioè procurarsi dei valori stimati dei parametri b, ϕ , τ , procedere al *calcolo* delle due *correlazioni* parziali $z_1[0]$ e $z_2[0]$ relative alle due versioni del codice di spreading provenienti dai due cammini del canale, e infine *combinare* le due correlazioni parziali con degli opportuni coefficienti, rispettivamente 1 e $be^{-j\phi}$, pari al complesso coniugato dei guadagni complessi dei vari cammini.

Anche se abbiamo derivato la struttura del ricevitore ottimo per il caso di 2 raggi, questa struttura si generalizza facilmente al caso di un numero di raggi N_p generico come nel modello generale (7.2), prevedendo la stima di tutti i $3N_p$ parametri di canale a_i , ϕ_i , τ_i , $i = 1, ..., N_p$, il calcolo di N_p correlazioni parziali $z_1[0],...,z_{N_p}[0]$, e la combinazione delle correlazioni parziali secondo i coefficienti $a_i e^{-j\phi_i}$. La struttura del ricevitore risultante è mostrata nella Fig. 9.25 e prende il nome di "Rake" (rastrello, intendendo che ogni ramo di correlazione rappresenta un dente del rastrello da giardiniere).

Abbiamo usato la parola-chiave *combinare* relativamente alla somma pesata delle varie correlazioni $z_i[0]$ perché di fatto il Rake è un ricevitore che sfrutta la *diversità temporale*

DS/SS IN AMBIENTE MULTIPATH 34



Figura 9.25 Architettura del ricevitore "Rake"

dei cammini multipath e che può sfruttare un guadagno d'array - i coefficienti del Rake sono quelli relativi alla strategia MRC (10.6). Questa diversità è sfruttabile non appena i vari cammini del canale possono essere *risolti*, cioè distinti l'uno dall'altro, e cioè quando i ritardi differiscono tra di loro di *più di un chip*; se così non è, i ritardi non possono essere chiaramente distinti (risolti), la stima dei coefficienti del multipath è imperfetta, e il guadagno di array è inferiore al teorico.

9.4.2.2 Calcolo delle prestazioni La struttura del ricevitore Rake è ottima su canale AWGN perchè riesce a sfruttare *tutta* la potenza utile di segnale proveniente da tutti i cammini multipli, ma prescinde dalla presenza di interferenza intersimbolica che potrebbe nascere considerando, oltre ad a[0], anche tutti gli altri simboli dati. In pratica, il ricevitore Rake viene implementato con un numero di "finger" abbastanza limitato (da 3 a 5). La complessità, più che nel processing dei vari rami, sta nella *stima dei parametri di canale* che deve essere inizializzata e mantenuta (rinfrescata) per tutta la durata del collegamento, e può risultare problematica per alta mobilità del trasmettitore/ricevitore.

Le prestazioni del ricevitore Rake possono essere messe a confronto con quelle del semplice ricevitore a correlazione (despreading-filtro adattato) calcolando il miglioramento del rapporto SNR relativo a z[0] nell'un caso e nell'altro, limitandoci per semplicità al caso del canale a due raggi. Per un segnale BPSK con ricevitore a correlazione in cui la cancellazione del multipath è perfetta (perchè $\tau > T_c$), abbiamo che

$$z[0] = \sqrt{2P_s}a[0] + n[0] \quad \Rightarrow \quad SNR_{cor} = \frac{2P_s}{N_0/T_s} \tag{9.43}$$

dove P_s è la potenza di segnale *ricevuta soltanto dal raggio diretto* e dove N_0/T_s è la potenza della sola parte reale della componente di rumore n[0]: la costellazione BPSK è

349

reale essa stessa, e la componente in quadratura di rumore non deve essere considerata. Per il ricevitore Rake invece, dobbiamo esaminare la forma di z[0]:

$$z[0] = z_1[0] + be^{-j\phi} z_2[0] =$$

$$= \left(\sqrt{2P_s}a[0] + n_1[0]\right) + be^{-j\phi} \left(be^{j\phi}\sqrt{2P_s}a[0] + n_2[0]\right)$$

$$n_1[0] \stackrel{\triangle}{=} \frac{1}{T_s} \int_0^{T_s} w(t)c(t) dt \ , \ n_2[0] \stackrel{\triangle}{=} \frac{1}{T_s} \int_{\tau}^{\tau+T_s} w(t)c(t-\tau) dt \tag{9.44}$$

Le due variabili aleatorie $n_1[0]$ ed $n_2[0]$ sono congiuntamente Gaussiane (perchè derivanti da trasformazioni lineari di un processo Gaussiano w(t)), entrambe a media nulla e con componenti I/Q indipendenti aventi varianza $\sigma_1^2 = \sigma_2^2 = N_0/T_s$. Quest'ultimo risultato deriva dal fatto che w(t)c(t) e $w(t)c(t-\tau)$ hanno le stesse proprietà statistiche di w(t) (vedi (9.12)). La variabile di decisione finale è

$$z[0] = \sqrt{2P_s}a[0](1+b^2) + N[0] , \quad N[0] \stackrel{\triangle}{=} n_1[0] + be^{-j\phi}n_2[0]$$
(9.45)

Per calcolare SNR_{rake} dobbiamo considerare la sola parte reale di z[0] dato che la costellazione è BPSK, e in particolare dobbiamo calcolare le statistiche della parte reale $N_I[0]$ della componente di rumore totale N[0] a partire da quelle di $n_1[0]$ ed $n_2[0]$. Ovviamente, N[0]ha statistiche Gaussiane perchè combinazione lineare di variabili Gaussiane; considerando che $E\{n_1[0]\} = E\{n_2[0]\}$, troviamo che $E\{N[0]\} = 0$. Per quanto riguarda la varianza invece,

$$E\left\{N_{I}^{2}[0]\right\} = E\left\{\left(n_{1,I}[0] + b\cos(\phi)n_{2,I}[0] + b\sin(\phi)n_{2,Q}[0]\right)^{2}\right\}$$
$$= \sigma_{1}^{2} + b^{2}\cos^{2}(\phi)\sigma_{2}^{2} + b^{2}\sin^{2}(\phi)\sigma_{2}^{2} = \frac{N_{0}}{T_{*}}\left(1 + b^{2}\right)$$
(9.46)

Tutti i doppi prodotti nello sviluppo del quadrato della prima riga della (9.46) danno contributo nullo per le proprietà di correlazione dei vari termini. Infatti le componenti I e Q sono statisticamente incorrelate, quindi $E\{n_{1,I}[0]n_{2,Q}[0]\} = E\{n_{2,I}[0]n_{2,Q}[0]\}0$; inoltre i c(t) è temporalmente incorrelato con $c(t - \tau)$ e quindi anche il termine $E\{n_{1,I}[0]n_{2,I}[0]\}$ vale 0. In conclusione,

$$SNR_{rake} = \frac{2P_s(1+b^2)^2}{(1+b^2)N_0/T_s} = \frac{2P_s(1+b^2)}{N_0/T_s}$$
(9.47)

e il miglioramento dell'SNR del ricevitore Rake è quindi

$$\gamma \stackrel{\triangle}{=} \frac{SNR_{rake}}{SNR_{cor}} = \frac{2P_s(1+b^2)}{N_0/T_s} \frac{N_0/T_s}{2P_s} = 1+b^2$$
(9.48)

del tutto analogo al guadagno di potenza di un ricevitore in diversità 50. Nella Fig. 9.26 vediamo che il guadagno γ può essere rilevante, anche maggiore di 3 dB quando b > 1. Il fatto che il raggio secondario sia più forte del raggio diretto può verificarsi in pratica in una situazione di collegamento tra mezzi mobili in cui inizialmente il raggio LOS è il più forte, quindi b < 1; successivamente, con il mutare dello scenario di propagazione, li raggio LOS tende a diminuire e il raggio riflesso ad aumentare fino ad avere b > 1. Un eventuale ricevitore a correlazione resterebbe comunque "agganciato" al raggio diretto, cancellando del tutto il raggio riflesso e quindi non godendo del guadagno del Rake.



Nel caso di multipath ricco, se i vari raggi sono risolvibili, cioè la differenza tra i ritardi è maggiore di T_c , il guadagno del Rake si può facilmente generalizzare ed è pari a

$$=1+\sum_{i=2}^{N_p}\frac{|a_i|^2}{|a_1|^2}$$

(9.49)

9.5 CDMA Riveduto e Corretto

Alla luce della conoscenza (più) approfondita dei formati spread-spectrum derivante dai paragrafi precedenti, possiamo riprendere in considerazione alcuni argomenti collegati al CDM/CDMA già introdotti in generale nel Cap. 2, principalmente l'influenza della interferenza da accesso multiplo e l'efficienza/capacità dei metodi di accesso multiplo.

9.5.1 Accesso Multiplo e Multiple Access Interference (MAI)

 γ =

Nel paragrafo 2.8.3 sulle tecniche di accesso multiplo abbiamo utilizzato il modello semplificato (2.147) del segnale BPSK/CDMA ricevuto dal ricevitore centralizzato, trascurando per semplicità eventuali costanti di ampiezza e fase per concentrarci sul problema di sincronizzazione. Possiamo meglio specificare adesso tale modello, includendo anche ulteriori parametri:

$$s_{CDMA}(t) = \sum_{i=1}^{U} \sqrt{2P_i} e^{j\phi_i} d_i (t - \tau_i) \cdot c_i (t - \tau_i) + w(t)$$
(9.50)

dove di nuovo $d_i(t) \in c_i(t)$ sono rispettivamente il segnale dati e il codice dell'utente *i*-esimo, e P_i , τ_i , e ϕ_i sono rispettivamente la potenza, il ritardo di propagazione e lo sfasamento dell'*i*-esimo utente, e U è il numero degli utenti attivi. Non abbiamo considerato propagazione multipath per semplicitaà, ma abbiamo esplicitamente aggiunto il disturbo AWGN. Come già precedentemente sottolineato, non è possibile realizzare una rete CDMA sincronizzata - i codici dei vari utenti non possono essere ortogonali e sono di tipo pseudonoise. Immaginando che l'utente desiderato sia il # 1, riscriviamo la (9.50) evidenziando la MAI I(t):

$$s_{CDMA}(t) = \sqrt{P_1} e^{j\phi_1} d_1(t - \tau_i) \cdot c_1(t - \tau_i) + \sum_{i=2}^U \sqrt{P_i} e^{j\phi_i} d_i(t - \tau_i) \cdot c_i(t - \tau_i) + w(t)$$

$$= \sqrt{P_1} e^{j\phi_1} d_1(t - \tau_i) \cdot c_1(t - \tau_i) + I(t) + w(t)$$
(9.51)

Nel momento in cui effettuiamo il despreading-filtraggio con il codice dell'utente # 1, i segnali degli altri utenti costituiscono *interferenza a larga banda* del tutto analoga a quella che abbiamo considerato per il caso del jammer a banda larga J(t) nel paragrafo 9.3.1. Ragionando nella stessa maniera, l'interferente *i*-esimo contribuisce alla potenza interferente totale all'uscita del filtro adattato con un termine di potenza (vedi (9.34)) $I_i = P_i/M$ derivante dall'integrazione effettuata dal filtro adattato di uno spettro di interferenza a banda larga sostanzialmente bianco, e con una d.s.p. pari a $2P_iT_c$. Osserviamo poi che i vari interferenti sono mutuamente indipendenti (perchè caratterizzati da flussi dati e codici di firma diversi e indipendenti) e quindi la potenza interferente totale si può valutare come la somma delle potenze parziali

$$P_{MAI} = \sum_{i=2}^{U} P_i / M \tag{9.52}$$

Da questa modellistica possiamo trarre due conclusioni semplici. *In primis* possiamo calcolare il rapporto SIR dell'utente # 1 come segue:

$$SIR_1 = \frac{P_1}{P_{MAI}} = \frac{U-1}{M} \frac{P_1}{\sum_{i=2}^{U} P_i/M} = \frac{U-1}{M}$$
(9.53)

L'ultima espressione è relativa al caso, sempre verificato in pratica, in cui la rete applica un controllo di potenza, cioè si assicura che le potenze ricevute P_i dei vari utenti al ricevitore centralizzato siano bilanciate: $P_i = P_r \forall i$, e quindi il SIR di tutti gli utenti è il medesimo. Ovviamente, il SIR è tanto più favorevole quanto più il numero di utenti attivi è piccolo nei confronti dello spreading factor M.

La seconda conclusione che possiamo trarre dal modello ci permette di calcolare la BER del collegamento dell'utente # 1: come già commentato nel paragrafo 2.8.3, se il numero di utenti attivi U è abbastanza grande, diciamo maggiore di 10, le statistiche della MAI I(t) possono essere considerate con buona approssimazione *Gaussiane* in virtù del teorema-limite centrale. Considerando anche che lo spettro dell'interferenza è a banda larga rispetto al ricevitore a correlazione, possiamo ri-modellare il segnale ricevuto come

$$s_{CDMA}(t) = \sqrt{P_1} e^{j\phi_1} d_1(t - \tau_i) \cdot c_1(t - \tau_i) + N(t)$$
(9.54)

dove N(t) è il distrubo totale equivalente, risultante dalla somma di MAI e di rumore di ricezione, che, ai fini del calcolo della BER è equivalente a un AWGN con d.s.p. totale

$$S_N(f) = 2I'_0 = 2N_0 + 2I_0 \tag{9.55}$$

dove I_0 è la d.s.p. di ciascuna delle due componenti indipendenti I/Q della MAI ed è pari a (nel caso di controllo di potenza)

$$I_0 = (U - 1)P_r T_c (9.56)$$

Possiamo concludere adesso con il calcolo della BER del ricevitore (ottimo) per costellazione BPSK con rumore AWGN N(t) avente d.s.p I'_0 :

$$\operatorname{BER} = Q\left(\sqrt{\frac{2E_b}{I_0'}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0 + I_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}} \cdot \sqrt{\frac{1}{1 + \frac{U-1}{M}\frac{E_b}{N_0}}}\right) \quad (9.57)$$

cioè la (2.151), con la ben nota degradazione all'aumentare del numero degli utenti contemporaneamente attivi³.

9.5.2 Capacità di Shannon

Concludiamo il paragrafo sul CDMA riprendendo in discussione la questione dell'efficienza dell'accesso multiplo, già discussa precedentemente, e in particolare confrontiamo la capacità di Shannon multiutente di FDMA e CDMA. Per l'FDMA, immaginiamo di avere U canali (utenti) a banda limitata strettamente affiancati nell'ambito della frequenza, cioè come nella Fig. 2.43, ma senza banda di guardia e con $\beta = 0$ (è la configurazione a massima efficienza teorica). Ogni canale ha capacità $C_i = B \log(1 + P/(N_0B))$, dove supponiamo che il livello di rumore e la potenza ricevuta siano rispettivamente gli stessi per tutti gli utenti. I canali sono ortogonali, quindi dal punto di vista della teoria dell'informazione paralleli, e la capacità (multiutente) totale si può calcolare come somma delle capacità parziali:

$$\mathcal{C}_{FDMA} = U \cdot B \log \left(1 + \frac{P}{N_0 B} \right) \tag{9.58}$$

e quindi dovremo imporre per buon pilotaggio che $UR_b \leq C_{FDMA}$. Per valutare l'efficienza spettrale del protocollo d'accesso, immaginiamo che vi sia una banda totale allocata al servizio pari a quella occupata dal numero massimo di utenti $U_{MAX} = M$, cioè $B_{FDMA} =$ $M \cdot B$, con $U \leq M$. Misuriamo ora l'efficienza spettrale ρ come il rapporto tra il bit-rate totale UR_b e la banda totale MB, trovando immediatamente, per la condizione di buon pilotaggio,

$$\rho \stackrel{\triangle}{=} \frac{UR_b}{MB} \le \frac{UB}{MB} \log\left(1 + \frac{P}{N_0B}\right) = \frac{U}{M} \log\left(1 + \frac{(UR_b)P \cdot M}{(UR_b)N_0B \cdot M}\right)$$
$$= \frac{U}{M} \log\left(1 + \frac{\rho}{U/M}\frac{E_b}{N_0}\right) \Rightarrow \frac{E_b}{N_0} \ge \frac{2^{\frac{\rho}{U/M}} - 1}{\frac{\rho}{U/M}}$$
(9.59)

che è simile alla relazione per il singolo utente (4.103) (infatti E_b/N_0 è appunto il parametro di efficienza del singolo canale), corretta in ragione dell'efficienza di utilizzo della banda nell'accesso multiplo, U/M. Lo stesso identico risultato si ottiene valutando la capacità del TDMA (ortogonale).

Per il CDMA, la situazione è più complicata, perchè la formula (4.100) della capacità del canale AWGN a tempo continuo e banda limitata non è direttamente applicabile a causa dello spreading, che non appare nel modello di canale che abbiamo utilizzato a suo tempo. Inoltre, a causa della MAI, non è corretto al 100% dire che i vari canali (utenti) sono indipendenti (e infatti l'accesso multiplo non è ortogonale...). Possiamo però dire che ogni utente lavora in regime Gaussiano quando la rete è abbastanza affollata, considerando

³La confusione dei segnali spread-spectrum che dà luogo alla MAI richiama la confusione delle lingue della Torre di Babele...

la MAI come Gaussiana. Accettando quest'ipotesi, per ricavare correttamente capacità ed efficienza, dobbiamo considerare il canale Gaussiano equivalente a tempo discreto a valle del despreading/filtro adattato, in cui il SINR (Signal-to-Interference-plus-Noise Ratio), considerando il controllo di potenza e inclusa la MAI, è uguale per ogni utente e vale

$$SINR = \frac{P}{(N_0 + (U - 1)PT_c)/T_s}$$
(9.60)

La capacità (normalizzata) del canale è per ogni utente

$$c_{DS/SS} = \frac{1}{2} \log \left(1 + \frac{P}{N_0/T_s + (U-1)PT_c/T_s} \right)$$
(9.61)

Questo canale viene utilizzato con una cadenza di $1/T_s$ usi di canale/s e quindi la condizione di buon pilotaggio è, per ogni utente,

$$R_b \le \frac{1}{2T_s} \log \left(1 + \frac{P}{N_0/T_s + (U-1)PT_c/T_s} \right)$$
(9.62)

Per l'aggregato CDMA il bit-rate totale è UR_b e quindi

$$UR_{b} \leq \frac{U}{2T_{s}} \log \left(1 + \frac{P}{N_{0}/T_{s} + (U-1)PT_{c}/T_{s}} \right)$$
(9.63)

Il punto è adesso che per lo spread-spectrum la banda occupata *non* è $1/2T_s$ ma è invece $1/2T_c$ sia per il singolo utente che per tutto l'aggregato:

$$\rho \stackrel{\triangle}{=} \frac{UR_b}{1/T_c} \le \frac{UT_c}{T_s} \log \left(1 + \frac{P}{N_0/T_s + (U-1)PT_c/T_s} \right) \tag{9.64}$$

Sviluppando ulteriormente, e tenendo conto che $T_c = T_b \cdot \rho/U$, otteniamo infine

$$\frac{E_b}{N_0} \ge \frac{2^{\frac{\rho}{U/M}} - 1}{\frac{\rho}{U/M}} \frac{1}{1 - \frac{U-1}{M} (2^{\frac{\rho}{U/M}} - 1)}$$
(9.65)

Il primo dei due termini di quest'espressione è identico a quello trovato per l'FDMA e tiene conto dell'utilizzazione parziale della banda quando U < M. L'efficienza energetica è però peggiorata rispetto all'FDMA dal secondo fattore, sempre > 1, che tiene conto della presenza della MAI.

I limiti di Shannon per FDMA/TDMA e CDMA sono confrontati nella Fig. 9.27, dove però si nota una cosa interessante: se immaginiamo di assegnare una certa banda massima per il servizio pari proprio a $1/T_c = M/T_s$, con l'FDMA siamo vincolati ad avere al massimo U = M utenti ad "alta" efficienza, pari a quella della curva di Shannon. Con il CDMA, possiamo *oltrepassare* M e avere un numero di utenti anche maggiore, caratterizzati però da una efficienza inferiore perché limitata dalla MAI - la Fig. 9.27 riporta le curve anche con U = 256 e 512. Questa proprietà suggerisce per esempio uno scenario simile all' *Internet delle cose*, con un grande numero di utenti a bassa efficienza ciascuno, per il quale il CDMA è la tecnica di accesso ideale.

CDMA RIVEDUTO E CORRETTO 355



Figura 9.27 Limite di Shannon per FDMA/TDMA e CDMA



CAPITOLO 10

L'UNIONE FA LA FORZA -COMUNICAZIONI MIMO



"La Divinità Indiana Durga"

-Le molte braccia danno la forza di sconfiggere il demone-bisonte Mahishasura

Avere molte braccia conferisce alla Dea poteri sovrannaturali... Il motto di questo capitolo è infatti "L'Unione fa la Forza" ovvero, come migliorare un collegamento wireless convenzionale aggiungendo più antenne e/o coppie trasmettitore/ricevitore. Il problema è che tutte le antenne trasmettono contemporaneamente, sulla stessa banda, e magari con lo stesso codice di spreading.... E allora? Scopriamo i segreti delle comunicazioni MIMO!



Figura 10.1 Schema-base di un collegamento wireless MIMO 2×2

10.1 L'A-B-C delle Comunicazioni MIMO (Multiple-Input, Multiple-Output

Sappiamo che la ricerca di sempre maggior efficienza spettrale di un collegamento digitale affidabile si scontra con un limite fisico, e cioè con la capacità di Shannon. Fissato il rapporto segnale-rumore SNR, non è possibile avere un bit-rate affidabile maggiore del ben noto valore $B \log_2(1 + \text{SNR})$ avendo a disposizione una certa banda *B*. Nelle comunicazioni radio, la banda e/o lo *SNR* possono avere limitazioni anche forti, e di conseguenza anche la capacità del collegamento è limitata.

L'idea delle comunicazioni MIMO sembra piuttosto intuitiva: se non ce la faccio con un solo collegamento, posso provare ad aggiungerne un altro, come mostrato nella Fig. 10.1. Il flusso digitale d'ingresso viene suddiviso in due e inviato a due modulatori e antenne separate di trasmissione (TX) che usano *la stessa banda sulla stessa portante*, in modo che il bit-rate totale R_b è suddiviso tra le due. All'altro capo del collegamento, vengono utilizzate due antenne di ricezione (RX), e i due flussi ricevuti vengono ri-multiplati per recuperare il flusso originale. La prima impressione che si ha è quella di poter *raddoppiare* con il collegamento 2×2 l'efficienza del link, perché possiamo trasmettere tanto quanto due singoli collegamenti 1×1 allo stesso tempo. Purtroppo questa conclusione è prematura o superficiale: le antenne in Fig. 10.1 sono normalmente *omnidirezionali*, cioè che trasmettono/ricevono in/da ogni direzione e quindi, come suggerito dalla figura, il flusso trasmesso dall'antenna TX1 viene ricevuto sia dalla RX1 che dalla RX2 (stessa cosa per la TX2). Si crea dunque *interferenza* tra i due collegamenti 1×1 , che può diminuire l'affidabilità e/o alterare la capacità complessiva.

Prima di analizzare questo problema abbastanza complicato e capire che cosa possono fare "mostri" come l'access point 802.11 ax con 8 antenne di Fig. 10.2, vediamo che cosa è possibile ottenere con sistemi multiantenna più semplici, tradizionalmente utilizzati da anni nelle comunicazioni wireless (ben prima cioè degli anni '90).

10.2 La Ricezione in Diversità (tradizionale) diventa Trasmissione in Diversità

L'idea di usare più di un'antenna per migliorare un link è da sempre stata nelle corde dell'ingegeria delle radiocomunicazioni. In particolare, sono da sempre state usate più di un'antenna *in ricezione* per migliorare la sensibilità del ricevitore. La Figura illustra il principio della *ricezione in diversità*, cioè un MIMO semplificato 2×1 con due antenne RX e







una singola in trasmissione. Il trasmettitore invia un simbolo s[m] di una certa costellazione all'istante $t_m = mT_s$, dove T_s è il tempo di simbolo, e il ricevitore ha due antenne con due sezioni di ricezione radio e conversione in banda base separate. Supponendo canale piatto in frequenza, ricezione con filtro adattato e assenza di ISI, nonché recupero perfetto del ritardo del segnale, le uscite delle due sezioni del ricevitore corrispondenti al simbolo trasmesso sono, come indicato in Fig. 10.3,

$$r_1[m] = h_1 s[m] + w_1[m] , r_2[m] = h_2 s[m] + w_2[m]$$
 (10.1)

Nel ricevitore abbiamo anche un'ulteriore sezione di elaborazione in banda base che *combina* le due uscite di cui sopra per ottenere infine una variable *soft* di decisione z[m] relativa al simbolo s[m].

Nella (10.1), i due coefficienti $h_1 = |h_1| \exp\{j \angle h_1\}$ e $h_1 = |h_2| \exp\{j \angle h_2\}$ rappresentano i coefficienti di ampiezza/fase introdotti sul segnale ricevuto dai due canali radio (non selettivi in frequenza) tra l'antenna trasmittente e le due riceventi. I due termini $w_1[m]$ e $w_2[m]$ sono le due componenti di rumore di ricezione che risultano statisticamente indipendenti, perché sono generate da due diversi apparati, cioè le due sezioni radio del ricevitore. Queste ultime sono normalmente realizzate in tecnologia integrata e sono molto

simili tra di loro, e quindi potremo ritenere le varianze dei due termini di rumore identiche tra loro. Sono invece in generale differenti i rapporti segnale-rumore SNR alle due antenne:

$$SNR_{i} = \frac{S_{2}|h_{i}|^{2}}{2\sigma_{w}^{2}} , \quad i = 1, 2$$
(10.2)

dove $S_2 \stackrel{\triangle}{=} E\{|s[m]|^2\}$ è la potenza della costellazione trasmessa, e σ_w^2 è la varianza della componente I e/o della componente Q di entrambi i termini di rumore.

10.2.1 Strategie di combinazione

Come si combinano $r_1[m]$ e $r_2[m]$ per ottenere z[m]? L'idea più semplice è quella della cosiddetta *selezione*: il ricevitore valuta su di un certo periodo lo SNR sulle due antenne e *seleziona* il segnale proveniente dalla migliore, ignorando l'altro. Il criterio è molto semplice, si deve in teoria essere in grado di stimare in qualche modo il valore di $|h_{1,2}|$ per ricavare SNR_{1,2}. In pratica, è sufficiente selezionare l'antenna la cui potenza di uscita è la massima, dato che il contributo di rumore sulle due antenne è il medesimo. Questa strategia fornisce un collegamento che non è mai peggiore di quello del caso 1×1 , e in particolare compensa molto bene le possibili variazioni nel tempo della qualità dei collegamenti perchè trova sempre il "migliore" dei due, supponendo che la velocità di selezione sia adeguatamente commisurata al tempo di coerenza del canale. In formule, SNR_z = max(SNR₁, SNR₂).

Possiamo fare meglio della semplice selezione? Supponiamo di realizzare una combinazione lineare delle uscite delle antenne con coefficienti *complessi* a_1 e a_2 :

$$z[m] = a_1 \cdot r_1[m] + a_2 \cdot r_2[m]$$

Vogliamo trovare i valori dei coefficienti che forniscono il *massimo SNR*, e per far questo partiamo da (10.1) ottenendo

$$SNR = \frac{S_2 |a_1 h_1 + a_2 h_2|^2}{2\sigma_w^2 (|a_1|^2 + |a_2|^2)}$$
(10.4)

(10.3)

Dalla disuguaglianza di Cauchy-Schwarz sappiamo che

$$|a_1h_1 + a_2h_2|^2 \le (|a_1|^2 + |a_2|^2) \cdot (|h_1^*|^2 + |h_2^*|^2)$$

dove l'uguaglianza si ottiene quando $a_{1,2} = c \cdot h_{1,2}^*$, con $c \in \mathbb{R}$ una qualche costante. Dunque

$$SNR_{opt} = \frac{S_2 \cdot (|h_1|^2 + |h_2|^2)}{2\sigma_w^2} = SNR_1 + SNR_2$$
(10.5)

dove il valore di c non influisce sul SNRopt, per cui possiamo scegliere per semplicità

$$a_1 = h_1^*$$
, $a_2 = h_2^*$ (10.6)

ottenendo la strategia cosiddetta maximal ratio combining (MRC) perchè massimizza lo SNR post-combinazione. Se ad esempio $|h_1| = |h_2|$, il miglioramento sullo SNR rispetto al link 1×1 è un fattore 2, cioè 3 dB. Questo fattore di miglioramente dello SNR viene chiamato guadagno di potenza o guadagno di array (dove array indica la schiera della due antenne di ricezione).

Esempio 10.50

Il valore dei coefficienti ottimi per il criterio MRC è $a_{1,2} = c \cdot h_{1,2}^*$ dove $c \in \mathbb{R}$ è una costante reale arbitraria che non influisce sullo SNR. Troviamo il valore di c per il quale l'*errore quadratico medio* tra la variabile di decisione soft z[m] e il simbolo di costellazione s[m] è *minimo*, così che si possa direttamente procedere alla decisione su z[m] applicando le zone usuali per la costellazione trasmessa:

$$c_{opt} = \arg\max_{c} \mathbb{E}\{|z[m] - s[m]|^2\} = \arg\max_{c} \mathbb{E}\{|c(h_1^*r_1[m] + h_2^*r_2[m] - s[m]|^2\}$$
$$= \arg\max_{c} \mathbb{E}\{|c[(|h_1|^2 + |h_2|^2)s[m] + h_1^*w_1[m] + h_2^*w_2[m]] - s[m]|^2\}$$

Ponendo per semplicità $\alpha \stackrel{\triangle}{=} |h_1|^2 + |h_2|^2$ e $W \stackrel{\triangle}{=} h_1^* w_1[m] + h_2^* w_2[m]$ e osservando che il rumore e i simboli di costellazione sono statisticamente indipendenti si ha

$$c_{opt} = \arg \max_{c} E\{|(\alpha c - 1)s[m] + cW|^{2}\}$$

=
$$\arg \max_{c} E\{(\alpha c - 1)^{2}|s_{m}|^{2} + c^{2}|W|^{2} + 2(\alpha c - 1)c\Re[s_{m}W^{*}]\}$$

=
$$\arg \max_{c} \{(\alpha c - 1)^{2}S_{2} + c^{2}\alpha \cdot 2\sigma_{w}^{2}\}$$
 (10.7)

Derivando rispetto a c e uguagliando a zero otteniamo

$$2\alpha(\alpha c - 1)S_2 + 4c\sigma_w^2 \alpha = 0 \quad \to \quad (\alpha c - 1)S_2 + 2c\sigma_w^2 = 0$$
$$c_{opt} = \frac{1}{|h_1|^2 + |h_2|^2 + 2\sigma_w^2/S_2}$$
(10.8)

e quindi i coefficienti MRC che minimizzano anche l'errore quadratico medio tra $\boldsymbol{z}[m]$ e $\boldsymbol{s}[m]$ sono

$$a_1 = \frac{h_1^*}{|h_1|^2 + |h_2|^2 + 2\sigma_w^2/S_2} \quad , \quad a_2 = \frac{h_2^*}{|h_1|^2 + |h_2|^2 + 2\sigma_w^2/S_2} \tag{10.9}$$

Il denominatore $|h_1|^2 + |h_2|^2 + 2\sigma_w^2/S_2$ dei coefficienti ha la funzione di controllare l'ampiezza della combinazione per renderla il più possibile vicina in media all'ampiezza della costellazione data, e come si nota dipende anche dal livello del rumore sulle antenne relativamente alla potenza della costellazione stessa.

Il criterio MRC (e anche quello a selezione) si può estendere a un numero arbitrario N_R di antenne di ricezione - lasciamo lo sviluppo al lettore come esercizio, insieme con la constatazione che si può ottenere un guadagno di potenza massimo pari a N_R . Osserviamo invece in conclusione che MRC richiede la stima completa dei coefficienti di canale $h_{1,2}$ cioè parte reale/immaginaria o modulo/fase. Una versione pià semplice ma sub-ottima del criterio prevede di usarne *solo la stima della fase*, adottando quindi i coefficienti $a_i = \exp\{-j \angle h_i\}$ - questo criterio viene chiamato *equal gain combining* (EGC) ed ha prestazioni quasi buone come il MRC quando lo sbilanciamento nelle ampiezze dei due canali è limitato.

10.2.2 Diversità contro il Fading di Rayleigh

Abbiamo introdotto la ricezione in diversità per migliorare il rapporto segnale rumore sul canale AWGN, guadagnando in potenza ricevuta (*power* o *array* gain). In realtà il beneficio

362 L'UNIONE FA LA FORZA - COMUNICAZIONI MIMO

di questa tecnologia è ancora maggiore quando consideriamo anche il *fading di canale*, in particolare con statistiche di Rayleigh (7.26).

Partiamo subito con l'analisi generale di un ricevitore che applica la diversità di ricezione di ordine N, e che riceve un singolo simbolo di costellazione attraverso N canali "paralleli" con fading non selettivo (piatto) di Rayleigh, che supponiamo *indipendenti*. Quest'ultima ipotesi è un po' forzata, ma tutto sommato è abbastanza ben verificata quando le antenne RX sono distanziate di qualche lunghezza d'onda del segnale radio. Dunque i segnali ricevuti all'istante mT_s sono

$$r_i[m] = h_i \cdot s[m] + w_i[m] , \quad i = 1, ..., N$$
(10.10)

dove i coefficienti di fading sono tali per cui $E \{|h_i|^2\}=1$, così da ricevere mediamente su ogni antenna la stessa potenza ricevuta attraverso un canale AWGN ad ampiezza unitaria privo di fading. Se il ricevitore riesce a stimare (perfettamente) i guadagni (complessi) di canale (cioè i coefficienti del fading) h_i , possiamo combinare i segnali ricevuti nella variabile di decisione z[m] usando il criterio MRC:

$$z[m] = \sum_{i=1}^{N} h_i^* r_i[m]$$
(10.11)

da cui

$$z[m] = s[m] \cdot \sum_{i=1}^{N} |h_i|^2 + \sum_{i=1}^{N} h_i^* w_i[m] \stackrel{\triangle}{=} s[m] \cdot \sum_{i=1}^{N} |h_i|^2 + W[m]$$
(10.12)

La ricezione in diversità adesso ha *due* effetti: in primis, la consueta riduzione della varianza di rumore dato che le componenti di disturbo $w_i[m]$ sono comunque statisticamente indipendenti tra le varie antenne; inoltre, si nota un "effetto di media" dei coefficienti di fading apportato dalla combinazione $(1/N) \sum_{i=1}^{N} |h_i|^2$. Questo effetto di media *attenua le variazioni profonde* del fading, rendendole (assai) meno probabili che in un ricevitore a singola antenna, migliorando sia la BER che la probabilità di fuori sevizio.

Supponiamo infatti per semplicità che s[m] sia un simbolo BPSK, i.e., $s[m] \in \{-1; 1\}$. Se con E_s indichiamo l'energia per simbolo *totale*, ricevuta cioè da *tutte* le antenne, troviamo che la varianza normalizzata σ_w^2 delle componenti I/Q del rumore $w_i[m]$ non è pari a $(2E_s/N_0)^{-1}$ come nel ricevitore a singola antenna (2.111), ma diventa $\sigma_w^2 = (2(E_s/N)/N_0)^{-1}$ perché appunto ogni antenna riceve 1/N-esimo della potenza totale. Le componenti I/Q del disturbo W[m] in (10.12) hanno entrambe una varianza $\sigma_W^2 = \sigma_w^2 \sum_{i=1}^N |h_i|^2$, e possiamo calcolare il valore della BER in diversità *condizionato* al vettore (*array*) **h** dei guadagni di canale:

$$BER_D |\mathbf{h} = Q\left(\sqrt{\frac{2E_b}{N_0} \frac{\sum_{i=1}^N |h_i|^2}{N}}\right)$$
(10.13)

dove E_b , come precedentemente E_s , rappresenta l'energia per bit *totale* ricevuta attraverso le N antenne. Il valore cercato BER_D della BER si trova *mediando* BER_D |**h** sulle statistiche (Gaussiane) del vettore aleatorio **h**. Riportiamo unicamente il risultato finale:

$$BER_D = \left[\frac{1}{2}\left(1 - \sqrt{\frac{E_b/N_0}{N + E_b/N_0}}\right)\right]^N \sum_{i=0}^{N-1} \binom{N-1+i}{i} \left[\frac{1}{2}\left(1 + \sqrt{\frac{E_b/N_0}{N + E_b/N_0}}\right)\right]^N$$
(10.14)



Figura 10.4 Curve di BER del ricevitore in diversità su canali di Rayleigh indipendenti

Quando E_b/N_0 è sufficientemente grande, diciamo > 10 dB,

$$BER_D \simeq \binom{2N-1}{N} \left[\frac{1}{2} \left(1 - \sqrt{\frac{E_b/N_0}{N + E_b/N_0}} \right) \right]^N \tag{10.15}$$

da paragonarsi con il corrispondente valore (7.29) senza diversità:

$$BER = \frac{1}{2} \left(1 - \sqrt{\frac{E_b/N_0}{1 + E_b/N_0}} \right)$$
(10.16)

La Fig. 10.4 mette a confronto le curve di BER per diversi valori del numero delle antenne (fattore di diversità N). L'effetto è evidente: la BER "nativa" del ricevitore a singola antenna migliora esponenzialmente nel numero delle antenne N. Questo miglioramento del ricevitore viene chiamato guadagno di diversità (diversity gain) ed è tipico della ricezione su canali multipli con fading di Rayleigh indipendenti.

Considerando di nuovo (10.13), possiamo intuire che cosa accade per un numero di antenne ricevente molto grande: per la legge dei grandi numeri, la quantità $\sum_{i=1}^{N} |h_i|^2$ tende al suo proprio valor medio, cioè N, e quindi

$$BER_D \to Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$
 (10.17)

L'effetto del fading viene *asintoticamente* "cancellato" e il canale viene ricondotto a un semplice AWGN con la stessa energia per bit totale ricevuta E_b - un risultato che rincontreremo successivamente.

Esempio 10.51





Dopo la BER, è interessante valutare la probabilità di fuori servizio del ricevitore in diversità sul canale di Rayleigh. Ragionando liberamente in termini "ergodici", cioè scambiando medie temporali e statistiche, vogliamo ricavare la *frazione di tempo* durante la quale la BER rispetta un vincolo di *Qualità del Servizio* (QoS) BER₀ derivante dalla nostra applicazione, cioè si ha BER_D \leq BER₀. Considerando (10.13), la probabilità di fuori servizio é dunque

$$\Pr\{\text{BER}_{D} \ge \text{BER}_{0}\} = \Pr\left\{\sum_{i=1}^{N} |h_{i}|^{2} \le N/M\right\}$$
(10.18)

dove intendiamo che BER₀ è la BER che si ottiene dal ricevitore su canale AWGN puro (cioè con $|h_i| \equiv 1 \forall i$), e dove abbiamo incluso il consueto *margine* di sicurezza *M* sul rapporto segnale-rumore.

La variabile aleatoria $A \stackrel{\triangle}{=} \sum_{i=1}^N |h_i|^2$ ha una pdf chi-quadro a 2N gradi di libertà, per cui:

$$\Pr\left\{\sum_{i=1}^{N} |h_i|^2 \le N/M\right\} = \int_0^{N/M} \frac{1}{2^N(N-1)!} \alpha^{N-1} e^{-\alpha/2} d\alpha \tag{10.19}$$

L'espressione esatta di questo integrale è complicata e contiene funzioni non elementari; ricorriamo quindi a un limite di Chernoff:

$$\Pr\left\{\mathsf{BER} \ge \mathsf{BER}_0\right\} \le \left(\frac{\exp(1 - 1/M)}{M}\right)^N \tag{10.20}$$

La probabilità di fuori servizio decresce di nuovo esponenzialmente con N, come si nota in Fig. 10.5, e può essere ridotta a piacere secondo una specifica di QoS, a patto di aumentare il margine M e quindi la potenza media trasmessa. Se $N \to \infty$ la probabilità di fuori servizio tende naturalmente a 0 perchè non vi è alcuna variabilità nel rapporto segnale-rumore momentaneo: il canale è tramutato in un AWGN e la BER è costantemente uguale al proprio valor medio.



Figura 10.6 MIMO 2×1 , ovvero Trasmissione in Diversità

10.2.3 Diversità in trasmissione

Come già detto, i ricevitori in diversità sono noti fin dai tempi delle comunicazioni analogiche e sono un esempio di multi-antenna in ricezione. Assai più recentemente, è stata introdotta l'idea della diversità in *trasmissione*, che è strettamente legata alle comunicazioni digitali, con una doppia antenna al trasmettitore.

La Fig. 10.6 (a) rappresenta il collegamento con diversità TX: due simboli diversi $s_1[0]$ e $s_2[0]$ di una stessa costellazione vengono trasmessi allo stesso istante m = 0 dalle due antenne TX1 e TX2 e arrivano al ricevitore (a singola antenna) combinate attraverso due coefficienti di canale diversi:

$$r[0] = h_1 s_1[0] + h_2 s_2[0] + w[0]$$
(10.21)

dove w[0] è AWGN con varianza σ_w^2 per componente. Non si vede modo di recuperare separatamente i due simboli $s_1[0]$ e $s_2[0]$, ne' si vede alcun principio di diversità.

In ricezione, la parola *diversità* si riferisce alle antenne che ricevono due *diverse* copie dello stesso segnale trasmesso. Nel nostro caso andiamo a cercare di sfruttare la diversità delle antenne di trasmissione (che continua a sussistere visti i diversi coefficienti di canale) attraverso una diversità *temporale*, cioè ripetendo di fatto i due simboli già trasmessi e osservandoli nel ricevitore una seconda volta. In particolare, anziché ripetere banalmente $s_1[0] e s_2[0]$ rispettivamente dalle antenne TX1 e TX2, si scambiano i simboli sulle due antenne, se ne prende il complesso coniugato, e uno dei due viene anche cambiato di segno, come indicato in Fig. 10.6 (b), cosicché il segnale ricevuto al tempo m = 1 diventa

$$r[1] = h_1 s_2^*[0] - h_2 s_1^*[0] + w[1]$$
(10.22)

dove w[1] presenta le stesse statistiche (varianza) di w[0] ma ne è indipendente per le caratteristiche del filtro adattato privo di ISI che presumiamo in ricezione. Adesso, con due osservati, è potenzialmente sfruttabile la diversità dei due canali per cercare di recuperare i due simboli - non vi è aumento di efficienza del link (come del resto nella ricezione in diversità) perché complessivamente vengono trasmessi due simboli distinti $s_1[0]$ e $s_2[0]$ in due intervalli di segnalazione consecutivi m = 0 e m = 1.

Adesso dobbiamo ricavare lo schema di "combinazione", cioè l'elaborazione da applicare a r[0] e r[1] per ottenere due variabili di decisione soft $z_1[0]$ per demodulare $s_1[0]$ e $z_2[0]$ per $s_2[0]$, ottenute ciascuna attraverso opportuna combinazione lineare:

$$z_1[0] = a_{0,0}r[0] + a_{0,1}r^*[1] , \quad z_2[0] = a_{1,0}r[0] + a_{1,1}r^*[1]$$
(10.23)

Nella combinazione lineare abbiamo introdotto una piccola modifica da subito: poiché r[1] contiene la versione *complessa coniugata* dei due simboli, abbiamo usato $r^*[1]$ invece che direttamente r[1] per "riallinearci" con r[0].

Per trovare i coefficienti, possiamo cercare di ispirarci al criterio MRC della precedente sezione, applicando $a_{0,0} = h_1^* e a_{0,1} = (-h_2^*)^* = -h_2$ (il coefficiente di $s_1[0]$ in $r^*[1] e -h_2^*$) nel formare la variabile $z_1[0]$ relativa a $s_1[0]$, ignorando per il momento la presenza dell'*altro* simbolo. Analogamente, possiamo tentare con $a_{1,0} = h_2^* e a_{1,1} = h_1$ per $z_2[0]$:

 $z_1[0] = h_1^* r[0] - h_2 r^*[1] , \quad z_2[0] = h_2^* r[0] + h_1 r^*[1]$ (10.24)

Utilizzando (10.21) e (10.22) in (10.24), abbiamo

$$z_{1}[0] = |h_{1}|^{2}s_{1}[0] - h_{1}^{*}h_{2}s_{2}[0] + h_{1}^{*}w[0] + h_{1}^{*}h_{2}s_{2}[0] + |h_{2}|^{2}s_{1}[0] - h_{2}w[1]^{*}$$
$$= (|h_{1}|^{2} + |h_{2}|^{2})s_{1}[0] + W_{1}$$
(10.25)

e analogamente

$$z_2[0] = (|h_1|^2 + |h_2|^2)s_2[0] + W_2$$
(10.26)

dove notiamo una piacevole sorpresa: entrambi i termini di interferenza sono scomparsi dalle due variabili di decisione! Questa caratteristica di *ortogonalità*, è il punto saliente dello schema di trasmissione in diversità che viene chiamato *codifica di Alamouti*. Le componenti di rumore W_1 e W_2 in (10.25)-(10.26) sono poi (ancôra) Gaussiane perchè combinazione lineare di Gaussiane, e hanno entrambe varianza $\sigma_W^2 = (|h_1|^2 + |h_2|^2)\sigma_w^2$ per componente.

Dunque possiamo effettivamente recuperare i due simboli di costellazione trasmessi senza'alcuna mutua interferenza - resta però da capire l'efficacia dello schema di diversità. Come già accennato, non vi è alcun guadagno di bit-rate rispetto al caso 1×1 , come nella ricezione in diversità. Calcoliamo i rapporti segnale-rumore: su entrambe le variabili di decisione $z_1[0]$ o $z_2[0]$ lo SNR risulta pari a

$$\operatorname{SNR}_{Ala} = \frac{(|h_1|^2 + |h_2|^2)^2 S_2}{2\sigma_W^2} = \frac{(|h_1|^2 + |h_2|^2) S_2}{2\sigma_w^2}$$
(10.27)

cioè formalmente lo stesso risultato (10.2) della ricezione in diversità.

Possiamo allora dire che anche stavolta $\text{SNR}_{Ala} = \text{SNR}_1 + \text{SNR}_2$ e che le due diversità si equivalgono? SI e NO. SI, se supponiamo che entrambe le due antenne TX trasmettano contemporaneamente la *medesima* potenza radio del collegamento convenzionale 1×1 link - otteniamo un guadagno di potenza (array gain) pari (fino) ai consueti 3 dB. NO, se al contrario la potenza radio *totale* RF deve essere la stessa nel caso 1×1 come nel caso 1×2 della diversità TX - e questo purtroppo è quel che si verifica in pratica a causa di norme di emissione massima su di una certa banda radio. Dunque è più corretto paragonare i) cosa otteniamo con la codifica di Alamouti (10.27) a ii) cosa otteniamo con una singola antenna TX che emette la stessa potenza totale delle due di Alamouti. In tal caso $SNR_1 = 2|h_1|^2S_2/2\sigma_w^2$ e $SNR_2 = 2|h_2|^2S_2/2\sigma_w^2$, per cui $SNR_{Ala} = (SNR_1+SNR_2)/2$ e purtroppo non c'è alcun guadagno di potenza.

Perchè allora abbiamo studiato la diversità di trasmissione? Se anzichè semplice AWGN consideriamo un canale con fading di Rayleigh come in (7.26), allora h_1 e h_2 sono variabili



Figura 10.7 Modello completo di un collegamento MIMO 2 × 2

Gaussiane complesse a potenza unitaria. Notiamo dalla (10.26) che siamo nella stessa situazione della (10.12) con N = 2. In conclusione, se la potenza trasmessa è la medesima per una o per due antenne TX, abbiamo sul canale di Rayleigh un *guadagno di diversità* di ordine 2 dovuto proprio alla diversità tra le due antenne come in (10.13)-(10.15), e la qualità complessiva del canale comunque migliora.

10.3 Modellistica e Capacità di Shannon di un Collegamento MIMO

10.3.1 Modellistica di canale MIMO e utilizzo intuitivo

Dopo aver considerato separatamente che cosa possiamo ottenere con due antenne RX o due antenne TX, adesso dobbiamo finalmente esaminare il caso in cui abbiamo sia due antenne TX che due antenne RX - un collegamento MIMO 2×2 propriamente detto come in Fig. 10.7. Adesso abbiamo *quattro* canali wireless derivanti dalla ricezione da parte di ciascuna dell antenne RX di entrambe le antenne TX - come già accennato le antenne sono omnidirezionali e disposte relativamente vicine l'una all'altra nel trasmettitore e nel ricevitore, quindi la reciproca interferenza é inevitabile.

Nella consueta ipotesi di canali non selettivi in frequenza, ricezione a filtro adattato priva di ISI, recupero ideale del parametro di ritardo dei segnali, il modello dei segnali ricevuti all'istante mT_s è il seguente:

$$\begin{cases} r_1[m] = h_{1,1}s_1[m] + h_{1,2}s_2[m] + w_1[m] \\ r_2[m] = h_{2,1}s_1[m] + h_{2,2}s_2[m] + w_2[m] \end{cases}$$
(10.28)

dove il significato dei vari elementi è evidente dopo le sezioni precedenti. Peraltro, il modello 2×2 (10.28) si può generalizzare immediatamente a un numero generico di antenne trasmittenti N_T e antenne riceventi N_R per ottenere il modello MIMO $N_R \times N_T$:

$$\begin{cases} r_1[m] = h_{1,1}s_1[m] + h_{1,2}s_2[m] + \dots + h_{1,N_T}s_{N_T}[m] + w_1[m] \\ \vdots \\ r_{N_R}[m] = h_{N_R,1}s_1[m] + h_{N_R,2}s_2[m] + \dots + h_{N_R,N_T}s_{N_T}[m] + w_{N_R}[m] \end{cases}$$
(10.29)

Possiamo facilmente formulare (10.29) in forma matriciale concisa:

$$\mathbf{r} = \mathbf{H}\mathbf{s} + \mathbf{w} \tag{10.30}$$

dove abbiamo trascurato per semplicità la dipendenza dall'istante di tempo m, con

$$\mathbf{r} \stackrel{\triangle}{=} \begin{bmatrix} r_1[m] \\ \vdots \\ r_{N_R}[m] \end{bmatrix} , \ \mathbf{s} \stackrel{\triangle}{=} \begin{bmatrix} s_1[m] \\ \vdots \\ s_{N_T}[m] \end{bmatrix} , \ \mathbf{w} \stackrel{\triangle}{=} \begin{bmatrix} w_1[m] \\ \vdots \\ w_{N_R}[m] \end{bmatrix}$$
(10.31)

e dove

$$\mathbf{H} \stackrel{\triangle}{=} \begin{bmatrix} h_{1,1} & \dots & h_{1,N_T} \\ \vdots & \ddots & \vdots \\ h_{N_R,1} & \dots & h_{N_R,N_T} \end{bmatrix}$$
(10.32)

è la *matrice di canale MIMO*, non dipendente dal tempo. Il vettore di rumore **w** è a media nulla, Gaussiano e *spazialmente* bianco, cioè le varie componenti del vettore sono indipendenti al variare dell'antenna (nello spazio), cosicché la sua matrice di covarianza è

 $\mathbf{C}_{\mathbf{w}} \stackrel{\Delta}{=} \mathbf{E}\{\mathbf{w}\mathbf{w}^{\dagger}\} = 2\sigma_{w}^{2}\mathbf{I}_{N_{R}}, \text{ dove } \mathbf{I}_{N_{R}} \text{ indica la matrice identica } N_{R} \times N_{R}.$

Dal modello generale MIMO (10.30) vediamo che, senza considerare il disturbo w, il vettore di simboli trasmessi s è preda di una ragnatela di termini di mutua interferenza, che ne rendono problematico il recupero. Proprio vero? Vediamo il caso particolare $N_R = N_T$; allora, sempre trascurando il rumore, molto semplicemente

$$\mathbf{r} = \mathbf{H}\mathbf{s} \rightarrow \mathbf{s} = \mathbf{H}^{-1}\mathbf{r} \tag{10.33}$$

purché la matrice di canale sia *invertibile*. Per ricevere correttamente i simboli di costellazione, il ricevitore MIMO deve quindi essere in grado i) di stimare la matrice di canale, ii) invertirla, e iii) applicarla al vettore soft ricevuto. Questo modo di funzionamento è la *multiplazione spaziale (spatial multiplexing*), perche vengono "multiplati" N_T flussi dati (estratti in realtà da un unico flusso a velocità più alta) nello *spazio*, cioè su differenti antenne disposte in punti diversi dello spazio, ma sfruttando la *stessa risorsa trasmissiva* il multiplexing cioè avviene sulla stessa banda, allo stesso tempo, e se del caso con lo stesso codice di spreading/canalizzazione.

La multiplazione spaziale sembra incrementare il bit-rate del collegamento di un fattore $N_R = N_T$. Per capire se questa affermazione è corretta riprendiamo in considerazione la componente di disturbo del collegamento w. Applicando la matrice inversa di canale al vettore ricevuto **r**, incluso il rumore, otteniamo un vettore soft $\mathbf{z} = \mathbf{H}^{-1}\mathbf{r}$ come segue:

$$z = H^{-1}r = H^{-1}(Hs + w) = s + H^{-1}w = s + n$$
 (10.34)

Il vettore dei simboli è adesso accompagnato da una (nuova) componente di rumore (ancora) Gaussiano **n** che ha però statistiche diverse da **w**. In particolare, **n** è ancora a media nulla, ma la sua matrice di covarianza è

$$\mathbf{C}_{\mathbf{n}} = \mathbf{E}\left\{\mathbf{n}\mathbf{n}^{\dagger}\right\} = \mathbf{E}\left\{\mathbf{H}^{-1}\mathbf{w}\mathbf{w}^{\dagger}\left(\mathbf{H}^{-1}\right)^{\dagger}\right\} = 2\sigma_{w}^{2}\left(\mathbf{H}^{\dagger}\mathbf{H}\right)^{-1}$$
(10.35)

La (10.35) evidenzia chiaramente che la varianza del rumore può essere incrementata dalla ricezione MIMO (l'applicazione di \mathbf{H}^{-1}). Infatti nella matrice di covarianza di \mathbf{n} è "nascosto" un fattore $1/|\det(\mathbf{H})|^2$ derivante dalla $(\mathbf{H}^{\dagger}\mathbf{H})^{-1}$, che fa aumentare la potenza di rumore se e quando $|\det(\mathbf{H})|$ è piccolo. Se è vero che la componente di segnale in \mathbf{z} è ricevuta correttamente, è anche vero che può essere annegata in una componente di rumore molto forte. In tal caso l'aumento del bit rate è ingannevole, perchè magari il rumore troppo

forte impedisce di ricevere correttamente tutti i flussi multiplati e/o costringe a ridurne il bit-rate, vanificando la multiplazione. La strada maestra per risolvere il problema è riuscire a calcolare la *capacità di Shannon end-to-end del collegamento MIMO*.

Esempio 10.52

Consideriamo il consueto "esempio giocattolo" di collegamento 2×2 in cui i coefficienti di canale sono *reali* (ad esempio, trasmissione BPSK con rivelazione perfettamente coerente). In molti casi la configurazione geometrica del collegamento è tale per cui la distanza r tra le antenne TX1-RX1 è sostanzialmente la stessa della TX2-RX2 (questo perchè le TX1-TX2 e RX1-RX2 sono a una distanza reciproca d molto più piccola rispetto alla distanza r tra TX e RX); in prima approssimazione possiamo allora porre $h_{1,1} = h_{2,2}$. Ragionando in modo simile, concludiamo che anche $h_{1,2} = h_{2,1}$. Inoltre, se è vero che $d \ll r$, allora anche $|h_{1,1}| \simeq |h_{2,1}|$ - vi sarà però una differenza di fase dovuta alla piccola differenza di percorso Δr tra i due cammini TX1-RX1 rispetto a TX1-RX2 che induce uno sfasamento arbitrario $\phi = 2\pi\Delta r/\lambda_0$ ($\lambda_0 = c/f_0$ è la lunghezza d'onda della frequenza portante, c velocità della luce). La matrice risultante è *bisimmetrica* (cioè simmetrica):

$$\mathbf{H} = \begin{bmatrix} a & b \\ b & a \end{bmatrix}$$
(10.36)

Sappiamo che se fosse $|a| \gg |b|$, allora la matrice sarebbe a forte dominanza diagonale, e avremmo in pratica due canali radio *indipendenti* - a quel punto la capacità totale praticamente sarebbe pari al doppio della capacità dei due canali singoli. Se invece $|a| \simeq |b|$ l'intereferenza è sensibile, e dobbiamo capire l'effettivo aumento di capacità. Nell'ulteriore caso (pessimo) di $a \simeq b$ la matrice di canale è *mal condizionata*, il suo rango è praticamente 1, il determinante è vicino a zero e la vera capacità del collegamento deve essere di nuovo valutata.

Per il momento, seguendo il nostro modello di cui sopra,

$$\mathbf{H}^{-1} = \frac{1}{a^2 - b^2} \begin{bmatrix} a & -b \\ -b & a \end{bmatrix}$$
(10.37)

e la varianza della componente di rumore post-rivelazione è (per entrambe le antenne)

$$\sigma_n^2 = \sigma_w^2 \cdot (a^2 + b^2) / (a^2 - b^2)^2 \tag{10.38}$$

Si vede chiaramente che se i valori di |a| e |b| sono vicini tra loro, l'aumento di varianza è intollerabile. Se ad esempio a = 0.1 e b = 0.09 allora

$$\mathbf{H}^{-1} = \begin{bmatrix} 52.632 & -47.368\\ -47.368 & 52.632 \end{bmatrix}, \quad \sigma_n^2 = 5013.9 \cdot \sigma_w^2$$
$$\mathbf{z} = \begin{bmatrix} s^{(1)} + 52.632w^{(1)} - 47.368w^{(2)}\\ s^{(2)} - 47.368w^{(1)} + 52.632w^{(2)} \end{bmatrix}$$
(10.39)

e confrontiamo gli SNR *con* e *senza* MIMO. Senza l'antenna TX2, il rapporto segnalerumore all'antenna RX1 è is SNR₁ = $a^2 S_2/2\sigma_w^2 = a^2 \text{SNR}_n = 0.01 \text{SNR}_n$ dove SNR_n è un SNR di riferimento indipendente dal canale pari a $S_2/2\sigma_w^2$. Aggiungendo l'antenna TX2, lo SNR su *entrambe* le antenne è SNR = $(a^2 - b^2)^2/(a^2 + b^2)\text{SNR}_n = 2 \cdot 10^{-4}\text{SNR}_n = 0.02\text{SNR}_1$ con un peggioramento, a parità di potenza trasmessa da ciascuna antenna, di ben 17 dB ! Tornando per un attimo alla multiplazione spaziale, è chiaro che se $N_R < N_T$ non c'è speranza di recuperare i simboli trasmessi elaborando semplicemente il vettore ricevuto **r**. Esaminiamo brevemente che succede invece se $N_R > N_T$. In questo caso, il numero di antenne RX è maggiore del minimo indispensabile (N_T) per recuperare il vettore di simboli **s**, e quindi si può ottenere un certo "effetto diversità" aggiuntivo.

Lo scenario $N_R > N_T$ suggerisce una trasmissione uplink verso un punto di accesso Wi-Fi o verso una stazione radio base di una rete cellulare: il terminale mobile ha un numero di antenne minore (al limite una) rispetto a quello di una stazione radio base o un access point che può averne un certo numero. La questione da risolvere sta nell'algoritmo da utlizzarsi nella demodulazione dati, visto che in questo caso abbiamo piu "equazioni" (i segnali ricevuti, ovvero le N_R righe della matrice di canale) che "incognite" (gli N_T simboli trasmessi) - la matrice di canale ha dimensione $N_R \times N_T$ (formato "ritratto"), non è quadrata e non può essere invertita.

L'approccio più semplice è quello della stima dei simboli al minimo errore quadratico medio (MMSE): cercare un demodulatore lineare, cioè una matrice **G** di dimensione $N_T \times N_R$ da applicarsi al vettore $N_R \times 1$ ricevuto **r** per ottenere il vettore di decisione soft $\mathbf{z} = \mathbf{Gr}$ in modo che

$$\mathbf{E}\left\{\left\|\mathbf{z}-\mathbf{s}\right\|^{2}\right\} = \mathbf{E}\left\{\left\|\mathbf{Gr}-\mathbf{s}\right\|^{2}\right\} = \min$$
(10.40)

(10.41)

La soluzione si trova facilmente dalla teoria della stima:

$$\mathbf{G} = \left(\mathbf{H}^{\dagger}\mathbf{H} + \frac{1}{\mathrm{SNR}_n}\mathbf{I}_{N_T}\right)^{-1}\mathbf{H}^{\dagger}$$

Quando il rapporto segnale-rumore di lavoro è molto alto, il rivelatore ottimo **G** é la cosiddetta *pseudo-inversa di Moore-Penrose* $(\mathbf{H}^{\dagger}\mathbf{H})^{-1}\mathbf{H}^{\dagger}$ che annulla l'interferenza tra le antenne (come nel caso di matrice quadrata) senza considerare l'effetto del rumore, e che per motivi storici è anche chiamata soluzione *zero-forcing* (coincide con \mathbf{H}^{-1} se la matrice è quadrata). Se invece lo SNR è molto basso, la soluzione è semplicemente \mathbf{H}^{\dagger} cioè un criterio di tipo MRC ottimo rispetto al rumore, ma del tutto ignaro dell'interferenza. In generale, il rivelatore ottimo MMSE richiede (anche) la conoscenza del rapporto segnale-rumore di riferimento SNR_n = $S_2/(2\sigma_w^2)$.

Esempio 10.53

Riprendiamo in considerazione l'esempio, 52, e ricaviamo il ricevitore MMSE nelle stesse condizioni. Supponiamo che $SNR_n=40 \text{ dB}$, cosicché lo SNR_1 ad una antenna è un buon 20 dB, mentre lo SNR dopo ricezione MIMO con \mathbf{H}^{-1} è un misero 3 dB (SNR_1 -17 dB a causa dell'incremento di rumore).

Con i nostri dati abbiamo

$$\mathbf{H}^{\dagger}\mathbf{H} = \begin{bmatrix} 0.0181 & -0.018\\ -0.018 & 0.0181 \end{bmatrix}$$
$$\frac{1}{\mathbf{SNR}_{n}}\mathbf{I}_{N_{T}} + \mathbf{H}^{\dagger}\mathbf{H} = \begin{bmatrix} 0.018132 & -0.018\\ -0.018 & 0.018132 \end{bmatrix}$$
$$\left(\frac{1}{\mathbf{SNR}_{n}}\mathbf{I}_{N_{T}} + \mathbf{H}^{\dagger}\mathbf{H}\right)^{-1} = \begin{bmatrix} 3812.6 & 3784.8\\ 3784.8 & 3812.6 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} 40.61 & 35.36\\ 35.36 & 40.61 \end{bmatrix} , \quad \mathbf{GH} = \begin{bmatrix} 0.8786 & -0.1189\\ -0.1189 & 0.8786 \end{bmatrix}$$
(10.42)

$$\mathbf{z} = \mathbf{G}\mathbf{r} = \mathbf{G}\mathbf{H}\mathbf{s} + \mathbf{G}\mathbf{w} =$$

$$\begin{vmatrix} 0.8786s^{(1)} - 0.1189s^{(2)} + 40.61w^{(1)} + 35.36w^{(2)} \\ 0.8786s^{(2)} - 0.1189s^{(1)} + 35.36w^{(1)} + 40.61w^{(2)} \end{vmatrix}$$
(10.43)

Piuttosto che un semplice SNR dobbiamo adesso calcolare un SINR, cioè un Signal-to-Interference-plus-Noise Ratio visto che è presente un termine residuo di interferenza che non viene annullato: la matrice MMSE non realizza l'inversione della matrice di canale, ma semplicemente minimizza la potenza dell'errore rispetto al simbolo trasmesso, includendo nell'errore anche una possibile interferenza residua dell'altra antenna. Il risultato è

$$\operatorname{SINR}_{z} = \frac{(0.8786)^{2}S_{2}}{(0.1189)^{2}S_{2} + ((40.61)^{2} + (35.36)^{2}) 2\sigma_{w}^{2}} =$$
$$= \operatorname{SNR}_{n} \cdot \frac{0.772}{0.0141 \cdot \operatorname{SNR}_{n} + 2900} = 0.0003 \cdot \operatorname{SNR}_{n} = 0.03 \cdot \operatorname{SNR}_{1} \quad (10.44)$$

Il SINR è migliore dello SNR dell'esempio 52 di 2 dB - non molto, d'altro canto lo SNR_n è relativamente grande, e quindi la matrice G non è molto diversa dalla \mathbf{H}^{-1} .

Aggiungiamo adesso due ulteriori antenne in ricezione, arrivando a $N_R = 4 > N_T = 2$ e immagianiamo che gli ulteriori coefficienti di propagazione siano simili ai primi due:

1 - 0.0	9
.09 0.1	
12 -0.00	8
.08 0.12	JAN MARK
	$ \begin{array}{rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr$

Con i soliti calcoli troviamo

$$\mathbf{H}^{\dagger}\mathbf{H} = \begin{bmatrix} 0.0389 & -0.0372 \\ -0.0372 & 0.0389 \end{bmatrix}$$
$$\frac{1}{\mathrm{SNR}_{n}}\mathbf{I}_{N_{T}} + \mathbf{H}^{\dagger}\mathbf{H} = \begin{bmatrix} 0.038932 & -0.0372 \\ -0.0372 & 0.038932 \end{bmatrix}$$
$$\left(\frac{1}{\mathrm{SNR}_{n}}\mathbf{I}_{N_{T}} + \mathbf{H}^{\dagger}\mathbf{H}\right)^{-1} = \begin{bmatrix} 295.314 & 282.180 \\ 282.180 & 295.314 \end{bmatrix}$$
$$\mathbf{G} = \begin{bmatrix} 4.135 & 1.640 & 12.86 & 10.24 \\ 1.640 & 4.135 & 10.24 & 12.86 \end{bmatrix} , \quad \mathbf{GH} = \begin{bmatrix} 0.991 & -0.00892 \\ -0.00892 & 0.991 \end{bmatrix}$$
(10.46) cosicché

$$\mathbf{z} = \mathbf{G}\mathbf{r} = \mathbf{G}\mathbf{H}\mathbf{s} + \mathbf{G}\mathbf{w} =$$

 $\begin{bmatrix} 0.991s^{(1)} - 0.00892s^{(2)} + 4.135w^{(1)} + 1.640w^{(2)} + 12.86w^{(3)} + 10.24w^{(4)} \\ 0.991s^{(2)} - 0.00892s^{(1)} + 1.640w^{(1)} + 4.135w^{(2)} + 10.24w^{(3)} + 12.86w^{(4)} \end{bmatrix}$ (10.47)

Il (nuovo) SINR è adesso

$$\operatorname{SINR}_{z} = \frac{(0.991)^{2} S_{2}}{(0.00892)^{2} S_{2} + ((4.135)^{2} + (1.640)^{2} + (12.86)^{2} + (10.24)^{2}) 2\sigma_{w}^{2}} = \frac{(0.991)^{2} S_{2}}{(0.00892)^{2} S_{2} + ((4.135)^{2} + (1.640)^{2} + (12.86)^{2} + (10.24)^{2}) 2\sigma_{w}^{2}}$$

371

(10.45)

372 L'UNIONE FA LA FORZA - COMUNICAZIONI MIMO

 $\frac{0.982}{7.96 \cdot 10^{-5} \cdot \text{SNR}_n + 290.0} = 0.00336 \cdot \text{SNR}_n = 0.336 \cdot SNR_1 \quad (10.48)$ = SNR_n \cdot

Vediamo una degradazione rispetto a SNR1 di soli 5 dB, molto meglio dei 17 dB o 15 dB del ricevitore con sole due antenne - abbiamo realizzato un eccellente miglioramento con le due antenne aggiuntive e con il ricevitore MMSE. È vero che lo SNR è ancora degradato, e che quindi non potremmo usare costellazioni molto grandi, ma è anche vero che abbiamo a disposizione due canali per aumentare il bit-rate totale. Qual è allora la configurazione migliore?

Ciò che adesso abbiamo chiaramente capito, anche con l'ausilio di esempi numerici, è che semplicemente usando (fino a) N_T antenne in multiplazione spaziale, non possiamo pretendere di moltiplicare per N_T la capacità del collegamento sic-et-simpliciter. Questo sarebbe sicuramente accaduto se i collegamenti fossero stati indipendenti senza interferenza reciproca (matrice di canale *diagonale*), ma il caso generale di una matrice di canale H completa è ben più complicato. Sarebbe estremamente utile ricavare allora un sistema equivalente in cui, attraverso qualche procedimento, si riuscisse ad annullare l'interferenza reciproca tra le antenne: la capacità (di Shannon) totale sarebbe banalmente la somma delle capacità parziali dei diversi (sotto) canali - vediamo come riuscirci.

10.3.2 Capacità di Shannon del canale MIMO

I canali indipendenti "nascosti" all'interno della matrice H possono essere portati alla luce del sole attraverso la cosiddetta scomposizione ai valori singolari (Singular-Value Decomposition, SVD). La SVD permette di scomporre ogni matrice complessa $N_R \times N_T$ **H** (supponiamo per fissare le idee $N_R > N_T$) come rappresentato in Fig. 10.8, e cioè come il prodotto di tre matrici speciali:

 $\mathbf{H} = \mathbf{U}\mathbf{D}\mathbf{V}^{\dagger}$ (10.49)

dove **U** (di dimensione $N_R \times N_R$) e **V** ($N_T \times N_T$) sono *matrici unitarie*, cioè la loro Hermitiana (trasposta coniugata) è uguale all'inversa: $\mathbf{U}^{-1} = \mathbf{U}^{\dagger}$, $\mathbf{V}^{-1} = \mathbf{V}^{\dagger}$. Le matrici unitarie definiscono delle trasformazioni di coordinate che, considerate nello spazio \mathbb{R}^N , sono delle rotazioni che non alterano le dimensioni dei vettori. Ciò significa che se applichiamo una matrice unitaria a un vettore aleatorio a valor medio nullo e spazialmente bianco, il vettore mantiene la stessa matrice di covarianza (resta spazialmente bianco):

$$\mathbf{n} = \mathbf{U}\mathbf{w} \to \mathbf{C}_{\mathbf{n}} = \mathbf{E}\left\{\mathbf{n}\mathbf{n}^{\dagger}\right\} = \mathbf{E}\left\{\mathbf{U}\mathbf{w}\mathbf{w}^{\dagger}\mathbf{U}^{\dagger}\right\} = 2\sigma_{w}^{2}\mathbf{U}\mathbf{I}\mathbf{U}^{\dagger}$$
$$= 2\sigma_{w}^{2}\mathbf{U}\mathbf{U}^{-1} = 2\sigma_{w}^{2}\mathbf{I} = \mathbf{C}_{\mathbf{w}}$$
(10.50)

Inoltre, la matrice **D** ha le seguenti proprietà: i) ha le stesse dimensioni $N_R \times N_T$ di **H**; ii) la sottomatrice principale $N_T \times N_T$ è diagonale, come si vede in Fig. 10.8; iii) tutti gli elementi fuori da questa diagonale sono 0, e iv) gli elementi sulla diagonale (i valori singolari) riassumono il comportamento della matrice originale H perchè sono pari alla radice quadrata degli N_T autovalori reali della matrice Hermitiana $N_T \times N_T$ definita non negativa $\mathbf{H}^{\mathsf{T}}\mathbf{H}$.

Che relazione ha la SVD con la capacità di Shannon? Osserviamo che la scomposizione (10.49) ci permette di trovare un sistema equivalente al MIMO originario (10.30), tenendo conto che una trasformazione nota e invertibile sull'osservato non modifica la capacità del



Figura 10.8 Rappresentazione della SVD della matrice H

collegamento. Detto questo, usando (10.49) nella (10.30) otteniamo

$$\mathbf{r} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathsf{T}}\mathbf{s} + \mathbf{w} = \mathbf{U}\mathbf{D}\bar{\mathbf{s}} + \mathbf{w}$$
(10.51)

dove $\bar{\mathbf{s}} \stackrel{\triangle}{=} \mathbf{V}^{\dagger} \mathbf{s}$ è un vettore di simboli "ruotato" con le stesse statistiche del secondo ordine (i simboli di costellazione trasmessi \mathbf{s} sono per ipotesi mutuamente indipendenti). Se, come sempre, il ricevitore conosce la matrice di canale e/o la sua SVD, può costruire un ricevitore MIMO che applica al vettore \mathbf{r} la matrice \mathbf{U}^{\dagger} ottenendo

$$\bar{\mathbf{r}} \stackrel{\triangle}{=} \mathbf{U}^{\dagger} \mathbf{r} = \mathbf{U}^{\dagger} \mathbf{U} \mathbf{D} \bar{\mathbf{s}} + \mathbf{U}^{\dagger} \mathbf{w} = \mathbf{D} \bar{\mathbf{s}} + \bar{\mathbf{w}}$$
(10.52)

dove $\bar{\mathbf{w}}$ è un vettore di rumore Gaussiano con le stesse statistiche di \mathbf{w} - il canale di comunicazione che abbiamo ottenuto ha la stessa capacità di quello originale. D'altronde, possiamo scrivere esplicitamente le N_R equazioni che caratterizzano gli N_R segnali ricevuti attraverso (10.52), osservando che per le proprietà di \mathbf{D} , le ultime $N_R - N_T$ non hanno senso perchè non contengono alcun simbolo di informazione (gli elementi di \mathbf{D} che compaiono in quelle equazioni sono tutti uguali a 0). Ma \mathbf{D} è diagonale (nelle prime N_T righe), e le N_T equazioni di ricezione sono disaccoppiate rispetto ai simboli - sono di fatto equazioni scalari che descrivono N_T canali paralleli, indipendenti, Gaussiani derivanti da componenti di rumore indipendenti, e senz'alcuna mutua interferenza:

$$\bar{r}^{(i)} = d_i \bar{s}^{(i)} + \bar{w}^{(i)} , \quad i = 1, ..., N_T$$
(10.53)

Allora la capacità totale del collegamento c_{MIMO} è banalmente la somma delle capacità parziali c_i (4.94) dei vari collegamenti 1 × 1 paralleli (10.53):

$$c_{MIMO} = \sum_{i=1}^{N_T} c_i = \sum_{i=1}^{N_T} \log_2\left(1 + \text{SNR}_i\right) = \sum_{i=1}^{N_T} \log_2\left(1 + \frac{d_i^2 S_2}{2\sigma_w^2}\right) \text{ bit/c.u.} \quad (10.54)$$

L'interpretazione di (10.54) è semplice: il nostro collegamento MIMO ha "attivato" N_T canali virtuali paralleli con *differenti rapporti segnale-rumore* (e quindi differenti capacità individuali) derivanti dai diversi *guadagni* intrinseci dei canali paralleli $d_i \ge 0$. Se la matrice H non ha rango pieno, (almeno) uno dei $d_i \ge 0$, la (sub-)capacità relativa a quel canale virtuale va a 0; se è malcondizionata, (almeno) un d_i sarà (molto) più piccolo degli altri e apporterà un contributo di capacità trascurabile.

Dunque l'effettivo guadagno di multiplazione, cioè l'incremento di capacità dovuto alla multiplazione spaziale è fortemente dipendente (oltre che ovviamente dallo SNR di

374 L'UNIONE FA LA FORZA - COMUNICAZIONI MIMO

riferimento del link) dai valori dei d_i e in ultima analisi dalle proprietà della matrice di canale. Nel caso in cui $N_R < N_T$, la capacità è data formalmente ancora dalla (10.54), ma i valori singolari, cioè i canali paralleli sono in numero N_R e la sommatoria corre da 1 a N_R - in generale, da 1 a min (N_T, N_R) . Ragionando in modo un po' euristico, il min (N_T, N_R) rappresenta il numero di "canali virtuali" contenuti nel collegamento MIMO, e rappresenta sostanzialmente il guadagno di multiplazione che ci possiamo aspettare dalla tecnologia - non posso aumentare il bit-rate al di là del minimo numero di antenne a disposizione al TX o all'RX. Aumentare il numero di antenne in ricezione oltre N_T , può servire però ad aumentare ulteriormente la capacità per il miglioramento dell'SNR dovuto al guadagno di potenza.

Esempio 10.54

Consideriamo il caso irrealistico di un collegamento 2×2 con antenne direttive che non causano interferenza mutua, cioè con matrice di canale *diagonale*, e supponiamo per semplificare $h_{1,1} = h_{2,2} = 1$. I rapporti segnale-rumore sono SNR₁ = SNR₂ = SNR_n. Rispetto a un link a singola antenna avente la stessa potenza trasmessa che le due antenne trasmettono individualmente e capacità $c = \log_2(1 + \text{SNR}_n)$, la capacità MIMO presenta seccamente un aumento di un fattore 2, pari al guadagno di multiplazione: $c_{MIMO} = 2 \log_2(1 + \text{SNR}_n)$. Se però si mantiene costante la potenza *totale* trasmessa, allora la capacità MIMO ha sì un guadagno di multiplazione 2, ma rispetto a una capacità del link singolo con SNR ridotto di 3 dB per la riduzione della potenza trasmessa individualmente dalle 2 antenne. Perciò usare MIMO è conveniente solo se $2 \log_2(1 + \text{SNR}_n) > \log_2(1 + \text{SNR}_n)$ e cioè... *sempre* (verificare), e tanto più quanto lo SNR è alto!

Che succede se aggiungiamo altre due antenne (direttive) di ricezione (per arrivare a $N_R = 4$) in modo che l'antenna RX3 riceva solo la TX1, e la RX4 riceva solo la TX2? La matrice di canale sarà



Per trovare la capacità, dobbiamo trovare la SVD, che lasciamo per esercizio al lettore. Il risultato è che $d_1 = d_2 = \sqrt{2}$, quindi oltre al guadagno di multiplazione pari a 2 come sopra, abbiamo anche un guadagno di *potenza* pari a $d_1^2 = d_2^2 = 2$ che, a parità di potenza trasmessa, dà una capacità $c_{MIMO} = 2 \log_2(1 + \text{SNR}_n)$, recuperando anche la suddivisione di potenza tra le due antenne trasmittenti.

E se invece aggiungiamo due antenne in trasmissione, cioè TX3 che comunica con RX1 e TX4 con RX2 per arrivare a un sistema 2×4 ? La matrice è stavolta

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

e i valori singolari sono ovviamente gli stessi $d_1 = d_2 = \sqrt{2}$ del caso precedente. Il guadagno di multiplazione resta pari a 2, ma che accade della capacità, resta forse la stessa? Naturalmente no se manteniamo la potenza totale trasmessa costante, dimezzando giocoforza quella del caso $N_T = 2$ e ottenendo stavolta $c_{MIMO} = 2 \log_2(1 + \text{SNR}_n/2)$ esattamente pari al caso 2×2 - le due antenne addizionali non hanno portato benefici nel senso dello spatial multiplexing. Vedremo in seguito la maniera corretta di usare molte antenne in trasmissione.

Nell'Esempio 54, del tutto concettuale, abbiamo evidenziato il legame tra guadagno di multiplazione e di potenza quando la matrice di canale ha rango pieno. Come già discusso, la capacità può invece diminuire se il rango del canale è insufficiente, per la diminuzione del guadagno di multiplazione.

Esempio 10.55

Troviamo la capacità del collegamento 2×2 MIMO dell'Esempio 52 con

$$\mathbf{H} = \begin{bmatrix} 0.1 & -0.09\\ -0.09 & 0.1 \end{bmatrix}$$
(10.55)

e con SNR_n=45 dB. Calcolando la SVD troviamo

$$\mathbf{H} = \begin{bmatrix} -0.7071 & 0.7071 \\ -0.7071 & -0.7071 \end{bmatrix} \begin{bmatrix} 0.01 & 0 \\ 0 & 0.19 \end{bmatrix} \begin{bmatrix} -0.7071 & -0.7071 \\ 0.7071 & -0.7071 \end{bmatrix}$$
(10.56)

(poiché **H** è quadrata, **U=V**) e vediamo che i due autovalori, come ci attendevamo, sono di ordini di grandezza differenti. La capacità MIMO è

$$c_{MIMO} = \log_2 \left(1 + 10^{-4} \text{SNR}_n \right) + \log_2 \left(1 + 0.0361 \text{SNR}_n \right)$$
$$= 2.057 + 10.16 = 12.22 \text{ bit/c.u.}$$
(10.57)

e il diverso contributo dei due canali virtuali è chiaro. La capacità si misura come sempre in *bit per uso di canale*, dove con "uso di canale" si intende un atto completo di trasmissione parallela MIMO.

Su un canale 1×1 che trasmette la stessa potenza della sola antenna TX1 MIMO si ottiene

$$c_{1-to-1} = \log_2 \left(1 + 0.01 \text{SNR}_n \right) = 8.31 \text{ bit/c.u.}$$
 (10.58)

e il guadagno di multiplazione è evidente (circa un fattore 1.5) anche se il secondo canale virtuale MIMO è degradato. È però più corretto fare il confronto con il caso 1×1 in cui si *raddoppia* la potenza trasmessa per concentrare nella TX1 la stessa potenza trasmessa in totale dal MIMO. In questo caso

$$c_{1-to-1} = \log_2 \left(1 + 0.02 \text{SNR}_n \right) = 9.31 \text{ bit/c.u.}$$
 (10.59)

e comunque il MIMO risulta superiore, anche se di un fattore inferiore.

Valutiamo anche la capacità del link 4×2 dello stesso esempio. La SVD è

$$\mathbf{U} = \begin{bmatrix} -0.4870 & -0.1715 & -0.8539 & 0.06549\\ 0.4870 & -0.1715 & -0.3047 & -0.8003\\ -0.5127 & -0.6860 & 0.4057 & -0.3194\\ 0.5127 & -0.6860 & -0.1160 & 0.5031* \end{bmatrix}, \quad \mathbf{V}^{\dagger} = \begin{bmatrix} -0.7071 & 0.7071\\ -0.7071 & -0.7071 \end{bmatrix}$$
$$\mathbf{D} = \begin{bmatrix} 0.2759 & 0\\ 0 & 0.04123\\ 0 & 0\\ 0 & 0 \end{bmatrix}$$
(10.60)

La capacità è aumentata non poco rispetto al caso 2×2 - entrambi gli autovalori sono

maggiori:

$$c_{MIMO} = \log_2 \left(1 + 0.0017 \text{SNR}_n \right) + \log_2 \left(1 + 0.0761 \text{SNR}_n \right)$$
$$= 5.775 + 11.23 = 17.01 \text{ bit/c.u.}$$
(10.61)

)

Possiamo esprimere la relazione della capacità MIMO (10.54) in forma sintetica, osservando che se d_i^2 è un autovalore di $\mathbf{H}^{\dagger}\mathbf{H}$, allora $1 + d_i^2 S_2/2\sigma_w^2$ è un autovalore di $\mathbf{I}_{N_T} + S_2/2\sigma_w^2\mathbf{H}^{\dagger}\mathbf{H}$. Sapendo anche che il prodotto degli autovalori di una matrice è uguale al suo determinante, concludiamo da (10.54) che

$$c_{MIMO} = \log_2 \prod_{i=1}^{N_T} \left(1 + \frac{d_i^2 S_2}{2\sigma_w^2} \right) = \log_2 \det \left(\mathbf{I}_{N_T} + \frac{S_2}{2\sigma_w^2} \mathbf{H}^{\dagger} \mathbf{H} \right) \quad bit/c.u.$$
(10.62)

Concludiamo questa sezione molto fruttuosa osservando che se la potenza radio totale di trasmissione P_T è distribuita equamente su tutte le N_T antenne, allora $S_2/2 = P_T/N_T$ e

$$c_{MIMO} = \log_2 \det \left(\mathbf{I}_{N_T} + \frac{P_T}{N_T \sigma_w^2} \mathbf{H}^{\dagger} \mathbf{H} \right) \quad bit/c.u. \tag{10.63}$$

che useremo in seguito.

A questo proposito, cerchiamo di precisare ulteriormente la questione della potenza trasmessa, potenza ricevuta e rapporto E_s/N_0 per il collegamento MIMO (10.29). Riguardo la potenza trasmessa P_T , è chiaro che, come già accennato, $P_T = N_T S_2/2$ nel caso (che si verifica quasi sempre in pratica) in cui i simboli di costellazione sulle varie antenne sono incorrelati. Se la potenza è equiripartita sulle antenne di trasmissione, dalla (10.29) si ricava facilmente la potenza ricevuta all'antenna RX#*i*:

$$P_{R,i} = \frac{S_2}{2} \sum_{\ell=1}^{N_T} |h_{i,\ell}|^2$$

per cui la potenza ricevuta totale è

$$P_R = \sum_{i=1}^{N_R} P_{R,i} = \frac{S_2}{2} \sum_{i=1}^{N_R} \sum_{\ell=1}^{N_T} |h_{i,\ell}|^2 = \frac{S_2}{2} \left(\|\mathbf{H}\|_2 \right)^2 = \frac{S_2}{2} \operatorname{tr} \left(\mathbf{H} \mathbf{H}^{\dagger} \right)$$
(10.64)

dove $\|\mathbf{H}\|_2$ è la *norma di Frobenius* della matrice di canale e l'operatore tr(·) indica la *traccia* di una matrice, cioè la somma degli elementi lungo la diagonale principale. Se indichiamo con T_s il tempo di segnalazione sul canale MIMO, l'energia per simbolo viene definita come usualmente, cioè $E_s = P_R \cdot T_S$ e si riferisce al simbolo *vettoriale* MIMO cioè al set di simboli di costellazione $s_1, ..., s_{N_T}$ ricevuti nell'intervallo $mT_s \leq t < (m+1)T_s$ "in parallelo" da *tutte* le N_T antenne di trasmissione. Infine, volendo valutare il rapporto E_s/N_0 in ricezione, si deve intendere nel modello (10.29) che la varianza delle componenti I e/o Q di rumore di ogni antenna è pari a $\sigma_w^2 = N_0/T_s$ derivante dalla ricezione a filtro adattato, come ricavato nella (2.53).

Combinando la (10.62) e la (10.64) otteniamo infine

$$c_{MIMO} = \log_2 \det \left(\mathbf{I}_{N_T} + \frac{E_s}{N_0} \frac{1}{\left(\|\mathbf{H}\|_2 \right)^2} \mathbf{H}^{\dagger} \mathbf{H} \right) \quad bit/c.u. \quad N_T \le N_R$$
(10.65)



Figura 10.9 Pre-codifica e ricezione con CSI al TX

$$c_{MIMO} = \log_2 \det \left(\mathbf{I}_{N_R} + \frac{E_s}{N_0} \frac{1}{\left(\|\mathbf{H}\|_2 \right)^2} \mathbf{H} \mathbf{H}^{\dagger} \right) \quad bit/c.u. \quad N_T \ge N_R \tag{10.66}$$

10.3.3 Capacità con allocazione della potenza al trasmettitore

Se (anche) il trasmettitore conosce la matrice di canale, è possibile migliorare la capacità del link rispetto alla (10.63). In questo caso infatti, possiamo materialmente *pre-codificare* il vettore dei simboli s attraverso la matrice V che il TX conosce, e quindi trasmettere i simboli pre-codificati in luogo degli originari. Fatto questo, possiamo decidere anche di *allocare la potenza radio totale* P_T in modo diseguale sui diversi simboli di s - e vedremo tra un attimo come e perché.

Secondo questa strategia, il TX invia il vettore dei simboli pre-codificati e con *controllo di potenza*

$$\mathbf{v} \stackrel{\triangle}{=} \mathbf{VAs} \ , \ \mathbf{A} \stackrel{\triangle}{=} \operatorname{diag}\{a_1, ..., a_{N_T}\}$$
 (10.67)

dove $a_i = \sqrt{2P_i/S_2}$, $i = 1, ..., N_T$ sono le varie ampiezze delle varie componenti del vettore di simboli inviati, in modo che ciascuno abbia la potenza prescritta P_i . Il ricevitore osserva il vettore soft

$$\mathbf{r} = \mathbf{H}\mathbf{v} + \mathbf{w} = \mathbf{U}\mathbf{D}\mathbf{V}^{\dagger}\mathbf{V}\mathbf{A}\mathbf{s} + \mathbf{w} = \mathbf{U}\mathbf{D}\mathbf{A}\mathbf{s} + \mathbf{w} = \mathbf{U}\mathbf{D}_{\mathbf{A}}\mathbf{s} + \mathbf{w}$$
(10.68)

e applica la matrice di ricezione \mathbf{U}^{\dagger} per ricavare

$$\bar{\mathbf{r}} = \mathbf{D}_{\mathbf{A}}\mathbf{s} + \bar{\mathbf{w}} \tag{10.69}$$

dove abbiamo indicato con $\mathbf{D}_{\mathbf{A}} = \text{diag}\{a_1d_1, ..., a_{N_T}d_{N_T}\}$ la matrice diagonale delle ampiezze ricevute, che tiene conto anche dei guadagni d_i dei canali virtuali.

Questa strategia di pre-codifica richiede ovviamente la conoscenza da parte del trasmettitore della cosiddetta *Channel Status Information* (CSI) nelle forma della **H** completa, o di **V** e **D**. In quest'ipotesi però il ricevitore materializza realmente le N_T equazioni disaccoppiate $\bar{r}_i = d_i a_i s^{(i)} + \bar{w}_i$, $i = 1, ..., N_T$ già viste in teoria, e la capacità in queste condizioni è

$$c_{MIMO} = \sum_{i=1}^{N_T} \log_2 \left(1 + \frac{d_i^2 a_i^2 S_2}{2\sigma_w^2} \right) = \sum_{i=1}^{N_T} \log_2 \left(1 + \frac{d_i^2 P_i}{\sigma_w^2} \right) \quad bit/c.u.$$
(10.70)

378 L'UNIONE FA LA FORZA - COMUNICAZIONI MIMO

dove ovviamente P_i è la potenza radio trasmessa dall'antenna # *i*.

Perché abbiamo introdotto l'allocazione di potenza tra le antenne, cioè la distribuzione di un certo budget totale massimo di potenza P_T tra i vari elementi radianti? La potenza massima trasmessa è sempre fissata, tipicamente dalla normativa di emissione radio in una certa banda frequenziale prevista dallo standard, e quindi le varie potenze alle antenne devono soddisfare il vincolo $P_1 + P_2 + ... + P_{N_T} = P_T$. Il criterio di allocazione per la multiplazione spaziale è ovvio: trovare la ripartizione delle potenze sulle antenne che massimizza la capacità di Shannon del collegamento. Dobbiamo cioè risolvere il seguente problema di massimo vincolato:

$$\max_{(P_1,\dots,P_{N_T})} = \sum_{i=1}^{N_T} \log_2 \left(1 + \frac{P_i}{\sigma_w^2/d_i^2} \right)$$

$$P_1 + P_2 + \dots + P_{N_T} - P_T = 0$$
(10.71)

Introduciamo allora un moltiplicatore di Lagrange λ e costruiamo la funzione Lagrangiana

$$\sum_{i=1}^{N_T} \log_2\left(1 + \frac{d_i^2 P_i}{\sigma_w^2}\right) + \lambda \left(P_1 + P_2 + \dots + P_{N_T} - P_T\right)$$
(10.72)

che deriviamo rispetto a P_i , $i = 1, ..., N_T$ e rispetto a λ , uguagliando a 0:

$$\frac{d_i^2/\sigma_w^2}{1 + \frac{d_i^2 P_i}{\sigma_w^2}} + \lambda = 0 \quad i = 1, ..., N_T$$

$$P_1 + P_2 + ... + P_{N_T} - P_T = 0 \tag{1}$$

Calcolando,

$$\frac{1}{\sigma_w^2/d_i^2 + P_i} + \lambda = 0 \quad \rightarrow \sigma_i^2 + P_i = -\frac{1}{\lambda} \forall i$$

$$P_1 + P_2 + \dots + P_{N_T} = P_T$$
(10.74)

dove abbiamo introdotto il rumore equivalente di canale $\sigma_i^2 \stackrel{\Delta}{=} \sigma_w^2/d_i^2$. Sommiamo adesso termine a termine tutte le N_T equazioni (10.74) per tutti i valori di *i*:

$$\sum_{i=1}^{N_T} \sigma_i^2 + \sum_{i=1}^{N_T} P_i = -N_T / \lambda \to -\frac{1}{\lambda} = \frac{\sigma_T^2 + P_T}{N_T} \stackrel{\triangle}{=} \bar{P}$$
(10.75)

per cui l'equazione di allocazione di potenza risulta

$$P_i = \bar{P} - \sigma_i^2 \tag{10.76}$$

Questa soluzione viene chiamata *criterio a riempimento d'acqua* (Water Filling, WF), che abbiamo già incontrato nella massimizzazione della capacità del canale ACGN (8.75). Per riassumere,

$$\begin{cases} P_i = \bar{P} - \sigma_i^2 , P_i > 0 \\ P_i = 0 \text{ altrimenti} \end{cases}$$
(10.77)

dove

$$\sigma_i^2 \stackrel{\triangle}{=} \frac{\sigma_w^2}{d_i^2} \quad , \quad \sigma_T^2 \stackrel{\triangle}{=} \sum_{i=1}^{N_T} \sigma_i^2 \quad , \quad \bar{P} = \frac{P_T + \sigma_T^2}{N_T} \tag{10.78}$$

In tal caso

$$c_{MIMO,wf} = \sum_{i=1}^{N_T} \log_2\left(\frac{d_i^2 \bar{P}}{\sigma_w^2}\right)$$
(10.79)

Nella soluzione WF, la quantità $\sigma_{w_i}^2$ è il rumore equivalente su ogni canale MIMO virtuale, cioè la varianza del rumore di ricezione σ_w^2 ri-scalata del guadagno di potenza intrinseco d_i^2 . Il criterio prescrive che la somma tra la potenza allocata P_i e il rumore equivalente σ_i^2 debba essere la stessa per ogni canale virtuale, e debba essere uguale alla potenza totale di segnale-più-rumore mediata su tutti i canali, amenoché su di un certo canale \bar{i} si abbia $\sigma_i^2 > \bar{P}$, nel qual caso la potenza sul canale va *azzerata* cioè l'antenna viene *spenta* ($P_i = 0$).

Il criterio WF è controintuitivo: notiamo che il rumore equivalente σ_i^2 sui vari sottocanali è variabile, e saremmo quindi tentati di allocare *più* potenza dove σ_i^2 è *maggiore*, controbilanciando così la rumorosità intrinseca del canale con una potenza maggiore, e ottenendo così sottocanali che lavorano più o meno allo stesso livello di SNR. Questo *non* è ciò che prescrive il criterio. Considerando la (10.77) vediamo che più grande è il livello di rumore, *minore* sarà la potenza del segnale - un criterio di esaltazione delle differenze. L'idea è che allocare molta potenza sulle antenne (più) rumorose significa sprecare energia per ottenere un contributo al bit-rate complessivo tutto sommato modesto. È meglio al contrario fare leva su canali già validi per sfruttarli al massimo, allocando su questi la maggior parte della potenza e quindi del bit-rate affidabile disponibile.

Esempio 10.56

Cerchiamo di valutare il miglioramento ottenibile con il criterio WF. Riprendiamo l'Esempio 55 e distribuiamo la potenza TX totale $P_T = 2S_2$ in modo diseguale, con $a_1^2 + a_2^2 = 2$. Calcolando la soluzione si trova

$$\sigma_T^2 = \left(\frac{1}{d_1^2} + \frac{1}{d_2^2}\right)\sigma_w^2 = 10,028\sigma_w^2 \quad , \quad \bar{P} = \frac{2S_2 + 10,028\sigma_w^2}{2} = S_2 + 5,014\sigma_w^2$$
(10.80)

e le potenze allocate sono

$$\begin{cases} P_1 = \bar{P} - \sigma_1^2 = S_2 + 5,014\sigma_w^2 - 10,000\sigma_w^2 = S_2 - 4,986\sigma_u^2 \\ P_2 = \bar{P} - \sigma_2^2 = S_2 + 5,014\sigma_w^2 - 27.7\sigma_w^2 = S_2 + 4,986\sigma_w^2 \end{cases}$$

Come ci attendevamo, $P_2 > P_1$ visto che $d_2^2 > d_1^2$. Naturalmente abbiamo ipotizzato $P_1 \ge 0$, altrimenti dovremmo spegnere TX1 - verifichiamo: $\text{SNR}_n = S_2/(2\sigma_w^2) = 45$ dB e quindi $S_2 = 63,250\sigma_w^2$ e $P_1 > 0$. Infine, dalla (10.79)

$$c_{MIMO,wf} = \log_2 \left(\frac{d_1^2 \bar{P}}{\sigma_w^2} \right) + \log_2 \left(\frac{d_2^2 \bar{P}}{\sigma_w^2} \right)$$

= $\log_2 \left(0.5 + 2 \cdot 10^{-4} \text{SNR}_n \right) + \log_2 \left(181 + 0.0722 \text{SNR}_n \right)$
= $2.77 + 11.27 = 14.04$ bit/c.u. (10.81)

e abbiamo incrementato di circa 2 bit/c.u. rispetto al caso di equidistribuzione della potenza. Il guadagno sarebbe stato maggiore se i due canali virtuali fossero stati ancora più *sbilanciati*.

380 L'UNIONE FA LA FORZA - COMUNICAZIONI MIMO

Al termine della procedura di allocazione della potenza, i canali virtuali si trovano ad operare con livelli di SNR considerevolmente diversi l'uno dall'altro. Come già osservato nel caso del canale ACGN, si deve procedere a un'operazione di *allocazione di bit* diseguale, utilizzando in teoria MOD/COD differenti sulle varie antenne a seconda delle relative capacità.

In pratica, fornire al TX la CSI completa è problematico, perché il RX deve re-inviare al TX molti parametri (coefficienti), impegnando non poco il canale di ritorno con dati di *controllo* che non fanno parte del collegamento, ma servono solo al funzionamento del modem. Per questo motivo si implementano strategie semplificate di pre-codifica che si basano su di un inseme pre-assegnato di matrici fissate dallo standard (ad esempio 4G/5G), scelte in tempo reale sulle base dell'indicazione (molto semplice e concisa: l'identificatore nell'elenco...) del RX.

10.3.4 Codifica Spazio-Tempo

Una volta ricavata la capacità del collegamento c, dobbiamo regolare il bit-rate di informazione secondo la consueta condizione $R_b \leq C = c/T_s$ (1/ T_s velocità di uso del canale) scegliendo codifica di canale e costellazione adatti (MOD/COD). La tecnologia MIMO offre un'ulteriore opportunità per effettuare questo "adattamento": i codici di canale convenzionali generano simboli codificati (di ridondanza o parità) che vengono distribuiti nel *tempo* insieme con i simboli di informazione (codici sistematici) per costruire il flusso codificato. In questo modo viene aumentata la velocità di segnalazione d'uscita, e appunto il rapporto tra i due clock rispettivamente di ingresso e di uscita costituisce il tasso r < 1 del codice.

Nel TX MIMO quel che si può fare è distribuire i simboli di ridondanza non (solo) nel *tempo*, ma (anche) nello *spazio*, cioè tra le antenne, realizzando uno schema di *codifica spazio-tempo*. Per fare un esempio, anche se non utilizzato in pratica, potremmo usare due antenne in TX con un codice sistematico a tasso 1/2, e decidere di inviare tutti e soli i bit di informazione sull'antenna TX1, e tutti e soli i bit di ridondanza sull'antenna TX2. Realizzeremmo una codifica puramente "spazio", diversa dalla codifica tradizionale puramente "tempo" dei collegamenti 1×1 .

Un codice Spazio-Tempo (ST) a blocco si costruisce creando un blocco A di N vettori di canale (in pratica, una matrice di canale $N_T \times N$):

TIME \rightarrow

$$\mathbf{A} \stackrel{\triangle}{=} [\mathbf{a}[0], \mathbf{a}[1], \dots, \mathbf{a}[N-1]] = \begin{bmatrix} a_1[0] & a_1[1] & \dots & a_1[N-1] \\ a_2[0] & a_2[1] & \dots & a_2[N-1] \\ \vdots & \vdots & \ddots & \vdots \\ a_{N_T}[0] & a_{N_T}[1] & \dots & a_{N_T}[N-1] \end{bmatrix} \quad \downarrow \text{SPACE}$$
(10.82)

Il blocco **A** contiene $N_T \times N$ simboli costituti da un mix di informazione e ridondanza. Nella multiplazione spaziale, non vi è alcuna ridondanza e il tasso di codifica è 1. I simboli di ridondanza possono essere distribuiti orizzontalmente (tempo) o verticalmente (spazio), o entrambi. Il tasso di codifica resta definito come il rapporto tra la velocità di informazione in presenza del codice e quella relativa al puro spatial multiplexing - se il codice è sistematico è pari al rapporto tra i simboli puramente informativi rispetto al numero totale $M \cdot N_T$.

Il codice spazio-tempo per antonomasia è la generalizzazione dello schema di Alamouti che abbiamo già incontrato come strategia 1×2 di trasmissione in diversità. Il codice

spazio-tempo di Alamouti 2×2 è infatti

$$\mathbf{A} = \begin{bmatrix} s_1[0] & s_2^*[0] \\ s_2[0] & -s_1^*[0] \end{bmatrix}$$
(10.83)

cioè esattamente lo schema già incontrato in Sect. 10.2.3. Dal nostro punto di vista attuale si tratta di un codice spazio-tempo con tasso r = 1/2. Si nota che le due *righe* del blocco di codice **A** sono *ortogonali* - la proprietà che permette di annullare l'intereferenza tra le due antenne nel decodificatore a blocco. Il codice di Alamouti è infatti un esempio di codice STOB *Space-Time Orthogonal Block*.

Il decodificatore per il codice di Alamouti 2×2 è la generalizzazione immediata della (10.24), che possiamo formalizzare secondo la notazione dei codici ST, a partire dall'equazione di canale:

$$\mathbf{R} = \mathbf{H}\mathbf{A} + \mathbf{W} \tag{10.84}$$

dove **R** e **W** sono le seguenti matrici $N_R \times N$:

$$\mathbf{R} \stackrel{\Delta}{=} [\mathbf{r}[0], \mathbf{r}[1], ..., \mathbf{r}[N-1]] \quad , \quad \mathbf{W} \stackrel{\Delta}{=} [\mathbf{w}[0], \mathbf{w}[1], ..., \mathbf{w}[N-1]]$$
(10.85)

Per il codice di Alamouti abbiamo N = 2 e $N_R = 2$:

$$\mathbf{R} = \begin{bmatrix} r_1[0] & r_1[1] \\ r_2[0] & r_2[1] \end{bmatrix} = \begin{bmatrix} h_{1,1}s_1[0] + h_{1,2}s_2[0] & h_{1,1}s_2^*[0] - h_{1,2}s_1^*[0] \\ h_{2,1}s_1[0] + h_{2,2}s_2[0] & h_{2,1}s_2^*[0] - h_{2,2}s_1^*[0] \end{bmatrix} + \mathbf{W} \quad (10.86)$$

Richiamiamo la decodifica 1×2 (10.24) riformulandola nel nostro contesto, e supponendo di avere a disposizione soltanto l'antenna RX1:

$$z_1^{(1)}[0] = h_{1,1}^* r_1[0] - h_{1,2} r_1^*[1] , \quad z_2^{(1)}[0] = h_{1,2}^* r_1[0] + h_{1,1} r_1^*[1]$$
(10.87)

che possiamo esprimere in forma matriciale come segue:

$$\mathbf{z}_{(1)}^{T} \stackrel{\triangle}{=} \begin{bmatrix} z_{1}^{(1)}[0] \ z_{2}^{(1)}[0] \end{bmatrix} = \begin{bmatrix} r_{1}[0] \ r_{1}^{*}[1] \end{bmatrix} \begin{bmatrix} h_{1,1}^{*} \ h_{1,2}^{*} \\ -h_{1,2} \ h_{1,1} \end{bmatrix}$$
(10.88)

L'antenna RX2 aggiunge diversità in *ricezione* alla diversità in trasmissione che già avevamo - per sfruttare RX2 implementiamo un *secondo* decodificatore 1×2 la cui uscita $\mathbf{z}_{(2)}$ ricavata analogamente alla (10.88) viene sommata alla precedente $\mathbf{z}_{(1)}$ ottenendo la variabile di decisione complessiva \mathbf{z} :

$$\mathbf{z}^{T} = (\mathbf{z}_{(1)} + \mathbf{z}_{(2)})^{T} = [r_{1}[0] \ r_{1}^{*}[1]] \begin{bmatrix} h_{1,1}^{*} & h_{1,2}^{*} \\ -h_{1,2} & h_{1,1} \end{bmatrix} + [r_{2}[0] \ -r_{2}^{*}[1]] \begin{bmatrix} h_{2,1}^{*} & h_{2,2}^{*} \\ -h_{2,2} & h_{2,1} \end{bmatrix}$$
(10.89)

ottenendo l'equazione finale di decodifica del codice di Alamuti 2×2 , che (si può dimostrare) fornisce le variabili di decisione soft per la stima MMSE e/o MV dei simboli di informazione trasmessi.

Esaminiamo le prestazioni ottenibili dal codice. L'analisi è lunga ma semplice - dobbiamo inserire (10.86) nella (10.89). Omettendo i dettagli, ma ricordando (10.25)-(10.26) e (10.89), otteniamo

$$z_{1}[0] = (|h_{1,1}|^{2} + |h_{1,2}|^{2} + |h_{2,1}|^{2} + |h_{2,2}|^{2})s_{1}[0] + W_{1}$$

$$z_{2}[0] = (|h_{1,1}|^{2} + |h_{1,2}|^{2} + |h_{2,1}|^{2} + |h_{2,2}|^{2})s_{2}[0] + W_{2}$$
(10.90)

dove W_1 e W_2 sono variabili aleatorie Gaussiane complesse con varianza per componente pari a $\sigma_W^2 = (|h_{1,1}|^2 + |h_{1,2}|^2 + |h_{2,1}|^2 + |h_{2,2}|^2)\sigma_w^2$. I termini di interferenza tra antenne sono ovviamente assenti, e su entrambe il rapporto-segnale rumore è

$$SNR_{ALA} = \frac{S_2(|h_{1,1}|^2 + |h_{1,2}|^2 + |h_{2,1}|^2 + |h_{2,2}|^2)}{2\sigma_w^2}$$
(10.91)

Abbiamo adesso *quattro* SNR 1 × 1 da confrontare con SNR_{*ALA*}. Se immaginiamo di non modificare le potenze ad ogni antenna TX passando da 1 a 2, allora SNR_{*ALA*} = SNR_{1,1} + SNR_{1,2} + SNR_{2,1} + SNR_{2,2} e possiamo ottenere fino a 6 dB di guadagno. Se però, più correttamente, immaginiamo che le due antenne TX trasmettano la stessa potenza totale di una unica antenna, otteniamo

$$SNR_{ALA} = \frac{SNR_{1,1} + SNR_{1,2}}{2} + \frac{SNR_{2,1} + SNR_{2,2}}{2}$$
(10.92)

Rispetto al caso 1×2 abbiamo in più un *guadagno di array* fino a 3 dB dovuto alle due antenne RX, che si combina con il *diversity gain* sul canale di Rayleigh che già avevamo nella trasmissione in diversità.

Lasciamo come esercizio per il lettore considerare un codice di Alamouti per un generico collegamento MIMO $N_R \times 2$, generalizzare la forma del decodificatore (10.89) e ricavare il rapporto segnale-rumore analogo a (10.92). Non è invece possibile generalizzare il codice per un numero di antenne di trasmissione $N_T > 2$, per l'impossibilità di creare codici STOB con costellazioni complesse con $N_T > 2$.

Nella pratica, la codifica ST (e in particolare il codice di Alamouti) è considerata come un'alternativa degradata rispetto alla multiplazione spaziale, da utilizzare a seconda dello stato del canale. Il presupposto è che il ricevitore sia in grado di i) valutare lo stato del canale, in particolare il rango della matrice del canale, ii) decidere il modo di funzionamento più opportuno dell'encoder MIMO, e iii) inviare tali informazioni al trasmettitore. Le reti cellulari 4G e 5G supportano entrambi il multiplexing spaziale e le diverse opzioni di codice di Alamouti.

10.3.5 MIMO OFDM

Facciamo un passo indietro verso le basi della tecnologia MIMO, in particolare verso la modellistica (10.29), basata sull'ipotesi di fondo che ciascuno degli $N_R \times N_T$ canali radio tra le varie antenne ha un comportamento piatto in frequenza,che può essere caratterizzato (solo) dal coefficiente di ampiezza/fase $h = |h|e^{j \angle h}$. Come può questa ipotesi verificarsi in pratica, data la banda molto larga delle comunicazioni wireless odierne? In un modello (più) generale dovremmo ammettere che il canale è caratterizzato da una *risposta impulsiva* $h_{i,\ell}(t)$ piuttosto che dalla *costante* $h_{i,\ell}$, e il segnale ricevuto all'antennas RX*i* dall'antenna TX ℓ dovrebbe essere ricavato dalla *convoluzione* di tutto il segnale trasmesso dall'antenna s $^{\ell}(t)$ con $h_{i,\ell}(t)$, piuttosto che il banale prodotto di s_m^{ℓ} con $h_{i,\ell}$ - torneremo su quest'argomento nel paragrafo 10.6.

La modellistica frequency-flat è esatta e applicabile se e quando la intendiamo applicata ad un segnale digitale OFDM (o in generale multiportante) con prefisso ciclico come nella (8.12). In tal caso, tutte le equazioni MIMO che abbiamo introdotto finora devono essere intese come *applicate a una singola sottoportante*, nel qual ambito la risposta in frequenza del canale è per costruzione *piatta*. Richiamiamo l'espressione (8.30) del segnale OFDM ricevuto sulla sottoportante k all'istante di simbolo OFDM #m per un collegamento 1×1
convenzionale:

 $z^{(k)}[m] = c^{(k)}[m] \cdot H_k + w^{(k)}[m] \quad k = 0, 1, ..., N - 1$ (10.93)

dove N è il numero delle sottoportanti, $c[m]^{(k)}$ è il simbolo di costellazione, $w[m]^{(k)}$ è la componente di rumore, e $H_k = H\left(\frac{k}{T_M}\right)$ è la risposta in frequenza del canale alla frequenza k/T_M della k-esima sottoportante. Se generalizziamo a un link MIMO 2 × 2 otteniamo

$$z_{1}^{(k)}[m] = c_{1}^{(k)}[m] \cdot H_{1,1;k} + c_{2}^{(k)}[m] \cdot H_{1,2;k} + w_{1}^{(k)}[m]$$
$$z_{2}^{(k)}[m] = c_{1}^{(k)}[m] \cdot H_{2,1;k} + c_{2}^{(k)}[m] \cdot H_{2,2;k} + w_{2}^{(k)}[m]$$
(10.94)

dove adesso abbiamo *quattro* risposte in frequenza relative ai quattro canali tra le 2×2 antenne. Le relazioni sono simili a quelle che compaiono nella (10.29), con un indice in più: il numero della sottoportante k. Assegnato un certo k, i coefficienti del canale MIMO $h_{i,\ell}$ sono i valori delle diverse risposte in frequenza $H_{i,\ell;k}$ del canale selettivo, valutate alla frequenza della sottoportante - e il modello MIMO resta valido.

Tenendo presente questo, tutti i risultati (capacità, codifica e decodifica ecc.) che abbiamo discusso finora devono essere intesi *per sottoportante*. Anche la stima del canale MIMO deve essere effettuata sottoportante per sottoportante con l'ausilio di simboli pilota come nella OFDM convenzionale. Ogni standard definisce particolari trame di pilota per identificare i vari $h_{i,\ell} = H_{i,\ell;k}$. L'idea di base per stimare $h_{i,\ell}$, $\ell = 1, ..., N_R$ è quella di attivare soltanto l'antenna TX*i* mantenendo tutte le altre spente, in modo che in ricezione non compaiano termini di interferenza e appunto sull'antenna RX ℓ si possa ricavare facilmente il valore del coefficiente di canale relativo, $h_{i,\ell}$. Tutte le tecnologie di comunicazione digitale wireless a banda larga hanno *de facto* adottato tecnologie multicarrier, e il modello a banda stretta che abbiamo adottato finora è ben giustificato e non costituisce minimamente una limitazione o approssimazione.

10.3.6 Capacità ergodica e fuori servizio sul canale di Rayleigh

Nei nostri calcoli di capacità, abbiamo finora ipotizzato che la matrice di canale **H** sia nota e costante in tutta la sessione di comunicazione. In generale però dovremmo considerarla variabile, o perchè vogliamo comprendere la variabilità da caso a caso dei possibili scenari di propagazione (e quindi valutare una capacità media statistica), o perchè si sta comunicando tra mezzi mobili e quindi, anche nell'ambito della stessa sessione, vi è lenta variabilità del canale a causa del movimento - e quindi necessità di considerare una media temporale.

Se dunque vogliamo considerare la variabilità del canale di propagazione, dobbiamo ammettere che **H** nell'espressione (10.63) della capacità di Shannon , piuttosto che una quantità nota è una *matrice aleaoria*, cioè una matrice i cui elementi sono variabili aleatorie che modellano il canale con fading tra coppie di antenne. Un modello semplice é che ogni elemento $h_{i,\ell}$ è una variabile aleatoria Gaussiana a media nulla e con componenti indipendenti aventi stessa varianza, ovvero con ampiezza di Rayleigh e fase uniforme. Un'ulteriore semplificazione è che gli elementi siano indipendenti tra di loro. Mentre il modello alla Rayleigh è molto ben verificato nella pratica, l'ipotesi sull'indipendenza dei guadagni del canale non è completamente giustificata: se la variabilità del canale è dovuta al movimento degli apparati, si può facilmente immaginare che i processi di fading derivanti dal movimento di due antenne riceventi (o trasmittenti) presentino una correlazione non trascurabile dato che le due antenne si muovono in modo solidale. La valutazione di questa



Figura 10.10 Capacità istantanea ed ergodica di un collegamento MIMO mobile (Schlegel)

dipendenza necessita di una modellazione fisica più sofisticata che non rientra nell'ambito di queste note. In generale, il presupposto di indipendenza porta a risultati *ottimistici*, dal momento che una matrice puramente casuale è con alta probabilità di rango *superiore* rispetto a una matrice con dipendenza statistica tra gli elementi.

La valutazione della capacità *media* di cui sopra necessita dunque di una operazione di media del valore ricavato supponendo la matrice **H** nota:

$$c_{E} = \mathbf{E}_{\mathbf{H}} \left\{ \log_{2} \det \left(\mathbf{I}_{N_{T}} + \frac{E_{s}}{N_{0}} \frac{1}{\left(\|\mathbf{H}\|_{2} \right)^{2}} \mathbf{H}^{\dagger} \mathbf{H} \right) \right\} \text{ bit/c.u. }, N_{T} \leq N_{R}$$

$$c_{E} = \mathbf{E}_{\mathbf{H}} \left\{ \log_{2} \det \left(\mathbf{I}_{N_{R}} + \frac{E_{s}}{N_{0}} \frac{1}{\left(\|\mathbf{H}\|_{2} \right)^{2}} \mathbf{H} \mathbf{H}^{\dagger} \right) \right\} \text{ bit/c.u. }, N_{T} \geq N_{R}$$

$$(10.95)$$

Questa capacità media è anche chiamata *capacità ergodica*, perché è generalmente intesa come la capacità media a lungo termine (media ergodica) sperimentata durante una lunga sessione di comunicazione da un collegamento MIMO in cui la variabilità è causata dalla variazione temporale del canale dovuta al movimento relativo di TX e RX. Il calcolo esplicito di questa quantità è complicato anche in caso di fading indipendente. Per darne un'idea, la Fig. 10.10 mostra un esempio (dovuto a Schlegel) di simulazione di un collegamento MIMO a lungo termine in condizioni tipiche. Le curve in blu rappresentano la capacità "istantanea" di un collegamento 2×2 (al di sotto) e 8×8 , confrontate con la capacità ergodica teorica (linee rosse) degli stessi collegamenti ma con elementi indipendenti della matrice di canale. La simulazione è stata condotta in condizioni tipiche (velocità = 50 km/h, E_s/N_0 =10 dB, f_0 =2.4 GHz, separazione degli elementi radianti TX e RX= $5\lambda_0$ =62.5 cm). La capacità è sensibilmente variabile nel tempo, a seconda dello stato del canale a un determinato istante, e talvolta si discosta notevolmente dalla media teorica (ergodica) calcolata con fading indipendenti.

Invece di calcolare la capacità ergodica e progettare un collegamento basato su questo valore pessimistico, può essere preferibile, come accade ogni volta che ci occupiamo delle prestazioni a lungo termine di un canale con fading, ragionare in termini di *probabilità di fuori servizio* (vedi anche l'Esempio 26), ovvero la frazione di tempo per il quale la capacità del collegamento risulta inferiore a un determinato valore di riferimento c_0 :

$$P_{out} \stackrel{\scriptscriptstyle \bigtriangleup}{=} \Pr\left\{c \le c_0\right\} \tag{10.96}$$

Questa probabilità di fuori servizio dipende ovviamente dalle statistiche della matrice di canale, e la sua formulazione esatta è di nuovo molto complicata - di solito viene valutata supponendo che c sia una variabile aleatoria Gaussiana con valor medio uguale a c_E e con una varianza che può essere calcolata in forma chiusa - non insisteremo su questo argomento.

10.4 MIMO Multiutente, Beamforming, Space-Division Multiple Access

Possiamo adesso scoprire un'ulteriore prospettiva nelle comunicazioni MIMO. Consideriamo l'uplink di una rete cellulare, in cui la stazione ricevente nella BTS è dotata di un numero relativamente grande di antenne $N_R \gg N_T$, e immaginiamo, anziché fare riferimento a un collegamento punto-punto con un *singolo terminale* mobile con N_T antenne, di avere a che fare con uno scenario pressoché equivalente in cui vi sono N_T terminali mobili con una *singola* antenna ciascuno - una specie di MIMO " distribuito". Abbiamo visto che nella modalità *spatial multiplexing* del collegamento punto-punto convenzionale, attraverso l'inversione della matrice di canale o comunque il ricevitore MMSE (10.41), si riesce ad annullare l'interferenza mutua tra le antenne - ragionando nella stessa maniera possiamo pensare a un nuovo tipo di accesso multiplo per servire N_T utenti convenzionali a singola antenna in una cella, *tutti operanti sulla stessa banda*, ma senz' alcuna mutua interferenza. La (10.54) ci dice che la capacità aggregata può essere sufficiente, dal momento che è sostanzialmente proporzionale al numero di utenti N_T . Chiamiamo questo scenario *MI-MO multiutente* - il nostro esempio si è concentrato sull'uplink, ma può essere facilmente applicato al downlink con lo stesso risultato.

Fondamentale in questa visione è la considerazione che la risorsa che sfruttiamo per effettuare l'accesso multiplo non è tempo, frequenza o codice, ma è ... *antenne*, quindi dobbiamo approfondire questo argomento, e nel far questo scopriremo anche nuovi punti di vista sul MIMO multiutente rispetto a quello che abbiamo appena introdotto.

10.4.1 Diagramma d'irradiazione di un'antenna

La possibilità di servire utenti multipli in un contesto simil-MIMO si può facilmente interpretare attraverso il concetto tradizionale di antenna a formazione di fascio (*beamforming antenna*). Per meglio comprendere questa tecnologia, dobbiamo fare una digressione sulle antenne in generale.

Come è noto, il campo lontano di un elemento radiante può essere caratterizzato dal cosiddetto diagramma di irradiazione. Supponiamo che la nostra antenna sia posizionata nell'origine di un sistema di riferimento come in Fig. 10.11, che stia irradiando una potenza totale P_T alla frequenza $f_0 = c/\lambda_0$ e che questa potenza radio sia uniformemente distribuita nello spazio. Ciò significa che a distanza r dall'antenna (con $r \gg \lambda_0$ in modo da essere in condizioni di *campo lontano*) il flusso superficiale di potenza dell'onda elettromagnetica è, nello spazio libero, $P_T/(4\pi r^2)$: la potenza totale irradiata (trasmessa) è distribuita uniformemente su di una superficie sferica di raggio r. In questo caso l'antenna è chiamata



Figura 10.11 Antenna nel proprio sistema di riferimento

isotropica o *omnidirezionale* perché invia potenza RF lungo qualsiasi direzione con la stessa intensità; si tratta in realtà di una configurazione di riferimento teorica molto difficile da realizzare in pratica, e spesso neanche necessaria.

Esempio 10.57

Supponiamo di voler progettare un'antenna per una stazione radio base di una rete cellulare che deve servire i telefonini in una cella da un sito posto sulla cima di una torre di sevizio. Per definizione, l'antenna deve essere omnidirezionale sul piano X-Y per poter servire efficacemente tutti gli abbonati, indipendentemente dalla collocazione del ricevitore; in altre parole, l'emissione dell'antenna deve essere uniforme rispetto all'angolo di *azimuth* ϕ (vedi Fig. 10.11). Tuttavia non è necessario irradiare verso il cielo o verso il centro della terra, e quindi è opportuno realizzare un'antenna la cui emissione *non* sia uniforme rispetto all'angolo di *elevazione* θ in Fig. 10.11. Al contrario, dovremo concentrare la potenza in un piccolo angolo di elevazione con θ che va da 0 (il livello del suolo) a pochi gradi, giusto per rivolgersi agli utenti negli edifici ad una certa altezza dal suolo, ma anche da 0 a -pochi gradi per servire gli utenti vicini alla torre. Se cerchiamo di visualizzare la distribuzione della potenza irradiata ci immaginiamo una specie di "ciambella" rotonda appoggiata sul piano orizzontale, come in Fig. 10.12.

Ciò che abbiamo descritto liberamente è il cosiddetto *diagramma d'irradiazione* di un'antenna, cioè la distribuzione della potenza irradiata rispetto agli angoli di azimuth ed elevazione. Il diagramma non è in generale uniforme, in modo da avere direzioni di emissione privilegiate, come nell'esempio della radiodiffusione.

Esempio 10.58

Un satellite per trasmissioni televisive domestiche orbita attorno alla terra in un'orbita cosiddetta geostazionaria equatoriale a 35,786 km di altitudine dalla superficie terrestre, rispetto alla quale appare in punto fisso del cielo (verso il quale puntano le parabole di ricezione). Considerando che il raggio della terra è pari a circa 6,378 km, l'antenna a bordo del satellite, per non sprecare inutilmente potenza radio, deve concentrare la propria emissione in un fascio (un cono) il cui angolo di apertura è $2 \sin^{-1}(6378/(35786 + 6378))=17.3$ gradi - vedi Fig. 10.13

MIMO MULTIUTENTE, BEAMFORMING, SPACE-DIVISION MULTIPLE ACCESS 387



Figura 10.12 Irradiazione di un'antenna per radiodiffusione



Figura 10.13 Emissione del satellite geostazionario

La Fig. 10.12 rappresenta un esempio di diagramma di irradiazione $G_T(\phi; \theta)$ che descrive come la potenza irradiata si distribuisce in una certa direzione individuata dalle coordinate angolari (ϕ =azimuth, θ =elevazione). Il diagramma è normalizzato in modo che

$$\int_{\theta=-\pi/2}^{\pi/2} \int_{\phi=-\pi}^{\pi} G_T(\phi;\theta) d\phi d\theta = 1$$
 (10.97)

e quindi il flusso superficiale di potenza nella direzione $(\phi; \theta)$ alla distanza r dall'antenna è

$$p_T(\phi;\theta) = \frac{P_T}{4\pi r^2} \cdot G_T(\phi;\theta)$$
(10.98)

Ci sono direzioni per le quali $G_T(\phi; \theta) > 1$, e cioè per le quali l'antenna presenta un vantaggio rispetto all'elemento isotropico ideale - per questo il diagramma di irradiazione è impropriamente chiamato anche guadagno dell'antenna. Più precisamente, di un'antenna si fornisce in pratica il massimo valore del diagramma $G_T \stackrel{\triangle}{=} \max_{(\phi;\theta)} G_T(\phi;\theta)$, che viene chiamato guadagno tout-court ed espresso in dB, e la direzione $(\phi_0; \theta_0) > 1$ in cui si realizza è la direzione principale di puntamento dell'antenna.

Il ragionamento appena sviluppato per un'antenna trasmittente si applica reciprocamente anche per un elemento ricevente: se l'antenna ricevente è idealmente isotropica ed è soggetta a un flusso di potenza p_T proveniente da una *qualunque* direzione (ϕ ; θ), allora la potenza radio raccolta è pari a $P_R = p_T \cdot A_e$ dove A_e è l'*area equivalente* dell'antenna - un parametro collegato alla dimensione fisica dell'elemento. Per un'antenna satellitare parabolica TV convenzionale, l'area equivalente è una frazione pari circa a 1/2 (che è rappresentativa di diversi fattori come la particolare geometria dell'antenna parabolica e l'efficienza elettrica complessiva) dell'area della "bocca" della parabola.

Anche l'antenna ricevente però non è in generale omnidirezionale, e quindi l'area equivalente dipende dalla direzione lungo la quale viene ricevuta la potenza radio: $A_e = A_e(\phi; \theta)$. Sfruttando una condizione di reciprocità elettromagnetica, si trova che l'area equivalente in ricezione è dipendente dal guadagno in trasmissione dell'antenna come segue:

$$A_e(\phi;\theta) = \frac{\lambda_0^2}{4\pi} \cdot G_R(\phi;\theta) \tag{10.99}$$

e quindi otteniamo la ben nota equazione di Friis per ricavare la potenza ricevuta su di un collegamento radio:

$$P_R(\phi;\theta) = \frac{P_T}{4\pi r^2} \cdot G_T(\phi;\theta) \frac{\lambda_0^2}{4\pi} \cdot G_R(\phi;\theta) = \frac{P_T}{(2\pi r/\lambda_0)^2} G_T(\phi;\theta) G_R(\phi;\theta) \quad (10.100)$$

10.4.2 Antenna ad array lineare e fattore d'array

Dalla sezione precedente abbiamo compreso che spesso è opportuno se non indispensabile realizzare antenne estremamente focalizzate o *direttive*, cioè che concentrano (il più possibi-le) la propria emissione lungo la direzione principale, e quindi minimizzano l'emissione nella altre direzioni (secondarie). Potremmo anche desiderare di variare su comando la direzione principale dell'antenna (per realizzare una comunicazione efficiente da/per un mezzo in movimento), possibilmente senza la necessità di parti meccaniche in movimento (antenna a *puntamento elettronico*).

La Figura 10.14 mostra la struttura-base di un sistema d'antenna ad *array* (schiera) lineare, composto da N elementi radianti identici (cioè con lo stesso diagramma $G(\phi; \theta)$) disposti in linea e intervallati da una distanza regolare d confrontabile con la lunghezza d'onda λ_0 - potrebbe rappresentare un classico sistema d'antenna per implementare un collegamento MIMO. Il segnale inviato ad ogni elemento può essere in generale variato in fase e/o ampiezza rispetto a un riferimento unico, ma per il momento supponiamo che questi parametri siano identici per tutti gli elementi. Analizzando un unico elemento radiante, il ricevitore alla (grande) distanza $r \ge 2L^2/\lambda_0$ è soggetto all'onda piana monocromatica (rappresentativa di un segnale radio modulato a banda stretta)

$$E_{RF}(z;t) = E_M \cos\left(2\pi f_0 t - \frac{2\pi}{\lambda_0} z\right) \tag{10.101}$$



Figura 10.14 Antenna ad array (schiera) lineare verticale

il cui equivalente in banda base è

$$E(z) = E_M \exp\left(-j\frac{2\pi}{\lambda_0}z\right) \tag{10.102}$$

Supponiamo ora che gli elementi dell'array siano disposti in verticale lungo l'asse Y, come in Fig. 10.15 (a), e consideriamo il generico angolo di elevazione θ lungo il quale si trova il ricevitore. Il campo ricevuto (equivalente in banda base) $E_A(z)$ è la sovrapposizione degli N campi $E_i(z)$, i = 0, ..., N - 1 generati separatamente dagli N elementi radianti. Dalla Fig. 10.15 (b) notiamo che il percorso dell'onda dall'elemento *i*-esimo fino al ricevitore è diverso per ogni elemento: l'i + 1-esimo è più lungo del "precedente" *i*-esimo di un "offset" $d \sin(\theta)$. Questo offset è molto piccolo rispetto a r, e quindi le varie emissioni dei vari elementi vengono ricevute praticamente con la stessa *ampiezza* (ricordiamo l'esempio 52). Al contrario, l'offset *non* può essere trascurato nel calcolo della *fase* dei segnali poiché è confrontabile con la lunghezza d'onda: lo sfasamento tra i segnali ricevuti da due elementi "adiacenti" dell'array è $2\pi d \sin(\theta)/\lambda_0$. Riassumendo, il campo totale ricevuto $E_A(z)$ è

$$E_A(z) = \sum_{i=0}^{N-1} E_0(z) \exp\left(-j2\pi i \frac{d\sin(\theta)}{\lambda_0}\right) = E_0(z) \frac{1 - \exp\left(-j2\pi N \frac{d\sin(\theta)}{\lambda_0}\right)}{1 - \exp\left(-j2\pi \frac{d\sin(\theta)}{\lambda_0}\right)}$$
$$= E_0(z) \exp\left(-j\pi N \frac{d\sin(\theta)}{\lambda_0}\right) \frac{\sin\left(\pi N \frac{d\sin(\theta)}{\lambda_0}\right)}{\sin\left(\pi \frac{d\sin(\theta)}{\lambda_0}\right)}$$
(10.103)

dove $E_0(z)$ è il campo ricevuto dal solo elemento #0.

La potenza ricevuta è proporzionale a $|E_A(z)|^2$, per cui dalla (10.103) vediamo che l'effetto dell'array sull'emissione totale è l'introduzione del seguente fattore di *direttività* nell'emissione :

$$A_F(\theta) \stackrel{\triangle}{=} \left| \frac{\sin\left(\pi N \frac{d\sin(\theta)}{\lambda_0}\right)}{N\sin\left(\pi \frac{d\sin(\theta)}{\lambda_0}\right)} \right|^2$$
(10.104)

390 L'UNIONE FA LA FORZA - COMUNICAZIONI MIMO



Figura 10.15 Funzionamento dell'array lineare



Figura 10.16 Esempio di fattore d'array - $d = \lambda_0/2$, N = 16 (a), e N = 32, (b)

dove abbiamo normalizzato tale fattore in modo che $A_F(0) = 1$.

Questa quantità prende il nome di *fattore d'array*, e ha l'effetto di focalizzare l'emissione dell'antenna lungo la direzione $\theta = 0$. La Fig. 10.16 mostra l'andamento del fattore d'array per N = 16 e N = 32, dove la direttività è chiarissima, e dove notiamo anche che il fascio tende a focalizzarsi sempre di più all'aumentare di N. L'andamento del fattore di array nello spazio è invece rappresentato in Fig. 10.17 - si nota la direttività in elevazione e l'omnidirezinalità in azimuth.

Osserviamo anche che il fattore d'array A_F non è l'unico contributo alla direttività dell'antenna. Infatti, il diagramma d'irradiazione complessivo è il prodotto tra il fattore dell'array lineare *e* il diagramma di irradiazione "nativo" degli elementi radianti, che finora abbiamo completamente trascurato:

$$G_A(\phi, \theta) = G_T(\phi, \theta) \cdot A_F(\theta)$$

Consideriamo delle antenne elementari a *dipolo* con lunghezza $\lambda_0/2$, cioè semplici elementi filari conduttori di lunghezza pari a metà della lunghezza d'onda radio, come in Fig. 10.18. Il digramma del dipolo a $\lambda_0/2$ è la classica "ciambella"

$$G_T(\phi;\theta) = \frac{\cos^2\left(\frac{\pi}{2}\sin(\theta)\right)}{\cos^2(\theta)}$$
(10.105)

omnidirezionale nel piano X-Y rappresentata in Fig. 10.18 (a). Con questi elementi, l'antenna ad array presenta il diagramma totale

$$G_A(\phi;\theta) = \left| \frac{\sin\left(\pi N \frac{d\sin(\theta)}{\lambda_0}\right)}{N\sin\left(\pi \frac{d\sin(\theta)}{\lambda_0}\right)} \right|^2 \frac{\cos^2\left(\frac{\pi}{2}\sin(\theta)\right)}{\cos^2(\theta)}$$
(10.106)



rappresentato in Fig. 10.18 (b).

La direttività dell'antenna è chiarissima: abbiamo ristretto il fascio radio realizzando un elemento selettivo che trasmette (o reciprocamente riceve) soltanto nella direzione θ =0 per servire utenti a terra (elevazione 0). Vediamo anche che l'antenna, oltre al fascio principale, presenta anche fasci *secondari* o *laterali* indesiderati. In generale, maggiore è *N*, più ridotti sono i fasci secondari. Ribadiamo che le considerazioni relativi al beamforming valgono sia in fase di trasmissione, nella quale il ruolo dell'array è "attivo" sia in fase di ricezione in cui l'array è "passivo".

Non insisteremo ulteriormente sullo studio e sui criteri di progettazione dei diversi tipi di array lineari, ne' introdurremo array 2D più complicati che possono conferire direttività sia sul piano *verticale* che su quello *orizzontale*, creando un fascio "a matita". Investigheremo invece un altro argomento: se dobbiamo comunicare con un aereo visto da terra sotto l'angolo di elevazione (ad esempio) θ_0 =15 gradi, possiamo puntare il fascio dell'array di Fig. 10.17 verso l'aereo, senza la necessità di fisicamente inclinare l'antenna? In altre parole, possiamo *sintetizzare* un fascio con un'elevazione arbitraria θ_0 ? La relativa tecnologia di *beamforming* è l'argomento della prossima sezione.

10.4.3 Beamforming e vettore di steering

La soluzione al problema di *beamforming* che abbiamo appena enunciato risiede nella possibilità di alimentare i diversi elementi dell'array con segnali di fasi α_i e ampiezze A_i , $i = 0, \ldots, N - 1$ appropriate. Questa tecnologia è utilizzata da decine di anni nel campo dei radar, della radioastronomia e delle comunicazioni con lo spazio profondo sotto il nome di *phased array*. Applicazione più recente e più evoluta è costituita dalle cosiddette *smart antennas* per stazioni radio base cellulari.

Pensiamo ad un *phased array* in cui per semplicità $A_i = 1 \forall i$, e supponiamo di voler "ruotare" il fascio principale, normalmente in θ =0, fino a $\theta = \theta_0$ aggiustando opportunamente le fasi degli elementi radianti α_i . Se consideriamo queste fasi di alimentazione nel



Figura 10.18 Array lineare di N=16 dipoli a $\lambda_0/2 \operatorname{con} d = \lambda_0/2$ - Configurazione e diagramma del singolo elemento (a); Fattore d'array e diagramma totale (b).



Figura 10.19 Esempi di fascio non ruotato (a) e ruotato (b) - $d = \lambda_0/2$, N = 16, $\theta_0 = \pi/4$

fattore di array otteniamo:

$$A_F(\theta) = \left| \sum_{i=0}^{N-1} \exp\left[-j \left(2\pi i \frac{d\sin(\theta)}{\lambda_0} - \alpha_i \right) \right] \right|^2$$
(10.107)

Si può banalmente ruotare (spostare) il fascio da 0 a θ_0 se imponiamo

A

$$\alpha_i = 2\pi i \frac{d\sin(\theta_0)}{\lambda_0} \tag{10.108}$$

Così facendo non si ottiene in realtà una vera e propria rotazione rispetto a θ del diagramma originario, ma il massimo del fattore d'array viene comunque a spostarsi su θ_0 :

$$_{F}(\theta) \stackrel{\triangle}{=} \left| \frac{\sin\left(\pi N \frac{d(\sin(\theta) - \sin(\theta_{0}))}{\lambda_{0}}\right)}{N \sin\left(\pi \frac{d(\sin(\theta) - \sin(\theta_{0}))}{\lambda_{0}}\right)} \right|^{2}$$
(10.109)

La Fig. 10.19 mostra il fattore d'array per la stessa antenna della Fig 10.16 con N=16, ma adesso con $\theta_0 = \pi/4$, cioè 45 gradi. I due fattori d'array (ruotato e non ruotato) sono messi a confronto nel digramma *polare* di Fig. 10.20 (chiamato anche "radar") comunemente usato per visualizzare i diagrammi d'irradiazione delle antenne.

Modificando in tempo reale i valori delle fasi α_i , possiamo rendere il fascio mobile e adattativo, ad esempio per seguire istante per istante la posizione nel cielo di un satellite e garantire un collegamento Internet ottimale per un apparato a bordo di un'imbarcazione o di un aereo - una tecnologia che viene chiamata *antenna a puntamento elettronico* (electronically steerable). Le fasi possono essere variate tramite opportuni componenti (phase shifters) che agiscono a frequenza radio, oppure lavorando in banda base mediante algoritmi anche sofisticati di Digital Signal Processing (DSP). I criteri di progetto dei phased array riguardo il numero N e la spaziatura d degli elementi radianti nonché (e soprattuto) le fasi α_i sono ormai consolidati e oggetto di testi specialistici. Un valore tipico per la spaziatura è $d = \lambda_0/2$ (adottato nei nostri esempi) perchè porta a semplificazioni di progetto e permette di ottenere sistemi di antenna efficienti e compatti.

Se l'algoritmo di *steering* viene implementato in banda base, con tutti i gradi di libertà derivanti dall'approccio DSP, ha senso pensare di variare anche le ampiezze A_i sui vari elementi (oltre alle fasi α_i), e di considerare il problema generale del *beamforming*:



"Dato un andamento desiderato (ideale) del fascio $A_D(\theta)$ come nell'esempio di Fig. 10.21

caratterizzato da un angolo di puntamento θ_0 e da una apertura $\Delta \theta$, ricavare l'insieme di coefficienti $g_i = A_i \exp(j\alpha_i)$ in modo che il fattore di array

$$A_F(\theta) = \left| \sum_{i=0}^{N-1} g_i \exp\left[-j2\pi i \frac{d\sin(\theta)}{\lambda_0} \right] \right|^2$$

(10.110)

risulti il più vicino possibile, secondo una certa metrica, a quello desiderato." Il parametro θ_0 regola il puntamento dell'antenna, mentre $\Delta \theta$ (l'apertura del fascio) impone la maggiore o minore direttività.

Osserviamo che se $d = \lambda_0/2$, allora

$$A_F(\theta) = \left| \sum_{i=0}^{N-1} g_i \exp\left[-j2\pi i \frac{\sin(\theta)}{2} \right] \right|^2 = \left| \sum_{i=0}^{N-1} g_i \exp\left[-j2\pi i \nu \right] \right|^2$$
(10.111)

dove $\nu \stackrel{\triangle}{=} \sin(\theta)/2$. Questa relazione è di fatto analoga al calcolo della risposta in frequenza di un filtro digitale FIR avente coefficienti (risposta impulsiva) g_i (2.82), a maggior ragione considerando che se θ varia in $[-\pi/2, \pi/2)$, allora ν varia in [-1/2, 1/2) proprio come la frequenza normalizzata f/f_c (f_c frequenza di campionamento) dello spettro di un segnale digitale.

Se dunque $d = \lambda_0/2$, possiamo (ri-)utilizzare per il progetto del cosiddetto *vettore di* steering $\mathbf{g} \stackrel{\triangle}{=} [g_0 \ g_1 \ ... \ g_{N-1}]$ (ovvero l'insieme dei coefficienti complessi dell'array) tutti i criteri ben noti per il progetto di filtri FIR, e tutte le relative astuzie. In particolare, così come possiamo progettare filtri FIR multi-banda (chiamati anche filtri *a pettine*), possiamo anche realizzare antenne *multi-fascio* (multi-beam) per "puntare" contemporaneamente più ricevitori a elevazioni diverse. Nelle comunicazioni wireless, un array multi-beam a puntamento elettronico viene chiamato *smart antenna*, ed è alla base dello *Space Division Multiple Access* (SDMA), intimamente collegata all'argomento principale di questo Capitolo, ovvero le comunicazioni MIMO.



Figura 10.21 Fattore d'array ideale nei dominî $\theta \in v$

10.4.4 Space-Division Multiple Access (SDMA) and Multiuser MIMO

Per capire meglio SDMA, ruotiamo il nostro array in modo che giaccia lungo l'asse X anziché l'asse Y degli esempi precedenti. Otteniamo un'antenna omnidirezionale sul piano verticale (in elevazione), ma (fortemente) direzionale sul piano X-Y (azimut), cosicché (in assenza di steering)

$$A_F(\phi) \stackrel{\triangle}{=} \left| \frac{\sin\left(\pi N \frac{d\sin(\phi)}{\lambda_0}\right)}{N\sin\left(\pi \frac{d\sin(\phi)}{\lambda_0}\right)} \right|^2$$

(10.112)

Il diagramma d'irradiazione è come in Fig. 10.20, ma giacente sul piano *orizzontale* (X-Y) - come se riportassimo il fascio su di una mappa. Immaginiamo poi che l'origine degli assi sia la posizione di una stazione radio base (BTS, Base Transceiver Station) di una rete cellulare che debba servire una popolazione di U utenti (mobili) distribuiti all'interno della cella come in Fig. 10.22. Ogni terminale utente #u, u = 1, ..., U, trasmette con un'antenna omnidirezionale convenzionale un simbolo di costellazione $s^{(u)}[m]$ al tempo mT_s , che viene ricevuto dalla BTS lungo una certa direzione ϕ_u .

Se il canale radio è frequency-flat (oppure immaginando di considerare come di consueto la sottoportante k di uno spettro OFDM), possiamo ricevere il simbolo dell'utente # u senza interferenza dagli altri utenti se applichiamo al segnale ricevuto dagli elementi dell'array di BTS un vettore di steering $\mathbf{g}^{(u)} = [g_0^{(u)} g_1^{(u)} \dots g_{N-1}^{(u)}]^T$ in modo da sintetizzare un fascio direttivo lungo la direzione ϕ_u di quell'utente: $g_i^{(u)} = \exp(j2\pi i d \sin(\phi_u)/\lambda_0)$. I diversi utenti possono essere separati facendo leva sulla diversità angolare della loro posizione spaziale 2D - ecco la ragione del nome SDMA - anche se "occupano" la stesso slot temporale o lo stesso canale frequenziale. Naturalmente, la BTS deve conoscere la posizione dell'utente nella cella per poter applicare il vettore di steering corretto, proprio come deve conoscerne lo slot temporale o la frequenza di canale usando i tradizionali TDMA o FDMA. Possiamo ricollegare questa interpretazione geometrico-intuitiva dello SDMA all'approccio rigoroso dei sistemi MIMO che già conosciamo? Dopotutto si sta parlando in entrambi i casi di sfruttare i molti elementi di un'antenna per separare utenti diversi.

Chiamiamo come sempre **r** l'uscita degli elementi dell'array ricevente della BTS; secondo il modello (10.103), abbiamo a che fare nell'uplink della cella con U collegamenti MIMO MULTIUTENTE, BEAMFORMING, SPACE-DIVISION MULTIPLE ACCESS 397



Figura 10.22 Funzionamento dello SDMA

MIMO $N_R \times 1$ (aka SIMO, single-input multiple-output) che si sovrappongono sulla stessa banda (tempo), cosicché

$$\mathbf{r} = \mathbf{H}\mathbf{s} + \mathbf{w} = \sum_{\ell=1}^{C} s^{(\ell)}[m] \cdot \mathbf{h}^{(\ell)} + \mathbf{w} = s^{(u)}[m]\mathbf{h}^{(u)} + \sum_{\ell \neq u} s(\ell)[m] \cdot \mathbf{h}^{(\ell)} + \mathbf{w} \quad (10.113)$$

dove la matrice di canale $N_R \times 1$ del generico utente # ℓ è il vettore colonna

$$\mathbf{h}^{(\ell)} = \begin{bmatrix} h_1^{(\ell)} \\ \vdots \\ h_{N_R}^{(\ell)} \end{bmatrix} \quad h_i^{(\ell)} = \bar{h}^{(\ell)} \exp(-j2\pi(i-1)d\sin(\phi_\ell)/\lambda_0) \quad , \quad i = 1, \dots, N_R$$

e $\bar{h}^{(\ell)}$ è il coefficiente di canale che si avrebbe nel caso convenzionale di antenna singola di ricezione - di fatto l'uscita del primo elemento dell'array (*i*=1).

Questa relazione mostra che in generale l'utente desiderato #u è disturbato da *interferenza* da accesso multiplo (MAI, Multiple Access Interference), derivante dai segnali degli altri utenti. Senza considerare questa MAI, ma sapendo che il segnale dell'utente proviene dalla direzione ϕ_u , un approccio di tipo beamforming richiede di usare un vettore di steering $\mathbf{g}^{(u)}$ con elementi secondo la (10.108), cioè, in breve,

$$\mathbf{g}^{(u)} = [1, \exp(j2\pi d\sin(\phi_u)/\lambda_0), ..., \exp(j2\pi (N-1)d\sin(\phi_u)/\lambda_0)]^T$$
(10.114)

e l'uscita del beamformer dell'utente #u, cioè la relativa variabile di decisione, è

$$z^{(u)}[m] = \mathbf{g}^{(u)T}\mathbf{r} = \sum_{i=1}^{N_R} r_i[m] \exp(j2\pi(i-1)d\sin(\phi_u)/\lambda_0)$$
(10.115)

Come si confronta quest'approccio con ciò che già sappiamo dalle tecniche MIMO? Da questo punto di vista, e trascurando la MAI, il nostro ricevitore uplink $N_R \times 1$ di BTS è un ricevitore *in diversità* che dovrà adottare una legge di combinazione MRC (10.11) con coefficienti $a_i^{(u)} = h_i^{*(u)}$:

$$\zeta^{(u)}[m] = \sum_{i=1}^{N_R} r_i[m] h_i^{*(u)} = \bar{h}^{*(u)} \sum_{i=1}^{N_R} r_i[m] \exp(j2\pi(i-1)d\sin(\phi_u)/\lambda_0) \quad (10.116)$$

Si vede che questa legge di combinazione, a meno del coefficiente $\bar{h}^{*(u)}$ è *equivalente* a quella implementata dal *beamforming* attraverso il vettore di steering dell'array. Il beamforming è quindi ciò che di fatto viene (anche) realizzato dal ricevitore MRC MIMO, che in più è ottimale rispetto al rumore purchè lo si corregga con il termine di canale $\bar{h}^{*(u)}$.

Tornando poi al punto di vista "geometrico", riconosciamo che l'approccio beamforming è anche ottimale *rispetto alla MAI* perchè il nostro linear array isola il segnale dell'utente u dagli altri per la proprietà di direttività dell'array - la stessa cosa che si ha con l'FD-MA, quando usiamo un filtro adattato *passa-banda* per isolare il canale dell'utente # u, e contemporaneamente rimuovere più rumore possibile.

Dunque nello SDMA utenti che sfruttano la stessa banda di frequenza allo stesso tempo possono essere *separabili* se occupano diverse posizioni nello spazio. La separazione degli utenti può essere effettuata sfruttando la diversità spaziale e ricevendo i segnali emessi dai diversi terminali utente mediante un'antenna MIMO (array) i cui elementi sono a loro volta separati spazialmente. È chiaro che, seguendo l'approccio del beamforming, gli utenti possono essere effettivamente separati (*ortogonalizzati...*) con trascurabile MAI (solo) se la loro distanza angolare $\Delta \phi$ è maggiore dell'apertura che l'antenna della BTS ricevente è in grado di sintetizzare, e che dipende come sappiamo dalla dimensione dell'array N_R .

In generale, per essere in grado di sfruttare appieno la diversità spaziale, il numero di antenne riceventi alla BTS N_R deve essere adeguatamente maggiore del numero di utenti U. Questa condizione è chiara (anche) dal punto di vista del beamforming o della smart antenna: come si vede in Fig. 10.22 la smart antenna per SDMA deve essere in grado di sintetizzare *tanti fasci* con apertura adeguatamente piccola *quanti utenti* devono essere serviti nella cella per mantenerli ortogonali, e ciò si può realizzare efficacemente solo se $N_R \gg U$.

Detto questo, che differenza c'è tra SDMA e MU-MIMO? Spesso le due cose vengono considerate come identiche; in realtà SDMA si basa soltanto su considerazioni geometriche e tenta di separare gli utenti soltanto sulla base dell'informazione sulla loro posizione. Nel MU-MIMO, i criteri per ottimizzare i vettori di beamforming possono essere *diversi* dalla semplice separazione geometrica. Consideriamo infatti (10.113) e (10.115) e ricaviamo l'espressione della variabile di decisione soft $z_m^{(u)}$ del ricevitore di BTS per l'utente (u) al tempo m:

$$z^{(u)}[m] = \mathbf{g}^{(u)T}\mathbf{r} = s^{(\ell)}[m]\mathbf{g}^{(u)T}\mathbf{h}^{(u)} + \sum_{\ell \neq u} s^{(\ell)}[m] \cdot \mathbf{g}^{(u)T}\mathbf{h}^{(\ell)} + W_m^{(u)}$$
(10.117)

dove $\mathbf{g}^{(u)}$ è il vettore di steering dell'utente # u e dove abbiamo distinto i termini di i) utente desiderato, ii) MAI, e iii) rumore $W_m^{(u)} = \mathbf{g}^{(u)T}\mathbf{w}$. Se ammettiamo di avere nella cella "molti" utenti indipendenti, possiamo dire che le statistiche della MAI sono con ottima approssimazione Gaussiane per il teorema limite centrale. Quindi, la somma di MAI e rumore ha statistiche Gaussiane, e possiamo calcolare la capacità di Shannon dell'utente #u dopo beamforming in modo abbastanza semplice, a partire dal rapporto segnale-rumore-più-interferenza SINR:

$$c^{(u)} = \log\left(1 + \text{SINR}_{u}\right) = \log\left(1 + \frac{S_{2}|\mathbf{g}^{(u)T}\mathbf{h}^{(u)}|^{2}}{2\sigma_{W}^{2,(u)} + S_{2}\sum_{\ell \neq u}|\mathbf{g}^{(u)T}\mathbf{h}^{(\ell)}|^{2}}\right)$$
(10.118)

dove $\sigma_W^{2,(u)} = \|\mathbf{g}^{(u)}\|^2 \sigma_w^2$. Il progetto del vettore di steering si può (ri)formulare come segue: trovare l'insieme dei vettori degli utenti che massimizzano una qualche capacità "collettiva" nella cella. Diciamo "collettiva" perchè ovviamente la scelta di un certo $\mathbf{g}^{(u)}$ influenza anche la capacità degli altri utenti $\neq u$ attraverso la MAI, e quindi una massimizzazione individuale della capacità non ha senso.

Una scelta molto usata per la capacità collettiva è la *somma pesata* (Weighted Sum Rate, WSR)

$$c_{wsr} \stackrel{\triangle}{=} \sum_{u=0}^{U-1} \alpha_u c^{(u)} \tag{10.119}$$

dove i coefficienti $0 \le \alpha_u \le 1$ sono le *priorità* dei vari utenti scelte a seconda della rilevanza della *u*-esima applicazione, $\sum \alpha_u = 1$, e soggetta ovviamente a un vincolo di potenza massima

$$\sum_{u=0}^{U-1} \|\mathbf{h}^{(u)}\|^2 \le H \tag{10.120}$$

Il vettore di steering normalizzato soluzione dell'ottimizzazione vincolata è di tipo MMSE:

$$\mathbf{g}_{wsr}^{(u)} = \frac{(\mathbf{I} + \sum_{\ell \neq u} q_{\ell} \mathbf{h}_{\ell} \mathbf{h}_{\ell}^{\dagger})^{-1} \mathbf{h}_{u}}{\|(\mathbf{I} + \sum_{\ell \neq u} q_{\ell} \mathbf{h}_{\ell} \mathbf{h}_{\ell}^{\dagger})^{-1} \mathbf{h}_{u}\|}$$
(10.121)

dove i coefficienti q_u dipendono dalle priorità α_u .

L'equazione (10.117) relativa all'uplink del nostro sistema multiutente $N_R \times 1$, è applicabile senza (quasi) nessuna modifica al caso $1 \times N_R$ del downlink. I vettori di steering $\mathbf{g}^{(u)T}$ devono adesso essere applicati da parte della BTS ai simboli $s^{(u)}[m]$ prima della trasmissione, ma la variabile di decisione $z^{(u)}[m]$ al terminale utente ha sostanzialmente la stessa forma - i due canali UL e DL sono reciproci. Nel contesto del multiuser MIMO per il downlink, applicare lo steering vector è chiamato precoding.

Oltre ai ricevitori appena esaminati, che sono basati su di un approccio di elaborazione lineare (vettore di steering), nel contesto del MU-MIMO si possono progettare anche ricevitori più complicati ma più efficaci. In particolare, i ricevitori a *cancellazione d'interfenza*, dopo avere effettuato un processing lineare simile a quello appena visto, identificano l'utente ricevuto con il massimo SINR, chiamiamolo \bar{u} , ne rivelano il dato $s^{(\bar{u})}[m]$ basandosi su $z^{(\bar{u})}[m]$, e poi *ricostruiscono* sulla base della stima di canale il contributo di \bar{u} alla MAI $\hat{s}_m^{(\bar{u})}\mathbf{h}^{(\bar{u})}$ nella (10.117). Tale contributo viene sottratto (*cancellato*) dalle uscite dei rivelatori lineari di tutti gli altri utenti $\ell \neq \bar{u}$, dopo pesatura con il vettore di steering $\mathbf{g}^{(\ell)T}$ relativo. Fatto questo, viene identificato, nei restanti U - 1 utenti con interferenza cancellata, quello che ha il miglior SINR, stimandone il dato, ricostruendo l'interferenza... e così via. L'analisi di questo approccio, in certe situazioni molto efficace, è complicata e non la tratteremo.

Non tratteremo neanche il caso in cui i vari link degli utenti MU-MIMO, anziché del tipo $N_R \times 1$ sono più in generale $N_R \times N_T$ con $N_T > 1$. Accenneremo a questa possibilità nella sezione successiva relativa ad una tecnologia in sviluppo, cioè il MIMO in versione *massiccia*.

10.5 La Nuova Frontiera: Massive MIMO

A questo punto niente ci impedisce di considerare uno scenario in cui anche i terminali mobili hanno un numero di antenne > 1. In questo caso si deve rivedere l'approccio di beamforming studiato nella sezione precedente, perché il collegamento potrà presentare anche una qualche forma di *multiplazione spaziale*. Restando alla considerazione dell'uplink, dovremmo stavolta accertarci che $N_R \gg U \cdot N_T$ e dovremmo ricalcolare la capacità di Shannon in queste condizioni - un po' complicato.

400 L'UNIONE FA LA FORZA - COMUNICAZIONI MIMO

D'altra parte, il vincolo MU-MIMO $N_R \gg U \cdot N_T$ porta naturalmente alla considerazione di sistemi di antenne con un numero molto, *molto* grande di elementi - forse fantascienza, in realtà una delle tecnologie studiate nel contesto delle reti cellulari 5G sotto il nome di *Massive MIMO* (MMIMO).

Riprendiamo l'espressione della capacità ergodica su canale di Rayleigh c_E (10.95). In generale, il calcolo di c_E è complicato, salvo il caso in cui il numero di antenne su di un lato del collegamento MIMO (cioè uno tra N_R o N_T) è molto grande - un risultato asintotico molto informativo. Vediamo il caso simil-downlink, nel quale cioè $N_T \gg N_R$:

$$c_{E} = \mathsf{E}_{\mathbf{H}} \left\{ \log_{2} \det \left(\mathbf{I}_{N_{R}} + \frac{E_{s}}{N_{0}} \frac{1}{\left(\|\mathbf{H}\|_{2} \right)^{2}} \mathbf{H} \mathbf{H}^{\dagger} \right) \right\}$$

e supponiamo che gli elementi di **H** siano Gaussiani indipendenti con $E\{|h_{i,\ell}|^2\} = 1$. L'elemento i, ℓ di \mathbf{HH}^{\dagger} $(i, \ell = 1, ..., N_R)$ è

$$\left(\mathbf{H}\mathbf{H}^{\dagger}\right)_{i,\ell} \stackrel{\Delta}{=} \sum_{p=1}^{N_T} h_{i,p} h_{\ell,p}^* = N_T \cdot \frac{1}{N_T} \sum_{p=1}^{N_T} h_{i,p} h_{\ell,p}^* \to N_T \mathbf{E}\{h_{i,p} h_{\ell,p}^*\} = N_T \delta[i-\ell]$$
(10.122)

dove la tendenza asintotica si verifica per la legge dei grandi numeri: la media aritmetica tende (con varianza piccolissima) alla media statistica (valore atteso). Nella stessa maniera possiamo dimostrare che $(||\mathbf{H}||_2)^2 \rightarrow N_T N_R$. In conclusione,

$$c_E \to \log_2 \det\left(\left(1 + \frac{E_s/N_R}{N_0}\right) \mathbf{I}_{N_R}\right) = N_R \log_2\left(1 + \frac{E_{s,r}}{N_0}\right) \quad \text{bit/c.u.} \quad (10.123)$$

dove $E_{s,r} = E_s/N_R$ è l'energia per simbolo ripartita su ciascuna antenna ricevente. Vediamo che con molte antenne in trasmissione si riesce a realizzare il massimo multiplexing gain, pari in questo caso a N_R , sacrificando il guadagno di array, che non è presente. L'effetto del fading di Rayleigh è però annullato: nel caso particolare $N_R = 1$, vediamo che la capacità ergodica del collegamento è quella di un canale AWGN con una potenza di segnale pari a quella totale ricevuta E_s .

Il risultato asintotico è molto semplice e interessante, ma ha una qualche rilevanza pratica? In realtà sì se, come già accennato, lo riferiamo al caso del downlink di un rate cellulare con una BTS dotata di molte antenne. E soprattutto perché il valore asintotico (10.123) non necessita di un valore particolarmente grande di N_T per essere praticamente raggiunto - $N_T \approx 10$ può già ritenersi sufficiente.

Avendo a disposizione molte antenna alla BTS, possiamo invertire il verso del collegamento e considerare l'*uplink*, dove adesso è N_R a tendere all'infinito. In questo caso,

$$c_E \to \log_2 \det\left(\left(1 + \frac{E_s/N_T}{N_0}\right) \mathbf{I}_{N_T}\right) = N_T \log_2\left(1 + \frac{N_R}{N_T} \frac{E_{s,r}}{N_0}\right) \quad \text{bit/c.u.} \quad (10.124)$$

Vediamo che di nuovo realizziamo il massimo guadagno di multiplazione, che è pari a N_T , e che inoltre cancelliamo l'effetto del fading di Rayleigh; in più abbiamo stavolta un guadagno di array considerevole, derivante dalle molte antenne di ricezione. Il guadagno è pari a N_R/N_T , cioè pari al numero di antenne riceventi per ogni antenna trasmittente, e si aggiunge al guadagno di multiplazione, facendo crescere la capacità con il $\log(N_R)$.

La tecnologia MMIMO diventa realizzabile per la banda delle *onde millimetriche* (f_0 maggiore di 30 GHz ovvero λ_0 minore di 1 cm), per la quale la lunghezza d'onda è

così piccola che anche gli elementi radianti dell'array sono di dimensioni ridotte, per cui realizzare un sistema d'antenna integrando decine o addirittura centinaia di elementi non è impossibile - per questo MMIMO è una delle tecnologie più promettenti per lo sviluppo della rete cellulare 5G.

Il Massive MIMO si declina immediatamente anche in versione multiutente. Immaginiamo una popolazione di U utenti nella cella, e concentriamoci sulla direzione uplink, supponendo che ogni utente abbia un terminale MIMO con N_T antenne (usualmente un numero inferiore a 4). Possiamo considerare tutti questi utenti come equivalenti a un unico utente *fittizio* con $U \cdot N_T$ antenne distribuite e una potenza totale trasmessa pari a $U \cdot P_T$, dove P_T è la potenza media *per utente*. Nel regime asintotico in cui $N_R \gg UN_T$, possiamo calcolare la capacità di questo link ad unico utente fittizio dalla (10.124), che risulterà pari alla capacità totale nella cella:

$$c_{MM} = N_T \cdot U \log_2 \left(1 + \frac{N_R}{N_T \cdot U} \frac{U \cdot E_{s,r}}{N_0} \right) = N_T \cdot U \log_2 \left(1 + \frac{N_R}{N_T} \frac{E_{s,r}}{N_0} \right) \quad \text{bit/c.u.}$$
(10.125)

e quindi la capacità per utente sarà

$$c_{MM} = N_T \log_2 \left(1 + \frac{N_R}{N_T} \frac{E_{s,r}}{N_0} \right) \quad \text{bit/c.u.}$$
(10.126)

esattamente come nel caso a singolo utente, dimostrando che con il MMIMO si ottiene massimo guadagno di multiplazione, guadagno di diversità, cancellazione del fading, e decorrelazione (ortogonalizzazione) degli utenti. Naturalmente il regime asintotico è più difficile da ottenere perchè deve essere $N_R \gg UN_T$ e il numero di antenne della BTS cresce molto.

Un'antenna MMIMO può esser realizzata nella forma di un array 2D lineare con elementi radianti disposti su di una griglia regolare, oppure un array circolare con elementi equispaziati su di una circonferenza, come negli esempi di Fig. 10.23. Con un array 2D si realizza direzionalità nel piano orizzontale e verticale, per servire ad esempio una popolazione di utenti disposti in modo molto denso anche in verticale (in un grattacielo) come rappresentato in Fig. 10.24.

Infine, osserviamo che l'effettiva capacità che si ottiene dal MMIMO è più piccola di quella prevista da (10.126). E non perché il regime asintotico non venga raggiunto, ma per il consueto difetto di modellazione della matrice di canale che *non* ha elementi indipendenti come nella formulazione teorica: $\mathbf{HH}^{\dagger} \nleftrightarrow N_T \mathbf{I}$ e la capacità è ridotta e di più complicata valutazione.

10.6 Modellistica ed emulazione del canale MIMO-OFDM

Il modello generico a banda larga nel dominio del tempo per un collegamento MIMO con N_T antenne trasmittenti ed N_R antenne riceventi di tipo omnidirezionale è complicato: come sappiamo, ciascuno degli N_R segnali ricevuti $r_i(t)$, $i = 1, ..., N_R$ è costituito dalla sovrapposizione di N_T componenti provenienti dagli N_T canali wireless relativi alla propagazione tra l'antenna TX#1, TX#2,...,TX# N_T verso la ricevente RX#i. Ognuno di questi N_T canali sarà caratterizzato dalla sua propria *risposta impulsiva* avente una struttura simile alla (7.6) - avremo in totale (ben) $N_T \cdot N_R$ risposte impulsive $h_{i,\ell}(t)$, una per ciascuna delle coppie TX# ℓ -RX#i, $\ell = 1, ..., N_T$, $i = 1, ..., N_R$ (vedi la Fig. 10.25 nel caso semplice $N_T = N_R = 2$).







Figura 10.24 3D-SDMA mediante MMIMO

(t)



Figura 10.25 Modello a larga banda di un collegamento MIMO 2×2

La relazione che lega i segnali trasmessi $s_{\ell}(t)$ e i segnali ricevuti $r_i(t)$ è in teoria semplice per quanto appena detto:

$$\begin{cases} r_1(t) = s_1(t) \otimes h_{1,1}(t) + s_2(t) \otimes h_{1,2}(t) + \dots + s_{N_T}(t) \otimes h_{1,N_T}(t) + w_1(t) \\ \vdots \\ r_{N_R}(t) = s_1(t) \otimes h_{N_R,1}(t) + s_2(t) \otimes h_{N_R,2}(t) + \dots + s_{N_T}(t) \otimes h_{N_R,N_T}(t) + w_{N_R}(t) \end{cases}$$
(10.127)

Le componenti di disturbo $w_i(t)$ sono processi AWGN complessi mutuamente indipendenti perché generati da sezioni radio del ricevitore fisicamente distinte (una per ciascuna antenna), aventi componenti I/Q (indipendenti) ciascuna con densità spettrale di potenza N_0 . Come però già sappiamo dal paragrafo 10.3.5, il modello si semplifica quando lo si applica a un segnale OFDM: la modellazione a banda larga (10.127) si riconduce a N modelli a banda stretta (10.94) (uno per ogni sottoportante) che operano di fatto del dominio della frequenza. Il coefficiente $h_{i,\ell}^{(k)}$ della matrice di canale $\mathbf{H}^{(k)}$ relativo alla sottoportante #k, k = 0, 1, ..., N - 1, è il valore $H_{i,\ell}(k/T_M)$ della risposta in frequenza del canale wireless che descrive la propagazione tra la coppia di antenne TX# ℓ -RX#i, $\ell = 1, ..., N_T$, $i = 1, ..., N_R$, valutata alla frequenza k/T_M della k-esima sottoportante. La modellazione dei singoli elementi di $\mathbf{H}^{(k)}$ si può dunque effettuare secondo quanto descritto nel paragrafo 8.4.4.

Fissata la sottoportante k, l'emulazione (simulazione) del coefficiente di canale $h_{i \ell}^{(k)} =$ $H_{i,\ell}(k/T_M)$ della matrice di canale MIMO **H** (d'ora in poi ometteremo l'indice k per brevità) è semplice: il valore è Gaussiano a media nulla e varianza assegnata e in prima approssimazione indipendente da tutti gli altri elementi della matrice - si tratta dunque di emulare $N_T \cdot N_R$ canali con le medesime statistiche, ma con parametri indipendenti.

D'altro canto, sappiamo che l'ipotesi di mutua indipendenza degli elementi di **H** non è completamente realistica. Serve quindi (anche) un modello che descriva le correlazioni tra i vari elementi $h_{i,\ell}$ in modo da rendere più verosimile l'emulazione.

Cominciamo con alcune premesse riguardo alle proprietà di vettori e matrici aleatorie a elementi complessi. Se \mathbf{g} è un vettore colonna aleatorio M-dimensionale a media nulla (cioè $E \{g\} = 0$) a componenti indipendenti, la sua matrice $M \times M$ di correlazione (covarianza) è

$$\mathbf{R}_g \stackrel{ riangle}{=} \mathrm{E}\left\{\mathbf{g}\mathbf{g}^{\dagger}\right\} = 2\sigma_g^2 \mathbf{I}$$

dove l'operatore [†] indica il trasposto coniugato (Hermitiano) di un vettore o matrice, dove σ_a^2

è la varianza di ciascuna delle due componenti reale/immaginaria mutuamente indipendenti del generico elemento complesso g_i di **g** e dove **I** indica la matrice identica. Applicando a **g** una trasformazione lineare con matrice $M \times M$ **A**, otteniamo il vettore **h** = **Ag** la cui matrice di covarianza è

$$\mathbf{R}_{h} = \mathbf{E}\left\{\mathbf{h}\mathbf{h}^{\dagger}\right\} = \mathbf{E}\left\{\mathbf{A}\mathbf{g}\mathbf{g}^{\dagger}\mathbf{A}^{\dagger}\right\} = \mathbf{A}\mathbf{E}\left\{\mathbf{g}\mathbf{g}^{\dagger}\right\}\mathbf{A}^{\dagger} = \mathbf{A}\mathbf{R}_{g}\mathbf{A}^{\dagger} \stackrel{\triangle}{=} 2\sigma_{g}^{2}\boldsymbol{\Psi}$$

dove $\Psi = \mathbf{A}\mathbf{A}^{\dagger}$ gode di simmetria Hermitiana ($\Psi = \Psi^{\dagger}$) e quindi possiede autovalori reali. Data la particolare forma di Ψ , useremo anche la notazione abbastanza evidente

$$\mathbf{A} = \sqrt{\mathbf{\Psi}} \Rightarrow \mathbf{\Psi} = \sqrt{\mathbf{\Psi}} \sqrt{\mathbf{\Psi}}^{\top}$$

Immaginiamo adesso di voler creare un vettore aleatorio Gaussiano complesso a media nulla **h** avente matrice di covarianza (correlazione) *assegnata* \mathbf{R}_h a partire da un vettore Gaussiano "bianco" **g**, facilmente generabile per simulazione. Da quanto appena visto, **h** si ottiene banalmente dalla trasformazione lineare $\mathbf{h} = \sqrt{\mathbf{R}_h \mathbf{g}}$. Resta da capire, assegnata \mathbf{R}_h , come se ne calcola la "radice quadrata" $\sqrt{\mathbf{R}_h}$. Tenendo conto che una matrice di correlazione è per definizione Hermitiana e semidefinita positiva, cioè presenta autovalori reali non negativi, essa può essere scomposta come segue:

$$\mathbf{R}_h = \mathbf{U}\mathbf{D}\mathbf{U}^\dagger \tag{10.128}$$

dove U è la matrice degli autovettori normalizzati di \mathbf{R}_h , e D ne è la matrice diagonale degli autovalori, $\mathbf{D} = \text{diag} \{\lambda_1, \lambda_2, ..., \lambda_M\}$. La normalizzazione degli autovettori garantisce che U è unitaria, cioè $\mathbf{U}\mathbf{U}^{\dagger} = \mathbf{I}$. Trovata la decomposizione ai valori singolari (10.128) di \mathbf{R}_h , il calcolo di $\sqrt{\mathbf{R}_h}$ è immediato:

 $\sqrt{\mathbf{R}_h} = \mathbf{U}\sqrt{\mathbf{D}}\mathbf{U}^\dagger \tag{10.129}$

dove evidentemente $\sqrt{\mathbf{D}} = \text{diag}\left\{\sqrt{\lambda_1}, \sqrt{\lambda_2}, ..., \sqrt{\lambda_N}\right\}$

Risolto il problema di generare *vettori* (Gaussiani) aventi matrice di correlazione data, veniamo adesso alla matrice di canale **H** di dimensione, in generale, $N_R \times N_T$. Come imporre una mutua correlazione assegnata tra gli elementi della *matrice*? L'idea è quella di *vettorizzare* la matrice **H** creando un vettore **h** di dimensione $M = N_R N_T$ e specificare le correlazioni attraverso una matrice \mathbf{R}_h di dimensione $M \times M$. Il vettore **h** si ottiene "impilando" le N_T colonne \mathbf{h}_i di **H** una sopra l'altra:

$$\mathbf{H}_{N_R \times N_T} = [\mathbf{h}_1 \, \mathbf{h}_2 \dots \mathbf{h}_{N_T}] \Rightarrow \mathbf{h}_{M = N_R \cdot N_T \times 1} \stackrel{\triangle}{=} \operatorname{vec}(\mathbf{H}) = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \vdots \\ \mathbf{h}_{N_T} \end{bmatrix}$$
(10.130)

e nella simulazione viene ottenuto per trasformazione da un vettore bianco **g** M-dimensionale attraverso la matrice $\sqrt{\mathbf{R}_h}$ di dimensione $M \times M$ come sopra:

$$\mathbf{h} = \sqrt{\mathbf{R}_h} \, \mathbf{g} \quad \Rightarrow \quad \mathbf{H} = \text{unvec}(\mathbf{h}) \tag{10.131}$$

dove gli operatori vec e unvec riferiscono rispettivamente alle operazioni di vettorizzazione diretta e inversa di una matrice. È chiaro che per implementare (10.131) è necessario calcolare la SVD della matrice \mathbf{R}_h data.



Figura 10.26 Correlazione "locale" al Ricevitore MIMO

Resta infine il problema principale di *modellizzazione*: che caratteristiche ha la matrice \mathbf{R}_h nei casi pratici di propagazione MIMO? Il problema non è semplice, perchè anche in casi MIMO elementari come 4 × 2, la matrice \mathbf{R}_h è abbastanza grande (8 × 8) e coinvolge la conoscenza o misura di parecchi parametri (i 64 elementi).

Immaginiamo che il nostro collegamento 4×2 sia come in Fig. 10.26, in cui le antenne di ricezione sono molto vicine l'un l'altra, mentre le 2 antenne di trasmissione sono molto distanti. Estremizzando la situazione, si intuisce che la matrice di canale è del tipo

$$\mathbf{H} = egin{bmatrix} h_1 & h_2 \ h_1 & h_2 \ h_1 & h_2 \ h_1 & h_2 \ h_1 & h_2 \end{bmatrix} riangleq [\mathbf{h}_{col,1} \; \mathbf{h}_{col,2}]$$

Infatti ogni antenna di ricezione "vede" l'antenna TX#1 nella stessa maniera, per cui $h_{i,1} = h_1 \forall i$; con lo stesso criterio, $h_{i,2} = h_2 \forall i$ con $h_1 \in h_2$ (molto) diversi tra loro: le antenne di ricezione sono molto *correlate*, mentre quelle di trasmissione sono sostanzialmente scorrelate. In generale non sarà esattamente vero che $h_{i,1} = h_1 \forall i$, ma gli $h_{i,1}$ saranno comunque correlati - nel nostro caso potremo dire che gli elementi di **H** sono correlati entro le colonne, ma le colonne tra di loro sono incorrelate. Definiamo allora la matrice $N_R \times N_R$ di correlazione *di ricezione*

$$\mathbf{R}_{rx} \stackrel{\triangle}{=} \mathbf{E} \left\{ \mathbf{H} \mathbf{H}^{\dagger} \right\} = \sum_{i=1}^{N_T} \mathbf{E} \left\{ \mathbf{h}_{col,i} \mathbf{h}_{col,i}^{\dagger} \right\}$$
(10.132)

Reciprocamente, possiamo immaginare di avere un collegamento con due antenne TX molto ravvicinate e 4 antenne di ricezione molto spaziate tra di loro; in questo caso, la matrice avrà l'aspetto

$$\mathbf{H} = \begin{bmatrix} h_1 & h_1 \\ h_2 & h_2 \\ h_3 & h_3 \\ h_4 & h_4 \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{h}_{row,1} \\ \mathbf{h}_{row,2} \\ \mathbf{h}_{row,3} \\ \mathbf{h}_{row,4} \end{bmatrix}$$

cioè stavolta ogni antenna di ricezione vede le due di trasmissione in modo identico (molto correlato), ma diverso (incorrelato) dall'una all'altra. Le correlazioni si caratterizzano



Figura 10.27 Modello alla Kronecker di canale MIMO

stavolta attraverso la matrice $N_T \times N_T$ di correlazione di trasmissione

$$\mathbf{R}_{tx} \stackrel{\triangle}{=} \mathbf{E} \left\{ \mathbf{H}^{\dagger} \mathbf{H} \right\} = \sum_{i=1}^{N_R} \mathbf{E} \left\{ \mathbf{h}_{row,i}^{\dagger} \mathbf{h}_{row,i} \right\}$$
(10.133)

Questa idea di "località" della correlazione si può generalizzare sulla base della considerazione (suffragata dalla Fig. 10.27 per una rete cellulari) che nel collegamento MIMO wireless la correlazione tra gli elementi della matrice di canale è scomponibile in una componente locale *di trasmissione* derivante dall'ambiente di propagazione in prossimità del trasmettitore, e in una componente locale *di ricezione* derivante dall'ambiente di propagazione in prossimità del ricevitore - i due fenomeni non interagiscono tra di loro per la (più) grande distanza tra il gruppo di antenne di trasmissione e il gruppo di antenne di ricezione (Fig. 10.27).

Se si accetta questo modello, la matrice $N_R N_T \times N_R N_T$ di correlazione totale \mathbf{R}_H si ottiene facilmente dal *prodotto di Kronecker* tra \mathbf{R}_{tx} e \mathbf{R}_{rx} :

$$\mathbf{R}_{H} = \mathbf{R}_{tx} \otimes \mathbf{R}_{rx} \stackrel{\triangle}{=} \begin{bmatrix} r_{tx,1,1} \mathbf{R}_{rx} & \cdots & r_{tx,1,N_{T}} \mathbf{R}_{rx} \\ \vdots & \ddots & \vdots \\ r_{tx,N_{T},1} \mathbf{R}_{rx} & \cdots & r_{tx,N_{T},N_{T}} \mathbf{R}_{rx} \end{bmatrix}$$
(10.134)

I parametri da identificare per la matrice scomposta secondo Kronecker sono $N_T \times N_T + N_R \times N_R$ (in pratica, le matrici "locali" \mathbf{R}_{tx} e \mathbf{R}_{rx}), assai meno degli $N_T N_R \times N_T N_R$ necessari in generale per la matrice completa \mathbf{R}_H .

Il risultato finale e fondamentale, che non dimostriamo, è il seguente, che deriva dalle proprietà del prodotto di Kronecker e della SVD: se la matrice di correlazione di canale \mathbf{R}_H ha struttura di Kronecker (10.134), allora non è necessario calcolare la $\sqrt{\mathbf{R}_h}$ per implementare la (10.131) e ottenere poi dopo unvec la matrice di canale **H**; quest'ultima si può ottenere direttamente dalla trasformazione (filtraggio *spaziale*)

$$\mathbf{H} = \sqrt{\mathbf{R}_{rx} \, \mathbf{G} \, \sqrt{\mathbf{R}_{tx}}} \tag{10.135}$$

dove $\mathbf{G} = \text{unvec}(\mathbf{g})$ è una matrice $N_R \times N_T$ a componenti Gaussiane indipendenti, e dove il calcolo delle matrici radici quadrate si effettua (solo) sulle matrici di correlazione locale,

che vengono anche chiamate matrici di correlazione destra e sinistra.

Esempio 10.59

Il gruppo di standardizzazione www.3gpp.org che emana e mantiene le normative per le reti cellulari 4G (tecnologia radio LTE, long-Term Evolution) e 5G (tecnologia radio NR, New Radio) indica esplicitamente una forma tipica della matrice di correlazione locale alla Base Transceiver Station (BTS) di rete o al terminale utente (UE, User Terminal). Quando si hanno 2 o 4 antenne la forma è la seguente:

$$\mathbf{R}_{2} = \begin{bmatrix} 1 & \alpha \\ \alpha^{*} & 1 \end{bmatrix} , \ \mathbf{R}_{4} = \begin{bmatrix} 1 & \alpha^{1/9} & \alpha^{4/9} & \alpha \\ (\alpha^{1/9})^{*} & 1 & \alpha^{1/9} & \alpha^{4/9} \\ (\alpha^{4/9})^{*} & (\alpha^{1/9})^{*} & 1 & \alpha^{1/9} \\ \alpha^{*} & (\alpha^{4/9})^{*} & (\alpha^{1/9})^{*} & 1 \end{bmatrix}$$
(10.136)

Il valore di α viene specificato per indicare maggiore o minor grado di correlazione; in particolare si consiglia l'utilizzo del valore 0.3 per bassa correlazione e 0.9 per alta correlazione, a seconda dello scenario che si intende emulare. Si consigliano anche in generale valori di correlazione *maggiori* per lo UE (antenne più vicine tra loro) che per la BTS.

Se ad esempio abbiamo un MIMO 2×2 con correlazione debole alla BTS e correlazione forte all'UE, nell'uplink avremo

$$\mathbf{R}_{tx} = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix} , \ \mathbf{R}_{rx} = \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix} ,$$
(10.137)

per cui

$$\mathbf{R}_{H} = \mathbf{R}_{tx} \otimes \mathbf{R}_{rx} = \begin{bmatrix} \mathbf{R}_{rx} & 0.9 \, \mathbf{R}_{rx} \\ 0.9 \, \mathbf{R}_{rx} & \mathbf{R}_{rx} \end{bmatrix} = \begin{bmatrix} 1 & 0.3 & 0.9 & 0.27 \\ 0.3 & 1 & 0.27 & 0.9 \\ 0.9 & 0.27 & 1 & 0.3 \\ 0.27 & 0.9 & 0.3 & 1 \end{bmatrix}$$
(10.138)

Una matrice di canale con queste correlazioni si può generare attraverso la (10.135) utilizzando le matrici radice quadrata locali, ottenibili facilmente dalla (10.137) usando la (10.129):

$$\sqrt{\mathbf{R}_{tx}} = \begin{bmatrix} 0.8473163 & 0.5310885\\ 0.5310885 & 0.8473163 \end{bmatrix}, \ \sqrt{\mathbf{R}_{rx}} = \begin{bmatrix} 0.1517577 & 0.9884177\\ 0.9884177 & 0.1517577 \end{bmatrix}$$
(10.139)



CAPITOLO 11

SOTTILI COME UN CAPELLO, GRANDI COME IL MONDO: LE FIBRE OTTICHE



La terra e le acque del nostro pianeta sono attraversate da centinaia di cavi per telecomunicazioni ad altissima capacità che connettono ad Internet tutto il mondo. Sono i manufatti più "grandi" costruiti dall'umanità perché coprono senza interruzione decine di migliaia di chilometri, ma allo stesso tempo sono costituiti da fili esili come un capello: le fibre ottiche.

-Cavo in fibra ottica SEA-ME-WE-3 (2000)

Comunicazioni Digitali, I Edizione. di M. Luise Copyright © 2023 University of Pisa I cavi in fibra ottica costituiscono la spina dorsale (e sempre più anche le ramificazioni) dell'infrastruttura mondiale cablata per le telecomunicazioni: Internet così come oggi lo conosciamo non sarebbe semplicemente possibile senza le tecnologie delle comunicazioni ottiche e in particolare senza i cavi in fibra, che permettono di trasportare enormi moli di dati su lunghissime distanze. Dunque la realizzazione di un collegamento digitale affidabile e ad altissima capacità (Tbit/s...) richiede di conoscere le tecnologie fondamentali delle comunicazioni in fibra ottica - il tema di questo capitolo.

11.1 Il come e il perché dei Collegamenti in Fibra Ottica

11.1.1 Comunicazioni Radio e Comunicazioni Ottiche

In ogni link per comunicazioni digitali, su cavo o wireless, e con qualunque tecnologia, le risorse di maggiore importanza sono la *banda* e la *potenza* del segnale necessarie per coprire efficacemente il collegamento. Senza entrare in grandi dettagli, sappiamo che per formati binari (cioè non multilivello), un flusso di informazione ad R_b bit/s necessita di una banda *B* dell'ordine di $R_b \div 2R_b$ Hz che il mezzo portante (cioè il mezzo fisico che permette la comunicazione) deve essere in grado di garantire. Inoltre, per effetto dei disturbi che accompagnano la trasmissione e la ricezione del segnale (distorsione, rumore elettronico, interferenze ecc.), ogni ricevitore è caratterizzato da una certa *sensibilità*, cioè da una potenza minima del segnale ricevuto che garantisca il raggiungimento di determinate specifiche di *qualità del servizio* - tipicamente un certo rapporto segnale-rumore, un fissato tasso d'errore sul bit, una determinata probabilità di perdita di pacchetti di bit o grandezze similari.

Mentre la banda supportata vincola la massima capacità del collegamento (vedi anche le formule di Shannon), la sensibilità del ricevitore pone un vincolo alla (minima) potenza trasmessa, date le caratteristiche di *attenuazione* del segnale da parte del portante utilizzato. Il dato di sensibilità può infatti essere riformulato come una limitazione sulla *distanza* massima L che il collegamento è in grado di coprire, assegnata la potenza massima che il trasmettitore è in grado di erogare (vuoi per limiti fisici, vuoi per vincoli relativi a un formato o standard di trasmissione).

Le due risorse citate possono essere sfruttate al meglio con un opportuno progetto dei formati di comunicazione: l'efficienza spettrale di una trasmissione può essere migliorata ricorrendo a modulazioni multilivello, mentre l'efficienza energetica può aumentare usando opportuni codici a protezione d'errore. Tuttavia, ogni mezzo portante è intrinsecamente caratterizzato da limitazioni fondamentali di carattere fisico sulla banda e/o sull'attenuazione che non possono essere oltrepassate.

Perché dovremmo utilizzare segnali *ottici* per trasportare l'informazione digitale? La Fig. 11.1 rappresenta la quasi totalità dello *spettro* della radiazione elettromagnetica. Notiamo che al di là delle onde centimetriche caratteristiche dei sistemi a *microonde* (satelliti, cellulari, WI-Fi ecc.) attualmente limitati a frequenze inferiori ai 30 GHz (λ_0 =1 cm), abbiamo le cosiddette *onde millimetriche* (λ_0 da 10 a 1 mm) usate nel campo del rilevamento ambientale e che si stanno per la prima volta affermando con le tecnologie cellulari 5G, e quindi la radiazione del cosiddetto *infrarosso lontano* (λ_0 circa compresa tra 1000 e 10 μ m) e *infrarosso vicino* (da 10 a 0.76 μ m) ove l'aggettivo si riferisce alla minore o maggiore prossimità alla zona compresa tra gli 0.38 e gli 0.76 μ m della radiazione *visibile*. La zona dell'infrarosso vicino è quella di interesse per le comunicazioni ottiche perché i cavi in fibra presentano in questa zona la massima trasparenza al segnale (cioè la

IL COME E IL PERCHÉ DEI COLLEGAMENTI IN FIBRA OTTICA 411



minima attenuazione). Dal punto di vista della distanza coperta dal collegamento, la fibra

Figura 11.1 Spettro della radiazione elettromagnetica.

allo stato attuale dell'arte, garantisce (come preciseremo in seguito) un'attenuazione del segnale di un fattore pari a

 $\alpha_{fibra} \cong 10^{0.02L}$ (*L* misurata in km) (11.1)

mentre l'attenuazione di un segnale radio alla frequenza $f_0 = 10 \text{ GHz}$ utilizzata dai satelliti geostazionari per televisione digitale (corrispondente a una lunghezza d'onda $\lambda_0 = c/f_0 = 0.03 \text{ m}$, dove $c = 3 \cdot 10^8 \text{ m/s}$ è la velocità di propagazione dell'onda elettromagnetica nel vuoto) è pari a

$$\alpha_{radio} = (4\pi L/\lambda_0)^2 \cong 1.75 \cdot 10^9 L^2 \qquad (L \text{ misurata in km}) \tag{11.2}$$

che risulta più favorevole di quella della fibra già ad esempio per L = 800 km.

Per quanto riguarda la banda però, osserviamo che ogni tecnologia di comunicazione di tipo passabanda a una certa frequenza portante f_0 è vincolata per motivi tecnologici a trattare segnali aventi banda pari al più a qualche percento di f_0 . La banda relativa $\beta \triangleq B/f_0$ dei segnali passa-banda portatori di informazione è cioè limitata in generale a qualche percento; poniamo per fissare le idee $\beta = 0.01$. Risulta quindi che la banda di un sistema radio con $f_0 = 10 \text{ GHz} (\lambda_0 = 3 \text{ cm})$ è dell'ordine dei 100 MHz (100 Mbit/s), valore indicativo dei transponder per televisione digitale via satellite (che sfruttano in realtà 30 MHz).

La superiorità delle comunicazioni ottiche adesso è chiara: le comunicazioni su fibra sono caratterizzati da lunghezze d'onda attorno a $\lambda_0 = 1.55 \,\mu\text{m}$, corrispondente a una frequenza portante pari a 194 THz. La banda relativa disponibile per trasferire informazione è quindi dell'ordine di qualche THz, *quattro ordini di grandezza* maggiore di quella caratteristica

412 SOTTILI COME UN CAPELLO, GRANDI COME IL MONDO: LE FIBRE OTTICHE

del satellite TV. I cavi in fibra di fatto costituiscono la spina dorsale (backbone) di Internet che non sarebbe così come lo conosciamo oggi senza questa tecnologia. Nella Fig. 11.2 mostriamo lo stato attuale (2021) dei cavi in fibra che trasportano il traffico Internet a grandi distanze. Un esempio di questa tecnologia neanche troppo recente, ma che ha fatto storia, è costituito dal cavo transcontinentale FLAG (Fiber-optics Loop Around the Globe), entrato in servizio nel 1997 che si estende ininterrottamente per 27,000 km dalla Gran Bretagna al Giappone collegando 12 paesi a 10 Gb/s full-duplex su due coppie di fibre a 5 Gb/s per coppia. Nell'introduzione a questo capitolo abbiamo già menzionato l'ancora più recente SEA-ME-WE-3: 39,000 km di collegamento, ad oggi (2021) il più lungo in esercizio.



Figura 11.2 Mappa dei cavi in fibra terrestri (rossi) e sottomarini (grigi) che trasportano il traffico Internet a lunga distanza (da https://www.itu.int/itu-d/tnd-map-public/).

11.1.2 Schema-base di un link su fibra ottica

Il raggiungimento delle attuali grandi capacità dei sistemi su fibra è stato reso possibile (principalmente) da due miglioramenti tecnologici fondamentali: la messa a punto di sorgenti ottiche a semiconduttore di dimensioni ridotte, alta efficienza e alta affidabilità, cioè i LASER (light amplification by stimulated emission of radiation) a semiconduttore (anni '80 del secolo scorso), e l'affinamento delle tecniche di produzione industriale delle fibre in vetro, che ha reso disponibili le fibre ad alta qualità usate oggigiorno per i collegamenti ad alta capacità (anni '70 del secolo scorso).

Lo schema rappresentato in Figura 11.3 rappresenta genericamente un collegamento tra una sorgente e un utente di informazione digitale collegati attraverso un cavo in fibra e relativi modem. Nella totalità dei casi pratici, l'informazione digitale che deve essere trasmessa è associata a un segnale *elettrico*; il trasmettitore può essere visto quindi sotto un duplice aspetto: esso può essere infatti considerato un *modulatore* che imprime l'informazione data su di una portante alle frequenze ottiche, oppure può essere considerato come una sorta di *trasduttore* che trasferisce l'informazione da un segnale in ambito *elettrico* a un segnale in ambito *ottico* (così come il trasmettitore del telegrafo di Morse trasferiva l'informazione da



un ambito meccanico a un ambito elettrico più adatto alla trasmissione).

Figura 11.3 Trasmissione dell'informazione su fibra.

Lo stesso duplice aspetto riguarda evidentemente anche il ricevitore, che svolge la funzione duale al trasmettitore, recuperando un segnale *elettrico* cui è associata l'informazione ricevuta, a partire dal segnale *ottico* all'uscita della fibra. La necessità delle conversioni elettroottica e optoelettronica rappresenta un "collo di bottglia" allo sfruttamento della enorme banda della fibra. Allo stato attuale (2021) della tecnologia infatti, è impensabile realizzare trasmissioni numeriche monocanale a velocità maggiori di circa 100 Gb/s, perché i componenti optoelettronici interfacciati alla fibra non consentono di trattare segnali con una velocità maggiore. La banda complessiva di decine di THz messa a disposizione dalla fibra deve quindi essere sfruttata in altra maniera, ad esempio attraverso più tramissioni contemporanee di più flussi dati su diverse lunghezze d'onda (WDM, wavelength division multiplexing), tecnologia sulla quale torneremo in seguito.



Figura 11.4 Schema generale di un link in fibra ottica.

414 SOTTILI COME UN CAPELLO, GRANDI COME IL MONDO: LE FIBRE OTTICHE

Lo schema semplificato di un tipico collegamento su fibra (a modulazione d'intensità e rivelazione diretta) è mostrato in Figura 11.4. La modulazione dati è impressa con la tecnica cosiddetta *on-off keying* (OOK) ovvero "a tutto o niente": la sorgente *ottica*, nel nostro caso un diodo LASER a semiconduttore, viene accesa alla massima potenza durante il periodo di bit T oppure completamente spenta variando la corrente di pilotaggio del dispositivo proporzionalmente a un segnale *elettrico* ON-OFF digitale, a seconda che il k-esimo bit di informazione b[k] assuma rispettivamente il valore logico "1" o il valore logico "0".

Il segnale modulato prodotto dal LASER a una certa lunghezza d'onda λ_0 viene quindi opportunamente inviato (accoppiato) alla fibra ottica - possiamo immaginare l'onda emessa dal LASER come un'onda piana con polarizzazione lineare oscillante alla frequenza $f_0 = c/\lambda_0$ di emissione del dispositivo. Stante la modulazione dati OOK, la potenza ottica $P_T(t)$ entrante in fibra all'estremità di trasmissione è

$$P_T(t) = P_0 \sum_{n = -\infty}^{+\infty} b[k] p(t - kT)$$
(11.3)

ove p(t) è il consueto impulso NRZ e P_0 è la potenza di picco utile, cioè quella effettivamente trasferita dal LASER alla fibra. Il segnale *ottico* viene raccolto all'estremità terminale della fibra da un dispositivo *fotorivelatore* (ancora un diodo a semiconduttore) che fornisce in uscita una corrente *elettrica* I(t) proporzionale all'intensità del segnale luminoso ricevuto attraverso un coefficiente R detto *responsivity* del dispositivo.

Il segnale all'uscita del fotorivelatore viene trattato da un filtro (*integrate and dump*) (2.58) per eliminare i disturbi introdotti nel processo di fotorivelazione, e viene infine rigenerato dal circuito latch (soglia) finale. Si ottiene in ricezione una replica del segnale digitale di pilotaggio del LASER di trasmissione, contenente l'informazione prodotta dalla sorgente. Poiché non viene effettuato alcun tipo di elaborazione del segnale ottico all'uscita dalla fibra prima della fotorivelazione, il ricevitore appena descritto è chiamato *a rivelazione diretta* e costituisce lo schema usato nella quasi totalità dei modem in fibra attualmente in esercizio.

Resta da capire il meccanismo secondo il quale il segnale ottico del LASER si propaga, cioè viene *guidato*, dalla fibra ottica per chilometri di percorso nel collegamento dal modem di trasmissione a quello di ricezione.

11.1.3 Che cos'é una fibra ottica?

11.1.3.1 Un po' di fisica: Riflessione e rifrazione Una fibra ottica è sostanzialmente un cavo sottilissimo di vetro ultra-puro e quindi ultra-trasparente. Il cavo funziona da "guida d'onda" cioè confina l'onda luminosa al suo interno e ne permette la propagazione dall'estremità di lancio a quella di arrivo. Questo fenomeno di guida è basato su di una variazione delle proprieà ottiche tra due mezzi all'interno dei quali la luce si propaga, variazione che provoca fenomeni di riflessione e/o rifrazione dell'onda. L'esempio più semplice di fibra ottica si trova nella lampada da tavolo *vintage* anni '70 della Fig. 11.5 in cui la luce entra dall'estermita in basso dei vari cilindri "snelli" di fibra di vetro ed esce solo dall'altra estremità (i puntini luminosi), senza disperdersi dalle pareti laterali. La variazione delle proprietà ottiche, e in particolare dell'*indice di rifrazione*, si verifica all'interfaccia tra il vetro e il mezzo fisico (aria) che circonda il cavo stesso.

Prima di approfondire lo studio delle fibre, premettiamo dunque alcuni richiami di carattere generale sui fenomeni di rifrazione/riflessione delle onde luminose, in un contesto semplificato di ottica geometrica. L'*indice di rifrazione n* di un mezzo trasparente alla luce

IL COME E IL PERCHÉ DEI COLLEGAMENTI IN FIBRA OTTICA 415



Figura 11.5 Lampada da tavolo a fibre ottiche anni '70

è il rapporto che intercorre tra la velocità di propagazione del raggio luminoso nel vuoto c e la corrispondente velocità di propagazione nel mezzo v:

$$a \triangleq c/v$$

(11.4)

che risulta un numero maggiore di 1 e costante in tutti i punti dello spazio all'interno di un mezzo omogeneo e isotropo. Quando un raggio luminoso si trova ad attraversare una superficie di interfaccia tra due mezzi fisici diversi "vede" una brusca variazione dell'indice di rifrazione e si ha la situazione schematizzata in Figura 11.6: il raggio incidente *a* proveniente dal mezzo numero 1 (avente indice di rifrazione n_1) viene parzialmente *riflesso* come raggio *a'* che resta nel mezzo 1 e viene parzialmente *rifratto* (trasmesso) come raggio *b* nel mezzo 2 (che ha indice di rifrazione n_2). Se φ_1 è l'angolo formato con la normale alla superficie di incidenza dal raggio a ($0 \le \varphi_1 \le \pi/2$), il raggio riflesso *a'* forma con la normale lo stesso angolo φ_1 , mentre il raggio rifratto *b* forma con la stessa retta un angolo φ_2 tale che

$$n_1 \cdot \sin \varphi_1 = n_2 \cdot \sin \varphi_2 \tag{11.5}$$

(*legge della rifrazione* o *di Snell*). Supponendo, come nell'esempio in Figura 11.6, che si abbia $n_2 < n_1$, è chiaro che il raggio rifratto tende ad "allontanarsi" dalla normale, poiché, dovendo essere $n_1/n_2 = \sin \varphi_2 / \sin \varphi_1$, sarà $\sin \varphi_2 > \sin \varphi_1$, cioè $\varphi_2 > \varphi_1$.

Se l'angolo di incidenza del raggio *a* tende ad aumentare, si arriva a una condizione per la quale dovrebbe aversi $\varphi_2 = \pi/2$; in pratica, questo significa che il raggio rifratto non si produce. Si ha in tal caso a che fare con il fenomeno della *riflessione totale*; l'angolo di incidenza φ_c al di là del quale si ha riflessione totale è dunque quello per cui $\varphi_2 = \pi/2$, ovvero sin $\varphi_2 = 1$:

$$\varphi_c \triangleq \arcsin\left(n_2/n_1\right)$$
 (11.6)

Tale angolo viene chiamato *angolo critico*. Si capisce il principio di funzionamento come guida ottica di una fibra: i raggi in fibra che incidono sull'interfaccia vetro/aria con un angolo maggiore di quello critico φ_c (dove evidentemente la fibra è il mezzo 1 e l'aria il mezzo 2), vengono riflessi totalmente e restano confinati all'interno della fibra stessa, così

416 SOTTILI COME UN CAPELLO, GRANDI COME IL MONDO: LE FIBRE OTTICHE





come schematizzato in Figura 11.7 per un raggio incidente che giace nel piano dell'asse della fibra (raggio *meridionale*): la riflessione totale si ripete multiple volte fino a che il raggio luminoso giunge all'estremità finale della fibra. Il vero e proprio cavo in fibra ad



Figura 11.7 Riflessione totale all'interno di una fibra ottica.

alta capacità aggrega *decine* di fibre singole (vedi l'esempio ad alta densità di fibre di Fig. 11.8), viene adeguatamente rinforzato (anche con cavetti d'acciaio se del caso) per dare sufficiente robustezza meccanica durante la posa, e viene rivestito con rivestimenti multipli in materiali plastici.

11.1.4 Fibre Multi-Mode Step-Index e Graded-Index

11.1.4.1 La fibra step-index Molti tipi di fibra usati in pratica nei sistemi di telecomunicazione non si discostano dal principio di funzionamento descritto nel paragrafo precedente. Poiché però è desiderabile poter controllare con precisione il valore di *entrambi* gli indici di rifrazione dei due mezzi coinvolti nel fenomeno della riflessione totale, le fibre effettivamente impiegate sono costituite da una zona cilindrica interna detta *nucleo* (nei paesi anglosassoni, *core*), che corrisponde al materiale 1 nella discussione precedente, circondato da una zona esterna a mo' di "guscio" cilindrico detto *mantello* (*cladding*), come mostrato in Figura 11.9. Sia il nucleo che il mantello sono costituiti dallo stesso vetro ultra-puro (SiO₂); l'indice di rifrazione n_1 del nucleo e quello del mantello n_2 possono però essere



Figura 11.8 Esempio di cavo multifibra ad alta capacità (da http://www.prysmiancable.co.nz).

variati e controllati con precisione aggiungendo quantità determinate di materiali estrinseci (ossidi di Ge, Pb o Al) durante la fabbricazione della fibra stessa.

Il diametro esterno tipico del mantello di una fibra per telecomunicazioni è pari a $125 \,\mu\text{m}$ (all'incirca quello di un capello), mentre il diametro del nucleo varia a seconda del tipo di fibra da qualche μ m fino a $50 \,\mu\text{m}$. Ovviamente la fibra viene protetta e irrobustita con ulteriori rivestimenti di materiali plastici. Per applicazioni particolari (ad esempio collegamenti corti in una rete locale o in ambiente industriale), vengono anche fabbricate fibre in materiale plastico con diametro maggiore di quello sopra indicato con caratteristiche di propagazione inferiore.



Figura 11.9 Fibra step-index.

Prendiamo dunque in considerazione la fibra di Figura 11.9, caratterizzata da una discontinuità *a gradino* dell'indice di rifrazione tra nucleo e mantello, e con nucleo largo: un tipo abbastanza diffuso è la 50/125 (cioè diametro del mantello 125 μm e del nucleo 50 μm) ed esaminiamo secondo i principi dell'ottica geometrica la propagazione di un raggio meridionale, cioè giacente su di un piano passante per l'asse della fibra stessa¹.

¹Ricordiamo che l'*ottica geometrica* può considerarsi valida fin quando le dimensioni caratteristiche della fibra (nel nostro caso, il raggio del nucleo) sono molto maggiori della lunghezza d'onda del segnale luminoso. Nei

418 SOTTILI COME UN CAPELLO, GRANDI COME IL MONDO: LE FIBRE OTTICHE

Consideriamo un raggio proveniente dalla sorgente di segnale e tale che, una volta all'interno del nucleo stesso, si trovi a incidere sull'interfaccia nucleo/mantello con un angolo minore o uguale all'angolo critico φ_c . Tale raggio viene parzialmente rifratto nel mantello e poi perduto per ulteriore rifrazione verso il mezzo circostante; la restante porzione riflessa subisce una successiva riflessione/rifrazione con ulteriore perdita, e così via *ad infinitum*. In pratica, il raggio originale si esaurisce dopo poche riflessioni interne successive, non viene cioè *guidato* all'interno della fibra e quindi non contribuisce all'intensità luminosa del segnale uscente dall'estremità terminale della fibra: in una parola, non viene *accettato* dalla fibra stessa.

Si definisce allora un *cono di accettazione* della fibra contenente tutti quei raggi che riescono a propagarsi per riflessione totale nel nucleo. Il vertice di questo cono giace su di un diametro della sezione del nucleo, e l'angolo al vertice θ_a (vedi Figura 11.10) è chiamato *angolo di accettazione* della fibra. Osservando la Figura 11.10, è facile mettere in relazione θ_a con n_1 ed n_2 . Consideriamo infatti un raggio che subisce riflessione interna totale, cioè avente $\varphi_1 > \varphi_c$; l'angolo θ_1 (complementare di φ_1) è l'angolo sotto cui viene rifratto un raggio meridionale entrato in fibra che forma con l'asse della medesima l'angolo θ , tale che

$$\sin\theta = n_1 \cdot \sin\theta_1 / n_0 \tag{11.7}$$

dove n_0 è l'indice di rifrazione del mezzo esterno alla fibra (aria). Essendo $n_0 \cong 1$ e $\theta_1 = \pi/2 - \varphi_1$, si ha

$$\sin \theta = n_1 \cdot \cos \varphi_1 \tag{11.8}$$

(11.9)

Poiché deve essere $\varphi_1 \ge \varphi_c$, cioè $\cos \varphi_1 \ge \cos \varphi_c$, segue

$$\sin\theta \le n_1 \cdot \cos\varphi_c = n_1 \sqrt{1 - \sin^2\varphi_c} = \sqrt{n_1^2 - n_2^2}$$

che definisce immediatamente l'angolo di accettazione

$$\theta_a \triangleq \arcsin\left(\sqrt{n_1^2 - n_2^2}\right) \tag{11.10}$$

I raggi che si presentano alla bocca della fibra con un angolo minore di θ_a rispetto



Figura 11.10 Angolo (cono) di accettazione della fibra.

sistemi per telecomunicazioni, la lunghezza d'onda della luce è dell'ordine del μ m, cosicché le conclusioni ricavate con l'ottica geometrica sono valide solo per fibre a nucleo largo, cioè con diametro maggiore della decina di μ m.
all'asse possono propagarsi all'interno del nucleo, gli altri vengono rifratti nel mantello e si esauriscono rapidamente.

L'angolo di accettazione è una delle specifiche fondamentali di ogni fibra. Molto più spesso anziché specificare θ_a , viene fornita la cosiddetta *apertura numerica* (Numerical Aperture) NA della fibra

$$NA \triangleq \sin \theta_a = \sqrt{n_1^2 - n_2^2} \tag{11.11}$$

Chiaramente, quanto maggiore è NA, tanto diventa efficiente è l'accoppiamento della sorgente ottica alla fibra, perché più ampio è il cono di accettazione dei raggi. Con i valori tipici di n_1 ed n_2 delle fibre per telecomunicazioni

$$n_1 = 1.50, \quad n_2 = 1.47 \tag{11.12}$$

otteniamo un'apertura numerica $NA \cong 0.3$ e un angolo di accettazione $\theta_a \cong 17^{\circ}$.

Un parametro la cui conoscenza identifica univocamente i valori di n_1 ed n_2 una volta nota l'apertura, è la variazione percentuale dell'indice di rifrazione:

$$\Delta_n \triangleq \frac{n_1 - n_2}{n_1} = 1 - n_2/n_1 \tag{11.13}$$

che, nell'esempio di cui sopra, vale 0.02 (2%). Poiché normalmente $\Delta_n \ll 1$ (cioè $n_1 \simeq n_2$), si ha

$$NA = \sqrt{(n_1 + n_2)(n_1 - n_2)} \cong \sqrt{2n_1(n_1 - n_2)} = \sqrt{2n_1^2 \frac{n_1 - n_2}{n_1}} = n_1 \sqrt{2\Delta_n}$$
(11.14)

Il tipo di fibra preso finora in considerazione presenta una variazione brusca dell'indice di rifrazione tra il nucleo ed il mantello. Per questo tale fibra è chiamata *step-index*: la variazione dell'indice di rifrazione n all'interno della fibra al variare della distanza radiale, si svolge "a gradino" tra i valori $n_1 e n_2$, ed è mostrata in Figura 11.11. La step-index con nucleo largo (ad esempio la 50/125 già menzionata) è il tipo più semplice di fibra, ma presenta un notevole inconveniente che viene preso in considerazione nel prossimo paragrafo.



Figura 11.11 Variazione dell'indice di rifrazione in una fibra step-index.

11.1.4.2 La dispersione intermodale nelle fibre SI Consideriamo i due raggi meridionali entranti in fibra con l'angolo di incidenza minimo ($\theta_0 = 0$, raggio parallelo all'asse della fibra) e massimo ($\theta_0 = \theta_a$) indicati in Figura 11.12. I due raggi viaggiano all'interno

del nucleo con la stessa velocità di propagazione $v = c/n_1$ ma coprono una stessa distanza L misurata lungo l'asse della fibra seguendo due percorsi che hanno lunghezza totale diversa. In particolare, il raggio 1 percorre una traiettoria di lunghezza $d_1 = L$, mentre la traiettoria del raggio 2 è lunga $d_2 = L/\sin \varphi_c$. Se i due raggi sono entrati in fibra allo stesso istante, giungono al punto a distanza L sulla fibra rispettivamente nei due istanti

$$t_1 = \frac{L}{v}, \quad t_2 = \frac{L/\sin\varphi_c}{v} \tag{11.15}$$

per cui l'intervallo intercorrente tra i due istanti è

$$\Delta t = t_2 - t_1 = \frac{L}{v} \left[\frac{1}{\sin \varphi_c} - 1 \right] = \frac{Ln_1}{c} \left[\frac{n_1}{n_2} - 1 \right] = \frac{Ln_1^2}{cn_2} \Delta_n \simeq \frac{Ln_1}{c} \Delta_n \quad (11.16)$$



Figura 11.12 Dispersione intermodale in una fibra step-index.

Questo fenomeno di ritardo temporale relativo tra diversi raggi prende il nome di *dispersione intermodale*. I vari cammini percorsi dai raggi possono essere considerati come i molti *modi di propagazione* dell'onda luminosa in fibra. Ad ognuno di questi modi può essere associata una velocità netta di propagazione lungo l'asse della fibra (velocità *assiale*) pari a $v_g = (c/n_1) \cdot \sin \varphi$ che dipende dall'angolo φ di incidenza all'interfaccia nucleo/mantello, e quindi dalla natura del modo. D'altronde, il segnale emesso da una qualunque sorgente luminosa accoppiata alla fibra sarà composto da raggi che si propagheranno in fibra secondo angoli diversi, con una distribuzione della potenza ottica totale caratteristica del tipo di sorgente stessa. In ogni caso, i vari raggi si propagheranno con velocità assiali diverse a seconda dell'angolo di incidenza, e giungeranno all'estremità terminale della fibra con ritardi reciproci variabili, dando luogo al fenomeno di dispersione sopra citato. Le fibre step a nucleo largo sono quindi dette *multimodo* e sono caratterizzate dal fenomeno della dispersione intermodale, che dobbiamo approfondire.

Immaginiamo di lanciare in fibra un impulso rettangolare di durata T e potenza ottica P_0 che rappresenta un bit di informazione (Fig. 11.13). Tale potenza P_0 si distribuisce tra tutti i possibili modi supportati dalla fibra, per cui la potenza ottica dell'impulso in uscita alla fibra avrà un andamento *diverso* da quello rettangolare. Infatti i raggi con il minimo angolo di

incidenza esterno (quelli cioè con $\theta = 0$) in corrispondenza al fronte di salita dell'impulso giungeranno a distanza L all'istante t_1 , mentre i raggi "più lenti" (quelli cioè con $\theta = \theta_a$) giungeranno a distanza L all'istante t_2 . L'impulso ricevuto avrà quindi un certo tempo di salita diverso da 0 (e di durata proprio pari a $\Delta t = t_2 - t_1$) come indicato in Fig. 11.13. Se questo impulso rappresenta il simbolo "1" in un collegamento digitale, come indicato in Figura 11.13, esso sarà seguito da altri impulsi a distanza T, 2T, ecc. relativi ai simboli successivi. Quando Δt è confrontabile con T, il generico impulso tende ad "allargarsi" e ad "invadere" gli intervalli di segnalazione adiacenti (vedi Figura 11.13). Nasce quindi *interferenza intersimbolica* (ISI, inter-symbol interference) per effetto della dispersione, e le prestazioni del sistema di trasmissione possono essere sensibilmente degradate.



Figura 11.13 Interferenza intersimbolica dovuta a dispersione intermodale.

Per questo motivo la dispersione intermodale impone un limite massimo alla velocità di trasmissione dell'informazione su una fibra multimodo step-index: in prima battuta possiamo immaginare che l'estensione massima dell'ISI Δt (la durata massima della "coda" dei bit) che non dà luogo a fuori servizio (che non causa cioà una BER del collegamento troppo alta) sia pari al tempo di bit T. Di conseguenza, affinché il link sia operativo si deve avere

$$\Delta t \le T \quad \to \quad \frac{Ln_1}{c} \Delta_n \le T \tag{11.17}$$

Indicando con $R_b = 1/T$ la velocità di trasmissione, si trova la limitazione

$$R_b \le \frac{c}{\Delta_n L n_1} \tag{11.18}$$

Si ottiene una diversa versione di questa relazione se si considerano i *due* parametri di merito del collegamento in fibra, e cioè bit-rate R_b e distanza L, e se si introduce il parametro di capacità $\delta = R_b \cdot L$:

$$\delta \le \frac{c}{\Delta_n n_1} \tag{11.19}$$

Con i valori (11.12), si trova $\delta \leq 10 \,\text{Mbit/s} \cdot \text{km}$ che indica una capacità assai modesta, insufficiente per qualunque esigenza moderna, sia essa di trasporto o di accesso.

È interessante notare che la capacità della fibra aumenta al diminuire di Δ_n . Infatti, riducendo Δ_n riduciamo l'apertura numerica NA della fibra e il suo angolo di accettazione, cosicché la diversità tra i modi che si propagano in fibra si riduce, riducendo il massimo ritardo relativo di propagazione e quindi la dispersione intermodale.

11.1.5 Le fibre a indice graduato (graded-index)

La dispersione intermodale è un fattore molto limitante delle fibre multimodo step-index (MM-SI). Per ovviare a questo inconveniente, si possono fabbricare fibre di differente tipo, in cui l'indice di rifrazione n_1 viene variato gradatamente all'interno del nucleo tra un valore massimo n_1 e il valore del mantello n_2 , man mano che ci si sposta dal centro della fibra verso il mantello. Si ottengono così le fibre multimodo graded-index (MM-GI, ad indice graduato) in cui i raggi non subiscono una riflessione brusca all'interfaccia nucleo/mantello, ma si "incurvano" seguendo la variazione graduale dell'indice di rifrazione (come chiariremo più avanti). La legge di variazione di n con la distanza radiale dal centro del nucleo ρ è il cosiddetto profilo α :

$$n(\rho) = \begin{cases} n_1 \sqrt{1 - 2\Delta_n (\rho/a)^{\alpha}} & \rho \le a \quad \text{(nucleo)} \\ n_2 & \rho > a \quad \text{(mantello)} \end{cases}$$
(11.20)

dove a è il raggio del nucleo, ed α è un parametro che viene fissato in fase di lavorazione (vedremo successivamente che in pratica α =2). L'andamento dell'indice di rifrazione per la fibra a profilo α è mostrato in Figura 11.14. Per capire intuitivamente perché i raggi vengono "incurvati", si deve considerare che la variazione teoricamente continua (11.20) di n_1 viene in realtà realizzata durante la manifattura della fibra depositando strati successivi di vetro con indici di rifrazione via via decrescenti a partire dal centro della fibra stessa. Il nucleo si presenta quindi come in Fig. 11.15 con strati di vetro aventi una piccola variazione Δn di n_1 . Il raggio di luce subisce quindi molteplici micro-rifrazioni "locali" ad ogni interfaccia da strato a strato, che su di una scala più grande possono essere considerate come un curvatura di tipo continuo.



Figura 11.14 Indice di rifrazione in una fibra graded-index.

Tornando alla modellistica più fine che prevede una variazione continua di n_1 con la distanza dal centro della fibra, la traiettorie di propagazione (raggi) possono essere ricavate mediante il *principio di Fermat* dell'ottica geometrica, secondo il quale il percorso scelto dal raggio per propagarsi dal punto P_1 di partenza fino al punto P_2 di arrivo è quello che minimizza il *tempo totale di percorrenza*. Il tempo dt necessario per percorrere un tratto di lunghezza elementare ds relativo al punto generico $\mathbf{r} = (x, y, z)^T$ della traiettoria, in cui l'indice di rifrazione vale n (\mathbf{r}) è pari a dt = ds/v = n (\mathbf{r}) ds/c; la traiettoria è allora tale da minimizzare il valore dell'*integrale curvilineo* $\int_{P_1}^{P_2} n$ (\mathbf{r}) ds che è proporzionale al tempo di propagazione totale - questo è il motivo per cui il percorso dei raggi è *rettilineo* in un mezzo otticamente omogeneo. Il principio di Fermat può essere riformulato in forma



Figura 11.15 Spiegazione semplice della curvatura dei raggi di luce nel nucleo graded-index.

differenziale (equazione di Eulero-Lagrange)

$$\frac{d}{ds}\left(n\left(\mathbf{r}\right)\frac{d\mathbf{r}}{ds}\right) = \nabla n\left(\mathbf{r}\right) \tag{11.21}$$

che, adottando un sistema di riferimento come in Figura 11.16, può essere semplificata come segue per ricavare l'equazione cartesiana y(z) del raggio:

$$\frac{d^2y}{dz^2} = \frac{1}{n}\frac{dn}{dy} \tag{11.22}$$

dove *n* è funzione di $\rho = |y|$ secondo il profilo α (11.20). Sostituendo tale relazione nella (11.22), nell'ipotesi $\Delta_n \ll 1$ (verificata nella pratica) e $\alpha = 2$ (profilo parabolico), si trova l'equazione di un *oscillatore armonico* (rispetto a *z*) la cui soluzione è

$$y(z) = y_0 \cos\left(\sqrt{2\Delta_n} \frac{z}{a}\right) + \frac{y'_0}{\sqrt{2\Delta_n}/a} \sin\left(\sqrt{2\Delta_n} \frac{z}{a}\right)$$
(11.23)

dove $y_0 e y'_0$ sono la posizione e la direzione iniziale del raggio.

Due raggi che partono dalla stessa posizione, ma con direzioni iniziali differenti si propagano seguendo traiettorie diverse, che seguono un andamento sinusoidale (anziché a zig-zag come nelle fibre SI) con diverse ampiezze (vedi la Figura 11.16). La dispersione intermodale viene però ridotta perché, per effetto della graduazione dell'indice di rifrazione, i raggi che "si allontanano" di più dal centro della fibra, seguendo quindi traiettorie più lunghe, si trovano a transitare in zone relativamente vicine al mantello in cui l'indice di rifrazione è *più piccolo* che al centro; la velocità di propagazione in tali zone risulta *maggiore*, compensando quindi l'allungamento del percorso ed "equalizzando" la risposta della fibra.

Nonostante ciò, un certo grado di dispersione residua è comunque presente perché l'equalizzazione dei ritardi non è perfetta. Si può dimostrare che le fibre a profilo α presentano il ritardo differenziale minimo pari a

$$\Delta t = \frac{n_1 \Delta_n^2}{8c} L \tag{11.24}$$

quando si sceglie $\alpha = 2(1 - \Delta_n) \simeq 2$, cioè il profilo parabolico come già accennato. La dispersione minima sopra indicata comporta un limite per la capacità δ della fibra MM-GI





pari a

L'incremento di capacità rispetto alla fibra MM-SI è pari a un fattore $8/\Delta_n$; con i valori (11.12), otteniamo $\delta = 4 \text{ Gb/s} \cdot \text{ km}$ che rende a tutt'oggi le fibre graded-index ancora utilizzate per applicazioni non impegnative, ad esempio le reti su area locale o metropolitana, in cui le tratte sono inferiori al km di lunghezza.

 $\delta \le \frac{8c}{n_1 \Delta_n^2}$

(11.25)

11.1.6 Le fibre Single-Mode (SM)

Una maniera per risolvere il problema della dispersione intermodale è quello di *inibire la* propagazione di modi multipli in fibra, lasciando un solo modo fondamentale; con ciò un impulso lanciato in fibra non può disperdersi perché tutti i "raggi" costituenti tale segnale sono obbligati a propagarsi secondo un'unica modalità. Sfortunatamente, la condizione di monomodalità per una fibra non può essere ricavata attraverso la semplice analisi basata sull'ottica geometrica sviluppata per la fibra multimodo nei paragrafi precedenti. Come vedremo in seguito, una fibra SI a modo singolo richiede una configurazione a nucleo stretto (pochi μ m); la dimensione caratteristica (diametro) del nucleo diventa così confrontabile con la lunghezza d'onda del segnale (1.55 μ m), ed è giocoforza ricorrere all'ottica elettromagnetica fondata sulle equazioni di Maxwell per analizzare la propagazione della luce.

In generale, il campo elettrico $\mathbf{E}(\mathbf{r};t)$ quasi-monocromatico (cioè con una eventuale modulazione dati passa-banda) in un certo punto dello spazio \mathbf{r} può essere espresso come

$$\mathbf{E}(\mathbf{r};t) = \Re \left[\tilde{\mathbf{E}}(\mathbf{r};t) \exp\left(j2\pi f_0 t\right) \right]$$
(11.26)

dove la funzione vettoriale complessa $\tilde{\mathbf{E}}(\mathbf{r};t)$ è l'*equivalente in banda base* rispetto alla frequenza f_0 del campo $\mathbf{E}(\mathbf{r};t)$ e come sempre $\Re[\cdot]$ indica la parte reale del numero complesso cui si riferisce - nel caso *strettamente* monocromatico (cioè LASER in modalità *Continuous Wave* (CW), ovvero non modulato) $\tilde{\mathbf{E}}(\mathbf{r};t) = \tilde{\mathbf{E}}(\mathbf{r})$ e parliamo di *fasore* dell'onda. L'analisi della propagazione di un'onda elettromagnetica nella fibra SI parte dalla considerazione delle equazioni di d'Alembert (delle onde)

$$\nabla^2 \mathbf{E} = \mu \varepsilon \frac{\partial^2 \mathbf{E}}{\partial t^2}, \quad \nabla^2 \mathbf{H} = \mu \varepsilon \frac{\partial^2 \mathbf{H}}{\partial t^2}$$
(11.27)

dove ε e μ sono rispettivamente la costante dielettrica e la permeabilità magnetica del mezzo, e **H** rappresenta il campo magnetico. La velocità di propagazione dell'onda nel mezzo risulta pari a $v = 1/\sqrt{\mu\varepsilon} \le 1/\sqrt{\mu_0\varepsilon_0} = c$.

Per un campo monocromatico in un materiale dielettrico omogeneo senza perdite, l'equazione di d'Alembert si semplifica nella cosiddetta *equazione di Helmoltz*:

$$\nabla^2 \tilde{\psi} \left(\mathbf{r} \right) + n^2 k_0^2 \tilde{\psi} \left(\mathbf{r} \right) = 0 \tag{11.28}$$

dove n = c/v è l'indice di rifrazione del mezzo (il vetro) e ψ è l'inviluppo complesso (fasore) di una qualunque delle sei componenti dei vettori complessi $\tilde{\mathbf{E}}$ ed $\tilde{\mathbf{H}}$ (e dove il tempo non compare perché parliamo appunto di fasori). Nella (11.28) $k_0 \triangleq 2\pi/\lambda_0 = 2\pi f_0/c$ è il *numero d'onda nel vuoto* dell'oscillazione del LASER inviata in fibra. Per la fibra SI, le equazioni vanno risolte all'interno del nucleo con indice di rifrazione n_1 e diametro 2a e anche nel mantello con indice di rifrazione n_2 e diametro esterno illimitato (cioè $\gg 2a$). Le due soluzioni vanno poi "raccordate" all'interfaccia nucleo-mantello con opportune condizioni di continuità delle componenti del campo EM.

La soluzione è piuttosto complicata e va al di là dello scopo di questo testo. Il risultato però è abbastanza semplice: quello che si scopre è che l'onda luminosa, per effetto della variazione dell'indice di rifrazione tra i due strati, resta quasi perfettamente "confinata" (cioè viene effettivamente guidata) all'interno del nucleo, e si può propagare secondo diversi modi, ciascuno caratterizzato da una diversa velocità di propagazione. Questi modi sono sempre e comunque in numero finito; se la fibra è a nucleo largo, come la MM-SI che già abbiamo analizzato con l'ottica geometrica, i modi sono in numero elevatissimo e le varie velocità di propagazione (lungo l'asse della fibra) sono molto vicine tra di loro. In pratica, si (ri-)ottengono i familiari modi dell'ottica geometrica, che si distribuiscono con continuità, e che hanno una velocità assiale di propagazione $(c/n_1) \cdot \sin(\varphi_1)$ parimenti variabile con continuità nell'intervallo $[(c/n_1) \sin(\varphi_c); c/n_1]$. Se però il nucleo è relativamente stretto nei confronti della lunghezza d'onda, i modi sono ben distinti l'uno dall'altro e assumono velocità di propagazione abbastanza differenti tra loro.

Per capire quanti modi differenti la fibra fa propagare definiamo la quantità

$$V \triangleq 2\pi a \cdot NA/\lambda_0 \tag{11.29}$$

che chiamiamo "parametro V" o anche "frequenza normalizzata" della fibra. Il numero di modi M supportati dalla fibra in funzione del parametro V è mostrato in Figura 11.17. Notiamo che per V < 2.405 sono indicati (soltanto) *due* modi di propagazione (uno ciascuno per le due diverse componenti di polarizzazione lineare H e V dell'onda) che indicano in realtà la presenza di una unica modalità di propagazione caratterizzata da una unica velocità ! Il numero "magico" 2.405 è un valore particolare derivante dalla soluzione dell'equazione di Helmoltz.

La condizione di mono-modalità sul parametro V che abbiamo appena ricavato si può ottenere creando fibre a nucleo (molto) stretto relativamente al diametro del mantello che resta fissato a 125 μm : con i consueti valori $n_1 = 1.50$ e $n_2 = 1.47$, per avere V < 2.405 si deve avere $a < 1.98 \ \mu m$ cioè un nucleo di 4 μm @ $\lambda_0 = 1.55 \mu m$. Di fatto, la stragrande maggioranza delle fibre usate nelle backbone e nella rete di accesso (ad esempio in architetture di rete come la PON, Passive Optical Network per la fornitura di connettività



Figura 11.17 Numero di modi supportati da una fibra step-index a guida debole ($\Delta_n \ll 1$).

a banda ultra larga fino agli utenti domestici) utilizza fibre SM-SI, cioè Single-Mode Step-Index o varie varianti tecnologiche di questa.

Le fibre MM-GI, come già detto, sono relegate ad applicazioni di rete locale o industriale dove la grande capacità e/o la grande distanza non sono necessarie. Naturalmente, anche le fibre SM-SI hanno una limitazione nella capacità, che non proviene però dalla dispersione intermodale, evidentemente annullata, e che dobbiamo esaminare.

11.1.7 Limiti di capacità del link: Dispersione Intramodale e Attenuazione della fibra SM

11.1.7.1 Ancora fisica: la dispersione intramodale nelle fibre SM-SI Le fibre SM risolvono la questione della dispersione intermodale, che costituisce la principale limitazione alla capacità delle fibre MM, anche in versione GI. Qual è dunque il limite di capacità delle SM? Da che cosa è determinato?

Il vetro è un materiale *dispersivo*, per il quale cioè l'indice di rifrazione dipende dalla lunghezza d'onda della oscillazione luminosa cui è sottoposto: $n = n(\lambda)$ ovvero, considerando le frequenza, n = n(f). Il fenomeno fisico che dimostra quest'affermazione sta in bella mostra sulla copertina di uno dei più venduti LP/CD di tutti i tempi: *The Dark Side of the Moon* del gruppo *Pink Floyd* (Fig. 11.18). Lasciamo a chi legge la spiegazione, basata ancora sulla legge di Snell e sul fatto che la luce bianca contiene tutti colori dello spettro.

Dunque, segnali a diverse lunghezze d'onda (colori) si propagano in un mezzo dispersivo con velocità diverse (o vengono rifratti all'interfaccia tra due mezzi diversi secondo angoli diversi...). D'altro canto, un segnale dati digitale con una certa occupazione di banda è costituito dalla sovrapposizione di una moltitudine di componenti spettrali (oscillazioni) a diverse frequenze (lunghezze d'onda), in un intervallo centrato attorno alla frequenza portante f_0 (ovvero attorno alla lunghezza d'onda portante λ_0) - in ottica e in meccanica ondulatoria, un segnale di questo tipo viene chiamato *pacchetto d'onda*.

In un pacchetto d'onda (impulso di bit...), la generica componente alla frequenza f si



Figura 11.18 The Dark Side of the Moon: dispersione cromatica della luce bianca in un prisma di vetro.

propaga nel vetro con una costante di propagazione

$$\beta(f) = \frac{2\pi n(f)}{\lambda} = \frac{1}{c} 2\pi f \cdot n(f) \tag{11.30}$$

alla quale corrisponde la velocità di propagazione, chiamata velocità di gruppo v_g definita da

$$\frac{1}{v_g(f)} \triangleq \frac{1}{2\pi} \frac{d\beta(f)}{df} = \frac{1}{c} \frac{d\left[f \cdot n\left(f\right)\right]}{df} = \frac{n\left(f\right)}{c} + \frac{f}{c} \frac{dn\left(f\right)}{df}$$
(11.31)

e quindi un tempo di propagazione su di una lunghezza totale L della fibra pari a

$$t_p(f) = \frac{L}{v_g(f)} \tag{11.32}$$

evidentemente variabile al variare di f. Le componenti, avendo tempi di propagazione diversi, giunte all'estremità di "uscita" della fibra presentano sfasamenti non semplicemente proporzionali alla frequenza (come si avrebbe con un ritardo identico per tutte le componenti): il segnale subisce *distorsione di fase*, dando luogo ad una dispersione temporale analoga a quella delle fibre MM, ma evidentemente causata da un fenomeno ben diverso.

Abbiamo, per riassumere, descritto il fenomeno della *dispersione cromatica*, causata cioè dalla diversità della lunghezza d'onda (colore) delle varie componenti del pacchetto d'onda (impulso dati). Qual è l'effetto della dispersione cromatica sulla qualità del collegamento? Non è dissimile da quello analizzato in precedenza per la dispersione intermodale: un impulso di durata T lanciato in fibra si "allarga" di una quantità Δt che provoca ISI e potenzialmente manda in fuori servizio il collegamento. È quindi (di nuovo) fondamentale valutare quantitativamente l'estensione dell'ISI, cioè ricavare Δt dai parametri della fibra e del collegamento.

Dato che la dispersione dipende dalla diversità di lunghezza d'onda delle componenti spettrali del segnale (pacchetto d'onda), si dovrà ammettere che la dispersione temporale (ISI) Δt risulti una funzione crescente di tale diversità, cioè in parole povere della *banda* del

segnale $\Delta\lambda$ valutata nel dominio della lunghezza d'onda λ . Come si passa dalla tradizionale banda del segnale Δf misurata sull'asse delle frequenze e quindi in Hz (kHz, MHz, GHz) alla banda misurata in m (μ m, nm) sull'asse delle lunghezza d'onda? Se, come sempre accade $\Delta f \ll f_0$, allora vale la relazione

$$\frac{\Delta f}{f_0} = \frac{\Delta \lambda}{\lambda_0} \tag{11.33}$$

cioè la banda relativa è la medesima.

Esempio 11.60

Valutiamo la banda in lunghezza d'onda di un collegamento ottico a velocità $R_b=100$ Gbit/s alla lunghezza d'onda standard $\lambda_0=1.55 \ \mu m$.

Alla velocità di 100 Gbit/s, la banda occupata (in banda passante...) attorno alla frequenza portante è pari a $B = \Delta f = 2R_b=200$ GHz. Considerando che $f_0 = c/\lambda_0=194$ THz $\gg \Delta f = 200GHz$, possiamo trovare subito la banda in lunghezza d'onda come

$$\Delta \lambda = \lambda_0 \frac{\Delta f}{f_0} = \frac{\lambda_0^2}{c} \Delta f = 1.6 \ nm \tag{11.34}$$

Da questo esempio si vede che l'unità di misura opportuna per le bande ottiche è il nm piuttosto che il μ m.

Ma torniamo al calcolo della dispersione. Come dicevamo, $\Delta t = \Theta(\Delta \lambda)$ dove $\Theta(\cdot)$ è una funzione al momento sconosciuta della banda. Poiché la dispersività del vetro è un effetto relativamente debole, approssimiamo $\Theta(\cdot)$ al primo ordine, cioè immaginiamo una diretta proporzionalità tra $\Delta \lambda$ e Δt :

$$\Delta t \cong \frac{dt_p}{d\lambda} \Delta \lambda = \frac{d\left(L/v_g\right)}{d\lambda} \Delta \lambda \tag{11.35}$$

ovvero, considerando il valore assoluto del ritardo relativo,

$$|\Delta t| \cong L \left| \frac{d \left(1/v_g \right)}{d\lambda} \right| \Delta \lambda = L \left| D \right| \Delta \lambda \tag{11.36}$$

dove $D \triangleq d(1/v_q) / d\lambda$ è il coefficiente di dispersione intramodale della fibra.

Il coefficiente di dispersione si chiama *intramodale* perchè la dispersione avviene entro l'unico modo di propagazione ed è un parametro di capitale importanza per le fibre SM. Esso viene fornito nell'unità di misura $ps/(nm \cdot km)$ e indica l'aumento (misurato in ps) della durata di un impulso caratterizzato da una certa larghezza spettrale $\Delta\lambda$ (misurata in nm) e che ha viaggiato in fibra per la lunghezza di 1 km. Per la fibra standard monomodo ITU-T G.652, *D* è pari a 17 ps/(nm \cdot km) alla lunghezza d'onda di 1.55 $\lambda_0 = 1.55 \,\mu$ m. Usando il valore 17 per *D*, esprimendo la lunghezza del collegamento *L* in km e la banda $\Delta\lambda$ in nm, otteniamo la dispersione temporale Δt (la lunghezza della coda di ISI dell'impulso) in ps.

Possiamo visualizzare l'effetto della dispersione intramodale tenendo conto che si tratta di un fenomeno *lineare* - possiamo cioè considerare la modificazione delle componenti del campo EM dall'ingresso all'uscita della fibra come dovuta al passaggio attraverso un *sistema lineare stazionario*. Allora possiamo cercare una relazione tra gli equivalenti in banda base di tali forme d'onda (quelli cioè sui quali effettuiamo una modulazione digitale) attraverso la risposta impulsiva o piuttosto la risposta in frequenza $H_F(f)$ (equivalente in banda base) dell'SLS che modella la fibra dispersiva. Nell'Appendice F mostriamo che

$$H_F(f) = \exp\left[-\jmath 2\pi f \frac{L}{v_g}\right] \exp\left[\jmath \frac{\pi D \lambda_0^2 L}{c} f^2\right]$$
(11.37)

evidenziando quindi che la fibra introduce una *distorsione di fase* a causa della dispersione intramodale. Il primo termine nella risposta in frequenza è soltanto il ritardo di propagazione sulla distanza L con la velocità v_g e *non* è distorcente. Il secondo termine, contenente appunto il coefficiente D è responsabile della distorsione selettiva in frequenza.

Esempio 11.61

Analizziamo la propagazione in una fibra G-652 @ $\lambda_0 = 1.55 \ \mu m$ di un impulso Gaussiano che rappresenta un bit di informazione di un generico flusso al bit-rate $R_b=10$ Gbit/s (quindi T=100 ps) - non utilizziamo la forma NRZ perché nelle trasmissioni ad alta velocità su fibra la forma d'onda del bit è sempre notevolmente limitata in banda e si discosta dall'NRZ ideale; usiamo quindi un impulso Gaussiano che, sebbene non rigorosamente limitato in banda, ha comunque componenti ad alta frequenza molto piccole e modella bene la situazione reale.

L'impulso viene lanciato con ampiezza unitaria e con un parametro di durata tale per cui nella forma d'onda Gaussiana si ha $6\sigma = T$, ed è rappresentato nella Fig. 11.19 - si nota che è perfettamente confinato nell'intervallo [-T/2, T/2]. Questa figura mostra anche (e soprattutto) l'*allargamento* per dispersione dell'impulso di bit dopo una propagazione di 10, 20, 50 km. Gli impulsi provocano ISI più o meno grande; ad esempio, per L = 20 km l'allargamento dell'impulso è $\Delta t = T$ al limite della tollerabilità. Nella figura, il termine di puro ritardo di propagazione L/v_g della (11.37) è stato compensato (cancellato).



Figura 11.19 Allargamento dell'impulso di bit per effetto della dispersione intramodale.

L'ISI dovuta alla dispersione intramodale crea evidentemente una limitazione al bitrate supportato dalla fibra. Se infatti imponiamo la consueta condizione $\Delta t \leq T$ (e cioè $R_b \leq 1/\Delta t$), tenendo conto della varie unità di misura in gioco otteniamo

$$R_b \le \frac{1}{L|_{km} |D| \Delta \lambda|_{nm}} \quad \text{(Tbit/s)} \tag{11.38}$$

dove D ha la consueta unità di misura. Utilizzando la (11.33) e tenendo conto che $\Delta f = 2R_b$, otteniamo infine

$$R_b \le \frac{1}{\lambda_0|_{\mu m}} \sqrt{\frac{c}{2000 \ L|_{km} |D|}} \quad \text{(Gbit/s)}$$
(11.39)

Questa relazione è rappresentata in Fig. 11.20 per $\lambda_0 = 1.55 \ \mu m$ e per la fibra G.652 con D = 17 e anche con $D = 2 \text{ ps/(nm \cdot km)}$.



Figura 11.20 Limitazione al bit-rate su fibra dovuta a dispersione intramodale.

Come si può ottenere un valore di D diverso dai fatidici 17 ps/(nm·km)? La Figura 11.21 mostra che il valore del coefficiente D è (fortemente) dipendente dalla lunghezza d'onda. Finora abbiamo sempre menzionato, senza particolare giustificazione la λ_0 =1.55 μm (la cosiddetta *III finestra*) come quella operativa, ma dalla figura si nota che esiste un'altra lunghezza d'onda per la fibra G.652, per la quale il coefficiente D è nullo: la λ_0 =1.3 μm chiamata λ_{ZD} o di zero dispersione (la cosiddetta *II finestra*). In realtà, un piccolissimo grado di dispersione è presente anche alla λ_{ZD} , dato che comunque lo spettro del segnale trasmesso ha un'estensione apprezzabile attorno a tale lunghezza d'onda, e quindi le varie componenti che vedranno valori diversi da 0 di D saranno in una qualche misura disperse, ma è chiaro che l'effetto sarà di un ordine di grandezza inferiore². La Fig. 11.21 giustifica anche perché il valore di D appare nella (11.36) in valore assoluto, fatto mai menzionato

²Per calcolare la dispersione nella banda attorno alla λ_{ZD} si introduce il *coefficiente di dispersione differenziale* $S \triangleq dD/d\lambda$ che vale per la G.652 0.092 ps/(nm²·km)

ne' giustificato finora: semplicemente perché il coefficiente può essere sia negativo che positivo - in entrambi i casi si ha comunque allargamento (broadening) netto dell'impulso di bit dovuto alla distorsione (dispersione) di un fattore pari al Δt dato dalla formula.



Figura 11.21 Coefficiente di dispersione intramodale per fibre SM-SI.

11.1.7.2 *Tipi di fibre SM non standard* La Fig. 11.21 suggerisce che per migliorare il comportamento della fibra basta cambiare la lunghezza d'onda di funzionamento del LASER - cosa che normalmente non viene fatta per il motivo che chiariremo nel prossimo paragrafo. Una maniera alternativa è quella di cambiare di nuovo tipo di fibra rispetto alla semplice SI. Nei paragrafi precedenti abbiamo indifferentemente parlato di dispersione cromatica o dispersione intramodale, me le due non sono in realtà esattamente la stessa cosa. La dispersione cromatica viene, come ampiamente discusso, da una proprietà fisica del vetro ed è una delle due componenti della dispersione intramodale totale. Nella dispersione totale vi è infatti una seconda componente oltre la cromatica, chiamata dispersione di guida d'onda (waveguide dispersion) che si avrebbe anche se fabbricassimo fibre con fantomatico vetro extraterrestre non dispersivo. Infatti, al variare della lunghezza d'onda λ delle componenti spettrali del segnale dati, la velocità di propagazione di tali componenti varia perchè varia il rapporto λ/a : ogni lunghezza d'onda "vede" un nucleo della fibra di dimensione relativa a se stessa diverso, e quindi l'effetto guida dell'onda (la velocità di propagazione) varia in ragione di questa diversità, anche se più debolmente rispetto alla variazione cromatica.

Per riassumere,

$$D = D_M + D_W \tag{11.40}$$

dove D_M è l'indice di dispersione dovuto alla sola dispersione cromatica del materiale, mentre D_W è la dispersione di guida d'onda. La scomposizione di D in tali contributi è mostrata in Figura 11.22, dove si nota che D_W tende a far spostare λ_{ZD} verso un valore maggiore di quello che competerebbe per effetto del solo termine di dispersione cromatica D_M . Intervenendo opportunamente sulla geometria della fibra attraverso a e Δ_n , e soprattutto sulla graduazione dell'indice di rifrazione nel nucleo è possibile ottenere fibre in cui il termine D_W è incrementato rispetto a quello di una fibra standard mostrato in

Figura 11.22, in modo da ottenere le *fibre a dispersione traslata* (o *dispersion-shifted*, *DS*) ITU G.653, la cui λ_{ZD} viene a spostarsi in III finestra ($\lambda = 1.55 \,\mu\text{m}$). Le fibre dispersion-shifted si ottengono imponendo un profilo di graduazione lineare (triangolare) dell'indice di rifrazione nel nucleo tra i valori n_1 ed n_2 . Per applicazioni particolari, specialmente



Figura 11.22 Coefficienti di dispersione intramodale totale, cromatica e di guida d'onda per fibre SM-SI.

collegamenti WDM, sono particolarmente indicate le fibre Non-Zero Dispersion Shifted NZ-DS ITU G.655 in cui il punto di zero dispersione viene imposto a una lunghezza d'onda leggermente inferiore a 1.55 μ m. In questo modo, un'ampia zona spettrale attorno a 1.55 presenta un coefficiente di dispersione piccolo e positivo, ma *non zero*, ad esempio 2 ps/(nm·km) come nella Fig. 11.20. Tale condizione favorisce la propagazione dei segnali multi-frequenza, multi-canale WDM ad altissimo bit-rate (vedi il paragrafo 11.3).

11.1.7.3 Caratteristiche di attenuazione delle fibre SM-SI. Le nostre considerazioni sulla propagazione del segnale ottico in fibra, sia nel caso semplificato dell'ottica geometrica, sia nel caso più approfondito dell'ottica elettromagnetica, non hanno minimamente considerato l'inevitabile *attenuazione* subita dal segnale stesso. Indicando con P(z) la potenza ottica totale su di una sezione trasversale del nucleo della fibra ad una distanza z dall'estremità di trasmissione della medesima, la legge dell'attenuazione è

$$\frac{dP(z)}{dz} = -\overline{\alpha} \cdot P(z) \tag{11.41}$$

che comporta un *decadimento esponenziale* della potenza all'aumentare della lunghezza L della tratta di fibra considerata. La perdita di potenza ℓ_F misurata in dB dall'ingresso all'uscita della fibra, cioè sull'intera tratta L è

$$\ell_F = 10 \log\left(\frac{P(L)}{P(0)}\right) = -10\overline{\alpha}\log(e) \cdot L = -\alpha \cdot L \ [dB]$$
(11.42)

dove α è il coefficiente di attenuazione della fibra, espresso in dB/km.

IL COME E IL PERCHÉ DEI COLLEGAMENTI IN FIBRA OTTICA 433

Nei primi tentativi di comunicazione su fibra della prima metà degli anni '60 del secolo scorso, Charles Kao (premio Nobel per la fisica nel 2009 per l'invenzione delle fibre ottiche, Fig. 11.23) aveva a che fare con fibre aventi un coefficiente di attenuazione dell'ordine di 1000 dB/km, a causa delle impurità nel vetro ai tempi utilizzato, che impediva ogni applicazione pratica. Nel 1970, la ditta statunitense Corning, leader assoluto nel trattamento del vetro, mise a punto un sistema di fabbricazione delle fibre, la *deposizione esterna di vapore* (OVD, outside vapor deposition), che consentì di produrre su scala industriale le prime fibre multimodo con una attenuazione "ragionevole", cioè 20 dB/km @ $\lambda_0 = 0.85 \,\mu\text{m}$ - la lunghezza d'onda della *I finestra*, quella di funzionamento per i LASER disponibili ai tempi. Il limite attuale per fibre SM convenzionali è di circa 0.2 (fino a 0.15 allo stato dell'arte) dB/km per la III finestra di trasmissione con $\lambda_0 = 1.55 \,\mu\text{m}$.





Già nella discussione generale di cui sopra, si è parlato di valori di α in corrispondenza di determinate lunghezze d'onda. Il coefficiente di attenuazione della fibra è infatti fortemente variabile con la lunghezza d'onda nell'ambito dell'infrarosso vicino. L'andamento tipico di α per una fibra monomodo standard al variare di λ è mostrato nella Figura 11.24. Notiamo, come è già stato ampiamente accennato, la presenza di un minimo assoluto (pari a circa 0.2 dB/km) nell'intorno di $\lambda_0 = 1.55 \,\mu\text{m}$, e di diversi minimi relativi, il più notevole dei quali è quello pari a $0.35 \div 0.5 \text{ dB/km}$ nelle vicinanze della lunghezza d'onda $\lambda_{ZD} = 1.3 \,\mu\text{m}$ di zero dispersione (si veda il 11.1.7.1).

La lettrice e/o il lettore avranno a questo punto compresa la ragione per la quale abbiamo nei paragrafi precedenti sempre citato la λ_0 =1.55 μ m come principale lunghezza d'onda di funzionamento dei collegamenti in fibra: le migliori caratteristiche di attenuazione, che la fanno preferire anche in presenza di dispersione non-zero. Torneremo sulle implicazioni dell'attenuazione della fibra dopo aver valutato la sensibilità dei modem per fibra e quindi della massima attenuazione tollerabile per il collegamento, data la potenza del LASER di trasmissione.

11.1.7.4 Ancóra dispersione: la PMD Per concludere, citiamo un'ulteriore limitazione dei cavi in fibra, che diventa rilevante per le dorsali (backbone) ad altissima capacità e su lunghe tratte, e cioè la *dispersione dei modi di polarizzazione* (PMD, Polarization Mode Dispersion).

Abbiamo già visto in Fig. 11.17 che la fibra a modo singolo supporta due modi di propagazione, corrispondenti ai due stati di polarizzazione ortogonale dell'onda luminosa prodotta dal LASER. In teoria, entrambi i modi si propagano con la stessa velocità e quindi



Figura 11.24 Coefficiente di attenuazione tipico α della fibra SMF-28 Corning (la fibra cessa di essere monomodo per $\lambda_c = 1.26 \,\mu\text{m}$).

la fibra può considerarsi single-mode. In realtà, sussistono piccole differenze nella velocità di propagazione dei due modi dovute ad asimmetrie costruttive del cavo in fibra; si ha a che fare con un ulteriore fenomeno di *dispersione*, la PMD appunto, che causa su lunghe distanze una distorsione dell'impulso di bit. Come di consueto, la PMD viene quantificata attraverso il parametro Δt di dispersione temporale chiamato anche DGD, Differential Group Delay), che può essere valutato (in ps) con la relazione

$$\Delta t = D_{PMD} \cdot \sqrt{L} \tag{11.43}$$

dove D_{PMD} è il coefficiente di PMD, misurato in ps/(km)^{-1/2} e L è misurata come sempre in km. Per la fibra standard G.652 il coefficiente di PMD è inferiore a 0.2 ps/(km)^{-1/2} e quindi l'allargamento (ISI) dell'impulso, tenendo conto della dipendenza dalla *radice* della distanza diventa significativo solo per grandi bit-rate e grandi distanze - la PMD è comunque un effetto di ordine superiore rispetto alla dispersione intramodale.

11.2 Modem per Fibra Ottica I: On-Off Keying (OOK) e Rivelazione Diretta (DD, Direct Detection)

11.2.1 Non ci stanchiamo della fisica: Interazioni materia/radiazione elettromagnetica

Il componente optoelettronico che genera i segnali digitali nei modem per fibra ottica è il *diodo LASER a semiconduttore*. Come tutti i LASER, il suo funzionamento è basato sui fenomeni di emissione della radiazione elettromagnetica da parte di un materiale (effetti fotoelettrici) che rivedremo rapidissimamente. Queste considerazioni saranno riprese anche nel paragrafo 11.2.4 che riguarda lo studio dei ricevitori per segnali ottici.

La Figura 11.25 mostra un esempio di LASER a semiconduttore il cui segnale ottico viene inviato in una coda (*pigtail*) in fibra, terminata a sua volta con un connettore ottico e quindi pronto ad essere connesso al veo e proprio cavo in fibra. Immaginiamo di misurare con un "power meter" ottico la potenza ottica P_0 in uscita dal LASER (cioè dalla pigtail) alla lunghezza d'onda $\lambda_0 = c/f_0$. Secondo la meccanica quantistica, questa potenza raccolta dal sensore è costituita da un flusso di moltissimi *fotoni*, ciascuno dei quali porta un quanto di energia $E_f = hf_0$, dove h è la costante di Planck. La maggiore o minore potenza è determinata dal maggiore o minore numero di fotoni/secondo prodotti dal LASER e raccolti dal meter: un LASER da telecomunicaizoni ha una potenza tipica nell'intorno di $P_0=10$ mW, che corrisponde, alla lunghezza d'onda di 1.55 μ m, a $P_0/(hf_0) \simeq 7.8 \cdot 10^{16}$ fotoni/s.





Restando su questo modello corpuscolare della luce, possiamo spiegare la generazione e la rivelazione della luce da parte di dispositivi a semiconduttore analizzando tre fenomeni fondamentali che sono rappresentati in Figura 11.26: l'assorbimento, l'emissione spontanea e l'emissione stimolata di un fotone da parte della materia. Nella Figura 11.26 sono rappresentati due livelli energetici E_1 ed E_2 che possono essere occupati da una certa popolazione di elettroni degli atomi del materiale di cui è costituito un dispositivo; in un gas, tali livelli sono singole righe ben isolate, mentre in un solido cristallino, e in particolare in un semiconduttore, i livelli sono ravvicinatissimi e si trasformano in pratica nelle bande continue mostrate in figura.

Quando un fotone di frequenza f_0 (ovvero energia $E_f = hf_0$) interagisce con il materiale, esso può essere *assorbito* provocando la transizione di un elettrone dal livello *base* E_1 al livello *eccitato* $E_2 = E_1 + hf_0 > E_1$ a energia più alta. Gli elettroni eccitati tornano in generale al loro stato iniziale (di minore energia) attraverso l'*emissione spontanea* di un fotone ad energia hf_0 . Se però un fotone interagisce con un atomo avente *già* un elettrone nello stato eccitato E_2 , si può avere emissione *stimolata* di un ulteriore fotone di energia hf_0 , che accompagna il fotone originario, mentre l'elettrone, ceduta la sua energia, transisce verso il livello base E_1 .

Nell'emissione stimolata, il fotone secondario è un "clone" del primario, ne ha cioè le stesse caratteristiche, in particolare *energia* (quindi frequenza), *quantità di moto* (quindi direzione di propagazione) e *stato di polarizzazione*. Ogni fotone prodotto per emissione stimolata concorre al campo totale macroscopico prodotto dal dispositivo con un contributo "in fase" temporale e spaziale con quello di tutti gli altri fotoni emessi secondo lo stesso meccanismo, dando luogo a quella che si chiama *emissione coerente*. È chiaro che questo



Figura 11.26 Fenomeni di interazione radiazione/materia.

processo di "clonazione" può essere iterato, nel senso che un fotone prodotto per emissione stimolata può a sua volta eccitare l'emissione di altri fotoni: sotto opportune condizioni si può quindi realizzare l'*amplificazione* di un segnale luminoso.

È questo il principio di funzionamento delle sorgenti LASER (Light Amplification by Stimulated Emission of Radiation, amplificazione luminosa per emissione stimolata

MODEM PER FIBRA OTTICA I: ON-OFF KEYING (OOK) E RIVELAZIONE DIRETTA (DD, DIRECT DETECTION) 437

di radiazione) in cui (parte del)l'emissione stimolata amplificata viene opportunamente convogliata all'esterno del dispositivo per realizzare sorgenti con un alto grado di coerenza temporale (fotoni tutti ad una singola frequenza o emissione *monocromatica*) e spaziale (fotoni che si propagano tutti lungo una singola direzione e quindi raggio di luce ad apertura strettissima). Al contrario, nell'emissione spontanea, i vari fotoni generati da transizioni *spontanee* verso stati ad energia inferiore hanno quantità di moto (cioè direzioni di propagazione) aleatorie; quando i contributi di ogni fotone si combinano per dare la radiazione totale, si ha una variazione istantanea di ampiezza e fase del campo totale (dovuta alla variabilità della fase dei vari contributi elementari) che risulta in una *emissione incoerente* (luminescenza), caratterizzata da uno spettro di emissione molto ampio, e da una scarsa coerenza spaziale. È questo il modo di funzionamento tipico del Light-Emitting Diode LED, che non è utilizzato nelle telecomunicazioni.

I semiconduttori sono tra i materiali più adatti ad essere impiegati come generatori di radiazione nell'infrarosso vicino caratteristico delle trasmissioni su fibra; la struttura a due livelli di Figura 11.26 (dove E_1 è l'energia E_v relativa alla *banda di valenza*, ed E_2 è quella E_c della *banda di conduzione*) è infatti tipica di tali materiali. La frequenza di emissione f_0 (lunghezza d'onda λ_0) è determinata dal valore della *energia di gap* $E_g = E_c - E_v$, che è una caratteristica del particolare materiale. Il LASER da telecomunicazioni è un diodo con una giunzione p - n in cui la corrente di alimentazione esterna I (chiamata corrente di *iniezione*) ha lo scopo di iniettare una grandissima popolazione di elettroni eccitati, e far sì che l'emissione stimolata prevalga sull'emissione spontanea. Il dispositivo viene realizzato in opportune leghe composte in percentuale variabile da Indio, Gallio, Arsenico e Fosforo (InGaAsP), una variazione sul materiale optoelettronico per eccellenza e di prima generazione, cioe l'Arseniuro di Gallio (GaAs).

11.2.2 I LASER per Telecomunicazioni

11.2.2.1 *il LASER a cavità o Fabry-Pérot* Il diodo LASER è intrinsecamente un dispositivo *amplificatore* di segnali ottici secondo il fenomeno dell'emissione stimolata. In particolare, il materiale sede dell'emissione stimolata (cioè sede di *amplificazione ottica*) può essere caratterizzato dal *guadagno di potenza ottica per unità di lunghezza g*, che varia al variare della lunghezza d'onda ed è una caratteristica del materiale e della corrente di alimentazione. In realtà, l'uso principale dei LASER è quello di *generatore ottico*, in cui cioè viene prodotto autonomamente un segnale ad una data lunghezza d'onda; nella terminologia delle telecomunicazioni, un LASER è di fatto un *oscillatore ottico* alla frequenza f_0 .

Il modo più semplice di trasformare un amplificatore in un oscillatore è quello di dotarlo di un certo grado di *reazione positiva*. Tale condizione viene raggiunta nel LASER quando si fa in modo di confinare il segnale ottico amplificato per emissione stimolata all'interno di una porzione ben definita del materiale amplificatore (cavità ottica), come mostrato nella Figura 11.27 che rappresenta lo schema super-semplificato di un LASER Fabry-Pérot o a cavità. In tal modo, il segnale all'interno della cavità subisce riflessioni multiple "avanti e indietro" e viene contestualmente amplificato in modo che si venga ad autosostenere una oscillazione (onda stazionaria). Se le due facce sono poi solo parzialmente riflettenti, è possibile estrarre dalla cavità una certa aliquota dell'emissione interna che sarà accoppiata alla pigtail di fibra, purché la perdita di potenza che si realizza in tal modo non sia tale da "disinnescare" l'oscillazione - dunque il guadagno dovuto all'emissione stimolata deve essere sufficiente. Inoltre, il meccanismo di "ping-pong" all'interno della cavità deve generare, in ogni riflessione successiva, un'onda *in fase con se stessa*, altrimenti la reazione che si realizza non è positiva e l'oscillazione non si innesca.



Figura 11.27 Funzionamento del LASER a cavità o Fabry-Pérot.

Per capirne di più, consideriamo un'onda monocromatica piana che si propaga all'interno della cavità secondo la direzione z indicata in Figura 11.27, e indichiamo con E_0 il fasore del campo elettrico nel punto di coordinata $z = 0^+$. Durante la propagazione, il campo viene amplificato per emissione stimolata; il fattore di amplificazione lungo tutto il dispositivo (cioè da $z = 0^+$ a $z = L^-$) è pari a $\sqrt{\exp(gL)}$. Considerando allora un percorso di andata e ritorno all'interno della cavità il fasore viene amplificato del fattore $\left[\sqrt{\exp\left(gL\right)}\right]^2 = \exp\left(gL\right)$, e subisce una rotazione di fase pari a $\exp\left(-\jmath\beta \cdot 2L\right)$, che equivale a $\exp(-j2\pi n_s/\lambda \cdot 2L)$, dove n_s è l'indice di rifrazione del semiconduttore. Inoltre, per effetto delle inevitabili perdite di propagazione, esso viene attenuato del fattore $\exp(-\alpha_{int}L)$, dove α_{int} è il coefficiente di perdita per unità di lunghezza (analogo al coefficiente α_L della fibra ottica) all'interno del semiconduttore. Per effetto della riflettività (intenzionalmente) non ideale delle facce della cavità, il campo viene ulteriormente attenuato del fattore $\sqrt{R} \cdot \sqrt{R}$, dove R < 1 è il coefficiente di riflettività³ di entrambe le facce semi-riflettenti in z = 0 e z = L - in questo modo l'aliquota di potenza (1 - R) viene trasmessa all'esterno (attraverso ciascuna delle due facce) e costituisce l'emissione utile del dispositivo.

Affinché l'oscillazione si autosostenga, il fasore del campo deve trovarsi *inalterato* al punto di partenza $z = 0^+$, dopo la seconda riflessione sulla faccia 1:

$$\tilde{\mathbf{E}}_{0}\exp\left(gL\right)\exp\left(-\alpha_{int}L\right)R\exp\left(-\jmath 4\pi nL/\lambda\right) = \tilde{\mathbf{E}}_{0}$$
(11.44)

per cui, considerando la (11.44) in modulo e fase, si hanno le condizioni

$$g = \alpha_{int} + \alpha_{cav}, \qquad \alpha_{cav} \triangleq \frac{1}{L} \ln\left(\frac{1}{R}\right)$$
 (11.45)

$$\lambda = \frac{2nL}{m} \triangleq \lambda_m \quad \Rightarrow \quad f = m \frac{c}{2nL} \triangleq m \cdot \Delta\nu = f_m \tag{11.46}$$

dove m è un intero arbitrario e $Delta\nu \triangleq c/(2nL)$ è la spaziatura tra le varie frequenze di oscillazione del dispositivo.

³Il coefficiente di riflettività R da un mezzo con indice di rifrazione n al vuoto è pari a $R = [(n - 1) / (n + 1)]^2$ per un segnale incidente lungo la normale all'interfaccia.

La prima equazione mostra chiaramente che il guadagno ottico deve essere sufficientemente alto da uguagliare la somma delle perdite interne e di quelle esterne dovute alla (intenzionale) non perfetta riflettività delle pareti. La seconda equazione indica invece che il LASER a cavità supporta un numero teoricamente *infinito* di modi di oscillazione (cosiddetti *longitudinali*), ovvero frequenze di oscillazione f_m nell'intorno della frequenza f_0 (Figura 11.28(a)). In realtà, il guadagno ottico g ha una certa curva di risposta al variare di λ , cosicché la condizione di "lasing" – cioè la (11.45) – può ottenersi solo per quei valori di λ_m nella (11.46) per cui g è sufficientemente alto (Figura 11.28(b)). Il LASER a cavità Fabry-Pérot è quindi un dispositivo *multifrequenza* che genera uno spettro a "pettine".





I LASER a cavità sono sempre meno utilizzati nelle telecomunicazioni perchè generano una fortissima dispersione in fibra. La banda $\Delta\lambda$ dello spettro del segnale è infatti pressoché indipendente dalla modulazione dati, e dipende soltanto dalla caratteristiche del dispositivo, cioè da quante righe spettrali genera e con che separazione spettrale $\Delta\nu$ le righe si presentano.Un esempio di spettro reale di un Fabry-Pérot è riportato in Fig. 11.29, dalla quale si può valutare una banda di emissione $\Delta\lambda$ = 7 nm. Se usiamo questo valore nella (11.36) otteniamo un ritardo differenziale Δt molto grande, incompatibile con alte capacità di trasmissione.

11.2.2.2 II LASER DFB a frequenza singola. L'equivalente ottico di un oscillatore radio è il LASER a *frequenza singola*, in cui il meccanismo di reazione positiva è diverso da quello





del Fabry-Pérot. Nel LASER monomodo a reazione distribuita (DFB, distributed feedback) raffigurato in Figura 11.30 (a), la reazione ottica per il sostentamento dell'oscillazione viene realizzata non mediante facce riflettenti come nel Fabry-Pérot, bensì attraverso la creazione di uno strato corrugato all'interfaccia tra gli strati p ed n della giunzione, il cosiddetto reticolo di Bragg, che crea una variazione periodica nell'indice di rifrazione del dispositivo subito al di sopra della zona sede di emissione (zona attiva) - la variazione si ha in quanto i due drogaggi p ed n danno luogo a una piccola diversità riguardo a questo parametro. Questa discontinuità, come illustrato nella Fig. 11.31, fa sì che in corrispondenza di ogni corrugamento si creino per rifrazione-riflessione due componenti, riflessa (azzurra) e rifratta (arancio) che viaggiano in due direzioni opposte all'interno del dispositivo (diffrazione di Bragg). La somma di tutte le componenti elementari per ciascuna delle due direzioni di propagazione costituisce l'emissione totale nelle due direzioni del LASER.

Come per il LASER Fabry-Pérot, l'emissione LASER si può avere soltanto quando le componenti multiple si combinano perfettamente in fase: si ha interazione costruttiva, e quindi reazione ottica *positiva* solo quando

$$f = f_m = m \cdot \frac{c}{2n_s\Lambda} = m\Delta\nu \tag{11.47}$$

dove Λ è il periodo spaziale del corrugamento, m è un intero arbitrario e n_s è l'indice di rifrazione.

Se confrontiamo questa relazione con la (11.46) relativa al LASER F-P, notiamo che qui compare il periodo Λ del reticolo di Bragg, che è dell'ordine del μ m (cioè dell'ordine della lunghezza d'onda di emissione), anziché la lunghezza totale L del LASER, che varia normalmente da 100 a 1000 μ m. In questo modo, la spaziatura $\Delta \nu$ tra le possibili righe del LASER DFB è molto grande e soltanto una *singola* frequenza "vede" un guadagno ottico sufficientemente elevato (cfr. la Fig. 11.28) da dar luogo a un'oscillazione LASER significativa: il dispositivo risulta a *frequenza singola*.



Figura 11.31 Spiegazione della Rifrazione di Brag.

11.2.3 La Modulazione d'Intensità (IM)

La tecnica più diffusa per realizzare un link ottico digitale è quella di associare direttamente l'informazione sul valore del generico bit del segnale messaggio alla *potenza* del segnale ottico trasmesso, realizzando la cosiddetta modulazione d'intensità (IM, Intensity Modulation). Il tipo più semplice di IM è la cosiddetta OOK (On-Off Keying), chiamata anche "modulazione a tutto o niente" in cui il LASER viene mantenuto ad una potenza costante P_p per la durata dell'intervallo di bit T quando si desidera trasmettere un simbolo logico pari ad 1, viceversa esso viene completamente spento per uno stesso periodo per trasmettere il simbolo logico 0. La *potenza ottica* generata dal LASER sarà dunque

$$P_0(t) = P_p \sum_{k} b[k] p(t - kT)$$
(11.48)

ove b[k] è il k-esimo simbolo trasmesso, p(t) è il consueto impulso rettangolare NRZ di durata T, e P_p è la potenza di picco del LASER. I simboli (bit) b[k] sono mutuamente indipendenti e assumono valori nell'alfabeto $\{0, 1\}$ con probabilità 1/2. La modulazione dati si realizza materialmente o agendo direttamente sulla corrente di alimentazione del LASER, oppure lasciando il laser in funzionamento CW e agendo su di un componente esterno: viene usata una porta ottica, interposta tra il LASER e la pigtail, che trasmette

(ON) o blocca (OFF) la luce generata dal LASER. La potenza ottica *media* trasmessa sul link è evidentemente $P_T = P_p/2$.

L'espressione (11.48) *non* è il segnale prodotto dal LASER - ne è la potenza istantanea, alla quale, come vedremo nella sezione a seguire, il ricevitore è sensibile. Volendo ricavare l'espressione dello spettro del segnale in fibra, dobbiamo fare un passo indietro e considerare il campo elettrico in polarizzazione lineare in uscita dalla codina di fibra del LASER:

$$\mathbf{E}(t) = \mathbf{E_0} \sum_{k} b[k] p(t - kT) \cos(2\pi f_0 t + \varphi(t) + \phi_0)$$
(11.49)

Il termine $\varphi(t)$ è il cosiddetto rumore di fase del LASER che in un DFB moderno è praticamente assente, e quindi (come già accennato) il dispositivo si comporta come un oscillatore ideale. Peraltro, la potenza istantanea del campo non dipende dal rumore di fase e quindi la (11.48) resta comunque valida.

Con un LASER DFB avente rumore di fase trascurabile, l'equivalente in banda base $S_E(f)$ dello spettro del segnale in fibra si ricava immediatamente come in (2.94)-(2.95):

$$S_{\mathbf{E}}(f) = \frac{\|\mathbf{E}_{\mathbf{0}}\|^2}{4T} \left(\delta(f) + |P(f)|^2\right) = \frac{\|\mathbf{E}_{\mathbf{0}}\|^2}{4T} \left(\delta(f) + \operatorname{sinc}^2(fT)\right)$$
(11.50)

Si nota la presenza di una componente discreta (riga) a frequenza 0 (cioè alla frequenza della portante) dovuta al fatto che i simboli OOK hanno media $\neq 0$ e pari a 1/2. A parte questa differenza, abbiamo il consueto spettro NRZ con una banda al primo nullo pari (in banda passante) a $2/T = 2R_b$, come già menzionato precedentemente.

11.2.4 Il diodo Fotorivelatore pin

I rivelatori di segnali ottici (o *fotorivelatori*) sono dispositivi a semiconduttore che producono una corrente elettrica proporzionale alla potenza utile totale P_R della radiazione luminosa incidente sulla propria area attiva, e basati sul meccanismo di assorbimento descritto nella sezione 11.2.1.

In un fotorivelatore ideale a semiconduttore, ogni fotone incidente viene assorbito, "liberando" un elettrone in banda di conduzione (Fig. 11.26) che, sotto l'azione di un campo elettrico esterno stabilito da un circuito di polarizzazione, contribuisce alla corrente di *fotoconduzione*. Quando la radiazione è assente, la corrente nel circuito esterno è praticamente nulla dato il basso valore di conduttività del semiconduttore intrinseco.

La corrente di fotoconduzione I_{id} prodotta in condizioni ideali di assorbimento totale della radiazione luminosa a frequenza f_0 si ricava facilmente, osservando che il numero di elettroni per unità di tempo che sostengono la corrente elettrica I_{id}/q (dove q è la carica elementare dell'elettrone) deve essere uguale al numero di fotoni per unità di tempo che sostengono la potenza ottica fotorivelata:

$$\frac{P_R}{hf_0} = \frac{I_{id}}{q} \quad \rightarrow \quad I_{id} = \frac{qP_R}{hf_0} \tag{11.51}$$

Per un fotorivelatore reale, la corrente I misurata è minore rispetto al valore ideale di un fattore $\eta < 1$ che prende il nome di *efficienza quantistica* del dispositivo:

$$I = \eta \frac{qP_R}{hf_0} \tag{11.52}$$

Introducendo la cosiddetta responsivity del rivelatore

$$R \triangleq \frac{\eta q}{h f_0} = 0.807 \eta \lambda_0 \qquad (\lambda_0 \text{ in } \mu \text{m})$$
(11.53)

misurata in A/W o in mA/mW, si ha la relazione

$$I = RP_R \tag{11.54}$$

che caratterizza il funzionamento del dispositivo.

Senza voler entrare in ulteriori dettagli, accenniamo solamente che i fotorivelatori per modem in fibra sono dei *diodi a tre strati* o *diodi p-i-n*, in cui l'assorbimento dei fotoni viene confinato all'interno dello strato *i* (intrinseco, cioè senza alcun drogaggio), che si trova intrappolato come in un sandwich tra i due strati con drogaggio p ed n (Fig. 11.32). Il dispositivo viene realizzato in opportune leghe composte in percentuale variabile da Indio, Gallio e Arsenico (InGaAs), una variazione del materiale optoelettronico per eccellenza, cioe l'Arseniuro di Gallio (GaAs).



Figura 11.32 Diodo pin fotorivelatore - (a) struttura di principio, (b) *eterostruttura* in InGaAs: gli strati p ed n sono realizzati in materiale diverso dallo strato i e per questo *non* assorbono fotoni alla frequenza f_0

Dunque la corrente di uscita del diodo *pin* soggetto ad una potenza ottica totale costante P_R è a sua volta costante ed è pari ad $I = RP_R$. In realtà, tale corrente di fotorivelazione è composta da una molteplicità di contributi elementari ognuno dovuto al passaggio nel circuito di uscita di un *singolo* elettrone prodotto per assorbimento di un singolo fotone incidente:

$$I(t) = \sum_{k=-\infty}^{\infty} h(t - t_k)$$
 (11.55)

ove h(t) rappresenta l'impulso di corrente elementare provocato da un singolo elettrone, per cui $\int h(t)dt = q$. La grandezza t_k rappresenta l'istante di comparsa del k-esimo elettrone

nel circuito di uscita ed è una quantità aleatoria, perché aleatorio è l'istante di assorbimento del fotone: gli istanti $\{t_k\}$ seguono una distribuzione di Poisson. Sotto questa ipotesi, I(t) rappresenta dal punto di vista statistico un *processo impulsi di Poisson filtrato*.

La Fig. 11.33 rappresenta una possibile realizzazione del processo I(t), che può essere vista come il risultato di una operazione di filtraggio con un filtro avente risposta impulsiva h(t) di un processo *impulsi di Poisson* $\Delta(t)$ che semplicemente "marca" con delle funzioni impulsive gli istanti di arrivo dei fotoni (elettroni). In questo senso si può identificare h(t) con una specie di "risposta impulsiva equivalente" del fotorivelatore, che è determinata dalle caratteristiche elettriche del dispositivo.



Da questo modello possiamo ricavare la corrente media fotorivelata (Fig. 11.33) attraverso il cosiddetto *teorema di Campbell* sui processi di Poisson:

$$E\{I(t)\} = \mu \int_{-\infty}^{\infty} h(t)dt = \mu q \qquad (11.56)$$

dove μ è l'*intensità* dei processi $\Delta(t)$ e I(t), cioè il numero medio di eventi di assorbimento nell'unità di tempo. D'altra parte, il valore medio $E\{I(t)\}$ coincide con la corrente di fotorivelazione (media) del dispositivo, già ricavata come $I = RP_R$, per cui

$$\mu = I/q = RP_R/q = \frac{\eta P_R}{hf_0} \tag{11.57}$$

e siamo quindi riusciti a legare le proprietà statistiche del processo di Poisson (cioè μ) ai parametri fisici del collegamento (potenza ricevuta P_R).

Calcolato il valor medio della corrente, possiamo riformulare l'uscita del fotorivelatore come segue:

$$I(t) = I + i(t)$$
(11.58)

ove i(t) è un processo di disturbo evidentemente a valor medio nullo detto *shot noise* o "rumore di granularità". Ancora dal teorema di Campbell, si trova che i(t) è stazionario in senso lato e ha densità spettrale di potenza (d.s.p.)

$$S_i(f) = \mu |H(f)|^2 \tag{11.59}$$

ove H(f) è la trasformata di Fourier di h(t). Quando la banda B del ricevitore che segue il fotorivelatore è (molto) più piccola della banda intrinseca del fotorivelatore B_f determinata da H(f), si ha la situazione rappresentata in Fig. 11.34, e il rumore shot ha d.s.p. costante nella banda e pari a

$$S_i(f) \cong S_i(0) = \mu |H(0)|^2 = \mu \left| \int h(t) dt \right|^2 = \mu q^2 = q R P_R$$
(11.60)

cioè il disturbo i(t) può considerarsi *bianco* con tale densità spettrale di potenza.



Figura 11.34 Densità spettrale di potenza del rumore shot

La (11.60) indica chiaramente che la densità spettrale di potenza del disturbo è dipendente dal livello del segnale utile: maggiore è P_R , maggiore è il valore medio I della corrente di uscita (che possiamo riguardare come componente utile), ma maggiore risulta anche la $S_i(f)$:

$$S_i(f) = qRP_R = q \cdot I \tag{11.61}$$

Resta il dubbio se per caso il rumore shot non tenda per questo motivo a sovrastare il segnale utile. Per convincerci del contrario, calcoliamo il rapporto segnale-rumore intrinseco di fotorivelazione SNR_d nella banda del ricevitore B, tenendo conto che lo shot noise è bianco:

$$SNR_d = \frac{I^2}{2B \cdot qI} = \frac{I}{2B \cdot q} = \frac{P_R}{2Bhf_0}$$
(11.62)

Notiamo che (fortunatamente) SNR_d è tanto maggiore (migliore) quanto maggiore è la potenza ricevuta, come è fisicamente ragionevole.

11.2.5 Prestazioni del Link OOK/DD su Fibra

11.2.5.1 *Ricezione Ottima di un Segnale OOK* Immaginiamo di assoggettare il nostro fotorivelatore direttamente al segnale OOK (11.49) in uscita alla fibra trascurando la dispersione (cioè attenuato ma non distorto). Cerchiamo di ricavare la forma del ricevitore ottimo

per un singolo bit di informazione, diciamo b[0] inviato nell'intervallo [0, T). In questo periodo di tempo, la corrente di fotorivelazione I(t) è un processo di Poisson avente una delle due possibili intensità condizionate $\mu^{(0)}(t)$ o $\mu^{(1)}(t)$ associate al rispettivo valore del simbolo trasmesso b[0], e ammettendo per maggiore generalità che la forma d'onda trasmessa sia anche arbitrariamente variabile nel tempo di bit. Il ricevitore ottimo secondo il criterio della Massima Verosimiglianza (MV) calcola la *funzione di log-verosimiglianza* $\Lambda(n)$, n = 0, 1 a partire dalla corrente di fotorivelazione I(t), e applica il seguente criterio per stimare il valore più verosimile $\hat{b}[0]$ di b[0]: "Decidere $\hat{b}[0] = 1$ quando $\Lambda_{\rm M}(1) > \Lambda_{\rm M}(0)$, *altrimenti decidere* $\hat{b}[0] = 0$ ".

L'espressione della funzione di log-verosimiglianza si trova con dei calcoli che non sono complicati ma un po' lunghi, e sono quindi riportati in Appendice G. Il risultato è il seguente:

$$\Lambda(n) = \frac{1}{q} \int_{0}^{T} I(t) \ln\left[1 + \mu^{(n)}(t)/\mu_d\right] dt - T\bar{\mu}^{(n)} \quad n = 0, 1$$
(11.63)

ove $\bar{\mu}^{(n)}$ è il valor medio temporale sull'intervallo di bit dell'intensità del processo condizionata al valore del simbolo b[0] = n:

$$\bar{\mu}^{(n)} = \frac{1}{T} \int_{0}^{T} \mu^{(n)}(t) dt = \frac{\eta}{h f_0 T} \int_{0}^{T} P_R^{(n)}(t) dt \quad n = 0, 1$$
(11.64)

Per comodità di calcolo, la funzione Λ è ricavata supponendo che il fotorivelatore, in assenza di segnale ottico in fibra, produca comunque una *corrente di buio*⁴ I_d corrispondente ad una intensità del processo I(t) in assenza di segnale ottico pari a $\mu_d = I_d/q$.

La particolare forma di $\Lambda(n)$ suggerisce un'implementazione del ricevitore ottimo MV come in Fig. 11.35 che ricalca molto da vicino lo schema del ricevitore a correlatori per canale Gaussiano. La "correlazione" della forma d'onda osservata I(t) viene però effettuata, piuttosto che con il possibile segnale trasmesso (proporzionale a $\mu^{(n)}(t)$), con una particolare funzione di tale segnale, e cioè con la quantità $\ln(1 + \mu^{(n)}(t)/\mu_d)$.

Dalla struttura generale di Fig. 11.35 è possibile ricavare la particolare forma del ricevitore ottimo OOK. Per questo formato di segnale si ha

$$\mu^{(0)}(t) \equiv 0 = \bar{\mu}^{(0)}$$

$$\mu^{(1)}(t) \equiv \frac{\eta P_M}{h f_0} = I/q = \bar{\mu}^{(1)}$$
(11.65)

 $(P_M$ =potenza di picco ricevuta) per cui la legge di decisione è:

$$\hat{b}[0] = 1 \quad \Leftrightarrow \quad \frac{1}{q} \ln(1 + I/I_d) \int_0^T I(t)dt - I \cdot T/q > 0 \tag{11.66}$$

ovvero

$$\hat{b}[0] = 1 \quad \Leftrightarrow \quad \frac{1}{q} \int_{0}^{T} I(t)dt > \frac{I \cdot T/q}{\ln(1 + I/I_d)} \hat{=} \xi \tag{11.67}$$

⁴La corrente di buio del fotorivelatore pin è di fatto la corrente di conduzione inversa di un diodo, quindi è piccolissima (inferiore al nA) ma non nulla



Figura 11.35 Ricevitore MV per Trasmissioni IM/DD

caratteristica di un ricevitore a soglia, con il valore della soglia ξ indicato. Supponendo che la corrente di buio del fotorivelatore sia trascurabile, cioè $I_d \rightarrow 0$, si ha immediatamente $\xi = 0$ e quindi il criterio di decisione diventa

$$\hat{b}[0] = 1 \iff z[0] \triangleq \frac{1}{q} \int_{0}^{1} I(t)dt > 0$$
(11.68)

rappresentato nello schema di ricevitore semplificato di Fig. 11.36.

Se la banda del fotorivelatore è (molto) maggiore della velocità di trasmissione, cioè se $B_f T \gg 1$, l' impulso di corrente elementare h(t) del dispositivo coincide praticamente con una funzione impulsiva avente integrale pari a q: $h(t) \cong q \cdot \delta(t)$. In quest'ipotesi la variabile di decisione z[0] nella (11.68) è pari al *numero di elettroni* N_T osservati nell'intervallo di bit T:

$$z[0] \triangleq \frac{1}{q} \int_{0}^{T} I(t)dt = \frac{1}{q} \int_{0}^{T} \sum_{k=1}^{N_{T}} q \cdot \delta(t-t_{k})dt = \frac{1}{q} \sum_{k=1}^{N_{T}} q = N_{T}$$
(11.69)

Se il fotorivelatore è ideale ($\eta = 1$), N_T è anche il numero di fotoni che hanno interagito con il fotorivelatore, e il ricevitore di Fig. 11.36 è conseguentemente chiamato a *conteggio di fotoni*. Dalla (11.69) si nota che viene deciso $\hat{b}[0] = 1$ non appena il ricevitore riesce a contare *almeno un fotone* nell'intervallo di segnalazione T.



Figura 11.36 Ricevitore a conteggio di fotoni per OOK

11.2.5.2 Il Limite Quantistico (Quantum Limit) Calcoliamo la probabilità di errore P(E) del ricevitore di Fig. 11.36, ricordando una proprietà generale dei processi di Poisson:

il numero N_T di impulsi di corrente originati in un qualunque intervallo temporale di durata T è una variabile aleatoria discreta di Poisson con parametro $\Lambda = \mu T$ (μ intensità del processo), e quindi con massa di probabilità

$$p_{N_T}(k) \triangleq \Pr\{N_T = k\} = \exp(-\mu T) \cdot \frac{(\mu T)^k}{k!} , \ k \ge 0$$
 (11.70)

Usando il teorema della probabilità totale, abbiamo

$$P(E) = \frac{1}{2}P(E \mid b[0] = 0) + \frac{1}{2}P(E \mid b[0] = 1)$$
(11.71)

In assenza di corrente di buio, l'intensità *condizionata* del processo di Poisson è $\mu^{(0)} = 0$ e

$$P(E \mid b[0] = 0) = \Pr\{N_T > 0 \mid b[0] = 0\} = \Pr\{N_T > 0 \mid \mu = 0\} = 0$$
(11.72)

cioè non è possibile commettere alcun errore sul simbolo 0 perché la potenza ottica incidente è nulla e quindi non può essere contato alcun fotone (il lettore o la lettrice ricorderà il canale Z del paragrafo 4.2.3...) ! Invece,

$$P(E \mid b[0] = 1) = \Pr\{N_T \le 0 \mid b[0] = 1\} = \Pr\{N_T = 0 \mid |\mu = \mu^{(1)}\}$$
$$= \exp\{-\mu^{(1)}T\} = \exp\{-I \cdot T/q\} = \exp\{-\eta P_M T/hf_0\}$$
(11.73)
$$P(E) = \frac{1}{-}\exp\{-\frac{\eta P_M T}{1-\frac{1}{2}}\}$$
(11.74)

da cui

$$P(E) = \frac{1}{2} \exp\left\{-\frac{\eta P_M T}{h f_0}\right\}$$
(11.74)

che costituisce il risultato cercato. Quando $\eta = 1$, $P_M T/hf_0$ è il numero di fotoni rivelati in media in un intervallo di bit in cui viene trasmesso un simbolo pari a 1. Il numero medio di fotoni per bit N_b è quindi pari a

$$N_b = \frac{1}{2} \cdot \frac{P_M T}{h f_0} + \frac{1}{2} \cdot 0 = \frac{1}{2} \cdot \frac{P_M T}{h f_0}$$
(11.75)

ed è un indicatore della maggiore o minore energia ottica per bit ricevuta. In definitiva,

$$P(E) = \frac{1}{2} \exp\{-2N_b\}$$
(11.76)

La (11.76) prende il nome di limite quantistico (QL, quantum limit) alle prestazioni del ricevitore DD per trasmissioni OOK, perché ogni fonte di non-idealità del ricevitore (efficienza quantistica < 1, corrente di buio, rumore termico, banda limitata del fotorivelatore) è stata trascurata, e l'unica fonte di errore è l'intrinseco rumore di fotorivelazione dovuto a fenomeni quantistici (alla distribuzione Poissoniana del flusso di fotoni fotorivelati nell'unità di tempo). La curva della BER al limite quantistico del ricevitore ideale DD per segnali OOK è rappresentata in Fig. 11.37.

Si definisce sensibilità (sensitivity) del ricevitore quel valore della potenza ottica media ricevuta \bar{P}_R (espressa generalmente in dBm) o equivalentemente quel numero medio di fotoni per bit \bar{N}_b necessario a raggiungere un valore di P(E) (o tasso d'errore, Bit Error Rate BER) fissato (ad esempio 10^{-9} o 10^{-12} a seconda dell'applicazione). Per il ricevitore ideale appena analizzato, fissando $P(E) = 10^{-9}$, si ha

$$\bar{N}_b = -\frac{1}{2}\ln(2\cdot 10^{-9}) \cong 10$$
 fotoni/bit (11.77)



MODEM PER FIBRA OTTICA I: ON-OFF KEYING (OOK) E RIVELAZIONE DIRETTA (DD, DIRECT DETECTION) 449

Figura 11.37 Limite quantistico del ricevitore DD per segnali OOK: BER (a) - Sensibilitá (b)

(Fig. 11.37 (a)) da cui, considerando che la potenza media in una trasmissione OOK è pari a $P_R = P_M/2$,

$$P_R = N_b h f_0 R_b \tag{11.78}$$

(Fig. 11.37 (b)) che fornisce ad esempio una sensibilità teorica al limite quantistico \bar{P}_R di un ricevitore DD per una trasmissione IM-OOK a 100 Gbit/s in III finestra pari a 0.129 μ W (-39.0 dBm). La sensibilità e/o il QL sono evidentemente dipendenti dalla BER desiderata (*target* BER): ripetendo il calcolo per $P(E) = 10^{-12}$ si ha $\bar{N}_b = 13.5$ e \bar{P}_R =-37.6 dBm.

11.2.5.3 Rumore nel Ricevitore - il fattore Q In realtà, i modem per fibra ottica hanno una sensibilità inferiore rispetto a quella prevista al QL, cioè richiedono una potenza ricevuta maggiore a parità di BER target. Il motivo è semplice: come avevamo già indicato nella Fig. 11.4, il segnale fotorivelato deve essere *amplificato* prima di subire il signal processing (11.69) necessario alla ricezione ottima. Con potenze ottiche ricevute vicine al QL, il segnale elettrico fotorivelato è molto debole e l'amplificazione porta con sé un notevole livello di *rumore Gaussiano* (termico, elettronico) provocato dall'amplificatore e ineliminabile - il consueto AWGN con d.s.p. $N_0/2$. Le prestazioni del ricevitore risultano quindi limitate dal rumore del ricevitore piuttosto che dal rumore shot, e in pratica il regime di funzionamento ricade nel familiare regime relativo alle statistiche del rumore di amplificazione: dopo filtraggio e campionamento, la variabile di decisione z[k] relativa al bit b[k] ha statistiche Gaussiane.

Se è vero che il QL ha un significato concettuale di benchmark importante e universale, la valutazione pratica delle prestazioni dei modem ottici IM/DD viene effettuata attraverso il cosiddetto "fattore Q" (dove Q sta per *Qualità*), la cui ragion d'essere si trova con un ragionamento un po' lungo. Si ammette *in primis* che il ricevitore funzioni in regime Gaussiano (vedi sopra...), e che i livelli medi del segnale ricevuto all'atto della decisione sui due valori 1 e 0 siano rispettivamente $\eta_1 e \eta_0$. Diversamente dal caso ideal del QL, si ammette che $\eta_0 \neq 0$ per eventuali difetti, corrente di buio, distorsione eccetera. Per lo stesso motivo si ammette che la varianza di rumore possa essere diversa sui due livelli (dove in questa varianza si include anche una eventuale ISI derivante da dispersione, il rumore shot diverso sui due livelli ecc., e si ipotizza che questo disturbo totale abbia comunque statistiche Gaussiane). In parole povere, si pone

$$\eta_1 \triangleq \mathbb{E}\{z[k] \mid b[k] = 1\} , \quad \eta_0 \triangleq \mathbb{E}\{z[k] \mid b[k] = 0\}$$
$$\triangleq \mathbb{E}\{(z[k] - \eta_1)^2 \mid b[k] = 1\} , \quad \sigma_0^2 \triangleq \mathbb{E}\{(z[k] - \eta_0)^2 \mid b[k] = 0\}$$
(11.79)

Detto questo, dobbiamo stabilire una soglia ξ di decisione in modo che $\hat{b}[k] = 1$ se $z[k] > \xi$ e $\hat{b}[k] = 0$ se $z[k] \le \xi$. In regime Gaussiano, la BER del ricevitore è

$$P(E) = \frac{1}{2}Q\left(\frac{\eta_1 - \xi}{\sigma_1}\right) + \frac{1}{2}Q\left(\frac{\xi - \eta_0}{\sigma_0}\right)$$
(11.80)

La soglia ottimale è quel valore ξ_{opt} che rende minima questa BER. Calcolando la derivata rispetto a ξ della (11.80) e uguagliando a 0 si ottiene il valore

$$\xi_{opt} = \frac{\sigma_0 \eta_1 + \sigma_1 \eta_0}{\sigma_0 + \sigma_1} \tag{11.81}$$

alla quale corrisponde il valore ottimo della BER

 σ_1^2

$$P(E) = Q\left(\frac{\eta_1 - \eta_0}{\sigma_1 + \sigma_0}\right) \tag{11.82}$$

Il fattore Q (raccomandazione ITU G-976) è proprio l'argomento della funzione Gaussiana $Q(\cdot)$ nella (11.82)⁵:

$$\mathbf{Q} \triangleq \frac{\eta_1 - \eta_0}{\sigma_1 + \sigma_0} \tag{11.83}$$

⁵assistiamo a un evidente conflitto di notazione, ma entrambe le quantitaà sono internazionalmente chiamate Q maiuscolo, evidentemente con diversi significati

e rappresenta una specie di rapporto segnale-rumore complessivo sul collegamento. Questo fattore si valuta empiricamente e facilmente dal *diagramma ad occhio* a valle del fotorivelatore, come suggerito dalla Fig. 11.38.





Le sensibilità dei modem in tecnologia IM/DD effettivamente utilizzati si discostano dal QL di 4-6 dB nelle migliori realizzazioni (backbone) fino a 10-20 dB in realizzazioni di minor pregio (reti FTTH o locali). I migliori risultati si ottengono dotando il ricevitore DD di un *preamplificatore ottico* oppure usando formati digitali differenti con *ricevitori coerenti*, tecnologie che saranno oggetto dei prossimi paragrafi.

11.3 Variazioni sul Tema: WDM e Amplificatori Ottici

11.3.1 Aumentare il bit-rate in fibra: Wavelength Division Multiplexing

Come abbiamo più volte menzionato, la capacità di trasporto dell'informazione di una fibra ottica eccede le decine di Tbit/s. D'altro canto, è impensabile riuscire a sfruttare tale banda attraverso una trasmissione a tale velocità con un modulatore IM/DD a causa delle limitazioni dei componenti elettronici e optoelettronici che in definitiva producono e trattano l'informazione che transita in fibra. Le massime velocità di trasmissione monocanale attualmente (2021) in uso corrente sono dell'ordine dei 100 Gbit/s. L'unica maniera per sfruttare la grande capacità della fibra è quella di ricorrere a sistemi *multicanale*, in cui ogni canale di comunicazione viene "aperto" indipendentemente dagli altri modulando una portante ottica su di una particolare lunghezza d'onda. Si realizza così lo schema di multiplazione a suddivisione di lunghezza d'onda (WDM, Wavelength Division Multiplexing) di Fig. 11.39, che aggrega più canali tributari ad alta velocità per realizzare collegamenti ad *altissima* capacità. Come è chiaro, non esiste alcuna differenza di principio tra la WDM e la FDM impiegata nei collegamenti wireless.

I primi collegamenti WDM effettivamente utilizzati presentavano una spaziatura tra le portanti molto grande, e per questo erano chiamati *Coarse* WDM (CWDM) - ITU-T ha standardizzato una spaziatura $\Delta\lambda$ tra le portanti ottiche CWDM di 20 nm (corrispondenti a ben 2500 GHz @ 1.55 μ m). Il numero di canali è forzatamente limitato e il segnale ottico viene demultiplato e poi ricevuto mediante una struttura di ricevitore con *filtro ottico*



Figura 11.39 Spettro del segnale WDM

che seleziona direttamente nel dominio ottico il canale desiderato, seguito da un ricevitore convenzionale DD (che non è accordato su alcuna lunghezza d'onda). Con il progredire delle tecnologie dei LASER e dei ricevitori, si è passati alla *Dense* WDM (DWDM) in cui la spaziatura standard ITU è 100 o 50 GHz - i canali sono molto più fitti e in numero maggiore, ma il ricevitore deve essere più selettivo e più agile. Vedremo che la tecnologia di ricezione ideale per il DWDM è la ricezione *coerente* del paragrafo 11.4.1.

11.3.2 Gli Amplificatori Ottici per aumentare la Sensibilità del ricevitore OOK

Migliorare la sensibilià del ricevitore, in particolare rispetto al rumore Gaussiano del frontend elettrico del ricevitore, significa poter coprire un collegamento in fibra su maggiori distanze - un fattore fondamentale nel progetto delle backbone di trasporto ad altissima capacità di Internet. La tecnologia-chiave da questo punto di vista è ed è stata a partire dalla fine anni '90 quella degli *amplificatori ottici* (optical amplifier, OA).

11.3.2.1 Generalità sugli amplificatori ottici Gli OA sono dispositivi attivi di varia natura che producono un segnale ottico a potenza incrementata rispetto a quella del segnale (ottico) accoppiato in ingresso, senz'alcun passaggio di dominio ottico/elettrico o viceversa. Parlando in generale, gli OA possono essere impiegati secondo quattro configurazioni base, riassunte schematicamente in Figura 11.40:

- Amplificatore di potenza (booster) per incrementare la potenza del segnale modulato da trasmettere in fibra (Figura 11.40(a)). Il booster risulta particolarmente utile quando non si desidera far operare il LASER di trasmissione ad alte potenze per problemi di affidabilità, stabilità e/o necessità di modulazione.
- Preamplificatore di segnale per incrementare la sensibilità del ricevitore amplificando il debole segnale ricevuto prima della rivelazione (Figura 11.40(b)). In questa configurazione, l'OA si rivela particolarmente utile per migliorare le prestazioni dei ricevitori DD.
- Compensatore delle perdite di suddivisione in reti di distribuzione completamente ottiche in fibra (Figura 11.40(c)). Nei sistemi punto-multipunto infatti, dopo alcuni stadi di suddivisione con componenti ottici passivi, il segnale (ottico) può risultare troppo debole per poter essere utilizzato senza una adeguata amplificazione (ottica).
- *Ripetitore non rigenerativo* (amplificatore di linea) per collegamenti a lunga distanza multitratta (Figura 11.40(d)). Nella Figura 11.41 viene confrontato lo schema di

VARIAZIONI SUL TEMA: WDM E AMPLIFICATORI OTTICI 453



Figura 11.40 Modi di impiego degli amplificatori ottici.

principio di un collegamento multitratta *rigenerativo* (Figura 11.41(a)), avente ripetitori costituiti dalla cascata di un ricevitore optoelettronico DD (che *rigenera* il flusso dati elettrico) e di un modulatore elettroottico con LASER (che rimodula il segnale ottico per la tratta successiva), e di un collegamento multitratta *trasparente* (Figura 11.41(b)) con OA. I vantaggi principali della seconda configurazione risiedono nella maggiore semplicità del ripetitore e soprattutto nella *trasparenza* del medesimo. I ripetitori rigenerativi sono infatti in grado di funzionare soltanto ad un (o ad un insieme limitato di) ben preciso bit-rate, e costituiscono quindi un impedimento ad eventuali incrementi di capacità del collegamento che possono essere desiderabili nel tempo di vita (decenni) del cavo. Al contrario, gli amplificatori ottici sono componenti analogici con una larghissima banda passante in grado di trattare segnali dei più vari formati e larghezza di banda, *in particolare anche segnali WDM*, e che non pongono pregiudiziali su possibili modifiche della capacità della tratta stessa con un semplice upgrade degli apparati in testa e in coda.

Chiaramente, il progetto del dispositivo terrà conto dell'effettivo impiego del medesimo; ad esempio, un OA impiegato come preamplificatore dovrà privilegiare la caratteristica di *bassa rumorosità* nell'amplificazione di piccoli segnali, mentre un booster dovrà fornire la massima *potenza* d'uscita senza fenomeni di saturazione.

Dal punto di vista costitutivo, gli OA si distinguono in tre grandi classi: i) amplificatori a nonlinearità in fibra; ii) amplificatori a semiconduttore; iii) amplificatori in fibra drogata. Gli OA utilizzati nelle backbone sono quelli in fibra, che descriviamo brevemente di seguito.



Figura 11.41 Ripetitori rigenerativi (a) e trasparenti (b) in un collegamento multitratta.

11.3.2.2 Amplificatori in fibra drogata Gli OA sono dei LASER propriamente detti, cioè *privi di retroazione ottica*; il fenomeno alla base dell'amplificazione ottica resta l'*emissione stimolata* all'interno della zona attiva del dispositivo.

Condizione necessaria all'emissione stimolata è realizzare l'inversione di popolazione, cioè fare in modo che la popolazione di elettroni nello stato eccitato E_2 di Fig. 11.26 sia molto maggiore di quella nello stato-base E_1 - come sappiamo, nei LASER a semiconduttore questa funzione viene svolta dalla corrente di alimentazione. Nell'amplificatore in fibra invece, il mezzo fisico sede ell'emissione stimolata è una sezione di qualche decina di metri di una fibra speciale, (vedi Fig. 11.42) nella quale il nucleo viene drogato con atomi della terra rara Erbio - per questo il dispositivo viene chiamato EDFA, erbium-doped fiber amplifier. L'Erbio è necessario perché possiede due livelli tra i quali avviene l'emissione stimolata che sono distanti un'energia hf_0 esattamente corrispondente alla lunghezza d'onda di 1.55 μ m alla quale vogliamo avere amplificazione. La condizione di inversione di popolazione si ottiene sfruttando l'energia fornita alla fibra drogata da un dispositivo esterno, per la precisione un generatore LASER a semiconduttore che emette alla lunghezza d'onda cosiddetta di *pompaggio* (degli elettroni verso l'alto) $\lambda_p = 0.98 \ \mu$ m). L'emissione del LASER pompa viene accoppiata alla sezione di fibra attiva, come rappresentato nella Figura 11.42 (b).

11.3.2.3 Premplificatore ottico per ricevitore IM/DD Usando un OA come preamplificatore per un ricevitore DD, la potenza P_R del segnale ottico ricevuto dalla fibra (accoppiata in ingresso all'OA, $P_{in} = P_R$) viene incrementata di un fattore pari al guadagno di potenza dell'amplificatore G (che può arrivare ad oltre i 30 dB), $P_{out} = G \cdot P_{in}$. Il segnale fotorivelato ha dunque una potenza $P_{out} = G \cdot P_R$ e la corrente di fotorivelazione è parimenti incrementata della stessa quantità rispetto al caso senza preamplificatore:

$$I_{OA} = RP_{out} = RGP_R = G \cdot I \tag{11.84}$$

Il rumore di amplificazione elettrica diventa in questa nuove condizioni *trascurabile* e non più limitante le prestazioni del ricevitore, che risultano effettivamente dominate, stavolta sì, dallo shot noise.

Sfortunatamente, l'amplificatore ottico, in quanto elemento attivo, è esso stesso rumoroso: in uscita, oltre al segnale di ingresso amplificato per emissione stimolata, vi è anche una componente di emissione spontanea, a sua volta amplificata (ASE, Amplified Spontaneous Emission), che viene naturalmente rivelata dal diodo p-i-n insieme com la parte utile




Figura 11.42 Esempio pratico (a, da http://www.fs.com)) e schema di principio (b) di un EDFA - gli *isolatori* sono componenti che impediscono la propagazione del segnale ottico in direzione opposta a quella indicata.

del segnale: il fotorivelatore ha una componente di rumore shot in eccesso dovuta alla fotorivelazione dell'ASE. L'effetto di questo rumore, così come nei modem radio, viene valutato attraverso un parametro caratteristico dell'amplificatore e cioèla *cifra di rumore* F_{OA} .

In conclusione, considerando anche le componenti di disturbo,il segnale amplificato e fotorivelato sarà approssimativamente⁶

$$I_{OA} = RGP_0 + i(t) \quad , \quad S_i(f) = qrGP_0 \cdot F \tag{11.85}$$

Il valore di F dipende dal dispositivo, ma, per le caratteristiche di funzionamento dell'EDFA, non può mai essere inferiore a 2. Dunque il limite di sensibilità del ricevitore IM/DD è il QL + 3 dB. In pratica, $F = 4 \div 6$ dB ma la relativa sensibilità QL + F (dB) viene raggiunta veramente e non è peggiorata dal rumore di amplificazione elettrica - rappresenta quindi un notevole miglioramento rispetto al ricevitore senza preamplificatore.

⁶Questa analisi semplificata non tiene conto di un ulteriore termine dipendente dal *quadrato* dell'ASE che nasce nella fotorivelazione

11.4 Modem per Fibra Ottica II: Modulazione di Fase/Ampiezza e Rivelazione Coerente

11.4.1 II Ricevitore Coerente con LASER Locale

La tecnologia IM/DD, super-consolidata e utilizzata nella stragrande maggioranza dei link ottici per qualunque applicazione, non brilla per raffinatezza ne' per prestazioni - rapportata ai modem radio, ha lo stesso grado di perfezionamento che avevano i ricevitori non accordati di Guglielmo Marconi. Manca infatti tutta la parte di pre-processing del segnale, come nei ricevitori eterodina o in quelli zero-IF rispettivamente delle Figg. 2.35 e 2.34; un primo esempio di preprocessing ottico è di fatto l'amplificazione ottica del paragrafo precedente. I perfezionamenti recenti della fotonica hanno però portato all'introduzione nell'ultimo decennio di modem commerciali con architetture di ricezione molto simili a quelle radio che abbiamo appena citato: i ricevitori ottici *coerenti* che aumentano la sensibilità del ricevitore e la capacità del collegamento.

11.4.1.1 Trasmettitore Coerente La prima tecnologia abilitante per i ricevitori coerenti è la disponibilità di LASER con bassissimo rumore di fase (vedi 11.49) che sono di fatto l'analogo ottico degli oscillatori radio. Inoltre, attraverso tecniche di fotonica integrata, l'oscillazione LASER può essere sfasata o regolata in ampiezza con componenti esterni pilotati dal segnale dati, realizzando quindi tutte le modulazioni di ampiezza e/o fase già utilizzate da tempo per le comunicazioni wireless. Se è vero che ancora formati come il 16QAM ottico non vengono (2021) utilizzati commercialmente, benché sufficientemente maturi come livello di affidabilità tecnologica, è vero che link coerenti DPSK (PSK con demodulazione differenziale, che esamineremo più avanti) sono disponibili commercialmente. Sappiamo che la PSK ha una prestazione in termini di BER migliore della ASK di fatto utilizzata nei sistemi convenzionali IM/DD, e quindi può contribuire al miglioramento della sensibilità del ricevitore.

D'altro canto, schemi di modulazione basati sulla variazione di fase di una portante ottica non possono essere demodulati con il ricevitore DD, che è soltanto sensibile alla potenza istantanea (cioè all'ampiezza)del segnale ottico ricevuto.

11.4.1.2 Combinazione con LASER locale Il secondo fattore abilitante dei modem coerenti è uno stadio di front-end del ricevitore realizzato in fotonica integrata in cui il segnale ottico ricevuto viene *combinato* con il segnale ottico di un LASER locale prima della fotorivelazione - un'operazione simile alla conversione di frequenza nei ricevitori radio, che è rappresentata nella Fig. 11.43 (a).

Consideriamo dunque il campo elettrico $\mathbf{E}(t)$ relativo al segnale ottico in uscita dalla fibra, quando il segnale trasmesso è generato da un LASER a singola frequenza ideale modulato in ampiezza/angolo:

$$\mathbf{E}(t) = \mathbf{E}_{\mathbf{0}} A(t) \cos\left(2\pi f_0 t + \phi(t)\right) \tag{11.86}$$

ove f_0 è come di consueto la frequenza portante associata alla lunghezza d'onda λ_0 di emissione del LASER di trasmissione, A(t) è la modulazione di ampiezza normalizzata $(0 \le A(t) \le 1), \phi(t)$ è la modulazione d'angolo, e infine **E**₀ è il vettore campo "di picco".

Il campo elettrico prodotto dal LASER CW locale ideale che emette ad una lunghezza d'onda λ_L è inoltre

$$\mathbf{E}_{\mathbf{L}}(t) = \mathbf{E}_{\mathbf{0}\mathbf{L}}\cos(2\pi f_L t + \phi_L) \tag{11.87}$$



MODEM PER FIBRA OTTICA II: MODULAZIONE DI FASE/AMPIEZZA E RIVELAZIONE COERENTE 457



Figura 11.43 Ricevitore Ottico Coerente - schema di principio (a) e realizzazione commerciale a 100 Gbaud (b, da http://www.fujitsu.com)

con significato delle varie quantità analogo alla (11.86). Il campo ricevuto e quello locale vengono *sommati* attraverso il combinatore di Fig. 11.43, che nella figura è rappresentato come uno specchio semiriflettente (dicroico), in realtà un componente integrato miniaturizzato - un esempio di ricevitore commerciale completo a 400Gbit/s è mostrato in Fig. 11.43 (b).Il campo risultante dalla combinazione è

$$\mathbf{E}_{\mathbf{R}}(t) = \mathbf{E}_{\mathbf{0}}A(t)\cos\left(2\pi f_0 t + \phi(t)\right) + \mathbf{E}_{\mathbf{0}\mathbf{L}}\cos(2\pi f_L t + \phi_L)$$
(11.88)

ed è questa quantitàà che giunge al fotorivelatore. Nel ricevitore si fa in modo che lo *stato di polarizzazione* dei due campi, entrambi polarizzati linearmente, sia allineato; nella (11.88) ci si può quindi riferire alla comune direzione di polarizzazione e passare a notazione scalare:

$$E_R(t) = E_0 A(t) \cos\left(2\pi f_0 t + \phi(t)\right) + E_{0L} \cos\left(2\pi f_L t + \phi_L\right)$$
(11.89)

La potenza ottica (istantanea) P(t) raccolta dal fotorivelatore, è proporzionale al *modulo* quadro del campo combinato; indicando con P_M la potenza di picco utile (cioè incidente sul

458 SOTTILI COME UN CAPELLO, GRANDI COME IL MONDO: LE FIBRE OTTICHE

fotorivelatore) del segnale modulato, e con P_L la potenza utile del LASER locale troviamo:

$$P(t) = \left[\sqrt{2P_M}A(t)\cos\left(2\pi f_0 t + \phi(t)\right) + \sqrt{2P_L}\cos(2\pi f_L t + \phi_L)\right]^2$$
(11.90)

Ricordando poi la (11.54) ricaviamo il valor medio statistico (variabile nel tempo) $\bar{I}(t)$ della corrente di fotorivelazione (processo impulsi di Poisson non stazionario):

$$\bar{I}(t) \triangleq E\{I(t)\} = RP(t)$$

$$= R[\sqrt{2P_M}A(t)\cos(2\pi f_0 t + \phi(t)) + \sqrt{2P_L}\cos(2\pi f_L t + \phi_L)]^2$$

$$= R\left[2P_M A^2(t)\cos^2(2\pi f_0 t + \phi(t)) + 2P_L \cos^2(2\pi f_L t + \phi_L) + 4\sqrt{P_M P_L}A(t)\cos(2\pi f_0 t + \phi(t))\cos(2\pi f_L t + \phi_L)\right]$$
(11.91)

Svolgendo i prodotti e i quadrati dei termini oscillatori, ed eliminando i termini risultanti alle frequenze ottiche $2f_0$, $2f_L$, $f_0 + f_L$ (evidentemente fuori dalla banda *elettrica* del fotorivelatore...), si ottiene

$$\bar{I}(t) = RP_M A^2(t) + RP_L + 2R\sqrt{P_M P_L} A(t) \cos(2\pi f_{IF} t + \phi(t) - \phi_L)$$
(11.92)

dove $f_{IF} = f_0 - f_L$ è la frequenza intermedia alla quale di fatto viene convertito il segnale ottico ricevuto. Tale frequenza deve essere contenuta nella banda del fotorivelatore - può essere dell'ordine dei GHz, ma può anche essere 0 (conversione diretta in banda base).

Se $f_{IF} \neq 0$, cioè scegliendo una conversione *eterodina*, si possono eliminare le componenti in banda base con un semplice filtraggio, e si ha

$$\bar{I}(t) = 2R\sqrt{P_M P_L} A(t) \cos(2\pi f_{IF} t + \phi(t) - \phi_L)$$
(11.93)

che mostra come la modulazione di ampiezza/angolo impressa sul segnale ottico ricevuto viene trasferita inalterata su di un segnale elettrico la cui frequenza portante f_{IF} è, come già detto, nell'ambito delle microonde, e che può essere quindi demodulato secondo le modalità consuete di un modem wireless.

Anziché in ampiezza/fase, la modulazione digitale può anche essere espressa con le consuete componenti I/Q $x_I = A \cos(\phi)$ e $x_Q = A \sin(\phi)$, come già sappiamo dalla (2.61):

$$\bar{I}(t) = 2R\sqrt{P_M P_L} \left[x_I(t) \cos\left(2\pi f_{IF} t - \phi_L\right) - x_Q(t) \sin\left(2\pi f_{IF} t - \phi_L\right) \right]$$
(11.94)

e quindi il ricevitore coerente è in grado di demodulare qualunque tipo di costellazione che il trasmettitore abbia utilizzato.

11.4.1.3 Ricevitore eterodina od omodina (zero-IF) Il ricevitore coerente che abbiamo appena analizzato è l'analogo di un ricevitore radio eterodina: il LASER locale oscilla ad una frequenza f_L diversa dalla frequenza f_0 di trasmissione e distante da questa proprio della frequenza intermedia f_{IF} . Esattamente come per i ricevitori wireless, il ricevitore ottico coerente eterodina è intrinsecamente agile in frequenza, nel senso che è in grado di ricevere trasmissioni su diverse lunghezze d'onda (frequenze), purché il LASER locale sia sintonizzabile: la frequenza intermedia elettrica f_{IF} è fissa, e quindi variando f_L si varia la $f_0(\lambda_0)$ sulla quale il ricevitore è è sintonizzato.

Oltre che sintonizzabile, il ricevitore coerente è anche estremamente selettivo, perché tale caratteristica è affidata allo stadio elettrico a frequenza intermedia, il cui filtro di ricezione ha una banda stretta: il ricevitore coerente è quindi ideale per demodulare flussi *Dense* WDM (DWDM) nei quali la spaziatura tra i canali, secondo la standardizzazione ITU, è pari a 50 GHz. Il ricevitore DD viceversa, per poter discriminare trasmissioni a diverse lunghezze d'onda coesistenti sulla medesima fibra, deve essere preceduto da un *filtro ottico*, che presenta però caratteristiche di sintonizzabilità e selettività molto inferiori a quelle ottenibili con la rivelazione coerente - queste architetture di ricezione sono più adatta al *Coarse* WDM (CWDM) in cui la spaziatura dei canali è di ben 20 nm.

Se nel ricevitore scegliamo invece $f_L = f_0$, otteniamo uno schema di ricevitore che chiamiamo *omodina*, in cui il LASER locale è mantenuto alla *stessa* frequenza di emissione del trasmettitore, cioè analogo allo schema *zero-IF* dei ricevitori radio. In tal caso il segnale elettrico fotorivelato è direttamente convertito in banda base, ma la ricezione comporta alcuni accorgimenti in più rispetto all'eterodina: *in primis*, le componenti inutili in banda base nella (11.92) risultano nella stessa banda del segnale utile e non possono essere banalmente filtrate come nello schema eterodina. La soluzione è l'adozione dello schema di front-end *bilanciato* mostrato in Fig. 11.44. Nell'ibrido ottico (chiamato anche accoppiatore a -3 dB), il segnale ricevuto viene suddiviso con un rapporto di potenza bilanciato (0.5/0.5) sui due rami e accoppiato come di consueto al LASER locale. All'interno dello stesso componente, e prima della combinazione con il segnale ricevuto, il segnale del LASER locale viene parimenti suddiviso nello stesso rapporto di potenza, ma con un opportuno sfasamento tra i due rami, in modo da presentarsi all'uscita dell'accoppiatore sulla linea superiore in *controfase* rispetto alla linea inferiore. I due segnali incidenti sui fotorivelatori



Figura 11.44 Parte frontale di un ricevitore coerente bilanciato

sono dunque dati da

$$E^{+}(t) = \frac{E_{0}A(t)\cos(2\pi f_{0}t + \phi(t)) + E_{0L}\cos(2\pi f_{L}t + \phi_{L})}{\sqrt{2}}$$
$$E^{-}(t) = \frac{E_{0}A(t)\cos(2\pi f_{0}t + \phi(t)) - E_{0L}\cos(2\pi f_{L}t + \phi_{L})}{\sqrt{2}}$$
(11.95)

e i due fotorivelatori sono poi connessi in modo che la corrente di fotorivelazione totale I(t) risulta la differenza delle due correnti di fotorivelazione parziali:

$$I(t) = I^{+}(t) - I^{-}(t)$$
(11.96)

Supponendo perfetto bilanciamento della responsivity dei due fotodiodi ($R^+ = R^- = R$), la componente media utile rivelata $\bar{I}(t)$ della corrente totale I(t), trascurando i consueti termini a frequenze ottiche, è pari a

$$\bar{I}(t) = \bar{I}^{+}(t) - \bar{I}^{-}(t)$$

$$= R \frac{[P_{M}A^{2}(t) + P_{L} + 2\sqrt{P_{M}P_{L}}A(t)\cos(2\pi f_{IF}t + \phi(t) - \phi_{L})]}{2}$$

$$-R \frac{[P_{M}A^{2}(t) + P_{L} - 2\sqrt{P_{M}P_{L}}A(t)\cos(2\pi f_{IF}t + \phi(t) - \phi_{L})]}{2}$$

$$= 2R\sqrt{P_{M}P_{L}}A(t)\cos(2\pi f_{IF}t + \phi(t) - \phi_{L}) \qquad (11.97)$$

e i termini in banda base vengono spontaneamente a cancellarsi, anche se $f_{IF}=0$. In pratica, abbiamo ottenuto l'analogo ottico del semplice mixer bilanciato che effettua la conversione di frequenza nei ricevitori radio.

La seconda questione che riguarda l'omodina è la seguente: se il trasmettitore usa una modulazione digitale con componenti I/Q, la parte frontale bilanciata del ricevitore deve essere ulteriormente sdoppiata, utilizzando una versione del LASER locale sfasata di 90 gradi, esattamente come in un ricevitore radio zero-IF, ottenendo l'architettura di ricezione omodina generale mostrata in Fig. 11.45, dove (tenendo conto che gli splitter *dimezzano* la potenza ottica dall'ingresso alle due uscite)

$$\bar{I}_{I}(t) = R\sqrt{P_{M}P_{L}A(t)\cos\left(\phi(t) - \phi_{L}\right)}$$

$$\bar{I}_{Q}(t) = R\sqrt{P_{M}P_{L}}A(t)\sin\left(\phi(t) - \phi_{L}\right)$$
(11.98)



Figura 11.45 Parte frontale di un ricevitore coerente bilanciato omodina I/Q

11.4.1.4 Rumore shot nel ricevitore coerente Già sappiamo che il rumore di fotorivelazione i(t) a media nulla sovrapposto alla componente utile $\overline{I}(t)$ è un processo di tipo shot-noise, in generale non stazionario perché l'intensità (11.92) del segnale ottico rivelato non è costante nel tempo (processo non-omogeneo). In realtà, nei ricevitori coerenti accade che la potenza del LASER locale sia *nettamente preponderante* rispetto a quella del segnale utile: $P_L \gg P_M$, cosicchè in pratica si può ritenere ai fini del calcolo dei disturbi che accompagnano la componente utile di segnale, che la potenza incidente sul fotorivelatore sia costante nel tempo, e pari essenzialmente a quella del LASER locale P_L . Questo porta a considerare il rumore di fotorivelazione come un processo stazionario bianco (nell'ipotesi di rivelatore a banda larga) caratterizzato dalla densità spettrale di potenza (vedi 11.60))

$$S_i(f) = qRP_L \tag{11.99}$$

Inoltre, l'ipotesi $P_L \gg P_M$ comporta che il processo shot noise i(t) ha grande intensità, cioè un alto numero di arrivi nell'unità di tempo. In queste condizioni, i(t) é assimilabile a un processo Gaussiano bianco.

In conclusione, il segnale elettrico nello stadio a frequenza intermedia di un ricevitore coerente eterodina, eliminate le componenti in banda base è

$$I(t) = 2R\sqrt{P_M P_L A(t)} \cos(2\pi f_{IF} t + \phi(t) - \phi_L) + i(t)$$
(11.100)

ove i(t) risulta in definitiva un processo Gaussiano bianco con densità spettrale di potenza $S_i(f) = qRP_L$ indipendente dal livello di potenza del segnale dati ricevuto. Nel ricevitore omodina I/Q abbiamo invece

$$\bar{I}_{I}(t) = R\sqrt{P_{M}P_{L}A(t)}\cos(\phi(t) - \phi_{L}) + i_{I}(t)$$
$$\bar{I}_{Q}(t) = R\sqrt{P_{M}P_{L}}A(t)\sin(\phi(t) - \phi_{L}) + i_{Q}(t)$$
(11.101)

dove i due processi di rumore $i_I(t)$ e $i_Q(t)$ sui due rami sono Gaussiani bianchi indipendenti, ciascuno con la stessa d.s.p derivante dalla rivelazione del LASER locale con potenza dimezzata per lo split: $S_I(f) = S_Q(f) = qRP_L/2$.

La ricezione coerente ha un ulteriore grande vantaggio sulla DD: le componenti di segnale fotorivelato sono intrinsecamente *amplificate* dalla combinazione con il LASER locale e successiva rivelazione rispetto al caso della rivelazione diretta. In questo modo, il rumore di amplificazione elettrica successivo è *trascurabile* rispetto allo shot-noise di rivelazione (11.99), e, contrariamente al ricevitore DD, *non* altera le prestazioni del link, che restano sostanzialmente *shot-noise limited* piuttosto che *electrical-noise limited*. Dobbiamo adesso vedere se tali prestazioni sono soddisfacenti.

11.4.2 Prestazioni dei Link Coerenti su Fibra

11.4.2.1 *Ricezione sincrona con recupero della fase della portante* Stante la modellistica del paragrafo precedente, le prestazioni in termini di BER dei link ottici coerenti sono immediate, richiamando i risultati del Capitolo 2 relativamente alla ricezione su canale AWGN e tenendo conto che il rumore di amplificazione elettrica non incide. Possiamo iniziare dal formato più performante, cioè il BPSK/QPSK, confrontando anche i due casi etorodina/omodina che, scopriremo, portano a risultati differenti. Supponiamo naturalmente che il ricevitore elettrico successivo al front-end sia in grado di stimare perfettamente la fase iniziale della portante residua ϕ_L (ricevitore sincrono).

Per calcolare la nostra BER, dobbiamo particolarizzare l'espressione generale del matched filter bound su AWGN (2.104). Dalla (11.100), nel caso di BPSK con rivelazione eterodina abbiamo $A(t) \equiv 1$, quindi la potenza media ricevuta è $P_R = P_M$ ed $E_b = 4R^2 P_M P_L \cdot T/2$ (trasmissione in banda passante); per quanto riguarda il rumore, tenendo conto di nuovo che il segnale è in banda passante attorno ad f_{IF} , abbiamo che $qRP_L = N_0/2$. Dunque

$$P(E)_{eter}^{BPSK} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = Q\left(\sqrt{\frac{2R^2P_MP_LT}{qRP_L}}\right) = Q\left(\sqrt{\frac{2RP_MT}{q}}\right) \quad (11.102)$$

Introducendo come di consueto il numero medio di fotoni per bit $N_b = P_M T/hf_0$ e supponendo unitaria l'efficienza quantistica del fotorivelatore p-i-n (cioè $R = q/(hf_0)$, si ha infine

$$P(E)_{eter}^{BPSK} = Q\left(\sqrt{2N_b}\right) \tag{11.103}$$

Esaminando il caso omodina, sempre immaginando di stimare perfettamente ϕ_L , possiamo considerare la sola componente I per la rivelazione BPSK, ponendo formalmente $\phi_L = 0$. Poiché stiamo considerando componenti I/Q *in banda base*, abbiamo che $E_b = R^2 P_M P_L T$ e $qRP_L/2 = N_0$. Dunque,

$$P(E)_{omod}^{BPSK} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = Q\left(2\sqrt{\frac{RP_MT}{q}}\right) = Q\left(2\sqrt{N_b}\right)$$
(11.104)

cioè *migliore di 3 dB* rispetto alla ricezione eterodina - questo vantaggio sul rapporto segnale-rumore è una condizione *generale* che si ouò applicare trasversalmente a qualunque formato digitale: la ricezione omodina porta comunque un vantaggio di 3 dB rispetto all'eterodina, necessitando però di un front-end più complicato.

Come si confrontano questi risultati con il QL della DD ? Osserviamo che quando $x \gg 1$, allora $Q(x) \cong e^{-x^2/2}$, si ha

$$P(E)_{omod}^{PSK} = Q(2\sqrt{N_b}) \cong \exp(-2N_b)$$
(11.105)

cioè il ricevitore PSK coerente omodina raggiunge il QL a parità di potenza media ricevuta, con in più la caratteristica che tale limite è (più facilmente) raggiungibile perchè il ricevitore non necessita di preamplificatore ottico ed è solo marginalmente influenzato dal rumore di amplificazione. Le prestazioni teoriche di alcuni formati con ricezione coerente sono mostrate per completezza nella Fig. 11.46.

11.4.2.2 Ricezione asincrona senza recupero della fase della portante La difficoltà di realizzazione di un demodulatore sincrono può essere aggirata realizzando schemi di demodulazione *asincroni*, che non necessitano cioè del recupero della fase residua della portante. Il formato più efficiente da questo punto di vista, usato in modem per backbone commerciali, è il DPSK (Differential PSK) con demodulazione, appunto, *differenziale*. Per poter evitare di recuperare la fase della portante, il segnale PSK trasmesso è il risultato di una codifica differenziale: il *k*-esimo simbolo BPSK di linea b[k] è ottenuto dal simbolo binario di sorgente d[k] mediante la seguente operazione:

$$b[k] = d[k] \oplus b[k-1], \quad (b[0] \triangleq 1),$$
 (11.106)

ove \oplus indica l'operazione di "somma logica" e il sopprassegno la negazione logica. Il generico simbolo di linea è dunque identico al precedente se il corrispondente simbolo di



Figura 11.46 Curve di BER dei demodulatori ottici coerenti

sorgente assume il valore "1", viceversa se il simbolo di sorgente vale "0". I valori logici vengono poi come di consueto mappati sui valori fisici simmetrici ± 1 .

Con la codifica differenziale, è possibile usare il demodulatore differenziale mostrato in Fig. 11.47 (a) per ricezione eterodina⁷ e (b) per ricezione omodina, ove le linee più spesse indicano il segnale I/Q, cioè l'inviluppo complesso $I_I(t) + \jmath I_Q(t)$, del corrispondente segnale ottico (passa-banda). Questo demodulatore effettua intrinsecamente anche la decodifica dei simboli di linea, fornendo direttamente i simboli di sorgente d[k]. La fase iniziale ϕ_L della portante si cancella automaticamente nell'operazione di moltiplicazione (reale in banda passante o complessa in banda base).

La BER del demodulatore differenziale eterodina/omodina è facilmente calcolabile, ed è pari rispettivamente a

$$P(E)_{eter}^{DPSK} = \frac{1}{2} \exp(-N_b) \quad , \quad P(E)_{omod}^{DPSK} = \frac{1}{2} \exp(-2N_b) \tag{11.107}$$

Le relative curve di BER sono visibili nella Fig. 11.46. Si nota che il DPSK omodina ha esattamente la stessa performance del QL IM/DD e peggiora rispetto al demodulatore sincrono di meno di 1 dB.

11.4.2.3 Altri formati digitali in link coerenti È chiaro che, stante la modellistica (11.100)-(11.101), il modem coerente è in grado di utilizzare qualunque formato di modu-

⁷Il filtro passa basso che precede il campionatore ha il solo scopo di eliminare le componenti a frequenza $2f_{IF}$ che nascono all'uscita del mescolatore. Per una demodulazione corretta, deve essere inoltre $f_{IF}T = 2k\pi$, con k un qualche intero, oppure $f_{IF}T \gg 1$.

SOTTILI COME UN CAPELLO, GRANDI COME IL MONDO: LE FIBRE OTTICHE





Figura 11.47 Demodulatore differenziale asincrono per DPSK - (a) in banda passante (eterodina); (b) in banda base (omodina)

lazione digitale I/Q, con le relative prestazioni su AWGN che già conosciamo, incluse le costellazioni QAM (che compaiono già opzionalmente in standard veloci per reti locali), e l'OFDM (confinato ancora al mondo della ricerca). Molte sperimentazioni sono anche state condotte su tecniche che prevedono di affidare l'informazione digitale allo stato di polarizzazione del segnale ottico (Polarization Shift Keying o PolSK), anche con demodulazione differenziale, e che richiedono schemi coerenti. Come per l'OFDM, tali formati non hanno (ancora) trovato applicazione commerciale.

11.5 Rete di Trasporto e Rete di Accesso: dalla Backbone all'FTTx

11.5.1 Le Tecnologie FTTx

Storicamente, i cavi in fibra (con le relative tecnologie di componentistica e modulazione digitale che abbiamo esaminato nei paragrafi precedenti) nascono come segmenti della rete di trasporto a grande capacità e lunga distanza. L'evoluzione di Internet e delle relative applicazioni porta però a richiedere capacità periferica sempre maggiore, in modo da rendere disponibili bit-rate notevoli (dell'ordine del Gbit/s) anche all'utente (domestico) finale. Inizialmente, queste esigenze di banda larga sono state soddisfatte con le tecnologie xDSL su doppino telefonico esaminate brevemente nel paragrafo 8.6. Quando però il requisito è quello di distribuire una capacità che si avvicina al Gbit/s, l'unico mezzo fisico adatto è la fibra ottica che si estende sempre di più verso la rete di accesso e l'utente finale. Ovviamente, tutto questo comporta uno sforzo economico e ingegneristico notevole perchè le (nuove) fibre per la rete di accesso devono essere materialmente interrate ex-novo (contrariamente ai vecchi doppini telefonici per xDSL), con tutti gli inconvenienti del caso.

Le tecnologie su fibra per la rete di accesso sono identificate con l'acronimo FTTx, che sta per Fiber-To-The-x, e dove x ha diversi significati (vedi Fig. 11.48): nella versione più moderna e performante della tecnologia x vale H (dunque FTTH) che significa Home: la

464

fibra giunge capillarmente come il doppino telefonico fino in casa dell'utente finale. La versione immediatamente meno performante è la FTTB, dove la fibra giunge a un *Building* multiresidenziale (condominio) e il flusso digitale viene poi distribuito su cavo Ethernet o (meno frequentemente) su coassiale, oppure parliamo di FTTC, dove la fibra arriva al Cabinet, cioè l'armadio in strada, e l'ultimo segmento di connessione si avvale di varie tecnologie di distribuzione: se l'utenza finale non è nelle immediate vicinanze del cabinet, deve essere collegata con una tecnologia VDSL e abbiamo un approccio *ibrido fibra/rame* con VDSL. Se la fibra non è disponbile, ricadiamo nelle vecchie tecnologie xDSL.



Figura 11.48 Tecnologie per la rete d'accesso (da http://www.tim.it)

La tendenza moderna è quella di estendere il più possibile la tecnologia FTTH per rendere disponibile (prima o dopo a seconda degli investimenti disponibili) la cosiddetta *banda ultra-larga* a tutti gli utenti, pubblici o privati, individuali o collettivi, che ne abbiano necessità o desiderio. La maniera che si sta affermando come la più efficace tecnicamente ed economicamente per fornire banda ultra-larga è quelle basata sulle tecnologie di rete ottica passiva (PON, Passive Optical Network). La Fig. 11.49 indica il grado di penetrazione delle tecnologie a banda larga (ADSL, FTTC) e ultra-larga (FTTH) nei vari paesi all'anno 2022.

11.5.2 Le PON per la Rete di Accesso

La PON è un'architettura di rete in fibra di tipo punto-multipunto per connettere un gruppo di utenti finali con un punto di presenza Internet (PoP, Point of Presence) di un operatore di rete (ISP, Internet Service Provider). La caratteristica principale della PON, la cui architettura di base è indicata nella Fig. 11.50, è che non vengono utilizzati dispositivi attivi lungo la connessione: gli apparati attivi risiedono soltanto nel PoP e presso gli utenti finali, mentre la rete in se stessa è costituita da segmenti di fibra single-mode e da una serie di *splitter*, cioè suddivisori di segnale di tipo passivo (Fig. 11.51) che "ramificano" il collegamento verso gli utenti finali. La PON è quindi intrinsecamente performante e non soggetta a interferenze (perché basata su fibra) e robusta (perché passiva).

A seconda del territorio e delle sue proprie esigenze, gli stadi di suddivisione con splitter possono essere più di uno in modo da servire un gruppo di utenti anche numeroso - un esempio è mostrato in Fig. 11.52 che rappresenta l'architettura di una PON multi-stadio. In questa architettura, il punto di splitting indicato con SFP comprende 20 splitter 1 x 16 ed è

466 SOTTILI COME UN CAPELLO, GRANDI COME IL MONDO: LE FIBRE OTTICHE







Figura 11.50 Architettura generale di una PON per servizi FTTH

inteso servire fino a 5 diversi *service provider* che vogliono fornire servizi all'utente finale attraverso la PON (le fibre che convergono in un unico splitter appartengono ad un unico provider). Gli ulteriori punti di splitting PFP con splitter 1 x 4 raccolgono ulteriormente le fibre derivanti dai vari SFP sul territorio per poi giungere al PoP con 80 fibre che servono un massimo di 4096 utenti finali.

La Fig. 11.50 contiene anche la nomenclatura tipica degli apparati della PON: gli ONT, *Optical Network Terminator* (chiamati anche ONU, *Optical Network Unit*), sono di fatto i modem lato utente, mentre l'OLT, *Optical Line Termination*, è l'analogo del DSLAM delle reti ADSL, cioè l'aggregatore di flussi e l'interfaccia verso la rete (normalmente IP) esterna. Come si nota, la rete supporta entrambe le direzioni di comunicazione su di una singola fibra mediante un Wavelength-Division Duplexing (WDD) in cui il downstream si svolge in





Figura 11.52 Esempio di Rete Ottica Passiva multi-stadio (fonte: OpenFiber) - POP=Point Of Presence, PFP=Primary Flexibility Point, SFP=Secondary Flexibility Point, U.I.=Unità Immobiliare - Cable Area=Area servita dallo stesso cavo da 80 fibre, Collection Area=Area servita da uno stesso PFP, Binding Area=Area servita da uno stesso SFP.

11.5.3 Un esempio di standard: la Gigabit PON (G-PON) ITU e le sue evoluzioni future

La Fig. 11.50 fa già riferimento allo standard per la PON che si sta maggiormente diffondendo: la Gigabit PON (G-PON) ITU-T G.984. Dal punto di vista trasmissivo, come già accennato, abbiamo modem IM/DD convenzionali su fibra single.mode standard G.652 con WDD per il full-duplex. I flussi digitali sono protetti con codici FEC a blocco di tipo Reed-Solomon (255,239). La capacità nominale offerta all'utente è 2.4 Gbit/s in downstream e 1.2 o 2.4 Gbit/s upstream a seconda delle implementazioni - ecco perché parliamo di banda *ultra-larga*. La massima distanza di copertura prevista (da OLT a ONU) è di 20 km, molto di più che nelle reti ADSL su rame. La tecnologia per garantire l'accesso multiplo





è il classico TDM/TDMA con un frame di 125 μ s downstream formattato centralmente dall'ONT, e burst multipli di dati TDMA upstream, organizzati in un frame ancora di 125 μ s upstream (Fig. 11.53). La sincronizzazione upstream delle ONU nella trama TDMA è molto semplice ed efficace e non verrà trattata - del resto, come sarà chiaro a questo punto a chi legge, le tecnologie-base di G-PON sono piuttosto consolidate; la questione più complessa nello sviluppo delle PON è *economica* e risiede nei grandi investimenti necessari per la posa delle fibre.

In chiusura, delineiamo brevemente (vedi la Fig. 11.54) l'evoluzione delle (G-)PON verso capacità per utente ancora maggiori - dal Gbit/s fino a 10 Gbit/s prima asimmetrico (XG) e poi simmetrico (XGS) fino ad arrivare ai 40 Gbit/s, al momento (2023) ancora non disponibile commercialmente in Europa. Allo stato attuale (2023) questo percorso è in via di svolgimento, con una maturità molto variabile da zona a zona all'interno di una nazione, e da nazione a nazione in Europa e nel mondo. Altre tecnologie da noi già prese in considerazione, come il WDM e la ricezione coerente, faranno parte di questa evoluzione.

GPON→XG-GPON→NG-PON2DownstreamDownstreamDownstreamDownstream

2,5 Gbit/s Upstream 1,25 Gbit/s 10 Gbit/s Upstream 2,5 Gbit/s Downstream 10 Gbit/s Upstream 10 Gbit/s Downstream 40 Gbit/s Upstream 10 Gbit/s



APPENDIX A OTTIMALITÀ DEL CODICE DI HUFFMAN

Come si può dimostrare che la procedura di Huffman porta ad un codice massimamente efficiente? La dimostrazione è ricorsiva (matematicamente si direbbe: condotta per induzione). Abbiamo già menzionato il concetto di "sorgente ridotta" S_i come risultato di ogni passo di accorpamento dei nodi. Segue da questo anche il concetto già menzionato di "codice ridotto" \mathcal{M}_i con *i* parole ($\mathcal{M}_M = \mathcal{M}$ è il codice sulla sorgente originaria) che la procedura di Huffman genera per la sorgente ridotta. Supponiamo allora che il codice \mathcal{M}_i ottenuto con la procedura di Huffman sulla sorgente S_i al passo i (che ha *i* parole di codice) sia ottimale, e dimostriamo che è ottimale anche il codice \mathcal{M}_{i+1} per la sorgente "espansa" S_{i+1} .

Partiamo a questo fine da alcune considerazioni preliminari. Innanzitutto, come è logico aspettarsi, i) se ho a che fare con un codice ottimo, e due simboli di sorgente s_i ed s_m sono tali per cui $p_i \leq p_m$, allora deve essere $l_m \leq l_i$. Se così non fosse, il codice non sarebbe ottimo perché basterebbe scambiare le due parole associate ai due simboli per ottenere un codice a lunghezza media minore. Infatti la differenza tra i contributi alla lunghezza media nei due casi sarebbe $(p_i l_i + p_m l_m) - (p_i l_m + p_m l_i) = (p_m - p_i)(l_m - l_i) \geq 0$.

Se poi, come abbiamo spesso fatto, ordiniamo i simboli di sorgente in modo che $p_1 \leq p_2 \leq ... \leq p_M$ (e quindi $l_1 \geq l_2 \geq ... \geq l_M$), scopriamo che ii) in un codice ottimo, di parole di lunghezza l_M ce ne devono essere (almeno) due. Se infatti ci fosse *una* sola parola μ_M avente lunghezza massima l_M , potremmo costruire un nuovo codice con la parola μ_M privata del suo ultimo simbolo. Il codice sarebbe ancora a prefisso (perché la parola μ_M ridotta sarebbe sempre a lunghezza $\geq l_{M-1}$, e quindi non potrebbe esser prefisso di nessun'altra) e avrebbe lunghezza media più piccola.

Infine, iii) delle (almeno due) parole di lunghezza l_M , necessariamente due devono

Comunicazioni Digitali, I Edizione. di M. Luise Copyright © 2023 University of Pisa

470 OTTIMALITÀ DEL CODICE DI HUFFMAN

condividere il prefisso di lunghezza l_M -1, e devono avere solo l'ultimo simbolo diverso (altrimenti potremmo di nuovo togliere l'ultimo simbolo di μ_M e ottenere un codice a prefisso più efficiente). Possiamo sempre supporre che, nella nostra numerazione, le due parole in questione siano $\mu_{M-1} e \mu_M$.

Detto questo, introduciamo una notazione un poco più complessa: chiamiamo $\mu_m^{(i)}$ la generica parola m-esima del codice di i parole costruito sull'albero per la sorgente (ridotta) S_i , e indichiamo con $p_m^{(i)}$ la sua probabilità. Supponiamo ora che \mathcal{M}_i sia ottimo e consideriamo il codice esteso \mathcal{M}_{i+1} secondo la relazione del procedimento di Huffman, e cioè

$$\mu_m^{(i+1)} = \mu_m^{(i)} \quad m = 1, \dots, i-1 \quad , \quad \mu_i^{(i+1)} = \mu_i^{(i)} 1 \quad , \quad \mu_{i+1}^{(i+1)} = \mu_i^{(i)} 0 \tag{A.1}$$

ovvero le prime M - i - 2 parole dei due codici sono le stesse, mentre le ultime due di \mathcal{M}_{i+1} hanno l'ultima parola di \mathcal{M}_i come prefisso e poi o 1 o 0 come terminazione (suffisso). Possiamo stabilire una relazione sulle lunghezze medie dei due codici:

$$l_{med}^{(i+1)} = l_{med}^{(i)} + 1 \cdot p_i^{(i+1)} + 1 \cdot p_{i+1}^{(i+1)}$$
(A.2)

dove il contributo di aumento della lunghezza media è dato dall'ulteriore cifra binaria di terminazione aggiunta alle due parole "estese" $\mu_i^{(i+1)} e \mu_{i+1}^{(i+1)}$ del codice \mathcal{M}_{i+1} . Ragioniamo per assurdo che supponiamo che quest'ultimo codice, con la sua lunghezza

media (A.2), non sia a sua volta ottimo. In tal caso, potremmo costruire un codice diverso media (A.2), *non* sia a sua volta ottimo. In tal caso, potremmo costruire un codice diverso $\overline{\mathcal{M}}_{i+1}$ avente $\overline{l}_{med}^{(i+1)} < l_{med}^{(i+1)}$ con le solite tre proprietà che abbiamo già discusso per i codici ottimi. Da $\overline{\mathcal{M}}_{i+1}$ potremmo poi costruire per riduzione un ulteriore codice $\overline{\mathcal{M}}_i$ per S_i avente le prime i-1 parole identiche, $\overline{\mu}_m^{(i)} = \overline{\mu}_m^{(i+1)}$, m = 1, ..., i-1 e poi come ultima parola il prefisso tolta l'ultima cifra binaria della $\overline{\mu}_i^{(i+1)}$ (o della $\overline{\mu}_{i+1}^{(i+1)}$ che sarebbe la stessa cosa). Le lunghezze medie di questi due codici modificati sarebbero legati da una relazione simile alla (A.2), e cioè $\overline{l}_{med}^{(i+1)} = \overline{l}_{med}^{(i)} + p_i^{(i+1)} + p_{i+1}^{(i+1)}$. Quindi, se fosse $\overline{l}_{med}^{(i+1)} < l_{med}^{(i+1)}$ come conseguente dall'ipotesi che \mathcal{M}_{i+1} non è ottimo, necessariamente risulterebbe anche $\overline{l}_{i}^{(i)} = c l^{(i)}$ il che à control parahé \mathcal{M} à parinetati iniziale attime. $\bar{l}_{med}^{(i)} < l_{med}^{(i)}$, il che è assurdo perché \mathcal{M}_i è per ipotesi iniziale ottimo! Per ricapitolare, se \mathcal{M}_i è ottimo, segue che necessariamente anche \mathcal{M}_{i+1} deve risultare

ottimo, il che completa la nostra dimostrazione.

APPENDIX B

TEOREMA DELLA CODIFICA DI CANALE

Riprendiamo il concetto di codice casuale per dimostrare il secondo teorema della codifica nel caso di codice binario con trasmissione su canale senza memoria. Useremo la notazione dei codici a blocco introdotta nel Cap. 6 in cui i bit di sorgente sono indicati con la lettera b, i simboli binari codificati con la lettera a e i simboli (binari) in uscita al canale sono indicati con d.

Indichiamo dunque con \mathbf{b}_k , $k = 0, 1, ..., 2^K - 1$ (tutte) le parole di sorgente (supposte equiprobabili) di lunghezza K, e con \mathbf{a}_k le corrispondenti 2^K parole di codice di lunghezza N che costituiscono il codebook \mathcal{A} . Ricevuta una parola \mathbf{d} all'uscita del canale (non necessariamente $\in \mathcal{A}$), il decodificatore utilizza il criterio di decisione MAP (di fatto equivalente al MV visto che le parole di sorgente sono tutte equiprobabili), cioè

$$\mathbf{b}_{\hat{k}} = \max_{\mathbf{a}} p(\mathbf{b}_{\ell} | \mathbf{d}) = \max_{\mathbf{a}} p(\mathbf{d} | \mathbf{b}_{\ell})$$
(B.1)

ovvero, ragionando sulle parole di codice invece che su quelle di sorgente senza perdere generalità vista la corrispondenza uno-a-uno tra le due,

$$\mathbf{a}_{\hat{k}} = \max_{\mathbf{a}_{\ell} \in \mathcal{A}} p(\mathbf{d}|\mathbf{a}_{\ell}) \tag{B.2}$$

471

La $p(\mathbf{d}|\mathbf{a}_{\ell})$ si calcola facilmente (sorgente e canale senza memoria) conoscendo le probabilità di sorgente e di canale.

Dunque ogni parola di codice $\mathbf{a}_k \in \mathcal{A}$ porta con se' una *zona di decisione* definita da

$$\mathcal{B}_{k} = \{ \mathbf{d} : p(\mathbf{d}|\mathbf{a}_{k}) \ge p(\mathbf{d}|\mathbf{a}_{\ell}) \ \forall \ell \neq k \}$$
(B.3)

Comunicazioni Digitali, I Edizione.

di M. Luise Copyright © 2023 University of Pisa

472 TEOREMA DELLA CODIFICA DI CANALE

che induce una probabilità di errore condizionata alla particolare parola di codice \mathbf{a}_k

$$P(E_m|\mathbf{a}_k) = \Pr\{\mathbf{d} \notin \mathcal{B}_k | \mathbf{a}_k\}$$
(B.4)

La probabilità di errore del codice si trova mediando la (B.4) sulle 2^K parole di codice $\mathbf{a}_k \in \mathcal{A}$ che sono equiprobabili perché associate alle equiprobabili parole di sorgente:

$$P(E_m) = \frac{1}{2^K} \sum_{\mathbf{a}_k \in \mathcal{A}} P(E_m | \mathbf{a}_k) = \frac{1}{2^K} \sum_{\mathbf{a}_k \in \mathcal{A}} \Pr\{\mathbf{d} \notin \mathcal{B}_k | \mathbf{a}_k\} = \frac{1}{2^K} \sum_{\mathbf{a}_k \in \mathcal{A}} \sum_{\mathbf{d} \notin \mathcal{B}_k} p(\mathbf{d} | \mathbf{a}_k)$$
(B.5)

Siamo adesso interessati a studiare le proprietà di un codice (N, K) con K/N = r < cnel quale il codebook A viene determinato come segue: la k-esima parola di codice $\mathbf{a}_k = [a_{k,0}, a_{k,1}, \dots, a_{k,N-1}]$ è individuata scegliendo in modo indipendente il simbolo n-esimo della parola $a_{k,n}$ secondo la distribuzione (p, 1 - p) che raggiunge la capacità cdel canale (*capacity-achieving* distribution). Quindi in questa costruzione una qualunque parola di lunghezza N **a** ha una certa probabilità $p(\mathbf{a})$ ben precisa di far parte di un codice non va confusa con la probabilità $1/2^K$ di presentazione di una data parola di codice \mathbf{a}_k che si verifica quando il codice viene effettivamente utilizzato per la trasmissione.

Formato (estratto...) il codice, possiamo ricavarne attraverso la (B.5) la probabilità di errore $P(E_m)$ che, valutata sull'insieme di tutti i codici, risulta essere una variabile aleatoria con una statistica indotta dalla distribuzione di probabilità capacity-achieving $p(\mathbf{a})$ che abbiamo usato nell'estrarre le parole aleatorie del codice. Di questa variabile aleatoria possiamo calcolare il valore atteso $E_a\{P(E_m)\}$:

$$\mathbf{E}_{\mathbf{a}}\{P(E_m)\} = \mathbf{E}_{\mathbf{a}}\left\{\frac{1}{2^{K}}\sum_{\mathbf{a}_k \in \mathcal{A}} P(E_m | \mathbf{a} = \mathbf{a}_k)\right\} = \frac{1}{2^{K}}\sum_{\mathbf{a} \in \mathcal{A}} \mathbf{E}_{\mathbf{a}}\left\{P(E_m | \mathbf{a})\right\}$$
(B.6)

Una volta effettuata la media E_a , tutte le $E_a \{P(E_m | \mathbf{a})\}$ sono identiche e quindi $E_a \{P(E_m)\} = E_a \{P(E_m | \mathbf{a})\}$ per qualunque \mathbf{a} .

Calcoliamo ora la media sui codici della probabilità di errore sulla singola parola **a** come richiesto dalla (B.6):

$$E_{\mathbf{a}}\{P(E_m|\mathbf{a})\} = \sum_{\mathbf{a}} p(\mathbf{a}) \cdot P(E_m|\mathbf{a})$$
(B.7)

dove $p(\mathbf{a})$ è la probabilità che **a** sia una parola di codice, valutata su tutto l'insieme dei codici casuali. Questa media *non* è la stessa media su \mathbf{a}_k che compare nella (B.5): là la media era su tutte le 2^K parole equiprobabili di *un* dato codice, qui la media è su tutte le 2^N possibili parole **a** di lunghezza N che appartengono ad un codice con una certa probabilità. Per questo, anche l'espressione della $P(E_m | \mathbf{a})$ è differente rispetto alla $P(E_m | \mathbf{a}_k)$:

$$P(E_m|\mathbf{a}) = \sum_{\mathbf{d}} p(\mathbf{d}|\mathbf{a}) \Pr\{\mathbf{d} \notin \mathcal{B}(\mathbf{a})|\mathbf{a}\}$$
(B.8)

Là nella (B.5) le parole di uscita dal canale **d** "scorrono" al di fuori della ben definita zona di decisione \mathcal{B}_k della parola di codice \mathbf{a}_k ; qui, **d** scorre su tutto lo spazio 2^N e per ogni punto si deve pesare la $p(\mathbf{d}|\mathbf{a})$ con la probabilità che quel punto, nella scelta aleatoria del codice, non stia nella zona di decisione di **a**, indicata con $\mathcal{B}(\mathbf{a})$ (cioè, così come **a** ha una certa probabilità di far parte del codice, anche ogni **d** ha una certa probabilità di (non) stare nella zona di decisione di quel certo **a**).

Inserendo la (B.8) nella (B.7) otteniamo

$$\mathsf{E}_{\mathbf{a}}\{P(E_m|\mathbf{a})\} = \sum_{\mathbf{a}} p(\mathbf{a}) \sum_{\mathbf{d}} p(\mathbf{d}|\mathbf{a}) \Pr\{\mathbf{d} \notin \mathcal{B}(\mathbf{a})|\mathbf{a}\}$$

TEOREMA DELLA CODIFICA DI CANALE 473

$$= \sum_{\mathbf{a}} \sum_{\mathbf{d}} p(\mathbf{d}, \mathbf{a}) \Pr\{\mathbf{d} \notin \mathcal{B}(\mathbf{a}) | \mathbf{a}\}$$
(B.9)

Concentriamoci adesso sull'evento $\{d \not\in \mathcal{B}(a) | a\}.$ Questo evento è equivalente al seguente:

$$\mathcal{G}(\mathbf{d},\mathbf{a}) = \bigcup_{\mathbf{a}' \neq \mathbf{a}} \{ (\mathbf{d},\mathbf{a}') : p(\mathbf{d}|\mathbf{a}') > p(\mathbf{d}|\mathbf{a}) \}$$

infatti si commette un errore, dato **a**, per tutti i casi in cui $p(\mathbf{d}|\mathbf{a}') > p(\mathbf{d}|\mathbf{a})$. Maggioriamo adesso la $\Pr{\mathcal{G}(\mathbf{d}, \mathbf{a})}$ attraverso un bound d'unione:

$$\Pr\left\{\bigcup_{\mathbf{a}'\neq\mathbf{a}} \{(\mathbf{d},\mathbf{a}'): p(\mathbf{d}|\mathbf{a}') > p(\mathbf{d}|\mathbf{a})\}\right\} \le \sum_{\mathbf{a}'\neq\mathbf{a}} \Pr\{(\mathbf{d},\mathbf{a}'): p(\mathbf{d}|\mathbf{a}') > p(\mathbf{d}|\mathbf{a})\}$$
$$= (2^{K}-1) \Pr\{(\mathbf{d},\mathbf{a}'): p(\mathbf{d}|\mathbf{a}') > p(\mathbf{d}|\mathbf{a})\} < 2^{K} \cdot \sum_{\mathbf{a}'\in\mathcal{G}(\mathbf{d},\mathbf{a})} p(\mathbf{a}')$$
(B.10)

avendo sfruttato il fatto che le $Pr\{(\mathbf{d}, \mathbf{a}') : p(\mathbf{d}|\mathbf{a}') > p(\mathbf{d}|\mathbf{a})\}$, al variare di \mathbf{a}' , sono identiche tra loro, visto che non c'è un punto di riferimento particolare nella costruzione del codice. Riassumendo,

$$\Pr\{\mathbf{d} \notin \mathcal{B}(\mathbf{a}) | \mathbf{a}\} < 2^{K} \cdot \sum_{\mathbf{a}' \in \mathcal{G}(\mathbf{d}, \mathbf{a})} p(\mathbf{a}')$$
(B.11)

che useremo tra breve.

Il vero cuore della dimostrazione sta adesso nella ripartizione delle *coppie* di parole (\mathbf{d}, \mathbf{a}) che si trovano rispettivamente all'uscita e all'ingresso del canale rumoroso in due insiemi disgiunti: le coppie cosiddette *compatibili* e le altre. Le coppie compatibili sono quelle per le quali

$$p(\mathbf{d}, \mathbf{a}) < p(\mathbf{d})p(\mathbf{a}) \cdot 2^{N(r+c)/2}$$
(B.12)

o anche

$$\frac{p(\mathbf{a}|\mathbf{d})}{p(\mathbf{a})} < 2^{N(r+c)/2} \tag{B.13}$$

Qual è il significato di questa relazione? Dalla (B.12) abbiamo

$$\log\left(\frac{1}{p(\mathbf{d},\mathbf{a})}\right) > \log\left(\frac{1}{p(\mathbf{d})}\right) + \log\left(\frac{1}{p(\mathbf{a})}\right) - N\frac{r+c}{2}$$
(B.14)

ovvero, calcolando i valori medi e tenendo conto che i singoli simboli di d e di a sono indipendenti,

$$NH(D,A) < NH(D) + NH(A) - N\frac{r+c}{2}$$
 (B.15)

cioè, richiamando la (4.13)

$$I(D,A) < \frac{r+c}{2} \tag{B.16}$$

Dunque le due sequenze sono compatibili, se r < c, quando l'informazione mutua non è superiore alla capacità del canale.

Indichiamo con \mathcal{P} l'insieme delle coppie di sequenze compatibili e torniamo al calcolo della probabilità di errore media: $E_a\{P(E_m|a)\}$:

$$\mathbf{E}_{\mathbf{a}}\{P(E_{m}|\mathbf{a})\} = \sum_{\mathbf{d},\mathbf{a}\in\mathcal{P}} \Pr\{\mathbf{d}\notin\mathcal{B}(\mathbf{a})|\mathbf{a}\}p(\mathbf{d},\mathbf{a}) + \sum_{\mathbf{d},\mathbf{a}\notin\mathcal{P}} \Pr\{\mathbf{d}\notin\mathcal{B}(\mathbf{a})|\mathbf{a}\}p(\mathbf{d},\mathbf{a})$$

474 TEOREMA DELLA CODIFICA DI CANALE

$$<\sum_{\mathbf{d},\mathbf{a}\in\mathcal{P}} p(\mathbf{d},\mathbf{a}) + \sum_{\mathbf{d},\mathbf{a}\notin\mathcal{P}} p(\mathbf{d},\mathbf{a}) \cdot 2^{K} \cdot \sum_{\mathbf{a}'\in\mathcal{G}(\mathbf{d},\mathbf{a})} p(\mathbf{a}')$$
(B.17)

dove abbiamo utilizzato la (B.11).

Studiamo il *secondo* termine nella (B.17), ripartendo dalla (B.11). Se $\mathbf{a}' \in \mathcal{G}(\mathbf{d}, \mathbf{a})$, per definizione $p(\mathbf{d}|\mathbf{a}') > p(\mathbf{d}|\mathbf{a})$ e quindi (applicando la regola di Bayes)

$$\frac{p(\mathbf{a}'|\mathbf{d})p(\mathbf{d})}{p(\mathbf{a}')} > \frac{p(\mathbf{a}|\mathbf{d})p(\mathbf{d})}{p(\mathbf{a})}$$
(B.18)

ovvero, usando la (B.13) e sapendo che (\mathbf{d}, \mathbf{a}) $\notin \mathcal{P}$,

$$\frac{p(\mathbf{a}'|\mathbf{d})}{p(\mathbf{a}')} > 2^{\frac{N(r+c)}{2}} \Rightarrow p(\mathbf{a}') < 2^{-\frac{N(r+c)}{2}} p(\mathbf{a}'|\mathbf{d})$$
(B.19)

allora

$$2^{K} \cdot \sum_{\mathbf{a}' \in \mathcal{G}(\mathbf{d}, \mathbf{a})} p(\mathbf{a}') < 2^{(K - \frac{N(r+c))}{2}} \sum_{\mathbf{a}' \in \mathcal{G}(\mathbf{d}, \mathbf{a})} p(\mathbf{a}' | \mathbf{d}) \le 2^{-\frac{N(c-r)}{2}}$$
(B.20)

avendo ricordato che K = rN. Utilizziamo questo risultato nel secondo termine della (B.17):

$$\sum_{\mathbf{d},\mathbf{a}\notin\mathcal{P}} \sum_{\mathbf{d},\mathbf{a}\notin\mathcal{P}} p(\mathbf{d},\mathbf{a}) \cdot 2^{K} \cdot \sum_{\mathbf{a}'\in\mathcal{G}(\mathbf{d},\mathbf{a})} p(\mathbf{a}') < \sum_{\mathbf{d},\mathbf{a}\notin\mathcal{P}} \sum_{\mathbf{d},\mathbf{a}\notin\mathcal{P}} p(\mathbf{d},\mathbf{a}) 2^{-\frac{N(c-r)}{2}} \le 2^{-\frac{N(c-r)}{2}}$$
(B.21)

Se r < c, chiaramente questo termine tende a 0 quando $N \rightarrow \infty$, cioè quando i blocchi di codice diventano grandi.

Vediamo adesso il *primo* termine della (B.17). Questo rappresenta per definizione la probabilità che le sequenze siano compatibili, cioè

$$\sum_{\mathbf{d},\mathbf{a}\in\mathcal{P}} p(\mathbf{d},\mathbf{a}) = \Pr\left\{\frac{p(\mathbf{a}|\mathbf{d})}{p(\mathbf{a})} < 2^{N(r+c)/2}\right\}$$
$$= \Pr\left\{\frac{1}{N}\log\frac{1}{p(\mathbf{a})} - \frac{1}{N}\log\frac{1}{p(\mathbf{a}|\mathbf{d})} \le \frac{r+c}{2}\right\}$$
$$= \Pr\left\{\frac{1}{N}\sum_{n=0}^{N-1}\log\frac{1}{p(a_n)} - \frac{1}{N}\sum_{n=0}^{N-1}\log\frac{1}{p(a_n|d_n)} \le \frac{r+c}{2}\right\}$$
$$\stackrel{\triangle}{=} \Pr\left\{\frac{1}{N}\sum_{n=0}^{N-1}(X_{1,n} - X_{2,n}) \le \frac{r+c}{2}\right\}$$
(B.22)

dove $X_{1,n}$ e $X_{2,n}$, n = 0, ..., N - 1 sono N variabili aleatorie con varianza finita e con valore atteso

$$\mathsf{E}\{X_{1,n}\} = \mathsf{E}\left\{\log\frac{1}{p(a_n)}\right\} = H(A) \ , \ \mathsf{E}\{X_{2,n}\} = \mathsf{E}\left\{\log\frac{1}{p(a_n|d_n)}\right\} = H(A|D)$$

e quindi $E{X_{1,n} - X_{2,n}} = I(A, D)$. Tenendo però conto che le parole del codice sono estratte seguendo la distribuzione *capacity-achieving*, segue anche che I(A, D) = c. Se r < c, in generale possiamo poi porre $(c + r)/2 = c - \varepsilon$, $\varepsilon > 0$ e la (B.22) diventa

$$\sum_{\mathbf{d},\mathbf{a}\in\mathcal{P}} p(\mathbf{d},\mathbf{a}) = \Pr\left\{\frac{1}{N}\sum_{n=0}^{N-1} (X_{1,n} - X_{2,n}) - c \le -\varepsilon\right\}$$
(B.23)

D'altronde

$$\Pr\left\{\frac{1}{N}\sum_{n=0}^{N-1}(X_{1,n} - X_{2,n}) - c \le -\varepsilon\right\} \le \Pr\left\{\left|\frac{1}{N}\sum_{n=0}^{N-1}(X_{1,n} - X_{2,n}) - c\right| \ge \varepsilon\right\}$$
(B.24)

e quest'ultima probabilità tende a 0 quando $N \rightarrow \infty$ per la legge dei grandi numeri (la media campionaria tende alla media statistica con probabilità 1):

$$\frac{1}{N} \sum_{n=0}^{N-1} (X_{1,n} - X_{2,n}) \xrightarrow{N \to \infty} I(A, D) = c$$
(B.25)

A questo punto resta dimostrato che la $E_a\{P(E_m)\} = E_a\{P(E_m|\mathbf{a})\}$, cioè la probabilità totale di errore di un codice aleatorio, tende a zero quando la dimensione dei blocchi di codice diventa (molto) grande - un codice random con parole molto lunghe si può considerare "mediamente buono". Se il valore atteso tende a zero, dovrà esistere *almeno un codice* la cui probabilità di errore, quando r < c può essere resa arbitrariamente vicina a 0 (cioè piccola a piacere) ingrandendo i blocchi - CVD.



APPENDIX C L'ALGORITMO BCJR (MODIFICATO)

Vogliamo sviluppare un algoritmo per calcolare i LAPPRs

$$L(b[k]) \stackrel{\triangle}{=} \ln \frac{\Pr\left\{b[k] = 1 | \mathbf{r}\right\}}{\Pr\left\{b[k] = 0 | \mathbf{r}\right\}}$$
(C.1)

per un blocco di K bit di informazione $\mathbf{b} = [b[0], b[1], ..., b[K-1]]$ codificati da un codificatore convoluzionale (eventualmente ricorsivo) a tasso 1/2, osservando un vettore soft ottenuto inviando i simboli codificati ri-mappati sui valori di canale -1 e +1 all'uscita di un canale AWGN. Dalla definizione di probabiloità condizionata otteniamo

$$\Pr\{b[k] = 1 | \mathbf{r}\} = f(b[k] = 1, \mathbf{r}) / f(\mathbf{r})$$
(C.2)

dove $f(\cdot)$ indica un densità di probabilità (pdf) congiunta o marginale (non indichiamo per semplicità i pedici di riferimento). Con gli stessi passaggi per il denominatore di (C.1) abbiamo

$$L(b[k]) = \ln \frac{f(b[k] = 1, \mathbf{r})}{f(b[k] = 0, \mathbf{r})}$$
(C.3)

Suddividiamo adesso l'*evento* $\{b[k] = 1(0)\}$ nell'unione di tutti gli eventi disgiunti dati da tutte le transizioni nel traliccio del codice che portano da un certo stato *i* al tempo k - 1 allo stato *j* al tempo *k*, e che sono comunque causate (sono "etichettate") da b[k] = 1(0). Con questa scomposizione, il LAPPR diventa:

Comunicazioni Digitali, I Edizione.

di M. Luise Copyright © 2023 University of Pisa





Figura C.1 Transizioni di stato nel traliccio corrispondenti a b[k] = 1 (a) e b[k] = 0 (b)

$$L(b[k]) = \ln \frac{\sum_{(i,j):b[k]=1} f\left(\xi[k-1]=i,\xi[k]=j,\mathbf{r}\right)}{\sum_{(i,j):b[k]=0} f\left(\xi[k-1]=i,\xi[k]=j,\mathbf{r}\right)} \triangleq \ln \frac{\sum_{(i,j):b[k]=1} f\left(i,j,\mathbf{r}\right)}{\sum_{(i,j):b[k]=0} f\left(i,j,\mathbf{r}\right)}$$
(C.4)

dove $\xi[k]$ rappresenta lo stato del decodificatore al tempo k. Prendendo come sempre il codificatore di Fig. 6.6 come caso di studio, l'insieme di transizioni nella somma al numeratore di (C.4) è mostrato in Fig. C (a), e quello coinvolto nel denominatore è in Fig. C (b). Il risultato fondamentale è dimostrare che le due pdf congiunte in (C.4) (che indicheremo per brevità con $f(i, j, \mathbf{r})$) possono essere calcolate con le due ricorsioni avanti-indietro che abbiamo menzionato nel testo principale.

In primis, usiamo la seguente notazione evidente (ricordiamo che r=1/2):

$$\mathbf{r} = \left[\overbrace{r[0], r[1], ..., r[2k-2], r[2k-1]}^{\mathbf{r}^{-}}, \overbrace{r[2k], r[2k+1]}^{\mathbf{r}_{k}}, \overbrace{r[2k+2], r[2k+3], ..., r[2N-2], r[2N-1]}^{\mathbf{r}^{+}}\right]$$
(C.5)

Poi, osserviamo che

$$f(i, j, \mathbf{r}) = f(i, j, \mathbf{r}^{-}, \mathbf{r}_{k}, \mathbf{r}^{+})$$

$$= f(\mathbf{r}^{+}|i, j, \mathbf{r}^{-}, \mathbf{r}_{k}) f(i, j, \mathbf{r}^{-}, \mathbf{r}_{k})$$

$$= f(\mathbf{r}^{+}|i, j, \mathbf{r}^{-}, \mathbf{r}_{k}) f(j, \mathbf{r}_{k}|i, \mathbf{r}^{-}) f(i, \mathbf{r}^{-})$$
(C.6)

Nel primo termine dell'ultima equazione (C.6), una volta che lo stato all'istante k è fissato al valore j, allora i valori "futuri" \mathbf{r}^+ risultano (condizionatamente) indipendente da tutto

ciò che è accaduto fino a k. Analogamente, nel secondo termine una volta che lo stato i al tempo k - 1 viene fissato, il "passato" \mathbf{r}^- è irrilevante. Quindi,

$$f(i,j,\mathbf{r}) = f(\mathbf{r}^{+}|j) f(j,\mathbf{r}_{k}|i) f(i,\mathbf{r}^{-})$$

$$\stackrel{\triangle}{=} \beta^{(j)}[k+1] \cdot \Gamma^{(j,i)}[k] \cdot \alpha^{(i)}[k-1]$$
(C.7)

cosicché

$$L(b[k]) = \ln \frac{\sum_{(i,j):b[k]=1} \beta^{(j)}[k+1] \cdot \Gamma^{(j,i)}[k] \cdot \alpha^{(i)}[k-1]}{\sum_{(i,j):b[k]=0} \beta^{(j)}[k+1] \cdot \Gamma^{(j,i)}[k] \cdot \alpha^{(i)}[k-1]}$$
(C.8)

Il risultato fondamentale e l'equazione ricorsiva per il calcolo di $\alpha^{(i)}[k-1] e \beta^{(j)}[k+1]$. Vediamo prima però come calcolare $\Gamma^{(j,i)}[k]$:

$$\Gamma^{(j,i)}[k] = f(j, \mathbf{r}_k|i) = f(i, j, \mathbf{r}_k) / p(i)$$

$$= \frac{f(i, j, \mathbf{r}_k)}{p(i, j)} \frac{p(i, j)}{p(i)}$$

$$= f(\mathbf{r}_k|i, j) p(j|i)$$
(C.9)

Condizionare ad i e j nel primo termine dell'ultima equazione qui di sopra significa sapere individuare i particolari valori di $a_0[k] e a_1[k]$, cosicché $f(\mathbf{r}_k|i, j) = f(\mathbf{r}_k|a_0[k], a_1[k])$; inoltre, p(j|i) è la probabilità della transizione assegnata, che vale 0 se la transizione non è compresa nel traliccio del codice, oppure è la probabilità che b[k] = 1 o b[k] = 0 se la transizione è permessa (e cioè vale 1/2 se i bit sono equiprobabili). Quindi, sui rami del traliccio (transizioni permesse),

$$\Gamma^{(j,i)}[k] = \frac{1}{2} f(\mathbf{r}_k | a_0[k], a_1[k])$$

$$= \frac{1}{4\pi\sigma^2} \exp\left[-\frac{(r[2k] - a_0[k])^2 + (r[2k+1] - a_1[k])^2}{2\sigma^2}\right]$$
(C.10)

dove $\sigma^2 = (2E_s/N_0)^{-1}$ è la varianza della componente di AWGN che disturba il simbolo binario codificato. Da qui in poi useremo una versione *normalizzata* di Γ che porta agli stessi risultati ed è più semplice da calcolare¹

$$\gamma^{(j,i)}[k] \stackrel{\triangle}{=} \exp\left[-\frac{(r[2k] - a_0[k])^2 + (r[2k+1] - a_1[k])^2}{2\sigma^2}\right]$$
(C.11)

Vediamo che γ è simile a una metrica di ramo del DV, e può essere calcolata facilmente sul traliccio a partire dai valori soft ricevuti **r** e conoscendo la varianza di rumore.

Torniamo al calcolo di $\alpha^{(i)}[k-1]$. Considerando un passo in più, cioè andando verso lo stato j al tempo k, abbiamo:

$$\alpha^{(j)}[k] = f(j, \mathbf{r}^{-}, \mathbf{r}_{k})$$

$$= \sum_{i} f(i, j, \mathbf{r}^{-}, \mathbf{r}_{k})$$

$$= \sum_{i} f(j, \mathbf{r}_{k} | i, \mathbf{r}^{-}) f(i, \mathbf{r}^{-})$$

$$= \sum_{i} \Gamma^{(j,i)}[k] \cdot \alpha^{(i)}[k-1] \qquad (C.12)$$

¹Si noti che qualunque costante di normalizzazione scompare nel calcolo del rapporto (C.8)

480 L'ALGORITMO BCJR (MODIFICATO)

che è un'equazione ricorsiva simile a quella per la metrica accumulata del DV (6.47). Da qui in poi, useremo la versione normalizzata

$$\alpha^{(j)}[k] = \sum_{i} \gamma^{(j,i)}[k] \cdot \alpha^{(i)}[k-1]$$
(C.13)

Nella (C.13), i $\gamma^{(j,i)}[k]$ hanno il ruolo delle metriche di ramo, e i valori α che otteniamo al tempo k sono uno per ogni stato nel traliccio, come nel DV.

Un trucco simile ma un po' più raffinato porta al calcolo di $\beta^{(j)}[k+1]$. Se ci muoviamo *indietro nel tempo* di un passo abbiamo

$$\beta^{(i)}[k] = f(\mathbf{r}_{k}, \mathbf{r}^{+}|i)$$

$$= \sum_{j} f(\mathbf{r}_{k}, \mathbf{r}^{+}, j|i)$$

$$= \sum_{j} f(\mathbf{r}^{+}|\mathbf{r}_{k}, i, j) f(j, \mathbf{r}_{k}|i)$$

$$= \sum_{j} f(\mathbf{r}^{+}|j) \cdot \Gamma^{(j,i)}[k] \qquad (C.14)$$

dove abbiamo considerato che assegnare lo stato j al tempo k significa cancellare la dipendenza da \mathbf{r}_k, i . Normalizzando,

$$\beta^{(i)}[k] = \sum_{j} \beta^{(j)}[k+1] \cdot \gamma^{(j,i)}[k]$$
(C.15)

Abbiamo ottenuto una equazione ricorsiva *retrograda*, cioè che procede all'indietro nel tempo con una metrica accumulata e (di nuovo) metrica di ramo $\gamma^{(j,i)}[k]$. Ecco perché nel testo abbiamo parlato dell'implementazione del BCJR attraverso due ricorsioni avantiindietro simili al DV. Dopo aver ricavato γ , α , e β per tutti gli istanti k del blocco di N bit di informazione seguendo rispettivamente (C.10), (C.13), and (C.15), si trovano immediatamente i LAPPRs attraverso (C.8). Ovviamente questo tipo di elaborazione, diversamente dal DV, non è causale e procede in tempo virtuale perché necessita della disponibilità di tutto il blocco di campioni ricevuti.

L'algoritmo appena ricavato deve però essere *modificato* per due motivi. In primis, andando a implementarlo così come appena descritto, si trova che esso è *mal condizionato* dal punto di vista numerico, e che migliora notevolmente considerando in tutta la derivazione *logaritmi* delle probabilità e/o log-rapporti di probabilità. Inoltre, per poter realizzare le iterazioni del turbo-decodificatore dobbiamo i) enucleare, nel calcolo del LAPPR relativo a un bit di informazione, l'eventuale *informazione estrinseca* sui bit di informazione che deve rappresentare un *ingresso aggiuntivo* del BCJR proveniente dal BCJR gemello (in generale, del decodificatore SISO gemello), nonché ii) individuare nel LAPPR calcolato dall'algoritmo quella componente che il SISO gemello considererà a sua volta come estrinseca.

Dunque ponendo $\overline{\gamma} \stackrel{\triangle}{=} \ln \gamma$, $\overline{\alpha} \stackrel{\triangle}{=} \ln \alpha$, $\overline{\beta} \stackrel{\triangle}{=} \ln \beta$, e $L[k] \stackrel{\triangle}{=} L(b[k])$, troviamo immediatamente

$$L[k] = \ln \sum_{(i,j):b[k]=1} \exp\left(\overline{\beta}^{(j)}[k+1] + \overline{\gamma}^{(j,i)}[k] + \overline{\alpha}^{(i)}[k-1]\right) - \ln \sum_{(i,j):b[k]=0} \exp\left(\overline{\beta}^{(j)}[k+1] + \overline{\gamma}^{(j,i)}[k] + \overline{\alpha}^{(i)}[k-1]\right)$$
(C.16)

Cerchiamo poi di capire come iniettare dentro l'algoritmo un'informazione estrinseca nella forma di un LAPPR *a-priori* (che proverrà ovviamente del SISO decoder gemello del turbo-decodificatore). Usando la notazione abbreviata $\mathbf{a}[k] = [a[2k], a[2+k1]]$ e $\mathbf{r}[k] = [r[2k], r[2+k1]]$, allora²

$$\gamma^{(j,i)}[k] = p(b[k]) \cdot f(\mathbf{r}_k | \mathbf{a}[k])$$

$$= p(b[k]) \exp\left[-\frac{\|\mathbf{r}[k] - \mathbf{a}[k]\|^2}{2\sigma^2}\right]$$
(C.17)

dove p(b[k]) è appunto l'informazione estrinseca (a-priori) e non è (più) il banale valore 1/2 della /C.10). In luogo delle probabilità, è comodo accettare questa informazione a-priori nella forma di un log-rapporto di probabilità (proveniente dall'esterno):

$$L_e[k] \stackrel{\triangle}{=} \ln \frac{\Pr\{b[k] = 1\}}{\Pr\{b[k] = 0\}} = \ln \frac{p_1}{p_0} = \ln \frac{p_1}{1 - p_1} = \ln \frac{1 - p_0}{p_0}$$
(C.18)

da cui, semplificando ancora la notazione,

.

$$p_1 = \frac{\exp(L^e/2)}{2\cosh(L^e/2)}$$
, $p_0 = \frac{\exp(-L^e/2)}{2\cosh(L^e/2)}$ (C.19)

I due valori in (C.19) sono dunque:

$$p(b[k]) = \frac{\exp(b_{\pm}[k]L_e[k]/2)}{2\cosh(L_e[k]/2)} = A(L_e[k])\exp\left(\frac{b_{\pm}[k] \cdot L_e[k]}{2}\right)$$
(C.20)

dove abbiamo introdotto il valore ri-mappato simmetrico del bit di informazione b[k], cioè $b_{\pm}[k] = 2b[k] - 1$ e dove $A(L_e[k]) = (2\cosh(L_e[k]/2))^{-1}$ non dipende dal bit. Sappiamo adesso come incorporare l'informazione estrinseca nell'algoritmo - dobbiamo solo modificare il calcolo della metriche di ramo γ come segue:

$$\gamma^{(j,i)}[k] = A(L_e[k]) \exp\left(\frac{b_{\pm}[k] \cdot L_e[k]}{2}\right) \exp\left[-\frac{\|\mathbf{r}[k] - \mathbf{a}[k]\|^2}{2\sigma^2}\right]$$
(C.21)

cosicché

$$\overline{\gamma}^{(j,i)}[k] = \ln\left[A(L_e[k])\right] + \frac{b_{\pm}[k] \cdot L_e[k]}{2} - \frac{\|\mathbf{r}[k] - \mathbf{a}[k]\|^2}{2\sigma^2}$$
(C.22)

Adesso, dobbiamo ri-derivare le ricorsioni per $\overline{\alpha} \in \overline{\gamma}$ nel dominio logaritmico per ricavare L(b[k]), inclusa l'informazione estrinseca, e dobbiamo anche enucleare entro L(b[k]) il contributo incrementale che viene apportato dall'esecuzione dell'algoritmo piuttosto che dall'informazione estrinseca d'ingresso. Tale contributo diventerà l'informazione estrinseca di uscita che sarà consegnata al decodificatore gemello.

Considerando (C.16), non vediamo particolari semplificazioni rispetto alla sua controparte non logaritmica (C.8), specialmente perchè si devono calcolare termini derivanti da operazioni nonlineari del tipo $\ln [\exp(a) + \exp(b)]$. In realtà, la semplificazione che cerchiamo (e che rende l'algoritmo robusto) sta proprio qui, e parte dalla seguente identità algebrica:

$$\ln [\exp(a) + \exp(b)] = \max(a, b) + \ln [1 + \exp(-|a - b|)]$$
(C.23)

²Ricordiamo che in tutta l'appendice abbiamo supposto che i simboli codificati a[n] siano binari simmetrici $\in \{\pm 1\}$.

482 L'ALGORITMO BCJR (MODIFICATO)

Da qui possiamo definire una nuova funzione di due variabili simile al max(a, b):

$$\max^*(a,b) \stackrel{\triangle}{=} \ln\left[\exp(a) + \exp(b)\right] = \max(a,b) + \ln\left[1 + e^{-|a-b|}\right] \tag{C.24}$$

che può essere generalizzata in modo ricorsivo a più variabili, godendo anche della proprietà *associativa*:

$$\max^*(a, b, c) \equiv \max^*(\max^*(a, b), c)$$
$$= \ln \left[\exp(a) + \exp(b) + \exp(c)\right]$$
(C.25)

Il max^{*} gode anche della proprietà *distributiva* rispetto alla somma (esercizio per il lettore), proprietà che useremo in seguito:

$$\max^{*}(z+a, z+b) = z + \max^{*}(a, b)$$
(C.26)

Per indicare la funzione max* di molte variabili, useremo la notazione compatta

$$\max_{i=1}^{N} {}^{*}(a_i) \stackrel{\triangle}{=} \max^{*}(a_1, a_2, \dots, a_N)$$
$$= \ln \left[\exp(a_1) + \exp(a_2) + \dots + \exp(a_N) \right]$$
(C.27)

Adesso torniamo alla (C.13) che dice

$$\alpha^{(j)}[k] = \sum_{i} \gamma^{(j,i)}[k] \cdot \alpha^{(i)}[k-1]$$

cosicché

$$\overline{\alpha}^{(j)}[k] = \ln\left(\sum_{i} \gamma^{(j,i)}[k] \cdot \alpha^{(i)}[k-1]\right)$$
$$= \max_{i}^{*} \left(\overline{\gamma}^{(j,i)}[k] + \overline{\alpha}^{(i)}[k-1]\right)$$
(C.28)

Analogamente, da (C.15),

$$\overline{\beta}^{(i)}[k] = \max_{j}^{*} \left(\overline{\beta}^{(j)}[k+1] + \overline{\gamma}^{(j,i)}[k] \right)$$
(C.29)

in modo che (C.16) diventa

$$L[k] = \max_{b[k]=1} {}^{*} \left(\overline{\beta}^{(j)}[k+1] + \overline{\gamma}^{(j,i)}[k] + \overline{\alpha}^{(i)}[k-1] \right) - \max_{b[k]=0} {}^{*} \left(\overline{\beta}^{(j)}[k+1] + \overline{\gamma}^{(j,i)}[k] + \overline{\alpha}^{(i)}[k-1] \right)$$
(C.30)

Dopo questi sviluppi, dobbiamo ancora enucleare in L[k] (C.30) il contributo *estrinseco* da inviare al decodificatore gemello - non sarà facile.

Ricominciamo da (C.22) e usiamo una versione leggermente modificata di $\overline{\gamma}^{(j,i)}[k]$:

$$\overline{\gamma}^{(j,i)}[k] = \ln\left[A(L_e[k])\right] + \frac{b_{\pm}[k] \cdot L_e[k]}{2} - \frac{\|\mathbf{r}[k]\|^2 + \|\mathbf{a}[k]\|^2 - 2\mathbf{r}[k] \cdot \mathbf{a}[k]}{2\sigma^2} \quad (C.31)$$

Quando utilizziamo (C.31) nella (C.30), tutti i termini che non dipendono dalla particolare transizione (i, j) si cancellano nelle differenza tra le due funzioni max^{*} (ricordiamo la

proprietà distributiva (C.26)), ad esempio il termine $\ln[A(L_e[k])]$, ma anche $\|\mathbf{r}[k]\|^2/2\sigma^2$ e $\|\mathbf{a}[k]\|^2/2\sigma^2$. Ciò che rimane e che effettivamente useermo nella (C.30) è $b_{\pm}[k] \cdot L_e[k]/2$ ed anche $r[2k]a_1[k]/\sigma^2 + r[2k+1]a_2[k]/\sigma^2$.

Infine, per un codice sistematico a tasso 1/2 come quelli del codificatore turbo PCCC, $a_1[k] = b_{\pm}[k]$, e $a_2[k] = p_{\pm}[k]$ (simboli simmetrici), cosicché

$$L[k] = \max_{b_{\pm}[k]=1} * \left(\overline{\beta}^{(j)}[k+1] + \frac{L_{e}[k] \cdot b_{\pm}[k]}{2} + \frac{r[2k]b_{\pm}[k]}{\sigma^{2}} + \frac{r[2k+1]p_{\pm}[k]}{\sigma^{2}} + \overline{\alpha}^{(i)}[k-1]\right) \\ - \max_{b_{\pm}[k]=-1} * \left(\overline{\beta}^{(j)}[k+1] + \frac{L_{e}[k] \cdot b_{\pm}[k]}{2} + \frac{r[2k]b_{\pm}[k]}{\sigma^{2}} + \frac{r[2k+1]p_{\pm}[k]}{\sigma^{2}} + \overline{\alpha}^{(i)}[k-1]\right) \\ = L_{e}[k] + 2\frac{r[2k]}{\sigma^{2}} + \max_{b_{\pm}[k]=1} * \left(\overline{\beta}^{(j)}[k+1] + \frac{r[2k+1]p_{\pm}[k]}{\sigma^{2}} + \overline{\alpha}^{(i)}[k-1]\right) \\ - \max_{b_{\pm}[k]=-1} * \left(\overline{\beta}^{(j)}[k+1] + \frac{r[2k+1]p_{\pm}[k]}{\sigma^{2}} + \overline{\alpha}^{(i)}[k-1]\right)$$
(C.32)

Considerando poi che r è l'uscita di un canale AWGN con ricezione a filtro adattato e privo di ISI, sappiamo che $\sigma^2 = N_0/(2E_s)$ (2.111). In conclusione,

$$L[k] = L_e[k] + \frac{4E_s}{N_0}r[2k] + \left\{ \max_{b[k]=1}^{*} \left(\overline{\beta}^{(j)}[k+1] + \frac{2E_s}{N_0}r[2k+1]p_{\pm}[k] + \overline{\alpha}^{(i)}[k-1] \right) - \max_{b[k]=0}^{*} \left(\overline{\beta}^{(j)}[k+1] + \frac{2E_s}{N_0}r[2k+1]p_{\pm}[k] + \overline{\alpha}^{(i)}[k-1] \right) \right\}$$
(C.33)

Commentiamo questa equazione che è proprio ciò che volevamo ricavare. Il LAPPR del bit di informazione b[k] è composto di *tre* termini: il primo, $(L_e[k])$, è la parte *estrinseca*, cioè a-priori, che proviene dall'esterno; il secondo termine $r[2k] \cdot 4E_s/N_0$ é direttamente collegato all'input proveniente dal canale rumoroso relativo al bit di informazione senz'alcuna codifica (ricordiamo che $a[2k] = b_{\pm}[k]$); il terzo termine tra parentesi graffe è ciò che viene calcolato dal BCJR modificato, cioè il decodificatore SISO adatto al turbo-codice, sulla base dei bit di parità rumorosi r[2k+1] = p[k] + w[2k+1]. Questo termine rappresenta proprio la parte che sarà consegnata al decoder gemello come ad esso *estrinseca*, perché calcolata sulla base di bit di parità che l'altro decoder *non* ha!

Con questa interpretazione, possiamo esprimere l'equazione finale (C.33) nella forma semplice e autoesplicativa seguente:

$$L[k] = L_{in}[k] + L_{ch}[k] + L_{out}[k]$$
(C.34)

dove $L_{ch}[k]$ è il termine di canale che viene calcolato una volta per tutte e non dipende dalle iterazioni di decodifica; $L_{in}[k]$ è il termine ricevuto come input dal decoder gemello e che non viene calcolato ma oslo aggiunto; $L_{out}[k]$ è il termine effettivamente calcolato dal decoder e che verrà inoltrato come informazione estrinseca al decoder gemello.



Figura C.2 Schema accurato del turbo-decoder

L'effetto turbo della decodifica iterativa si sviluppa attraverso il fatto che $L_{out}[k]$ è funzione di $L_e[k]$ (ovvero di $L_{in}[k]$) anche attraverso il calcolo delle metriche di ramo $\overline{\gamma}^{(j,i)}[k]$ in (C.28) e (C.29). Sfruttando questa osservazione, possiamo riassumere con maggiore esattezza di quanto fatto nel testo ciò che il turbo decoder effettivamente fa: dobbiamo aggiungere due indici al LAPPR (C.34): il primo (1,2) è l'identificatore dei due SISO gemelli, il secondo (i) è il numero dell'iterazione. Dunque all'iterazione i, i = $1, \ldots, N_{it}$, il decoder 1 calcola

$$L_{out}^{(1,i)}\left([k]; L_{in}^{(1,i)}[k]\right)$$
(C.35)

usando l'informazione estrinseca di ingresso

$$L_{in}^{(1,i)}[k] \stackrel{\triangle}{=} L_{out}^{(2,i-1)}[k]$$

(C.36)

calcolata dal decoder 2 alla precedente iterazione; successivamente, il decoder 2 calcola

$$L_{out}^{(2,i)}\left([k]; L_{in}^{(2,i)}[k]\right)$$
$$L_{in}^{(2,i)}[k] \stackrel{\triangle}{=} L_{out}^{(1,i-1)}[k]$$
(C.37)

e così via (due semi-iterazioni). La decodifica inizia con $L_{out}^{(2,0)}[k] = 0$. Il termine di "canale" $L_{ch}[k]$ viene aggiunto a L_{in} e L_{out} del decoder 1 soltanto al termine della iterazioni cio) energia di la constanta della di la constanta della di constanta della di la constanta della di constanta della di la constanta della di constanta della di constanta della di constanta della di constanta di constanta di constanta di constanta di constanta della di constanta di cons termine delle iterazioni, cioè quando si deve prendere la decisione finale su $L^{(1,N_{it})}(b[k])$. La Fig. C mostra infine una versione più accurata dello schema di Fig. 6.18 (b) dove vengono chiaramente indicati i vari termini del calcolo del LAPPR.

APPENDIX D

L'ALGORITMO "MESSAGE-PASSING" PER DE-CODIFICARE I CODICI LDPC

Lo sviluppo dell'algoritmo di message-passing delineato nel testo parte da una proprietà interessante ricavata da Gallager negli anni '60, e che è rilevante per i codici a controllo di parità. Considerando una parola di codice $\mathbf{a} = [a[0], a[1], ..., a[N-1]]$ avente simboli a-posteriori indipendenti caratterizzati da probabilità (a posteriori) di esser uguali a 1 in generale differenti l'una dall'altra e uguali rispettivamente a $[p_0, p_1, ..., p_{N-1}]$ (i.e., $p_n \stackrel{\triangle}{=} \Pr{a[n] = 1|r[n]}$, la probabilità che nella parola vi sia un numero *pari* di simboli uguali a 1 è

$$\frac{1}{2} + \frac{1}{2} \prod_{n=0}^{N-1} (1 - 2p_n)$$
(D.1)

che si dimostra facilmente per induzione (esercizio per il lettore). Ovviamente, la probabilità che vi sia un numero *dispari* di simboli 1 è

$$\frac{1}{2} - \frac{1}{2} \prod_{n=0}^{N-1} (1 - 2p_n) \tag{D.2}$$

Chiaramente questi "lemmi" tornano utili quando si deve considerare la probabilità che un parity-check sia verificato o meno.

Osservando come di consueto l'uscita soft di un canale AWGN con ricezione a filtro adattato e assenza di ISI, $r[n] = a_{\pm}[n] + w[n]$ e assumendo valori equiprobabili per i simboli

Comunicazioni Digitali, I Edizione.

di M. Luise Copyright © 2023 University of Pisa

485

486 L'ALGORITMO "MESSAGE-PASSING" PER DECODIFICARE I CODICI LDPC

simmetrici $a_{\pm}[n]\in\{-1,+1\},$ e varianza $\sigma^2=(2E_s/N_0)^{-1}$ per w[n], le probabilità a-posteriori sono

$$p_{n} \stackrel{\triangle}{=} \Pr \left\{ a_{\pm}[n] = 1 | r[n] \right\} = \frac{f_{r}(r[n]|a_{\pm}[n] = 1) \Pr \left\{ a_{\pm}[n] = 1 \right\}}{f_{r}(r[n])}$$
$$= \frac{\exp[-(r[n] - 1)^{2}/2\sigma^{2}] \cdot 0.5}{0.5 \exp[-(r[n] - 1)^{2}/2\sigma^{2}] + 0.5 \exp[-(r[n] + 1)^{2}/2\sigma^{2}]}$$
$$= \frac{\exp(r[n]/\sigma^{2})}{\exp(r[n]/\sigma^{2}) + \exp(-r[n]/\sigma^{2})} = \frac{1}{1 + \exp(-2r[n]/\sigma^{2})}$$
$$= \frac{1}{1 + \exp(-4r[n]E_{s}/N_{0})}$$
(D.3)

tenendo conto che la pdf $f_r(x)$ di r[n] è la mistura di due Gaussiane equiprobabili con medie condizionate ± 1 e varianza σ^2 .

Vogliamo adesso calcolare un rapporto di probabilità a-posteriori (APPR) del tipo

$$\frac{\Pr\left\{a[n] = 1 | \mathbf{r}, \{c_i = 0\}_{i \in C_{i,n}}\right\}}{\Pr\left\{a[n] = 0 | \mathbf{r}, \{c_i = 0\}_{i \in C_{i,n}}\right\}}$$
(D.4)

Usando due volte di teorema di Bayes si ha

$$\frac{p_n \cdot \Pr\left\{ \{c_i = 0\}_{i \in C_{i,n}} \middle| a[n] = 1, \mathbf{r} \right\}}{(1 - p_n) \cdot \Pr\left\{ \{c_i = 0\}_{i \in C_{i,n}} \middle| a[n] = 0, \mathbf{r} \right\}}$$
(D.5)

Considerando il denominatore, se a[n] = 0, verificare l'equazione di parità *i*-esima significa che tra tutti gli altri bit coinvolti nell'equazione deve esserci un numero *pari* di 1. Possiamo calcolare la probabilità di questo evento modificando (D.1) in modo da considerare solo i bit interessati:

$$\frac{1}{2} + \frac{1}{2} \prod_{n', i \in R_{i \setminus n}} (1 - 2p_{n', i})$$
(D.6)

dove abbiamo indicato con $R_{i\setminus n}$ l'insieme delle posizioni degli elementi uguali a 1 nella riga *i*-esima della matrice di parity-check, esclusa eventualmente la *n*-esima, e dove l'indice n', i indica l' n'-esimo bit the compone la *i*-esima equazione di parità. Al numeratore, a[n] = 1 e facciamo un ragionamento simile, ma usando la (D.8) anziché la (D.1). Inoltre, poiché gli a[n] sono mutuamente indipendenti, la probabilità che *tutte* le equazioni di parità siano verificate è il prodotto di queste probabilità. Allora, combinando (D.5) con (D.6), otteniamo

$$\frac{\Pr\left\{a[n]=1|\mathbf{r}, \{c_i=0\}_{i\in C_{i,n}}\right\}}{\Pr\left\{a[n]=0|\mathbf{r}, \{c_i=0\}_{i\in C_{i,n}}\right\}} = \frac{\prod_{i\in C_{i,n}}\left(1-\prod_{n',i\in R_{i\setminus n}}(1-2p_{n',i})\right)}{\prod_{i\in C_{i,n}}\left(1+\prod_{n',i\in R_{i\setminus n}}(1-2p_{n',i})\right)}$$
(D.7)

Ricolleghiamo adesso questo APPR ai vari messaggi $\rho_{i,n}$ e $q_{n,i}$ che vengono scambiati sul grafo di Tanner. Ricordando (6.60), vediamo che

$$\rho_{i,n}(0) = \frac{1}{2} + \frac{1}{2} \prod_{n',i \in R_{i \setminus n}} (1 - 2p_{n',i})$$

L'ALGORITMO "MESSAGE-PASSING" PER DECODIFICARE I CODICI LDPC 487

$$\rho_{i,n}(1) = 1 - \rho_{i,n}(0) = \frac{1}{2} - \frac{1}{2} \prod_{n', i \in R_{i \setminus n}} (1 - 2p_{n',i})$$
(D.8)

e quindi

$$\frac{\Pr\left\{a[n]=1|\mathbf{r}, \{c_i=0\}_{i\in C_{i,n}}\right\}}{\Pr\left\{a[n]=0|\mathbf{r}, \{c_i=0\}_{i\in C_{i,n}}\right\}} = \frac{p_n \prod_{i\in C_{i,n}} \rho_{i,n}(1)}{(1-p_n) \prod_{i\in C_{i,n}} \rho_{i,n}(0)}$$
(D.9)

Ricordando (6.59), ci accorgiamo anche che

$$q_{n,i}(0) = (1 - p_n) \prod_{i' \in C_{i' \setminus i,n}} \rho_{i',n}(1)$$
$$q_{n,i}(1) = p_n \prod_{i' \in C_{i' \setminus i,n}} \rho_{i',n}(0)$$
(D.10)

Infine, osserviamo che in in (D.6) $p_{n',i} = q_{n',i}$ e quindi

$$\rho_{i,n}(0) = \frac{1}{2} + \frac{1}{2} \prod_{n', i \in R_{i \setminus n}} (1 - 2q_{n',i}(0))$$
(D.11)

Trovate le relazioni che legano i vari ρ e q, vediamo i vari passi che costituiscono l'algoritmo iterativo *messagge-passing*:

1. Inizializzazione:

Partire con il numero di iterazione m = 0 e $\forall i$ calcolare

$$q_{n,i}^{(0)}(1) = \frac{1}{1 + \exp(-4r[n]E_s/N_0)}$$

$$q_{n,i}^{(0)}(0) = \frac{1}{1 + \exp(4r[n]E_s/N_0)}$$
(D.12)

Questo é il valore delle probabilità a posteriori (D.3) iniziali quando i nodi di check non possono (ancora) fornire alcuna informazione;

2. Scambio dei messaggi:

Aggiornare ρ :

$$\begin{split} \rho_{i,n}^{(m+1)}(0) &= \frac{1}{2} + \frac{1}{2} \prod_{n' \in R_{i \setminus n}} (1 - 2q_{n',i}^{(m)}(0)) \\ \rho_{i,n}^{(m+1)}(1) &= 1 - \rho_{i,n}^{(m+1)}(0) \end{split} \tag{D.13}$$

Aggiornare q:

$$q_{n,i}^{(m+1)}(0) = A_{n,i}^{(m+1)} \cdot (1 - p_n) \prod_{i' \in C_{i' \setminus i,n}} \rho_{i',n}^{(m)}(0)$$
$$q_{n,i}^{(m+1)}(1) = A_{n,i}^{(m+1)} \cdot p_n \prod_{i' \in C_{i' \setminus i,n}} \rho_{i',n}^{(m)}(1)$$
(D.14)

dove $A_{n,i}^{(m+1)}$ è una costante di normalizzazione per assicurarsi che $q_{n,i}^{(m+1)}(0) + q_{n,i}^{(m+1)}(1) = 1$ (sono probabilità);¹

3. *Stima dei bit*:

Calcolare infine la probabilità a posteriori dei simboli codificati a[n]:

$$Q_n^{(m+1)}(0) = A_n^{(m+1)} \cdot (1 - p_n) \prod_{i \in C_{i,n}} \rho_{i,n}^{(m)}(0)$$
$$Q_n^{(m+1)}(1) = A_n^{(m+1)} \cdot p_n \prod_{i \in C_{i,n}} \rho_{i,n}^{(m)}(1)$$
(D.15)

(di nuovo $A_n^{(m+1)}$ è una costante di normalizzazione) e calcolare la stima del simbolo

$$\hat{a}^{(m+1)}[n] = \begin{cases} 1 & Q_n^{(m+1)}(1) > Q_n^{(m+1)}(0) \\ 0 & \text{altrimenti} \end{cases}$$
(D.16)

4. Termine:

Se $\hat{\mathbf{a}}^{(m+1)}\mathbf{H}^T = \mathbf{0}$, oppure (m+1) è pari al numero massimo di iterazioni N_{it} , arrestare l'algoritmo e inoltrare il valore corrente di $\hat{\mathbf{a}}^{(m+1)}$ come stima della parola di codice. Altrimenti, incrementare *m* e riprendere dal passo 2. con una nuova iterazione.

Abbiamo aggiunto nelle varie quantità l'apice ${}^{(m)}$ per indicare esplicitamente il numero di iterazione nell'esecuzione dell'algoritmo.

Per maggior robustezza numerica, si usa in pratica un algoritmo modificato che opera sui log-rapporti di probabilità $L(q_{n,i}) \in L(\rho_{i,n})$, come già visto per il turbo-decoder - prima però di aggiornare la versione "pseudo-code" dell'algoritmo abbiamo bisogno di qualche passaggio preliminare.

Osserviamo che

$$\ln\left(\frac{1+e^x}{1+e^{-x}}\right) = \ln\left(\frac{e^{x/2}(e^{-x/2}+e^{x/2})}{e^{-x/2}(e^{x/2}+e^{-x/2})}\right) = x$$
(D.17)

cosicché, ad m = 0,

$$L^{(0)}(q_{n,i}) = \ln\left(\frac{1 + \exp(4r[n]E_s/N_0)}{1 + \exp(-4r[n]E_s/N_0)}\right) = \frac{4E_s}{N_0}r[n]$$
(D.18)

come già ricavato nel testo. La parte difficile è il calcolo dell'espressione di $L(\rho_{i,n})$ in funzione di $L(q_{n,i})$. Ripartiamo da (D.8):

$$\rho_{i,n}(0) = \frac{1}{2} + \frac{1}{2} \prod_{n' \in R_{i \setminus n}} (1 - 2p_{n',i})$$

$$\rho_{i,n}(1) = 1 - \rho_{i,n}(0) = \frac{1}{2} - \frac{1}{2} \prod_{n' \in R_{i \setminus n}} (1 - 2p_{n',i})$$
(D.19)

¹Da notare che nel calcolo di $q_{n,i}^{(m+1)}(x)$ si potrebbero anche usare i valori $\rho_{i,n}^{(m+1)}(x)$ che sono appena stati calcolati alla semi-iterazione precedente - è una scelta di progetto spesso legata all'architettura hardware

e osserviamo che, fissate due qualunque probabilità p_0 e p_1 con $p_0 + p_1 = 1$ abbiamo

$$p_{0} - p_{1} = 1 - 2p_{1} = \frac{p_{0} - p_{1}}{p_{0} + p_{1}} = \frac{1 - p_{1}/p_{0}}{1 + p_{1}/p_{0}} = \frac{\sqrt{p_{0}/p_{1}} - \sqrt{p_{1}/p_{0}}}{\sqrt{p_{0}/p_{1}} + \sqrt{p_{1}/p_{0}}}$$
$$= \frac{\exp(\ln\sqrt{p_{0}/p_{1}}) - \exp(-\ln\sqrt{p_{0}/p_{1}})}{\exp(\ln\sqrt{p_{0}/p_{1}}) + \exp(-\ln\sqrt{p_{0}/p_{1}})} = \tanh\left(\frac{1}{2}\ln(p_{0}/p_{1})\right)$$
(D.20)

e il log-rapporto di probabilità è

$$L \triangleq \ln(p_1/p_0) = -2 \tanh^{-1}(1 - 2p_1)$$

da cui anche

$$1 - 2p_1 = \tanh\left(-\frac{L}{2}\right)$$

Usando queste relazioni e la (D.19) otteniamo

$$L(\rho_{i,n}) = -2 \tanh^{-1} \prod_{n' \in R_{i \setminus n}} (1 - 2p_{n',i}) = -2 \tanh^{-1} \prod_{n' \in R_{i \setminus n}} \tanh\left(-\frac{L(q_{n',i})}{2}\right)$$
(D.21)

che può essere ulteriormente semplificata in modo da ottenere un'espressione priva di prodotti. Infatti la funzione tanh è *dispari* e *strettamente crescente*, per cui

$$L(\rho_{i,n}) = 2 \tanh^{-1} \left(\prod_{n' \in R_{i \setminus n}} \tanh(|L(q_{n',i})|/2) \left\{ -\prod_{n' \in R_{i \setminus n}} \operatorname{sgn}[-\tanh(L(q_{n',i}))] \right\} \right)$$
$$= 2 \tanh^{-1} \left(\prod_{n' \in R_{i \setminus n}} \tanh(|L(q_{n',i})|/2) \right) \cdot \left\{ -\prod_{n' \in R_{i \setminus n}} \operatorname{sgn}[-L(q_{n',i})] \right\} \quad (D.22)$$

Separiamo adesso il calcolo del *modulo* (ampiezza) e del *segno* di $L(\rho_{i,n})$ come segue. L'ampiezza è

$$|L(\rho_{i,n})| = 2 \tanh^{-1} \exp\left[\ln\left(\prod_{n' \in R_{i \setminus n}} \tanh(|L(q_{n',i})|/2)\right)\right]$$
$$= 2 \tanh^{-1} \exp\left[\left(\sum_{n' \in R_{i \setminus n}} \ln \tanh(|L(q_{n',i})|/2)\right)\right]$$
(D.23)

che possiamo ulteriormente semplificare introducendo la funzione di funzione

$$\phi(\alpha) \stackrel{\triangle}{=} -\ln \tanh(\alpha/2) \tag{D.24}$$

il cui grafico è mostrato in Fig. D, per cui

$$|L(\rho_{i,n})| = \phi^{-1} \left[\sum_{n' \in R_{i \setminus n}} \phi\left(|L(q_{n',i})| \right) \right]$$
(D.25)



Figura D.1 Funzione $\phi(\alpha)$

La proprietà saliente di questa funzione ϕ è che (per $\alpha > 0$) $\phi(\phi(\alpha)) = \alpha$ (ovvero $\phi = \phi^{-1}$). Usando questa nuova funzione, il log-rapporto di probabilità di $\rho_{i,n}$ diventa

$$L(\rho_{i,n}) = \left\{-\prod_{n' \in R_{i \setminus n}} \operatorname{sgn}[-L(q_{n',i})]\right\} \cdot \phi \left[\sum_{n' \in R_{i \setminus n}} \phi(|L(q_{n',i})|)\right]$$
(D.26)

Dopo questi passaggi complicati, le espressioni di $L(q_{n,i}) \in L(Q_n)$ sono semplici (anche perchè le costanti di normalizzazione nelle (D.13)-(D.14) scompaiono nel rapporto) e possiamo finalmente introdurre lo pseudo-code dell'algoritmo iterativo message-passing nel dominio logaritmico:

1. *Inizializzazione*: Settare m = 0 e

$$L^{(0)}(q_{n,i}) = \frac{4E_s}{N_0} r[n]$$
(D.27)

2. Scambio dei messaggi: Aggiornare $L(\rho)$:

$$L^{(m+1)}(\rho_{i,n}) = -\prod_{n' \in R_{i \setminus n}} \operatorname{sgn}[-L^{(m)}(q_{n',i})] \cdot \phi \left[\sum_{n' \in R_{i \setminus n}} \phi(|L^{(m)}(q_{n',i})|)\right]$$
(D.28)

Aggiornare L(q):

$$L^{(m+1)}(q_{n,i}) = \frac{4E_s}{N_0} r[n] + \sum_{\substack{i' \neq i \\ i' \in C_{i',n}}} L^{(m)}(\rho_{i',n})$$
(D.29)
3. Stima dei bit:

Calcolare il LAPPR del simbolo codificato a[n]:

$$L^{(m+1)}(Q_n) = \frac{4E_s}{N_0} r[n] + \sum_{i \in C_{i,n}} L^{(m)}(\rho_{i,n})$$
(D.30)

Stimare il simbolo binario

$$\hat{a}^{(m+1)}[n] = \operatorname{sgn}[L^{(m+1)}(Q_n)]$$
 (D.31)

4. Termine:

Se $\hat{\mathbf{a}}^{(m+1)}\mathbf{H}^T = 0$ oppure $(m+1) = N_{it}$ allora fine delle iterazioni e $\hat{\mathbf{a}} = \hat{\mathbf{a}}^{(m+1)}$. Altrimenti, incrementare *m* e tornare a 2. per una nuova iterazione.

Concludiamo con una ulteriore approssimazione/semplificazione dell'algoritmo, che è quella prevalentemente usata nelle implementazioni hardware, il cosiddetto algoritmo *normalized-min-sum* algorithm. Considerando la Fig. D, possiamo semplificare l'espressione di $L(\rho)$ e la ricorsione (D.28). In quest'ultima, dobbiamo calcolare la somma $\sum_{n' \in R_{i \setminus n}} \phi(|L^{(m)}(q_{n',i})|)$ di alcuni valori della funzione $\phi(\alpha)$, che risulta fortemente nonlineare e fortemente decrescente al variare di α . Possiamo quindi osservare che quando nella somma vi sono valori di ϕ relativi a diversi valori dell'argomento, l'unco termine che effettivamente conta nel valore finale è quello ottenuto in corrispondenza del *minimo* tra gli argomenti $|L^{(m)}(q_{n',i})|$, cioè

$$\phi\left[\sum_{n'\in R_{i\setminus n}}\phi(|L^{(m)}(q_{n',i})|)\right] \simeq \phi\left[\phi(\min_{n'\in R_{i\setminus n}}|L^{(m)}(q_{n',i})|)\right]$$

D'altronde $\phi(\phi(\alpha)) = \alpha$, e quindi

$$\phi \left[\sum_{n' \in R_{i \setminus n}} \phi(|L^{(m)}(q_{n',i})|) \right] \simeq \min_{n' \in R_{i \setminus n}} |L^{(m)}(q_{n',i})| \tag{D.32}$$

e non dovremo mai più calcolare la funzione ϕ . L'aggiornamento di $L(\rho)$ diventa

$$L^{(m+1)}(\rho_{i,n}) = -\prod_{n' \in R_{i \setminus n}} \operatorname{sgn}[-L^{(m)}(q_{n',i})] \cdot \min_{n' \in R_{i \setminus n}} |L^{(m)}(q_{n',i})|$$
(D.33)

la cui espressione ed interpretazione è discussa nel testo all'equazione (6.63).

Possiamo infine *normalizzare* questo algoritmo non considerando il fattore $4E_s/N_0$ nell'inizializzazione di L(q) (D.27) e in tutti i seguenti aggiornamenti iterativi (D.29) e (D.30) senz'alcuna perdita di performance ulteriore, ottenendo il cosiddetto algoritmo *normalized min-sum* (o semplicemente *min-sum*). L'uso del min-sum con la conseguente normalizzazione causa una piccola degradazione delle performance, ma permette di ottenere un algoritmo di decodifica più semplice e che, come il DV, non necessita della conoscenza del rapporto segnale-rumore. Lo pseudo-code del normalized-min-sum è il seguente:

1. *Inizializzazione*: Settare m = 0 e

$$L^{(0)}(q_{n\,i}) = r[n] \tag{D.34}$$

492 L'ALGORITMO "MESSAGE-PASSING" PER DECODIFICARE I CODICI LDPC

2. Scambio dei messaggi:

Aggiornare $L(\rho)$:

$$L^{(m+1)}(\rho_{i,n}) = -\prod_{n' \in R_{i \setminus n}} \operatorname{sgn}[-L^{(m)}(q_{n',i})] \cdot \min_{n' \in R_{i \setminus n}} |L^{(m)}(q_{n',i})|$$
(D.35)

Aggiornare L(q):

$$L^{(m+1)}(q_{n,i}) = r[n] + \sum_{\substack{i' \neq i \\ i' \in C_{i',n}}} L^{(m)}(\rho_{i',n})$$
(D.36)

3. *Stima dei bit*:

Calcolare il LAPPR del simbolo codificato a[n]:

$$L^{(m+1)}(Q_n) = r[n] + \sum_{i \in C_{i,n}} L^{(m)}(\rho_{i,n})$$
(D.37)

Stimare il simbolo binario

$$\hat{a}^{(m+1)}[n] = \operatorname{sgn}[L^{(m+1)}(Q_n)]$$
 (D.38)

4. *Termine*: Se $\hat{\mathbf{a}}^{(m+1)}\mathbf{H}^T = 0$ oppure $(m+1) = N_{it}$ allora fine delle iterazioni e $\hat{\mathbf{a}} = \hat{\mathbf{a}}^{(m+1)}$. Altrimenti, incrementare *m* e tornare a 2. per una nuova iterazione.

APPENDIX E DECODIFICA A CANCELLAZIONI SUCCES-SIVE DEI CODICI POLARI

Partiamo dal vettore *soft* $\mathbf{r} = \mathbf{a}_{\pm} + \mathbf{w}$ ottenuto all'uscita del canale AWGN con ricezione a filtro adattato e in assenza di ISI, e con simboli di codice simmetrici $a_{\pm}[n] \in \{\pm 1\}$. L'algoritmo SCD (Successive Cancelation Decoder) è per sua natura ricorsivo nei bit (niente a che vedere con la decodifica *iterativa* in passate successive su *tutti* i bit di Turbo ed LDPC), e deriva dalla costruzione ricorsiva del codificatore.

Per derivare la strategia di decodifica relativa al codificatore di ordine 2 della Fig. 6.35 consideriamo la funzione di verosimiglianza f(r[0], r[1] | c[0], c[1]). Dato che r[1] non dipende da c[0] abbiamo

$$f(r[0], r[1] \mid c[0], c[1]) = f(r[0] \mid c[0] \oplus c[1]) \cdot f(r[1] \mid c[1])$$

ed anche

$$f(r[0], r[1], c[1] \mid c[0]) = f(r[0] \mid c[0] \oplus c[1]) \cdot f(r[1] \mid c[1]) \cdot p(c[1])$$
(E.1)

dove come al solito $f(\cdot)$ indica una pdf, e $p(\cdot)$ una legge (massa) di probabilità. Marginalizzando la (E.1) abbiamo

$$f(r[0], r[1] \mid c[0]) = \frac{1}{2} \sum_{c[1]} f(r[0] \mid c[0] \oplus c[1]) \cdot f(r[1] \mid c[1])$$
(E.2)

Comunicazioni Digitali, I Edizione.

di M. Luise Copyright © 2023 University of Pisa

Dalla (E.2), con dei passaggi simili a quelli svolti nell'Appendice D, ricaviamo il valore del LLR per il bit c[0] come segue:

$$L^{(1)}(c[0]) \stackrel{\triangle}{=} \ln \frac{f(r[0], r[1] \mid c[0] = 1)}{f(r[0], r[1] \mid c[0] = 0)}$$
$$= -2 \tanh^{-1} \left(\tanh \left(\frac{L^{(0)}(c[0])}{2} \right) \tanh \left(\frac{L^{(0)}(c[1])}{2} \right) \right)$$
$$\simeq -\text{sgn} \left(L^{(0)}(c[0]) \right) \text{sgn} \left(L^{(0)}(c[1]) \right) \min \left(|L^{(0)}(c[0])|, |L^{(0)}(c[1])| \right)$$
(E.3)

(vedi anche la (D.33)) dove i due LLR $L^{(0)}(c[0])$ e $L^{(0)}(c[1])$ sono ricavati direttamente dalle uscite soft del canale rispettivamente r[0] e r[1], e valgono (vedi (D.18)):

$$L^{(0)}(c[n]) = \frac{4E_s}{N_0} \cdot r[n] , \quad n = 0, 1$$
(E.4)

L'apice nei valori delle LLRs indica lo stadio (m) di decodifica: per questo esempio, m = 0significa input, e m = 1 indica il primo (e unico) stadio di calcolo. Dopo che $L^{(1)}(c[0])$ è stato ricavato come sopra, si può procedere alla decisione su c[0] (evidentemente basata sull'osservazione di r[0] e r[1]) secondo il segno del LLR: $\hat{c}[0] = \text{sgn} (L^{(0)}(c[0]))$.

Il secondo stadio di decodifica comporta la *cancellazione* del bit $\hat{c}[0]$ che è stato appena decodificato¹:

$$L^{(1)}(c[1]) \stackrel{\triangle}{=} \ln \frac{f(r[0], r[1] \mid c[0], c[1] = 1)}{f(r[0], r[1] \mid \hat{c}[0], c[1] = 0)}$$

$$\ln \frac{f(r[0] \mid \hat{c}[0] \oplus 1) \cdot f(r[1] \mid c[1] = 1)}{f(r[0] \mid \hat{c}[0] \oplus 0) \cdot f(r[1] \mid c[1] = 0)} = \ln \frac{f(r[0] \mid \overline{c}[0])}{f(r[0] \mid \hat{c}[0])} + \ln \frac{f(r[1] \mid c[1] = 1)}{f(r[1] \mid c[1] = 0)}$$

$$= \operatorname{sgn}(\hat{c}[0]) \cdot L^{(0)}(c[0]) + L^{(0)}(c[1]) = \frac{4E_s}{N_0} \left(\operatorname{sgn}(\hat{c}[0]) \cdot r[0] + r[1]\right) \quad (E.5)$$

In generale, sgn $(\hat{c}[0]) \cdot L^0(c[0]) \neq |L^0(c[0])|$ perché $\hat{c}[0]$ è stato ricavato osservando r[0] e r[1], mentre $L^0(c[0])$ si basa *solo* su r[0] e può non esserci concordanza.

Questa procedura ricorsiva è schematizzata in Fig. E.1 - una specie di grafo di Tanner percorso in senso inverso al codificatore, in quanto l'elaborazione progredisce da destra a sinistra a partire dai due LLR di canale $L^0(c[0])$ and $L^{(0)}(c[1])$ (E.4). I nodi rappresentano processori che inoltrano l'ingresso oppure calcolano le due equazioni (E.3)-(E.5). Se



Figura E.1 Decodifica polare 2×2 in due stadi

¹In tutta questa appendice usiamo una funzione $sgn(\cdot)$ speciale, per la quale $sgn(0) \stackrel{\triangle}{=} -1$, significando così nella (E.5) e successive l'usuale (ri)mappatura dei valori logici 0 e 1 sui valori simmetrici rispettivamente -1 e +1.

qualcuno dei c[n] si rivela essere un bit congelato, non c'è ovviamente bisogno di calcolare le equazioni di decodifica.

Vediamo adesso l'estensione all'ordine 4, cercando di nuovo di "invertire" l'elaborazione del codificatore. Osservando la Fig. E.2, vediamo che a partire dall'input del decoder $L^{(0)}(c[n])$, n = 0, 1, 2, 3, possiamo calcolare i due LLR $L^{(1)}(c[0])$ e $L^{(1)}(c[1])$ come combinazione di due di questi input presi con offset 2 come segue:



Figura E.2 Decodifica polare di ordine 4

$$L^{(1)}(c[0]) = -2 \tanh^{-1} \left(\tanh\left(\frac{L^{(0)}(c[0])}{2}\right) \tanh\left(\frac{L^{(0)}(c[2])}{2}\right) \right)$$
$$L^{(1)}(c[1]) = -2 \tanh^{-1} \left(\tanh\left(\frac{L^{(0)}(c[1])}{2}\right) \tanh\left(\frac{L^{(0)}(c[3])}{2}\right) \right)$$
(E.6)

Fatto questo, possiamo effettuare una decodifica di ordine due come già visto, usando i due valori $L^{(1)}(c[0]) \in L^{(1)}(c[1])$ appena calcolati, per ottenere prima $\hat{c}[0]$ (avendo osservato tutti gli $L^{(0)}(c[n])$) e successivamente anche $\hat{c}[1]$ (avendo osservato tutti gli $L^{(0)}(c[n])$) e anche $\hat{c}[0]$). Dopodicheé, ci spostiamo nella parte inferiore del decoder e calcoliamo due nuovi LLR dopo *cancellazione* dei simboli appena stimati per trovare gli altri due:

$$L^{(1)}(c[2]) = L^{(0)}(c[2]) + \operatorname{sgn}(\hat{c}[0]) \cdot L^{(0)}(c[0])$$

$$L^{(1)}(c[3]) = L^{(0)}(c[3]) + \operatorname{sgn}(\hat{c}[1]) \cdot L^{(0)}(c[1])$$
(E.7)

Dopo la cancellazione, applichiamo infine una ulteriore decodifica di ordine 2 per ricavare $\hat{c}[2] \in \hat{c}[3]$.

496 DECODIFICA A CANCELLAZIONI SUCCESSIVE DEI CODICI POLARI

Questo ragionamento di tipo *divide et impera* si applica (ricorsivamente) anche all'ordine 8: si devono per prima cosa ricavare $L^{(1)}(c[n])$, n = 0, 1, 2, 3 pre-processando due a due le coppie $(L^{(0)}(c[n]), L^{(0)}(c[n+4]))$, n = 0, 1, 2, 3 per poi trovare $\hat{c}[0], \hat{c}[1], \hat{c}[2], \hat{c}[3]$ attraverso una decodifica di ordine 4 come sopra; poi, bisogna cancellare i bit appena stimati per trovare gli LLRs $L^{(1)}(c[n])$, n = 4, 5, 6, 7, e infine si devono trovare $\hat{c}[4], \hat{c}[5], \hat{c}[6], \hat{c}[7]$ attraverso una seconda decodifica di ordine 4. Iterando questa costruzione si raggiunge qualunque ordine di decodifica, secondo lo pseudo-code ricorsivo seguente:

Decodifica $(N; L_0, \ldots, L_{N-1})$:

Se N = 2, decodificare secondo (E.4-E.3-E.5) e terminare; altrimenti:

1. Combinatione: per i = 0, ..., N/2 - 1 calcolare

$$L^{(1)}(c[i]) = -2 \tanh^{-1} \left(\tanh\left(\frac{L^{(0)}(c[i])}{2}\right) \tanh\left(\frac{L^{(0)}(c[i+N/2])}{2}\right) \right)$$

- 2. Prima sub-decodifica: Decodifica $(N/2; L^{(1)}(c[0]), \dots, L^{(1)}(c[N/2 - 1])) \rightarrow \hat{c}[0], \dots, \hat{c}[N/2 - 1]$
- 3. Cancellazione dei bit: per $i = 0, \dots, N/2 - 1$ calcolare

$$L^{(1)}(c[N/2+i]) = L^{(0)}(c[N/2+i]) + \operatorname{sgn}(\hat{c}[i]) \cdot L^{(0)}(c[i])$$

4. Seconda sub-decodifica: Decodifica' $(N/2; L^{(1)}(c[N/2]), \dots, L^{(1)}(c[N-1])) \rightarrow \hat{c}[N/2], \dots, \hat{c}[N-1]$

Fine

APPENDIX F RISPOSTA IN FREQUENZA DI UNA FIBRA OTTICA DISPERSIVA

La dispersione cromatica può essere analizzata quantitativamente considerando un pacchetto d'onda (impulso di bit) che si propaga lungo l'asse z in un mezzo omogeneo isotropo lineare dispersivo semi-infinito. L'espressione del pacchetto d'onda è

$$\psi(z;t) = \Re\left[\tilde{\psi}(z;t)\exp\left(j2\pi f_0 t\right)\right] = \Re\left[\tilde{a}\left(z;t\right)\exp\left(j\left(2\pi f_0 t - \beta_0 z\right)\right)\right]$$
(F.1)

dove $\beta_0 = \beta(f_0)$ è la costante di propagazione alla frequenza f_0 e $\tilde{a}(z;t)$ è, fissato z, l'inviluppo complesso dell'onda, che varia lentamente nel tempo rispetto al periodo di oscillazione $T_0 = 1/f_0$. Come abbiano già visto, $\psi(z;t)$ rappresenta genericamente una qualunque delle sei componenti del campo elettromagnetico associato all'onda luminosa. Nota la forma d'onda lanciata $\tilde{a}(0;t)$, ci proponiamo di ricavare una relazione che leghi tale forma d'onda con quella osservabile ad una lunghezza z dall'imboccatura della fibra, $\tilde{a}(z;t)$. Poiché il fenomeno della dispersione è di tipo *lineare*, possiamo attribuire ad $\tilde{a}(0;t)$ ed $\tilde{a}(z;t)$ rispettivamente il significato di *ingresso e uscita di un sistema lineare stazionario* (*SLS*) che può essere completamente caratterizzato dalla propria risposta in frequenza $H(\nu)$. Per calcolare tale risposta in frequenza, supponiamo che $\tilde{a}(0;t) = \tilde{\psi}(0;t)$ sia esso stesso una oscillazione sinusoidale a frequenza ν :

$$\tilde{a}(0;t) = \tilde{\psi}(0;t) = \exp\left(j2\pi\nu t\right) \tag{F.2}$$

Comunicazioni Digitali, I Edizione.

di M. Luise Copyright © 2023 University of Pisa

498 RISPOSTA IN FREQUENZA DI UNA FIBRA OTTICA DISPERSIVA

Evidentemente, l'onda è in questo caso monocromatica a frequenza $f = f_0 + \nu$ e quindi si propagherà per definizione con la costante $\beta (f_0 + \nu)$; dunque

$$\psi(z;t) = \psi(0;t) \exp(-\jmath\beta(f_0 + \nu)z) = \tilde{a}(0;t) \exp(-\jmath\beta(f_0 + \nu)z)$$
$$= \exp(-\jmath2\pi\nu t) \exp(-\jmath\beta(f_0 + \nu)z)$$
(F.3)

D'altronde, quando l'ingresso di un sistema lineare è una oscillazione monocromatica, anche l'uscita è della stessa forma: il segnale d'uscita sarà una oscillazione alla stessa frequenza ν , modificata in ampiezza e fase in ragione del valore della risposta in frequenza alla frequenza di oscillazione ν :

$$\tilde{a}(z;t) = H(\nu) \exp\left(j2\pi\nu t\right) \tag{F.4}$$

per cui il pacchetto d'onda è anche esprimibile come

$$\tilde{\psi}(z;t) = H(\nu) \exp\left(\jmath 2\pi\nu t\right) \exp\left(-\jmath\beta_0 z\right) \tag{F.5}$$

Uguagliando la (F.3) e la (F.5) ricaviamo immediatamente la risposta in frequenza del mezzo dispersivo:

$$H(\nu) = \exp\left\{-\jmath\left[\beta\left(f_0 + \nu\right) - \beta\left(f_0\right)\right]z\right\}$$
(F.6)

Se l'indice di rifrazione *non* dipende da f (mezzo non dispersivo), cioè $\beta(f) = 2\pi n f/c$, otteniamo $\beta(f_0 + \nu) - \beta(f_0) = 2\pi \nu n/c$, e non si ha nessuna distorsione durante la propagazione, ma solo un ritardo proporzionale alla lunghezza del tragitto percorso. Supponiamo invece che il mezzo sia *debolmente dispersivo* (come è il caso della fibra), cioè che la variazione dell'indice di rifrazione nell'ambito della banda del pacchetto d'onda sia modesta. In tal caso è lecito approssimare l'andamento di $\beta(f) = \beta(f_0 + \nu)$ con uno sviluppo in serie di Taylor attorno ad f_0 arrestato al secondo ordine, ottenendo

$$H(\nu) \cong \exp\left[-\jmath \left. \frac{d\beta}{df} \right|_{f=f_0} \nu z \right] \exp\left[-\jmath \frac{1}{2} \left. \frac{d^2\beta}{df^2} \right|_{f=f_0} \nu^2 z \right]$$
(F.7)

La (F.7) viene di consuetudine espressa in funzione della velocità di gruppo v_g definita come in (11.31) e del *coefficiente di dispersione cromatica*

$$D \triangleq \frac{d\left(1/v_g\right)}{d\lambda} = \frac{d\left(1/v_g\right)}{df} \frac{df}{d\lambda} = -\frac{c}{\lambda^2} \frac{1}{2\pi} \frac{d^2\beta}{df^2}$$
(F.8)

come segue:

$$H(\nu) \cong \exp\left[-\jmath 2\pi\nu \frac{z}{v_g}\right] \exp\left[+\jmath \nu^2 \frac{\pi D\lambda_0^2 z}{c}\right]$$
(F.9)

In conclusione, se il mezzo è non dispersivo, le derivate di β rispetto ad f di ordine superiore al primo sono tutte nulle, e la $H(\nu)$ si riduce al consueto termine di ritardo di gruppo. In caso contrario, il mezzo introduce un termine di distorsione di fase tanto maggiore quanto maggiore è il coefficiente di dispersione D.

APPENDIX G CRITERIO DI DECISIONE OTTIMO E FUN-ZIONE DI LOG-VEROSIMIGLIANZA DEL RI-CEVITORE IM/DD

Immaginiamo di sviluppare il segnale osservato I(t) su di una base ortonormale M-dimensionale di funzioni $\{\varphi_m(t)\}_{m=1}^M$, ottenendo il vettore I che raccoglie le componenti della scomposizione:

$$\mathbf{I} = [I_1, I_2, ..., I_M], \quad I_m \triangleq \int_0^1 I(t)\varphi_m(t)dt$$
(G.1)

poiché I(t) è un processo aleatorio, I risulta un vettore aleatorio caratterizzato da una certa funzione densità di probabilità $f_{\mathbf{I}}(\mathbf{i})$. Una volta osservato un particolare valore i assunto da I, si possono definire le cosiddette probabilità a posteriori P_{AP} dei due valori di b[0]:

$$P_{AP}(0) \triangleq \Pr\{b[0] = 0 | \mathbf{I} = \mathbf{i}\}, \quad P_{AP}(1) \triangleq \Pr\{b[0] = 1 | \mathbf{I} = \mathbf{i}\}$$
(G.2)

Il ricevitore MAP sceglie come valore stimato $\hat{b}[0]$ del simbolo trasmesso b[0] quello tra i due valori possibili avente massima probabilità a posteriori: $\hat{b}[0] = 1 \Leftrightarrow P_{AP}(1) > P_{AP}(0)$. Se, come nelle nostre ipotesi, si ha $p_1 = p_0 = 1/2$, il criterio MAP di decisione viene a coincidere con il cosiddetto criterio della massima verosimiglianza (MV): "Decidere che il simbolo stimato $\hat{b}[0]$ è pari a 1 quando $f_{\mathbf{I}}(\mathbf{i}|b[0] = 1) > f_{\mathbf{I}}(\mathbf{i}|b[0] = 0)$,

Comunicazioni Digitali, I Edizione.

di M. Luise Copyright © 2023 University of Pisa



Figura G.1 Base ortonormale per la scomposizione della corrente di fotorivelazione

altrimenti decidere $\hat{b}[0] = 0$ ".

Il criterio MV viene spesso formulato per comodità di calcolo attraverso la funzione di log-verosimiglianza

$$\Lambda_{\mathrm{M}}(n) \triangleq \ln[f_{\mathbf{I}}(\mathbf{i}|b[0] = n)], \quad n = 0, 1$$
(G.3)

e recita finalmente come segue:

"Decidere $\hat{b}[0] = 1$ quando $\Lambda_{\rm M}(1) > \Lambda_{\rm M}(0)$, altrimenti decidere $\hat{b}[0] = 0$ ".

Detto questo, appare chiaro che le decisioni del ricevitore potrebbero in generale dipendere dal numero M di funzioni della base ortonormale di espansione, ovvero dalla dimensionalità di I. Questa dipendenza deve essere cancellata assicurandosi che $\{\varphi_m(t)\}_{m=1}^M$ risulti una base completa per l'insieme dei segnali osservati; ciò si ottiene passando al limite di $M \to \infty$.

Dunque il primo passo nel calcolo della funzione di logverosimiglianza $\Lambda(b[0])$ è quello di determinare la densità di probabilità del vettore aleatorio $\mathbf{I} = [I_1, I_2, ..., I_M]$ ottenuto espandendo la corrente di fotorivelazione I(t) su di una base ortonormale. Scegliamo come base M-dimensionale di sviluppo quella in cui l'm-esimo elemento φ_m è un impulso rettangolare diverso da zero sull'intervallo $[(m-1)\Delta, m\Delta), \Delta = T/M$, e, per normalizzazione, di ampiezza $1/\sqrt{\Delta}$. L'insieme di tali funzioni è rappresentato in Fig. G. Si ha quindi, nell'ipotesi di fotorivelatore a banda larga, cioè $h(t) = q\delta(t)$

$$I_m \triangleq \int_{0}^{T} I(t)\varphi_m(t)dt = \frac{1}{\sqrt{\Delta}} \int_{(m-1)\Delta}^{m\Delta} I(t)dt = \frac{q}{\sqrt{\Delta}} N_m$$
(G.4)

ove N_m è una v.a. di Poisson caratterizzata dalla seguente massa di probabilità condizionata al valore del simbolo b[0]:

$$p_{N_m}(n_m|b[0]) \triangleq \Pr\{N_m = n_m|b[0]\} = \exp[-\Lambda_m^{(b[0])}] \frac{\left[\Lambda_m^{(b[0])}\right]^{n_m}}{n_m!} \tag{G.5}$$

avendo definito

$$\Lambda_m^{(b[0])} \triangleq \int_{(m-1)\Delta}^{m\Delta} \left[\mu_d + \mu^{(b[0])}(t)\right] dt \tag{G.6}$$

che è il parametro della m-esima v.a. di Poisson N_m , condizionato alla trasmissione del simbolo b[0] e con una data corrente di buio $I_d = q\mu_d$. Come si nota, le v.a. N_m sono proporzionali alle componenti I_m del vettore aleatorio I, cosicché il problema della

decisione ottima può essere risolto considerando in luogo di I il vettore aleatorio equivalente $\mathbf{N} = [N_1, N_2, ..., N_m].$

Le *m* v.a. N_m sono mutuamente indipendenti perché ottenute da un processo impulsi di Poisson integrato su intervalli temporali disgiunti, e quindi la massa di probabilità del vettore aleatorio $\mathbf{N} = [N_1, N_2, ..., N_m]$ condizionata al simbolo trasmesso è data da

$$p_{\mathbf{N}}(\mathbf{n}|b[0]) = \prod_{m=1}^{M} p_{N_m}(n_m|b[0]) = \prod_{m=1}^{M} \exp[-\Lambda_m^{(b[0])}] \frac{\left[\Lambda_m^{(b[0])}\right]^{n_m}}{n_m!}$$
(G.7)

e la relativa funzione di log-verosimiglianza $\Lambda_M(b[0])$ è data da

$$\ln[p_{\mathbf{N}}(\mathbf{n}|b[0])] = -\sum_{m=1}^{M} \Lambda_m^{(b[0])} + \sum_{m=1}^{M} n_m \ln[\Lambda_m^{(b[0])}] - \sum_{m=1}^{M} \ln(n_m!)$$
(G.8)

Consideriamo adesso i tre termini a secondo membro nella (G.8): il terzo termine può essere scartato perchè, non dipendendo da b[0], non porta alcuna informazione utile ai fini della decisione (non interviene cioè nel problema di massimizzazione della funzione di log-verosimiglianza rispetto al simbolo presunto). Ricordando la (G.6), il primo termine risulta

$$\sum_{m=1}^{M} \Lambda_m^{(b[0])} = \sum_{m=1}^{M} \int_{(m-1)T/M}^{mT/M} [\mu_d + \mu^{(b[0])}(t)] dt$$
$$= \int_{0}^{T} [\mu_d + \mu^{(b[0])}(t)] dt = T\bar{\mu}^{(b[0])} + T\mu_d$$

(G.9)

per cui, scartando ulteriormente il termine ininfluente $T\mu_d$, la funzione di log-verosimiglianza equivalente è

$$\Lambda_{M}(b[0]) = -T \cdot \bar{\mu}^{(b[0])} + \sum_{m=1}^{M} n_{m} \ln[\Lambda_{m}^{(b[0])}]$$

$$= \sum_{m=1}^{M} n_{m} \ln \left[\int_{(m-1)T/M}^{mT/M} [\mu_{d} + \mu^{(b[0])}(t)] dt \right] - T \cdot \bar{\mu}^{(b[0])}$$

$$= \frac{\sqrt{\Delta}}{q} \sum_{m=1}^{M} i_{m} \ln \left[\int_{(m-1)T/M}^{mT/M} [\mu_{d} + \mu^{(b[0])}(t)] dt \right] - T \cdot \bar{\mu}^{(b[0])}$$
(G.10)

avendo tenuto conto che $I_m = qN_m/\sqrt{\Delta}$. Resta dunque l'ultimo passo, e cioè quello di considerare una base completa, con $M \to \infty$ (e quindi $\Delta \to 0$). In queste condizioni,

$$\int_{(m-1)T/M}^{mT/M} [\mu_d + \mu^{(b[0])}(t)] dt \cong \frac{T}{M} \left[\mu_d + \mu^{(b[0])} \left(\frac{mT}{M} \right) \right] = \frac{\mu_d T}{M} \left[1 + \frac{\mu^{(b[0])}(mT/M)}{\mu_d} \right]$$
(G.11)

e quindi

$$\Lambda_{M}(b[0]) \cong \frac{\sqrt{\Delta}}{q} \sum_{m=1}^{M} i_{m} \ln \left\{ \frac{\mu_{d}T}{M} \left[1 + \frac{\mu^{(b[0])}(mT/M)}{\mu_{d}} \right] \right\} - T \cdot \bar{\mu}^{(b[0])}$$
$$= \frac{\sqrt{\Delta}}{q} \sum_{m=1}^{M} i_{m} \ln \left\{ \frac{\mu_{d}T}{M} \right\} + \frac{\sqrt{\Delta}}{q} \sum_{m=1}^{M} i_{m} \ln \left\{ 1 + \frac{\mu^{(b[0])}(mT/M)}{\mu_{d}} \right\} - T \cdot \bar{\mu}^{(b[0])}$$
(G.12)

ovvero, scartando ancora i termini non dipendenti da $\boldsymbol{b}[0]$

$$\Lambda_{\rm M}(b[0]) \cong \frac{\sqrt{\Delta}}{q} \sum_{m=1}^{M} i_m \ln\left\{1 + \frac{\mu^{(b[0])}(mT/M)}{\mu_d}\right\} - T \cdot \bar{\mu}^{(b[0])} \tag{G.13}$$

Riprendendo l'espressione di i_m , si nota anche che, quando $M \to \infty$,

$$I_m \triangleq \frac{1}{\sqrt{\Delta}} \int_{(m-1)\Delta}^{m\Delta} I(t)dt \to \sqrt{\Delta} \cdot I(m\Delta)$$
(G.14)

per cui

$$\Lambda_{\rm M}(b[0]) \cong \frac{1}{q} \sum_{m=1}^{M} \Delta \cdot I(m\Delta) \ln \left\{ 1 + \frac{\mu^{(b[0])}(m\Delta)}{\mu_d} \right\} - T \cdot \bar{\mu}^{(b[0])} \tag{G.15}$$

Infine, quando $M \to \infty$, la somma a secondo membro tende ad un integrale, portando al risultato finale

$$\Lambda(b[0]) = \lim_{M \to \infty} \Lambda_{\rm M}(b[0]) = \frac{1}{q} \int_{0}^{T} I(t) \ln\left\{1 + \frac{\mu^{(b[0])}(t)}{\mu_d}\right\} dt - T \cdot \bar{\mu}^{(b[0])} \tag{G.16}$$

che rappresenta la forma della funzione di log-verosimiglianza citata nel testo. Si noti infine che nel passaggio dalla (G.13) alla (G.14) si è intenzionalmente scambiata la notazione del processo I(t) e le relative proiezioni aleatorie I_m con la particolare realizzazione osservata i(t) avente componenti determinate i_m .

BIBLIOGRAFIA

- 1. M. Luise, G. Vitetta, "Teoria dei Segnali", 3a Edizione, Mc Graw Hill International Editions, 2009.
- 2. B. Porat, "A Course in Digital Signal Processing", Wiley, 1996.
- 3. A. Papoulis, S. Pillai "*Probability, Random Variables and Stochastic Processes*", 4a Edizione, Mc Graw Hill International Editions, 2002.
- 4. G.H. Golub, C.F. van Loan, "Matrix Computations", 4a Edizione, JHU Press, 2013.
- 5. U. Spagnolini, "Statistical Signal Processing in Engineering", Wiley, 2018.
- J. Proakis, M. Salehi, "Digital Communications", 5th Edition, Mc Graw Hill International Editions, 2018.
- 7. T.M. Cover, J.A. Thomas, "Elements of Information Theory", 2nd Edition, Wiley, 2006.
- D.J.C. MacKay, "Information Theory, Inference, and Learning Algorithms", Cambridge University Press, 2003.
- 9. W. Ryan, S. Lin, "Channel Codes: Classical and Modern", Cambridge University Press, 2009
- 10. D.J.C. MacKay, "Fountain codes", IEE Proc.-Commun., Vol. 152, No. 6, December 2005
- E. Arikan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels", IEEE Transactions on Information Theory 55.7 (2009)
- A. Shokrollahi, M. Luby, "*Raptor Codes*", Foundations and Trends in Communications and Information Theory Vol. 6, Nos. 3-4 (2009)
- 13. A.F. Molisch, "Wireless Communications", Wiley, 2011.
- 14. A. Goldsmith, "Wireless Communications", Cambridge University Press, 2005.
- J.H. Holmes, "Spread Spectrum Systems for GNSS and Wireless Communications", Artech House, 2007

Comunicazioni Digitali, I Edizione. di M. Luise Copyright © 2023 University of Pisa

504 BIBLIOGRAFIA

- 16. S.K. Sharma, "*Characterization and modeling of MIMO wireless channels based on correlation tensor*", Computers and Mathematics with Applications 64 (2012).
- 17. C. Balanis, "Antenna theory analysis and design", 2nd Edition, Wiley, 1997.
- 18. B.E.A. Saleh, M.C. Teich, "Fundamentals of Photonics", 3rd Edition, Wiley, 2019.
- 19. G.P. Agrawal, "Fiber-Optic Communication Systems", 5th Edition, Wiley, 2021.

