Laurea Magistrale in *Ingegneria delle Telecomunicazioni*

The Viterbi (decoding) Algorithm

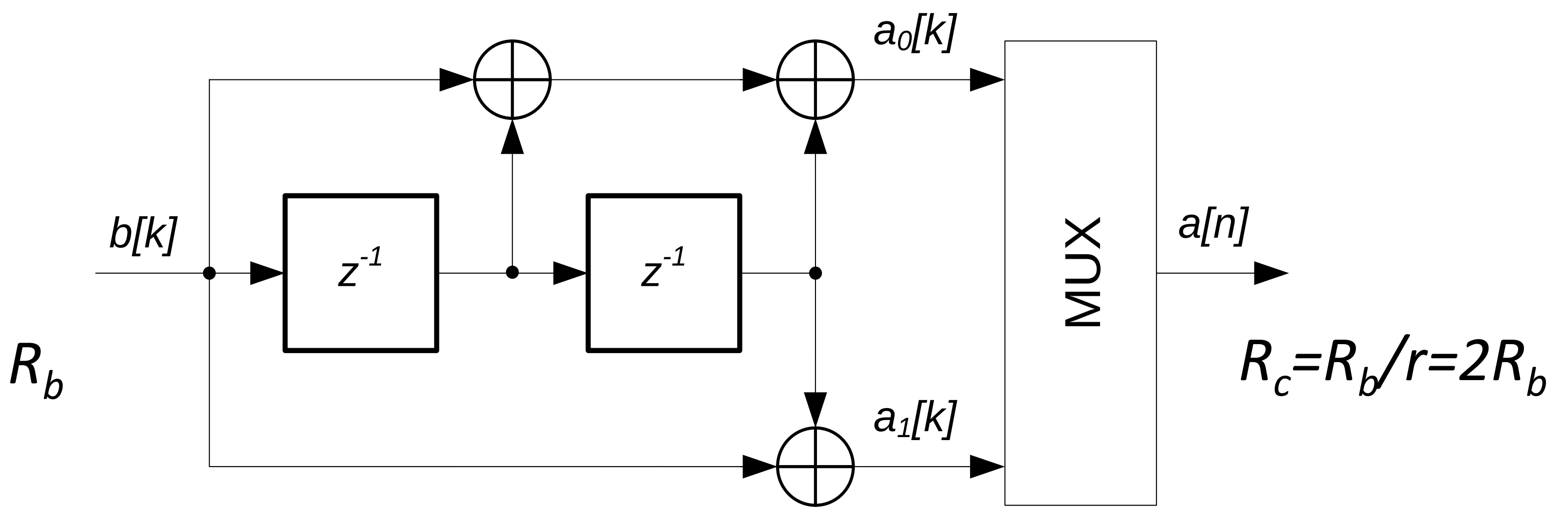
Marco Luise

marco.luise@unipi.it

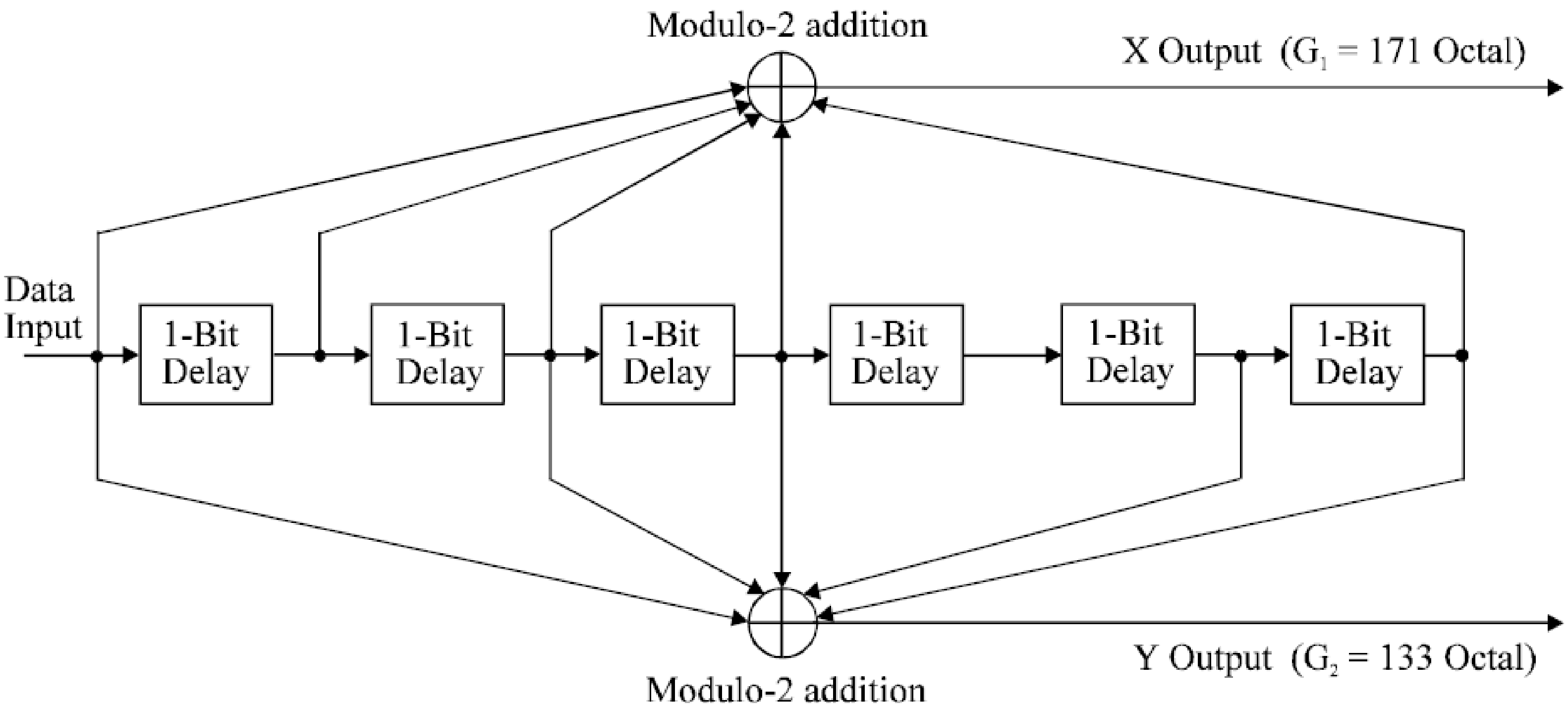
http://www.iet.unipi.it/m.luise/

Dip. Ingegneria dell’Informazione, University of Pisa, Italy

The r=1/2, K=3 Convolutional Encoder

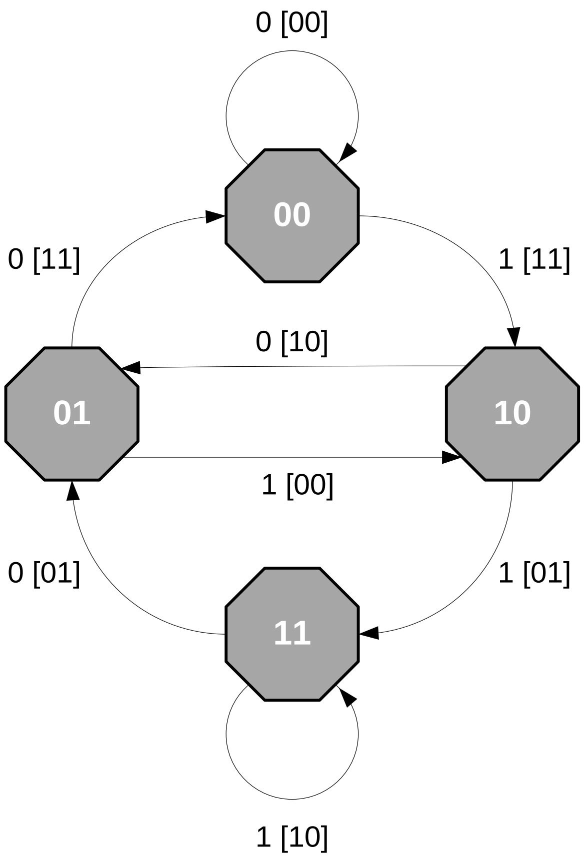


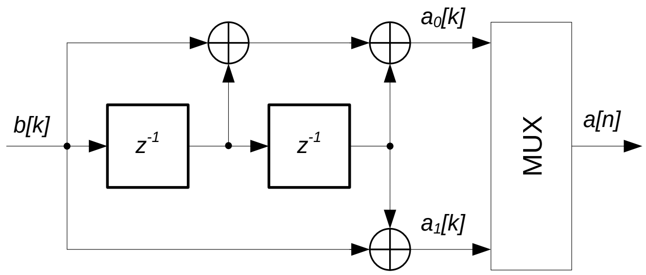
The *r*=1/2, *K*=7 Convolutional Encoder of DVB-T



Dip. Ingegneria dell'Informazione
University of Pisa, Pisa, Italy

The *State Diagram*

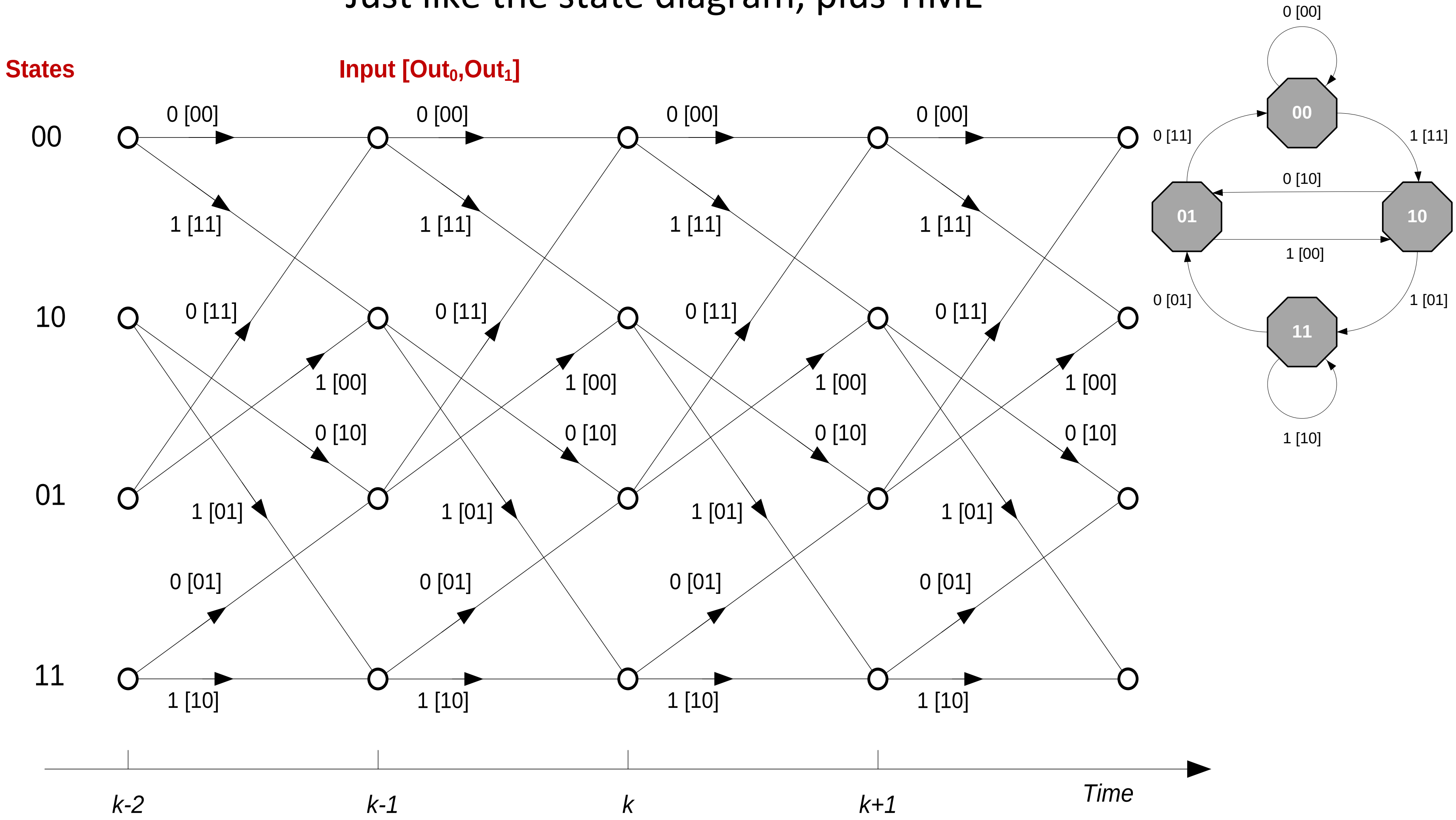




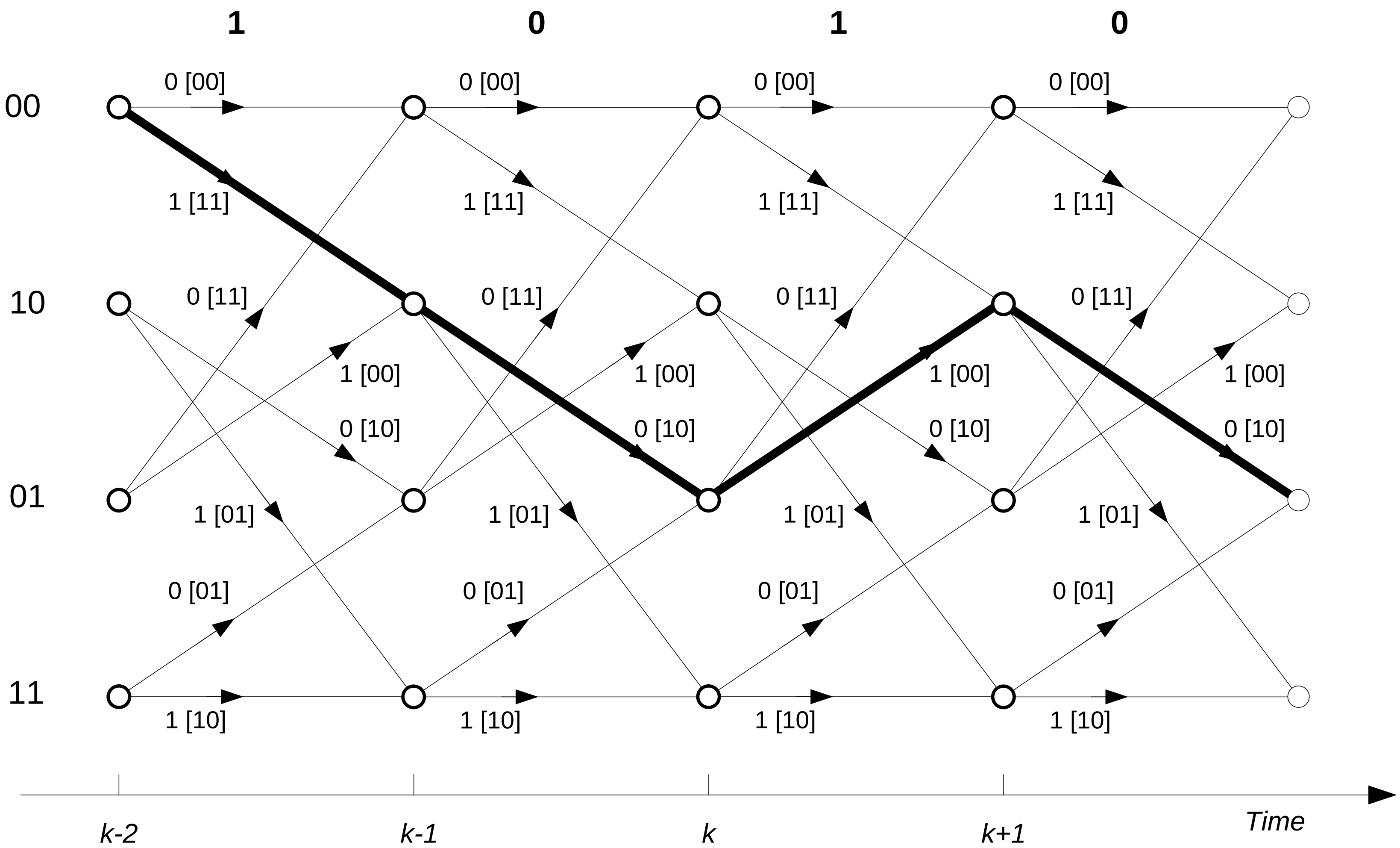
Describes the evolution of the FSM, but does not (explicitly) show *time*

The Trellis

Just like the state diagram, plus TIME



A Specific Path in the Trellis



r=1/2 Maximum Likelihood Sequence Estimation - HD

1. Observe the *binary* (Hard-Detected) channel output

$$d[n] = a[n] \oplus e[n] \, , \, n = 0, \ldots, N-1$$

$$a[2k] = a_0[k] \, , \, a[2k+1] = a_1[k] \, , \, k = 0, \ldots, M-1 \quad , \quad N = 2M$$

2. Find the *sequence* of source bits $\hat{\mathbf{b}} = \left[\hat{b}[0], \hat{b}[1], \ldots, \hat{b}[M-1] \right]$ such that

$$\hat{\mathbf{b}} = \arg\max_{\tilde{\mathbf{b}}} p(\mathbf{d} \,|\, \tilde{\mathbf{b}})$$

$$\mathbf{d} = \left[d[0], d[1], \ldots, d[2M-2], d[2M-1] \right]$$

r=1/2 Maximum Likelihood Sequence Estimation - SD

1. Observe the *soft* output of the AWGN (α =remapped *a*)

$$r[n] = \alpha[n] + w[n], \; n = 0,\dots,N-1$$

$$a[2k] = a_0[k], \; a[2k+1] = a_1[k], \; k = 0,\dots,M-1 \quad , \quad N = 2M$$

2. Find the *sequence* of source bits such that $\hat{\mathbf{b}} = \left[\hat{b}[0], \hat{b}[1], \dots, \hat{b}[M-1] \right]$

$$\hat{\mathbf{b}} = \arg\max_{\tilde{\mathbf{b}}} f_{\mathbf{R}}(\mathbf{r} \,|\, \tilde{\mathbf{b}})$$

$$\mathbf{r} = \left[r[0], r[1], \dots, r[2M-2], r[2M-1] \right]$$

Maximum Likelihood Sequence Estimation

1. Hard-detected, binary Input **d**

$$\hat{\mathbf{a}} = \arg\min_{\tilde{\mathbf{a}} \in \text{Trellis}} d_H(\mathbf{d}, \tilde{\mathbf{a}})$$

2. Soft Input, real-valued **r**

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\tilde{\boldsymbol{\alpha}} \in \text{Trellis}} \left\| \mathbf{r} - \tilde{\boldsymbol{\alpha}} \right\|^2$$

We have to *minimize a metrics* in both cases

Recursive Formulation of the VA: The *Survivors* at time *k-1*

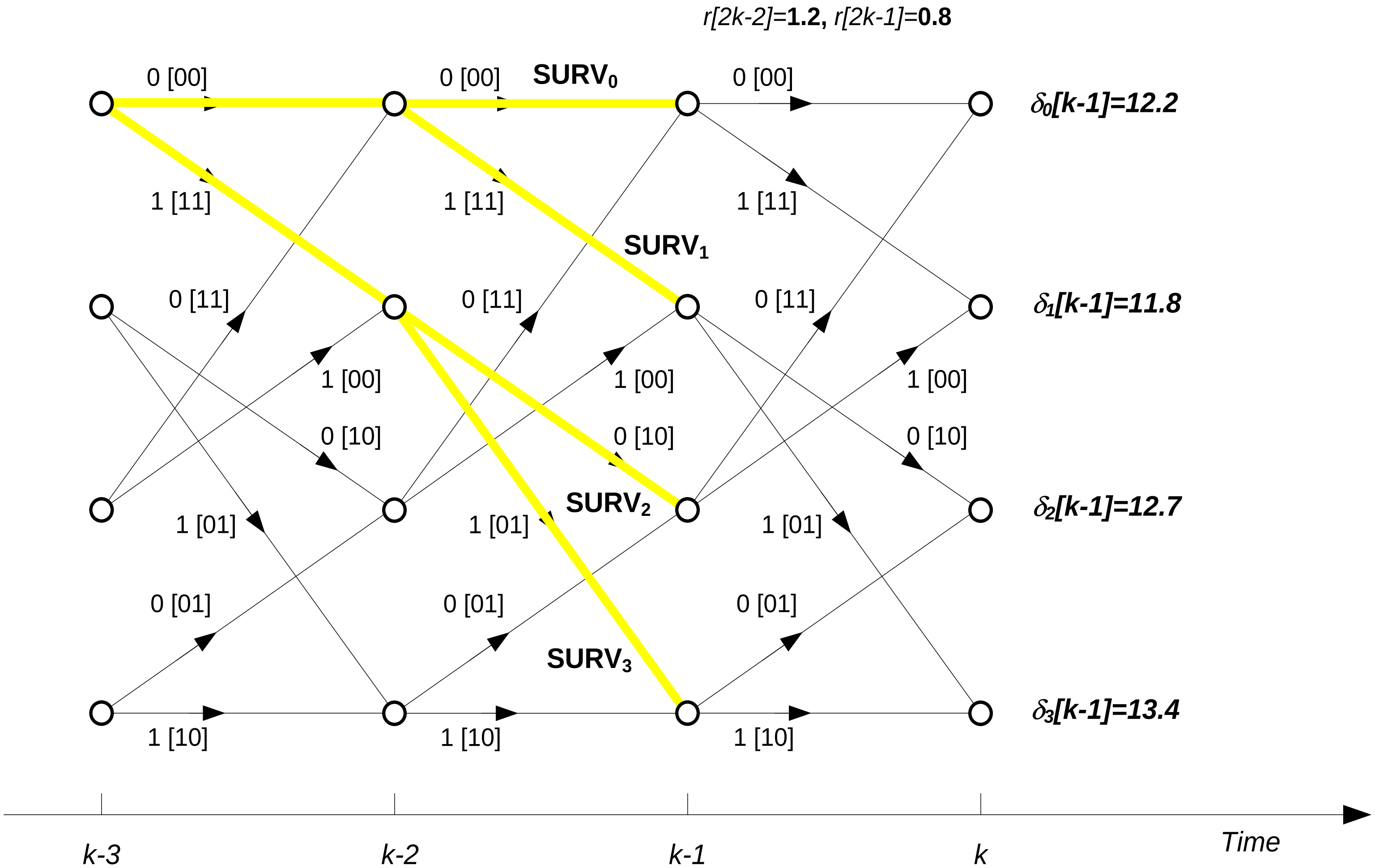
1. At time *k*, the algorithm has a got a sequence of tentative source symbols for every state *i* (the *Survivor* SURV_i)

$$\mathbf{b}_i[k-1]=\left[b_i^{(k-1)}[0],b_i^{(k-1)}[1],b_i^{(k-1)}[2],\ldots,b_i^{(k-1)}[k-1]\right]$$
$$i=0,\ldots,N_s-1$$

2. Each survivor also has an associated *accumulated metrics* (one per state)

$$\delta_i[k-1]$$
$$i=0,\ldots,N_s-1$$

The *Survivors* at time *k-1*



What does the VD do? 1/2

1. At time k-1 observe the received signal

$$r[2k-2]=\alpha_0[k-1]+w[2k-2]\ ,\ \ r[2k-1]=\alpha_1[k-1]+w[2k-1]$$

2. For each pair of states (*i,j*) connected at times (*k-1,k*) by a branch, respectively, compute the *branch metrics*

$$\lambda^{(i,j)}[k]=\left(r[2k-2]-\alpha_0^{(i,j)}[k-1]\right)^2+\left(r[2k-1]-\alpha_1^{(i,j)}[k-1]\right)^2$$

3. Extend the survivors and compute the new accumulated metrics by *adding* the *branch* metrics to the previous *accumulated* metrics

$$\delta_j^0[k]=\delta_{i_0}[k-1]+\lambda^{(i_0,j)}[k]$$

$$\delta_j^1[k]=\delta_{i_1}[k-1]+\lambda^{(i_1,j)}[k]$$

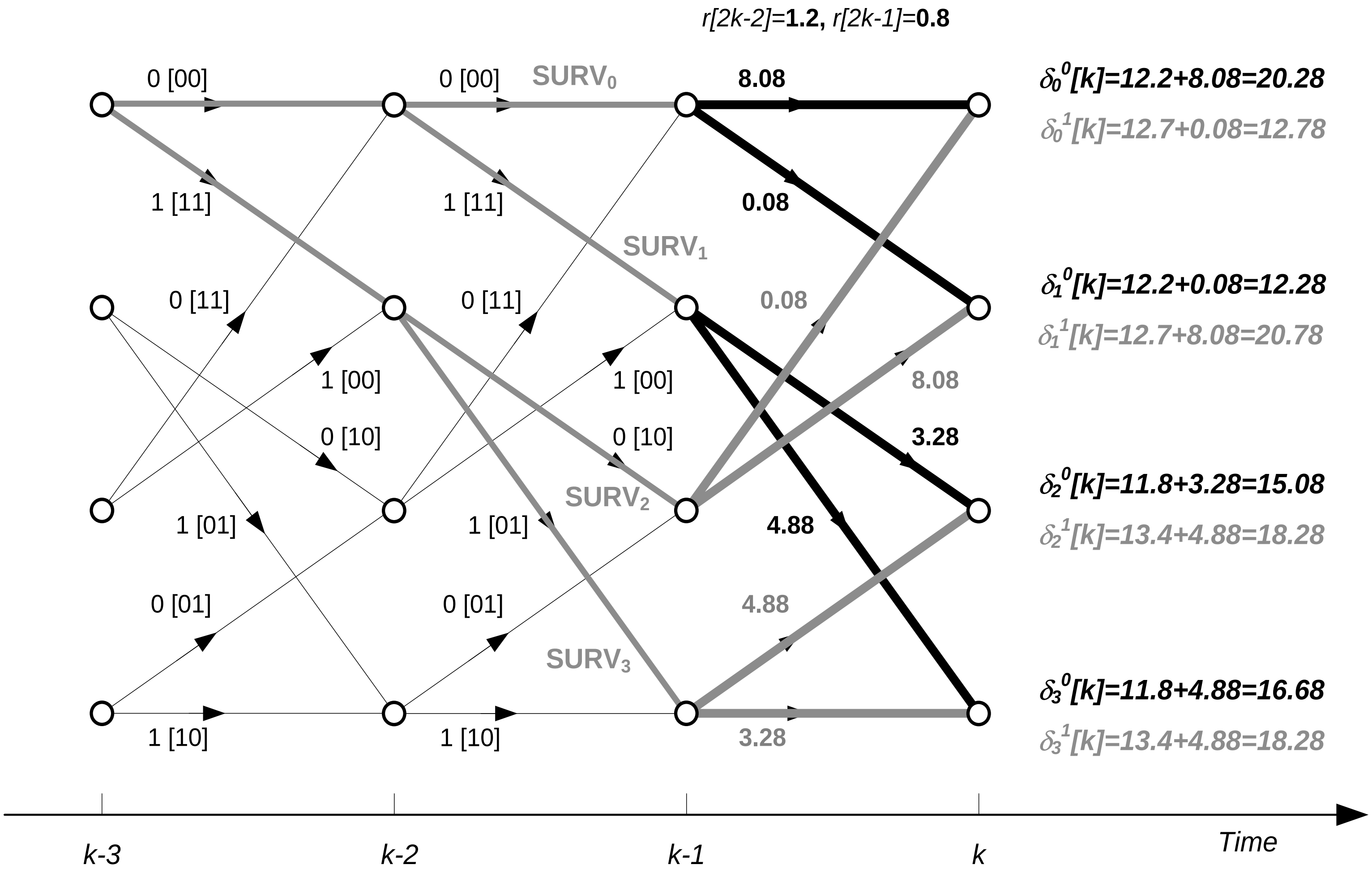
What does the VD do? 2/2

4. **For each state *i*, extend the survivors to time *k* by *comparing* the new accumulated metrics of the two survivors merging into that state, and then *selecting* the one with the smallest metrics:**

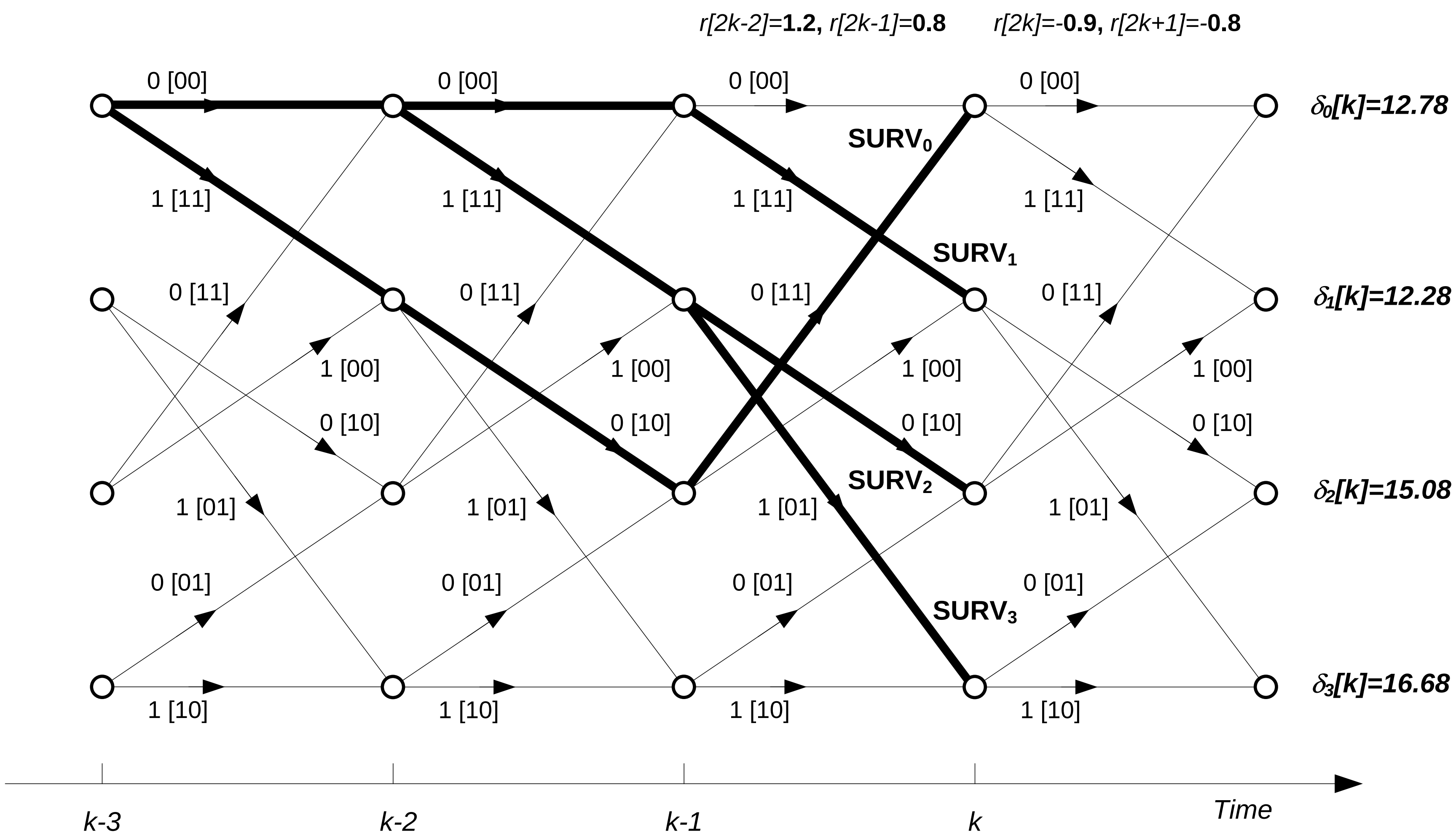
$$\delta_i[k] = \min\left(\delta_{i_0}[k-1] + \lambda^{(i_0,i)}[k], \delta_{i_1}[k-1] + \lambda^{(i_1,i)}[k]\right)$$

Add-Compare-Select

The Add-Compare-Select (ACS) Procedure



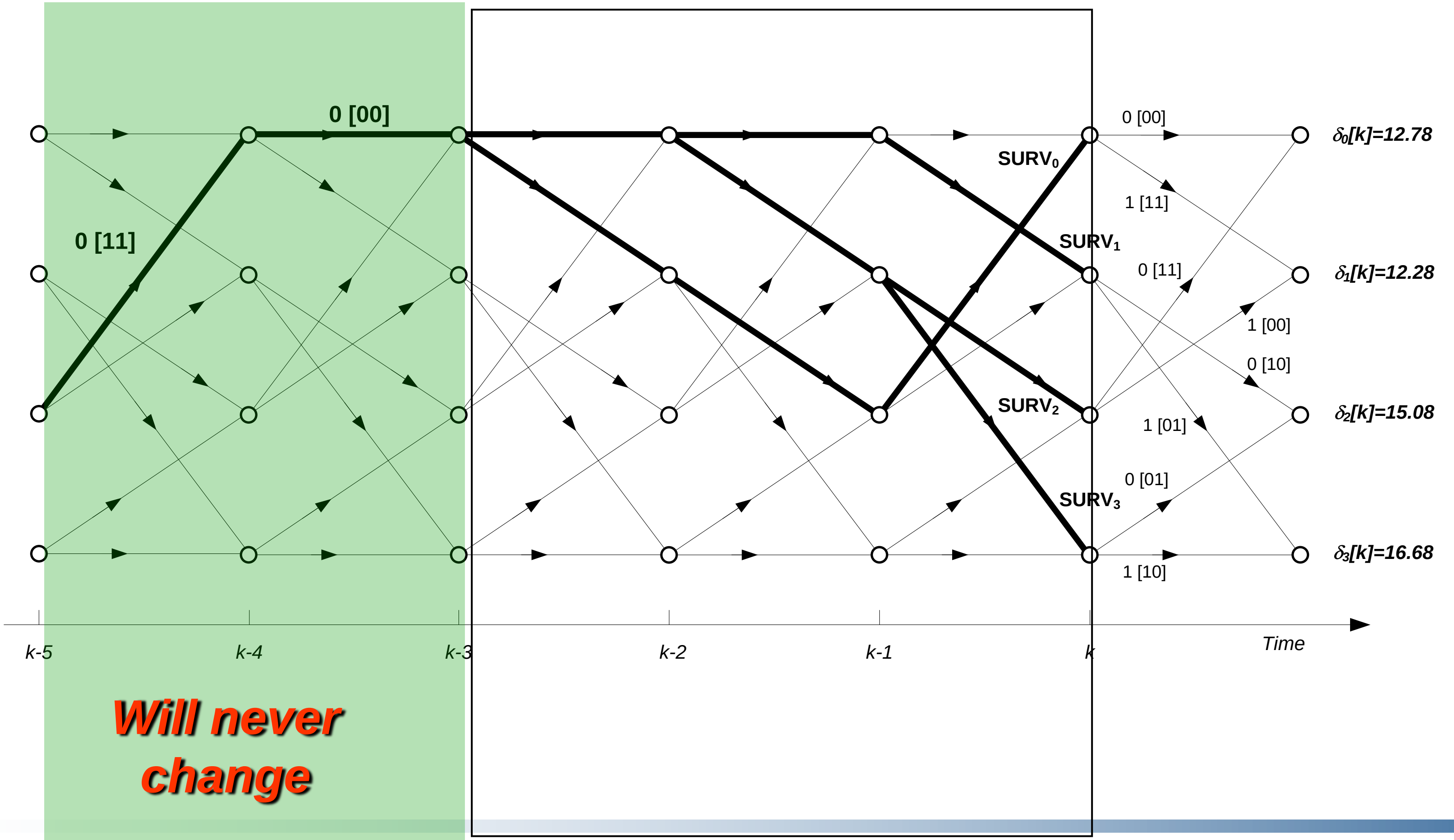
The Survivors at time *k*



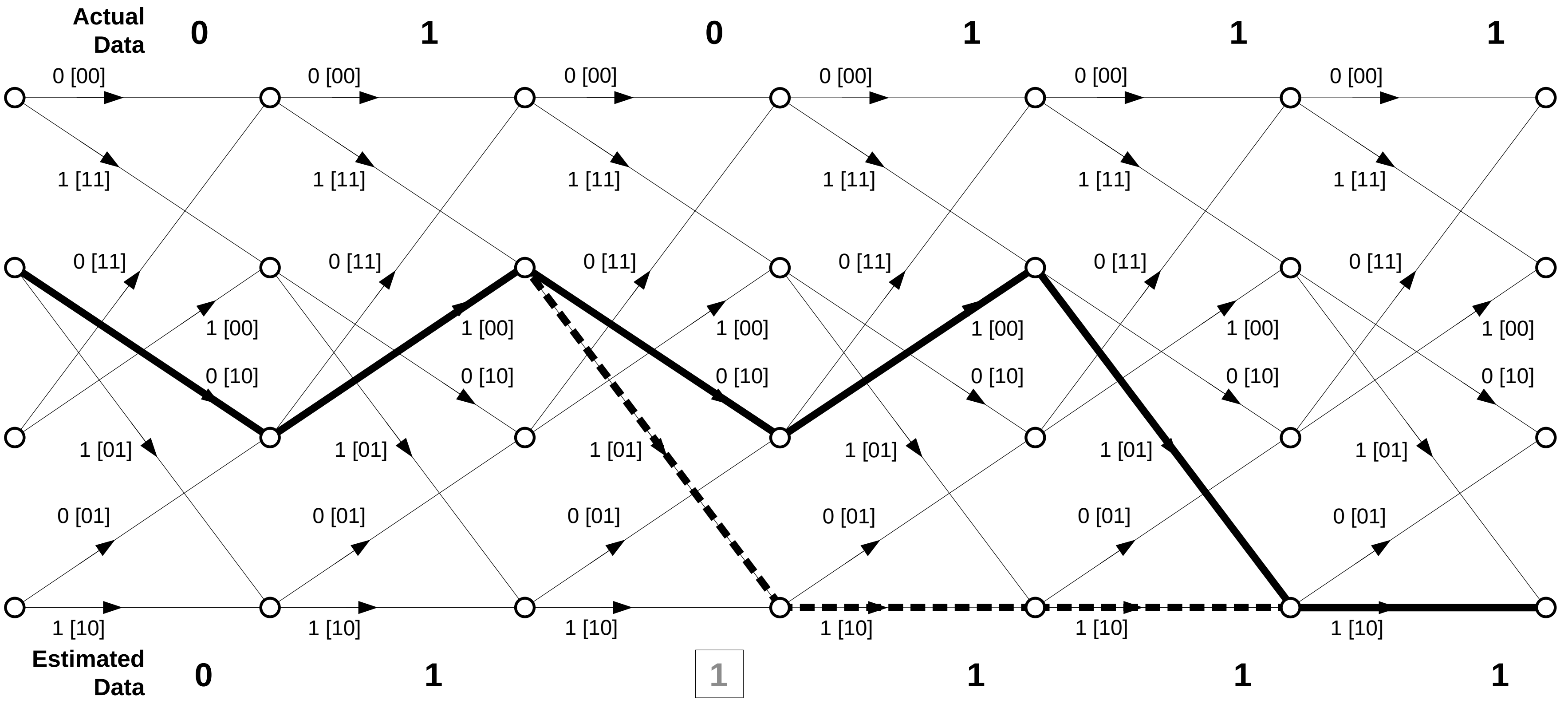
Dip. Ingegneria dell'Informazione
University of Pisa, Pisa, Italy

Back-Merging of Survivors

Decision Depth (4÷5)·K



Sample Error Event



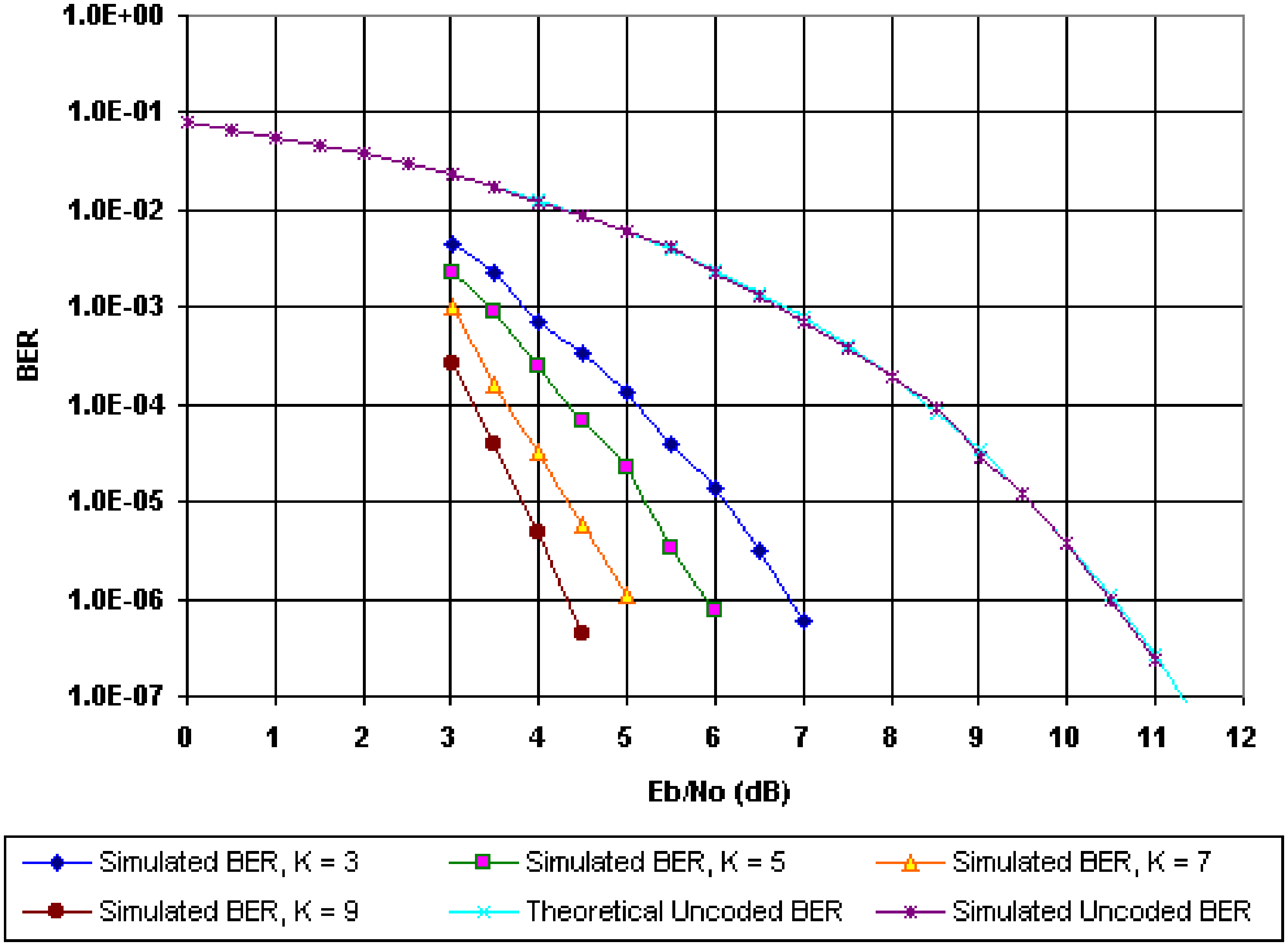
$$d_{free} \triangleq \lim_{M \to \infty} d_{\min}(M)$$

BER Performance

K	$(h_0[k])_8$	$(h_1[k])_8$	d_{free}
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

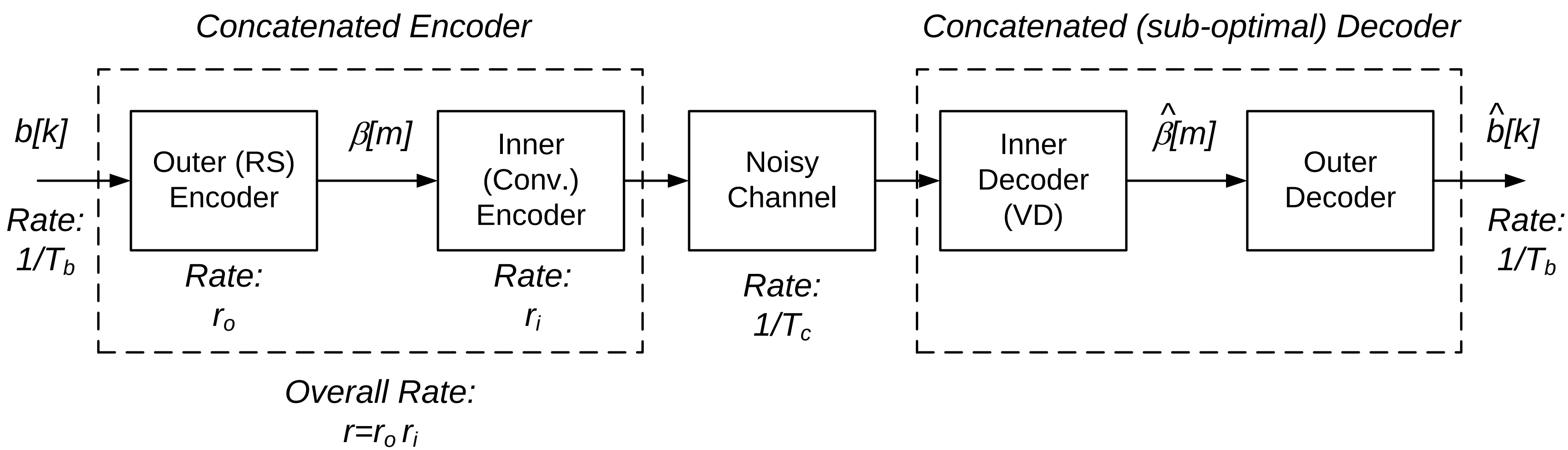
$$BER \geq N_{avg} Q\left(\sqrt{r \cdot d_{free} \frac{2E_b}{N_0}}\right)$$

BER of r=1/2 Best Conv Codes

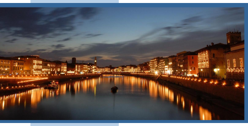


Concatenated Coding

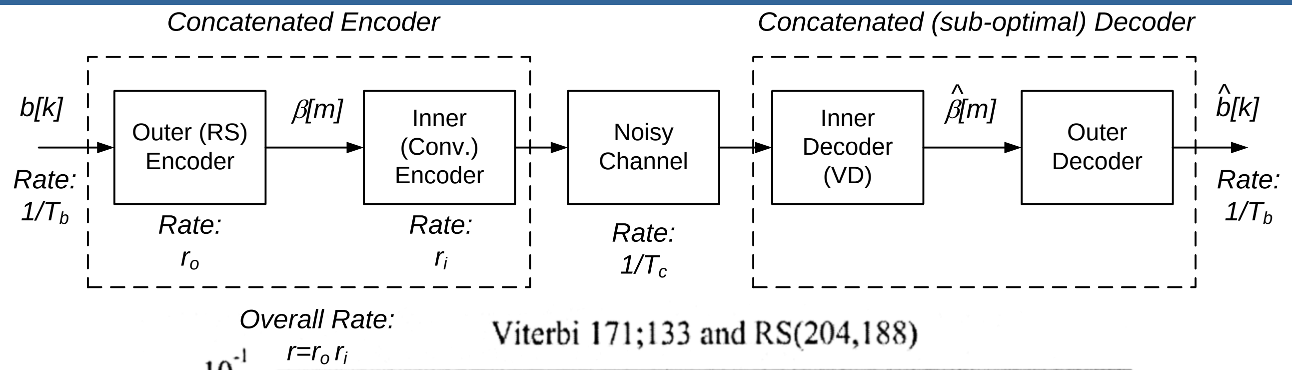
- **How to improve on the performance of one code?**
-
  - Let'us use TWO in cascade !

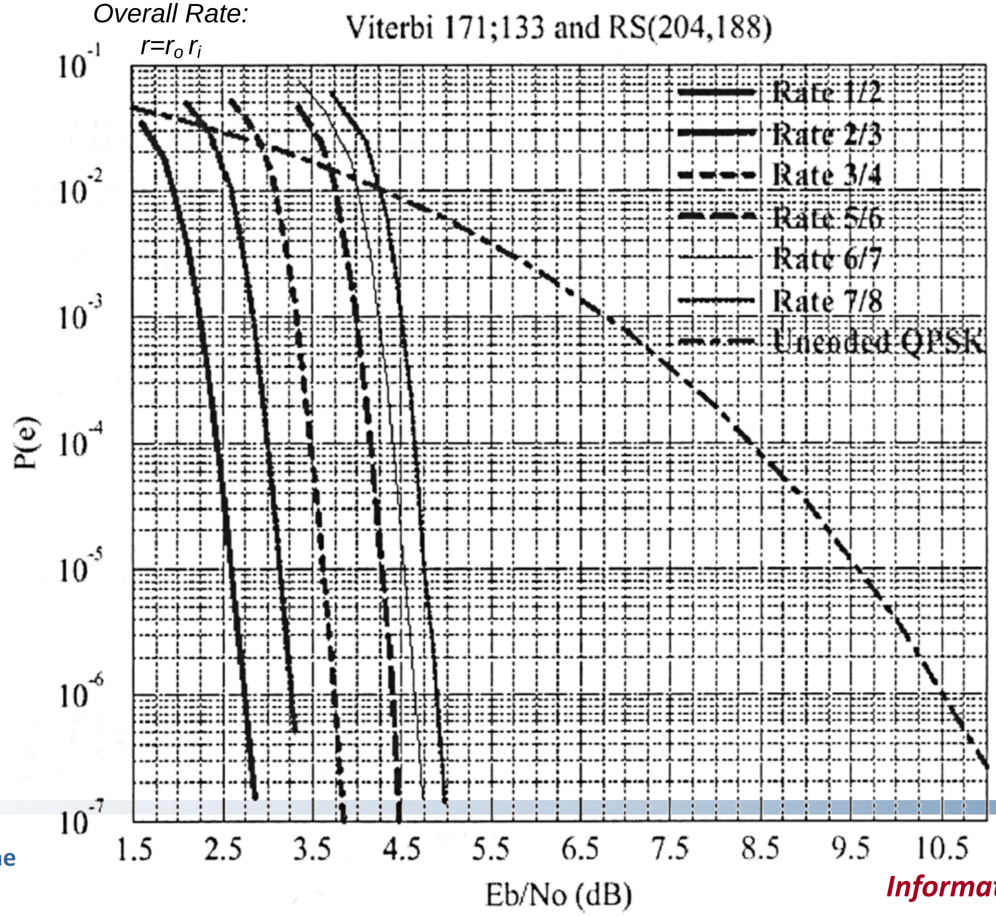


  - We call this arrangement *Concatenated Coding*



R-S & Convolutional Concatenated Coding

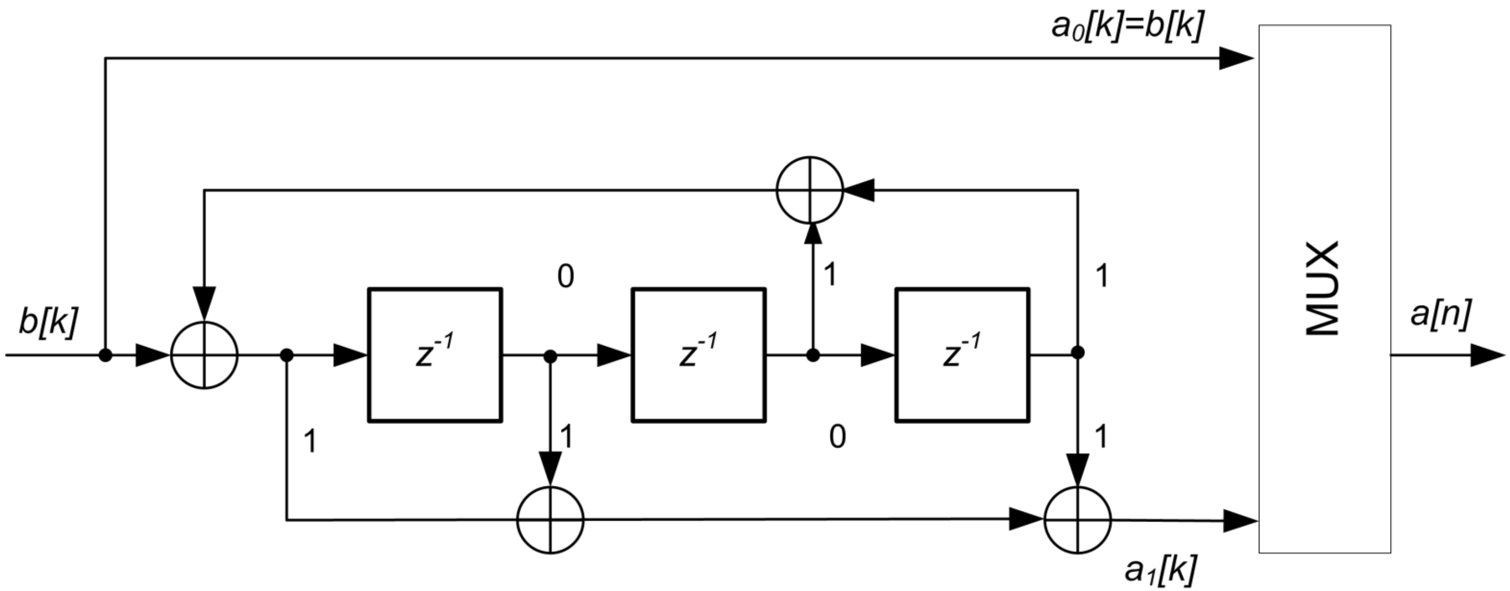




Dip. Ingegneria dell'Informazione
University of Pisa, Pisa, Italy

Marco Luise

Recursive Systematic Convolutional Codes



- It is not used as a *standalone* encoder – only concatenated with another encoder…