1 Data converters

1.1 Analog to Digital Converters: definitions and limits.

Definitions

An analog to digital converter (ADC) is a circuit that produces a numeric representation of an analog input quantity, typically a voltage or a current. In most cases, the output numeric representation consists in a digital binary code. In the following part of this document, we will consider that the input quantity is a voltage, indicated with v_{in} in Fig. 1.1.

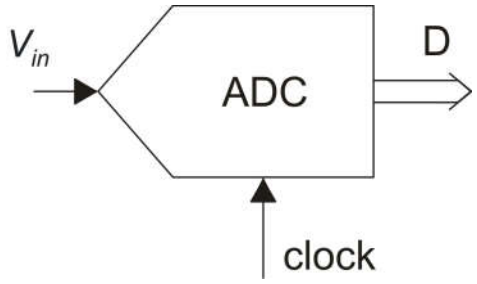

Fig.1.1. ADC symbol.

The output code D is formed by *n*-bits, therefore it can represent 2^n different numbers. We will assume that the reader is already familiar with the transfer function of an ADC and the main sources of error, such as offset, gain error, integral non-linearity (INL) and differential non-linearity (DNL). Furthermore, we will assume that the input voltage can be either unipolar (i.e. referred to ground) or differential.

The ADC is characterized by an input voltage range, which spans from two values indicated with V_{MIN} and V_{MAX}, where $V_{MIN} < V_{MAX}$ The magnitude of the interval is indicated with $V_{FS}=V_{MAX}-V_{MN}$.

We will refer to the following convention, where code *D* as a representation of an unsigned integer, varying between 0 and 2^n-1:

$$v_{in-dig} = V_{MIN} + \frac{V_{FS}}{2^n} D \tag{1.1}$$

In (1.1) $v_{in\text{-}dig}$ is the analog representation of code *D* in terms of input voltages. Voltage $v_{in\text{-}dig}$ must be a precise approximation of v_{in}. Clearly, due to the finite number of voltage levels that can be represented by all possible codes D, there will be a difference between v_{in} and $v_{in\text{-}dig}$. The minimum distance that, nominally, is present between two successive v_{in_dig} levels is generally indicated with LSB or with the symbol Δ. The LSB is given by:

$$\Delta \equiv LSB = \frac{V_{FS}}{2^n} \tag{1.2}$$

Note that, with this definition, LSB is a voltage, while the same acronym is used also to indicate the least significant bit in a binary code.

For the sake of the discussion that follows, it is important to divide the ADCs into two different categories:

- Nyquist-rate ADCs
- Oversampling ADCs

A Nyquist rate ADC performs the conversion without memory, thus the output code depends only on the present value of voltage v_{in}, with no contamination from previous conversions. In order to avoid signal distortion (aliasing), a Nyquist-rate ADC should respect the well know constraint on the sampling frequency (f_S):

$$f_S \geq 2B_S \tag{1.3}$$

where B_S is the signal bandwidth. If the sampling frequency is incremented over the minimum value, $2B_S$, there is no advantage in terms of resolution. Most widespread ADC solutions, such as the flash ADC, the SAR ADC and the counting ADC are practically Nyquist-rate converters.

An oversampling ADC operates at a frequency much higher than B_S and performs digital filtering of the output data. Due to filtering, the output code values depend also on the previous samples, not only on the present one. The benefit is an increase in resolution with respect to a Nyquist-rate converter in equal conditions of component precision. Generally speaking, an oversampling ADC shifts part of the complexity from the analog to the digital domain.

Factors that limit the resolution of a Nyquist-rate converter.

Figure 1.2 shows a generic architecture that may represent a wide class of Nyquist rate ADCs. The control logic produces a sequence of output codes that are converted into an analog voltage by the DAC (digital to analog converter) and compared with the input signal. At the end of the conversion cycle, the code that represents the result of the conversion remains on the output terminal. This code, processed by the DAC, produces the best approximation of the input voltage among all possible DAC outputs. In a SAR converter, the set of all possible codes is explored by means of a binary search. In a simple counting ADC, all possible codes are explored sequentially until the best match between v_{in} and the DAC output is found. A Flash converted can be considered as a counting DAC where comparison of v_{in} with all possible DAC outputs is performed in a parallel fashion.

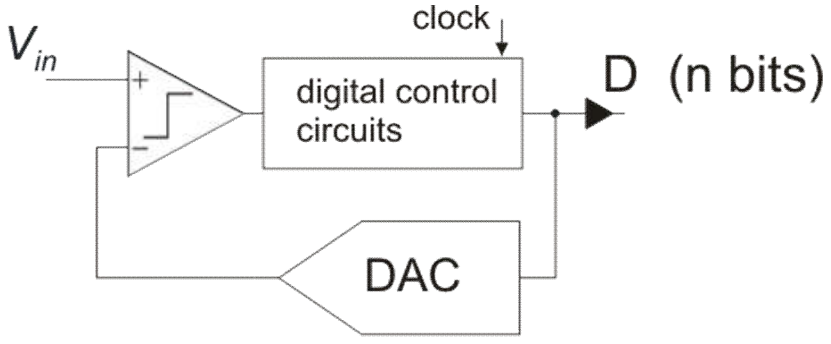

Fig. 1.2. Simplified architecture representing a SAR or a counting ADC.

In all the examined cases, the precision of the ADC is strongly related to the DAC precision. In particular, the ADC DNL and INL practically coincides with the DAC DNL and INL.

Now consider a popular DAC implementation, given by the programmable voltage divider of Fig. 1.3. The number of switch is equal to the number of resistors, which, in turn, is equal to 2^n. In practice, for each possible value of the DAC output voltage there is one resistor and a switch. The switches are controlled by a decoder in such a way that, for each possible value of the input code D, only one switch is on. This DAC architecture, indicated as "resistor string" DAC, is preferred for high-resolution converters since it guarantees a DNL < 1 LSB, or, in other terms a monotonic characteristic. However, like all other DAC architectures, the resistor string is affected by INL deriving from mismatch between the single elements that form the string. For simplicity, the DAC shown in Fig. 1.3 represents the case V_{MIN}=0, V_{MAX}=V_{ref}, then V_{FS}=V_{ref}.

In order to estimate the INL, let us consider the case depicted in Fig.1.3 (right), which refers to the case when the mid-range value, $V_{ref}/2$, is selected. This occurs when the selected switch picks up the voltage, which is located just in the central point of the voltage divider. Then, nominally, the voltage divider is split into two resistors, R_1 and R_2 in the figure, which connect the output to *gnd* and V_{ref}, respectively. We can calculate the error deriving from a mismatch between R_1 and R_2. As usual, we define $\overline{R}$ and ΔR such that:

$$R_1 = \overline{R} + \frac{\Delta R}{2}, \quad R_2 = \overline{R} - \frac{\Delta R}{2} \tag{1.4}$$

With these definitions:

$$V_{out} = V_{ref}\frac{R_1}{R_1 + R_2} = \frac{\overline{R} + \frac{\Delta R}{2}}{2\overline{R}} = \frac{V_{ref}}{2}\left(1 + \frac{\Delta R}{2\overline{R}}\right) \tag{1.5}$$

The INL error is the difference between the nominal value ($V_{ref}/2$) and the actual value given by (1.5). It is convenient to express the error in units of LSB:

$$\frac{\Delta V_{out}}{LSB} = \frac{V_{ref}}{4}\frac{\Delta R}{\overline{R}}\frac{2^n}{V_{ref}} = \frac{\Delta R}{\overline{R}}2^{n-2} \tag{1.6}$$

In order to have an error < 0.5 LSB, the relative matching error between R_2 and R_1 should be less than $1/2^{n-1}$. This means, for example, that in a 12 bit DAC the matching error between the two halves of the voltage divider should be less than 0.05 %. This is actually possible to obtain with a proper layout and the use of non-minimum area resistors. However, higher resolutions generally need complicated post-fabrication trimming techniques.

Since the DAC accuracy limits the accuracy of all ADCs corresponding to the block diagram of Fig. 1.2, we can conclude that the resolution (number of bits) of SAR, flash and counting ADCs is limited to 12-13 bits, unless trimming strategies are employed.

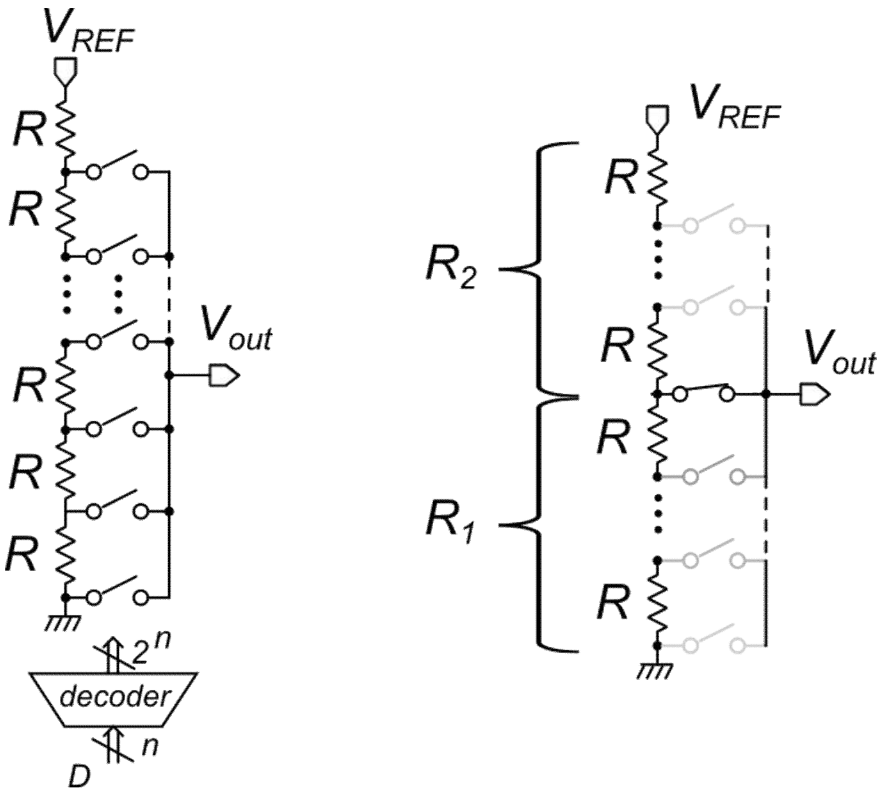

Fig. 1.3. (left) Resistor string DAC. (right) half range case (worst case for INL)

Oversampling ADCs allow overcoming this limitation without the need of increased component accuracy. Since oversampling ADCs base their function on filtering out the quantization noise in the digital domain, it is important to consider the relationship between effective resolution and noise. These arguments are presented in next paragraph.

Signal to noise ratios and effective resolution

First, let us consider an equivalent circuit that takes into account the errors introduced by the ADC. This simple equivalent circuit is shown in Fig.1.4 The output code D is transformed into the representation (v_{in_dig}) that it gives of the input voltage according to Eq. (1.1). The noise component v_n accounts for the difference between v_{in} (ideal) and the digital representation v_{in_dig}. The effects of discretization can be ascribed to a particular noise component: the quantization noise v_{nq}.

Quantization noise is illustrated by Fig.1.5 for a Nyquist-Rate converter, where a univocal relationship between input voltage and output code (or, equivalently, voltage representation of the code). Voltage v_{in1} represents a possible analog input voltage. Due to the finite number of output levels, voltage v_{in1} is converted to a representation (v_{in_dig}) that differs from the input value. The difference between the output voltage representation and the input voltage is just the quantization noise v_{nq}. Note that v_{nq} depends on the input signal: for particular values of the input voltages, such as v_{in2}, which coincides with one of the possible output representations, quantization error can be even zero.

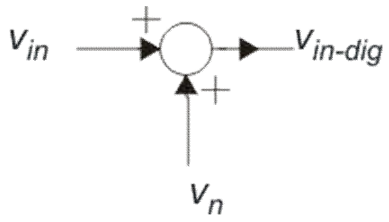

Fig.1.4. Equivalent model of the ADC

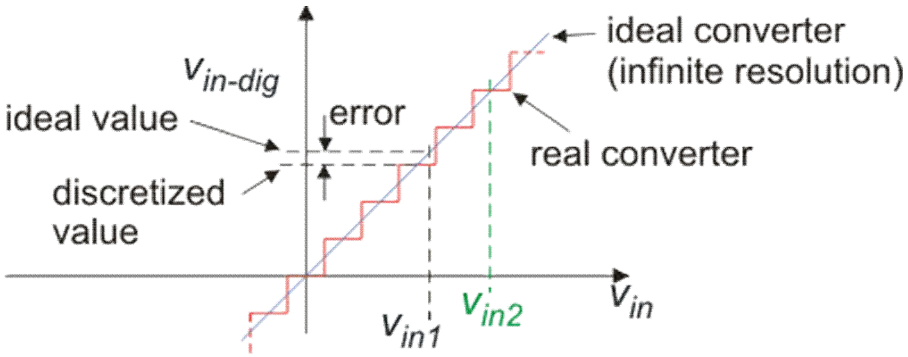

Fig. 1.5. Input-output characteristic of a Nyquist Rete converter, illustrating the origin of quantization noise.

Considering that the possible input values are uniformly distributed within the input range of the ADC, the mean square value of the quantization noise (i.e. the power of the quantization noise) is given by:

$$\left\langle v_{nq}^2 \right\rangle = \frac{\Delta^2}{12} \tag{1.7}$$

It is now important to consider an important parameter, which is the *SNR* (signal-to-noise ratio) of the ADC. This parameter is defined by:

$$SNR = \frac{P_{MAX}}{\left\langle v_n^2 \right\rangle} \tag{1.8}$$

where P_{MAX} is the power (i.e. mean square value) of the maximum sinusoidal waveform that can be represented by the ADC, while the noise at the denominator may change depending on the particular definition of the *SNR*. Generally, a sinusoidal waveform is also indicated as a “tone”, or “single tone”. Fig. 1.6 illustrates that the maximum peak-to-peak magnitude of a waveform that can be applied to the ADC without causing saturation is just $V_{MAX}–V_{MIN}=V_{FS}$ Then, for a sinusoidal waveform, the maximum power is:

$$P_{MAX} = \frac{V_{FS}^2}{8} \tag{1.9}$$

If the noise voltage considered in (1.8) is only the quantization noise, than, using (1.7), (1.9) and recalling (1.2), it is possible to derive the following expression:

$$SNR = \frac{V_{FS}^2}{8}\frac{12}{\Delta^2} = \frac{3}{2}2^{2n} \tag{1.10}$$

Expressing the SNR in decibels, we get a very popular equation:

$$SNR_{dB} = 10\log_{10}(SNR) \cong 6.02n + 1.76 \tag{1.11}$$

Expression (1.11) refers to the SNR calculated considering only quantization noise. However, the presence of other noise sources (such as electronic noise) may have a similar effect to quantization noise, since it limits the minimum difference of the input quantity that can be reliably detected. In other words, all source of noise limit the resolution. As a result, equation (1.11) is often used to determine a resolution (number of bits) from an SNR definition that takes also into account noise components other than quantization one.

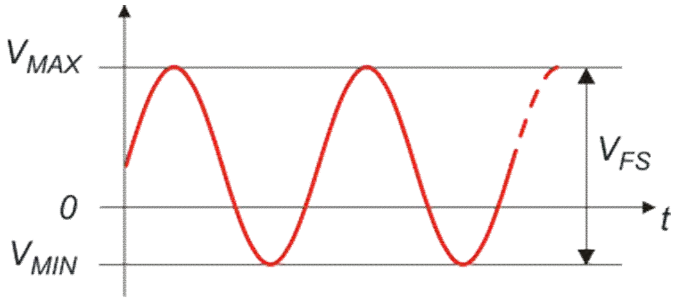

Fig. 1.6. A simusiod of maximum power.

In particular, it is possible to take into account even distortion through the parameter SINAD (signal to noise and distortion), also indicated as SNDR (signal to noise and distortion ratio), defined as:

$$SINAD = \frac{P_{MAX}}{\langle v_n^2 \rangle + P_D} \tag{1.12}$$

where P_D is the power of the harmonics other than the fundamental. Generally, P_D is calculated over a reduced number of harmonics. Using (1.11) with SNR=SINAD leads to the definition of the "effective number of bits" (ENOB) of a converter:

$$ENOB = \frac{SINAD_{dB} - 1.76}{6.02} \tag{1.13}$$

Another important consideration that will be used later descends from the relationship between SNR and resolution. Let us consider two different ADCs of resolution n_1 and n_2 and indicate with SNR_1 and SNR_2 their respective signal-to-noise ratios. From (1.10) we obtain:

$$\frac{SNR_2}{SNR_1} = 2^{2(n_2 - n_1)} \tag{1.14}$$

If the two converter has the same full-scale range, V_{FS}, then the maximum power of a single tone, P_{MAX}, is the same for the two converters. Then, using (1.8), the ratio SNR_2/SNR_1 becomes:

$$\frac{SNR_2}{SNR_1} = \frac{\left\langle v_{n1}^2 \right\rangle}{\left\langle v_{n2}^2 \right\rangle} \tag{1.15}$$

Obtaining $n_2 - n_1$ from (1.14) and using (1.15), the following expression can be derived:

$$n_2 - n_1 = \frac{1}{2}\log_2\left(\frac{\left\langle v_{n1}^2 \right\rangle}{\left\langle v_{n2}^2 \right\rangle}\right) \tag{1.16}$$

Equation (1.16) shows the relationship between the difference in resolution between two ADCs and the ratio of their respective noise powers. This equation is important because it explains how an increase in resolution can be obtained by reducing the noise power of a converter. In next section, it will be shown how this goal can be accomplished by means of oversampling.

1.2 Oversampling data converters

In order to understand the oversampling mechanism, let us consider Fig. 1.7, showing the result of sampling in the frequency domain.

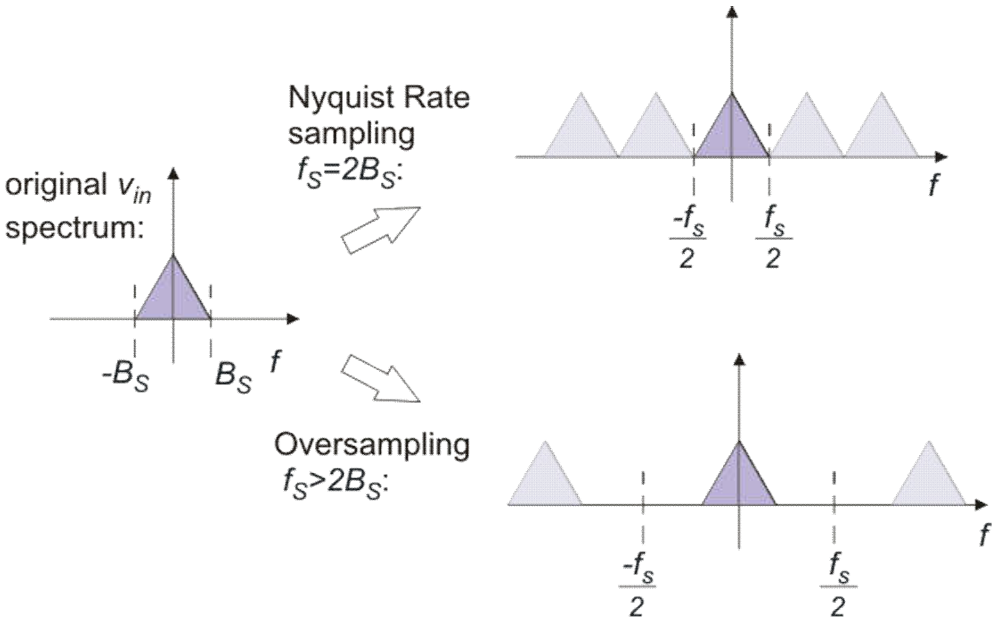

Fig. 1.7. Consequence of sampling in the frequency domain with and without oversampling

The frequency interval to be considered for the sampled signal (discrete time domain) extends only between $-f_S/2$ and $f_S/2$, where f_S is the sampling frequency. The replicas of the original signal spectrum, indicated in lighter color are used only to build the spectrum within the $[-f_S/2, f_S/2]$ interval. Since aliasing does not occur in both cases of Nyquist-Rate sampling and oversampling, replicas have no effect on the final spectrum and could be ignored. The main difference between the two cases are:

- In the case of Nyquist-Rate sampling, the frequency domain of the sampled signal coincides with the signal bandwidth, so that it is not possible to apply filtering in the discrete time domain (sampled data sequence) without altering the signal of interest.
- In the case of oversampling, the frequency domain is larger than the signal bandwidth, so that it is possible to apply a low pass filter to the output data stream with a cut-off frequency f_H, such that $B_S < f_H < f_S$, with no effect on the signal.

Indeed, filtering is applied to reduce quantization noise, improving the effective resolution. Fig. 1.8 shows the spectral density of quantization noise according to a widely accepted model that assumes:

1. The spectral density of quantization noise is constant over the whole discrete-time frequency domain.
2. Quantization noise and signal are uncorrelated.

This model is correct when the input signal is such that many different levels of the ADC are visited with a relatively high frequency. The model fails in describing the quantization noise density when the input signal is constant (i.e. a dc signal), of small amplitude (of the same order of the LSB or smaller) or slowly varying. In the following discussion, we will apply the uniform density model of quantization noise, while keeping in mind that it may lead to wrong results when the hypothesis of a large and fast input signal does not hold.

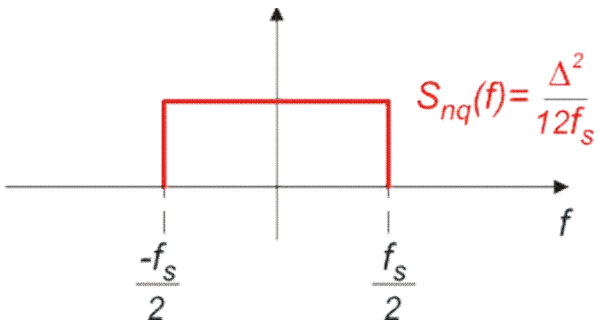

Fig. 1.8. Power spectral density of quantization noise according to the uniform density model

Note that the total power of quantization noise, given by (1.7), is independent on the sampling frequency. Therefore, if the sampling frequency is increased, then the density get smaller, since the same noise power spreads over a larger frequency interval. Fig. 1.9 shows the quantization noise PSDs, $S_{nq}(f)$ and $S'_{nq}(f)$ of the same ADC for two different sampling frequencies f_S and f_S', respectively. Since the total noise power is unchanged, then the following relationship should hold:

$$S'_{nq}(f) = S_{nq}(f)\frac{f_S}{f_S'} \tag{1.17}$$

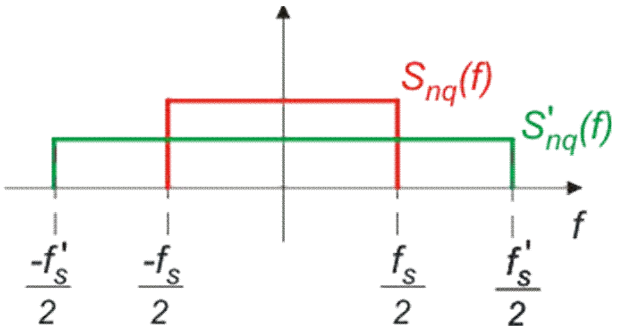

Fig. 1.9. Power spectral density of quantization noise for two sampling frequencies.

This phenomenon is used to reduce the spectral density in the signal bandwidth in order to reduce the total quantization noise. The principle is illustrated in Fig. 1.10. If a converter is operated at a sampling frequency equal to the Nyquist rate, i.e. $f_S=2B_S$, then the situation depicted in Fig. 1.10 (a) occurs. The spectral density in this case is indicated with $S_{n\text{-}NR}(f)$. Figure 1.10 (b) shows the same converter operated with oversampling. The sampling frequency, in this case is higher than the sampling rate by a factor r_{OS}, called "oversampling ratio", commonly indicated also with OSR.

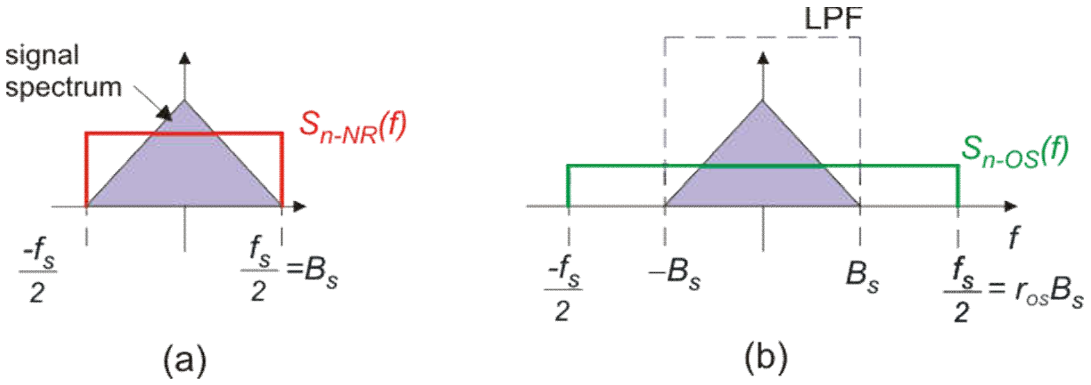

Fig. 1.10. Quanyization noise reduction by means of oversampling.

Since, in the case of oversampling, the discrete time frequency domain is wider than the signal bandwidth, it is now possible to apply a low pass filter (dashed line in the figure, showing an ideal low-pass filter). This filter selects the signal spectrum and rejects part of the noise components. This is not possible if the sampling frequency is equal to the sampling rate, since the width of frequency domain coincides with the signal bandwidth and, and anticipated earlier, any filtering function would alter the signal. The block diagram of a simple oversampling converter is shown in Fig. 1.11. The low pass filter operates on the digital data stream (D_{st}) produced by the original ADC. The total power of the quantization noise that affects the data at the output of the filter (D_{out}) is given by:

$$\left\langle v_{n-OS}^2 \right\rangle = 2B_S S_{n-OS}(f) \tag{1.18}$$

Applying (1.17) with f_S=Nyquist rate=$2B_S$ and $f_S'=f_S/r_{OS}$, we find that $S_{n\text{-}OS}=S_{n\text{-}NR}/r_{OS}$, then:

$$\left\langle v_{n-OS}^2 \right\rangle = 2B_S \frac{S_{n-NR}(f)}{r_{OS}} = \frac{\left\langle v_{n-NR}^2 \right\rangle}{r_{OS}} \tag{1.19}$$

Equation (1.19) shows that the quantization noise power at the output of the filter is diminished by the OSR factor.

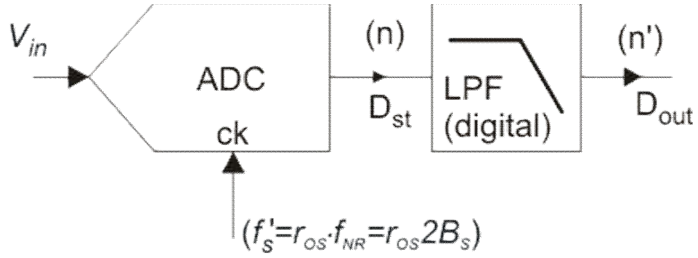

Fig. 1.11. Block diagram of an oversampling converter.

Note that the block indicated with ADC in Fig.1.11 is still a conventional Nyquist-rate converter (e.g. a SAR or a flash converter) which is operated at a sampling frequency higher than the Nyquist-rate. The oversampling converter is the combination of this "oversampled" ADC and the digital low pass filter. If n is the resolution of the original ADC and n' is the effective number of bits (due only to quantization noise) of the output data stream D_{out}, then the increment in resolution can be calculated from (1.16):

$$n'-n = \frac{1}{2}\log_2\left(\frac{\left\langle v_{n-NR}^2 \right\rangle}{\left\langle v_{n-OS}^2 \right\rangle}\right) = \frac{1}{2}\log_2\left(r_{OS}\right) \tag{1.20}$$

As a result, we gain one bit of resolution every time the r_{OS} is increased by a factor of four. It is important to stress the fact that this increase is not obtained by improving the component accuracy, but only at the cost of a sampling frequency much higher than the minimum value required for signal integrity, which is the Nyquist rate.

The simple oversampling approach has two important drawbacks. First, it is not efficient: in order to gain one bit we need to operate at a sampling frequency that is four times the Nyquist rate. If we need to achieve a more significant improvement, for example 4 bits, we need to operate at a frequency that is 4^4=256 higher than the Nyquist rate. This means a much higher power consumption for a modest resolution increase.

The second problem stems from the mentioned limits of the uniform density model of the quantization noise. Consider the block diagram of Fig. 1.11 with a constant input (dc voltage). Since the ADC is a Nyquist-rate converter, its output will be determined by only the present input, according to a deterministic law as that of Fig. 1.5. The output D_{st} of the ADC is then a constant code. If the input value does not coincide with one of the particular voltage levels that are represented with a zero error, then the output code will correspond to an input voltage v_{in_dig} which differs from the actual input v_{in}. This difference is the quantization error. Since it is constant, its spectrum is a Dirac delta function, placed at f=0. With such a noise PSD, the low pass filter is useless and the output stream D_{out} will be affected by the same error as D_{st}. As a result, for dc inputs the oversampling ADC does not have any

advantage in terms of resolution. To overcome the limitation occurring with dc inputs, electrical noise is sometimes artificially introduced at the input of the amplifier. This technique is called "dithering". Electrical noise, whose magnitude is at most a few LSB, cause the output of the ADC to switch across a few adjacent codes in a random fashion. This switching effect operates a sort of modulation of the dc quantization noise, spreading its energy across the whole discrete-time frequency interval. The LPF can now reject the noise even when the input signal is a dc or slowly varying signal. Since the noise is chosen in such a way to have most of its spectral components out of the signal bandwidth, the LPF remove also most of the noise introduced to implement the dithering approach.

The way the dithering approach works is illustrated in Fig. 1.12. V_k and V_{k+1} are two input voltages corresponding to two adjacent output codes of the ADC. The actual input voltage, supposed constant, is indicated with V_{dc}. The mean of V_k and V_{k+1} is the threshold, which, when crossed, determine the switching between the two codes. Since V_{dc} is below the threshold, it would cause the V_{k+1} level (i.e. the code that represent it) to be constantly present on the output of the ADC. As we have already explained, in this case the LPF is ineffective. Adding the dithering noise ($v_{n\text{-}dt}$ in the figure) makes the input signal to cross the threshold occasionally. Then, the ADC output ($v_{in\text{-}dig}$) will vary between the two levels and the LPF, which operates a sort of average, will yield an intermediate value between V_k and V_{k+1}. This corresponds to increase the resolution of the ADC, since it is now possible to represent input voltages that are located between two adjacent levels of the original ADC. Clearly, if Vdc is closer to one of the two levels V_k and V_{k+1} , then that level will appear at the ADC output more frequently than the other one, and the average produced by the LPF will be closer to that level.

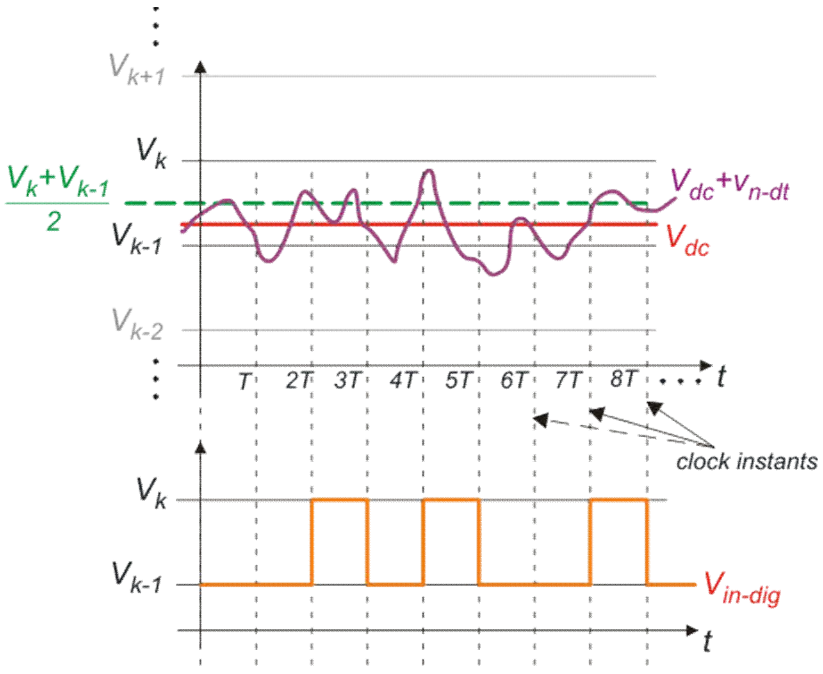

Fig. 1.12. Operating principle of the dithering method.

Although dithering solves the problems occurring with dc signals, oversampling alone is still an ineffective way to increase the ADC resolution, since, as we have seen, to gain a modest number of

bits, very large OSRs are required. Such a limitation is solved by means of the delta-sigma converter described in next section.

1.3 Delta sigma converters

A delta-sigma converter (Δ-Σ), often indicated also with sigma-delta, is a converter that combines two different principles, namely:

- Oversampling
- Noise shaping

Noise shaping refers to quantization noise. The effect of noise shaping is equivalent to high-pass filtering the quantization noise, in such a way that the total power is not altered, but most of the power is shifted out of the signal bandwidth.

Delta sigma converters include a large variety of different architectures. The most important parameter of delta-sigma converters is the order. The higher the order, the more aggressive is noise shaping, leaving less noise in the signal bandwidth. In this document, we analyze the simpler case, i.e. the first order, discrete-time delta-sigma ADC, represented by the block diagram of Fig. 1.13.

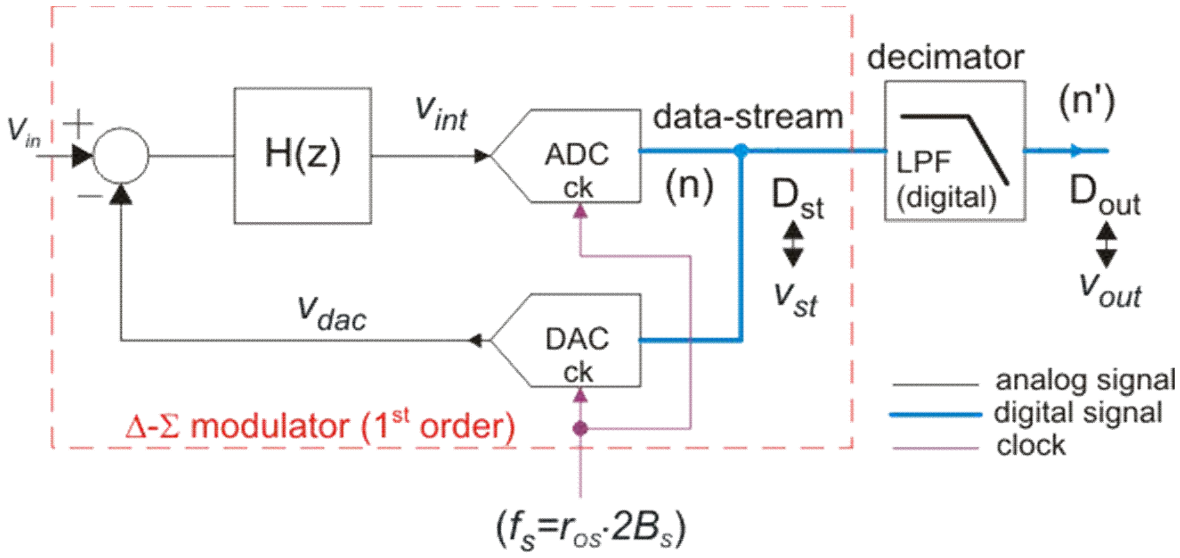

Fig. 1.13. Block diagram of a first-order delta-sigma ADC.

The section included into the red-dashed box is called delta-sigma modulator. It is an all-analog circuit, while the output LPF operates in the digital domain. The ADC and DAC have the same number of bits, indicated with n. This is also the resolution of the data stream (D_{st}) produced by the ADC. The representation of D_{st} digital codes in terms of input signals of the ADC are indicated with v_{st}. The DAC, ideally transforms the digital code into its voltage representation. In practice, as we have seen, the DAC is marked by errors, so that v_{dac} (the output voltage of the DAC) can be different from v_{st}. In the following analysis we will assume that the DAC operates in an ideal way, so that v_{dac}=v_{st}. DAC errors will be discussed later.

Note that the ADC is placed in the forward path of the modulator, while the DAC is in the feedback path. Voltage v_{dac} is often referred to as the feedback signal. We will also assume that the input (and feedback signal) are sampled at the input of block *H*. For this reason, block *H* (the "loop filter") operates in the discrete-time domain. As such, block *H* is characterized by a Z-domain transfer function.

It is possible to study the delta-sigma modulator using the linearized model shown in Fig. 1.14, where the digital code has been replaced by its voltage representation. The ADC is represented by the model of Fig. 1.4, where the error due to finite resolution is introduced by quantization noise v_{nq}.

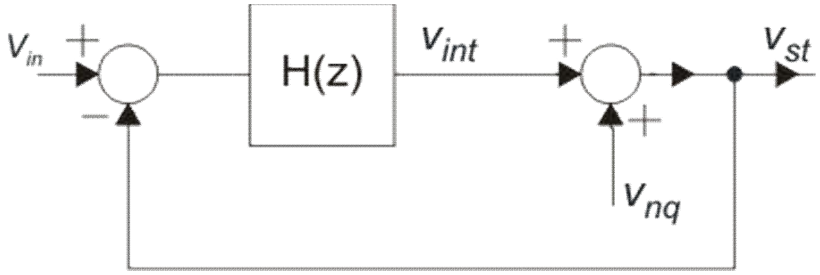

Fig. 1.14. Linearized model of the first-order delta-sigma modulator.

As we have stated earlier, the DAC is considered ideal and then it is not present in the feedback path (since $v_{dac}=v_{st}$). Analysis of the model in Fig. 1.14, produces the following expression of the output data stream v_{st} as a function of the input signal and the quantization noise:

$$v_{st}(z) = v_{in}(z) \cdot STF(z) + v_{nq}(z) \cdot NTF(z) \tag{1.21}$$

where:

$$STF(z) = \frac{H(z)}{1+H(z)}; \quad \text{"Signal Transfer Function"} \tag{1.22}$$

$$NTF(z) = \frac{1}{1+H(z)}; \quad \text{"Noise Transfer Function"} \tag{1.23}$$

Let us consider the signal bandwidth that, in an oversampling converter is located at frequencies much smaller than $f_S/2$, In the signal bandwidth, the target is obtaining a signal transfer function which is close to one and a noise transfer function as low as possible. Considering also stability issues (not analyzed in this document for simplicity), the ideal form for *H(z)* that allows satisfying both requirements is that of a discrete-time integrator. In particular:

$$H(z) = \frac{z^{-1}}{1-z^{-1}} \tag{1.24}$$

This function can be implemented with the simple block diagram in Fig. 1.15, where block "T" represents a delay of one clock cycle. With *H*(*z*) given in (1.24), the *STF* and *NTF* become:

$$STF(z) = \frac{\dfrac{z^{-1}}{1-z^{-1}}}{1+\dfrac{z^{-1}}{1-z^{-1}}} = z^{-1} \tag{1.25}$$

$$NTF(z) = \frac{1}{1+\dfrac{z^{-1}}{1-z^{-1}}} = 1-z^{-1} \tag{1.26}$$

The *STF* is simply a delay of one clock cycle and then it does not alter the signal, as required. On the other hand, the *NTF* coincides with the discrete-time derivative operation and as such, it rejects the dc component and strongly reduces the components at low frequencies.

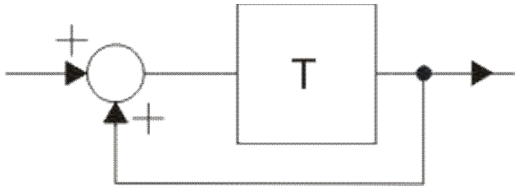

Fig. 1.15. Block diagram of the discrete-time integrator.

The denomination of the converter derives from the fact that the signal undergoes a difference ("delta") with the feedback signal and then the difference is integrated ("sigma", i.e. summed) by the integrator.

Let us now calculate the transfer function of the *NTF* in the frequency domain, by substituting z with $e^{j\omega T}$, where T is the clock period. We recall that the clock is at the sampling frequency f_S, which is r_{OS} times higher than the Nyquist-rate.

$$NTF(j\omega) = 1-e^{-j\omega T} = e^{-j\omega\frac{T}{2}}\left(e^{j\omega\frac{T}{2}} - e^{-j\omega\frac{T}{2}}\right) = e^{-j\omega\frac{T}{2}} \cdot 2j\sin\left(\omega\frac{T}{2}\right) \tag{1.27}$$

Replacing ω with $2\pi f$, we obtain the dependence of *NTF* on frequency, thus:

$$NTF(j\omega) = e^{-j\pi fT} \cdot 2j\sin(\pi fT) \tag{1.28}$$

The delta sigma-modulator produces a data stream with the same resolution of the original ADC included in the loop (see Fig. 1.13) but differently from a simple oversampling converter, the quantization noise of the data-stream is filtered (i.e. "shaped") by the NTF. The spectral density of the quantization noise at the modulator output, $S_{n\text{-}DS}(f)$ can be calculated considering by the following expression.

$$S_{n-DS}(f) = S_{n-OS}\left|NTF(f)\right|^2 = S_{n-OS} \cdot 4\sin^2\left(\pi\frac{f}{f_S}\right) \tag{1.29}$$

where $S_{n\text{-}OS}(f)$ is the spectral density produced by the original ADC operated at the oversampling frequency $f_S=1/T$. Figure 1.16 depicts the spectral density $S_{n\text{-}DS}(f)$, showing the shaping effect. Note the reduced density in the signal bandwidth, selected by the digital LPF. In practical cases the signal

bandwidth is much smaller than $fs/2$ (normally more than 100 timed smaller), thus the noise density in the bandwidth is orders of magnitude smaller than in the case of simple oversampling (density $S_{n\text{-}OS}$).

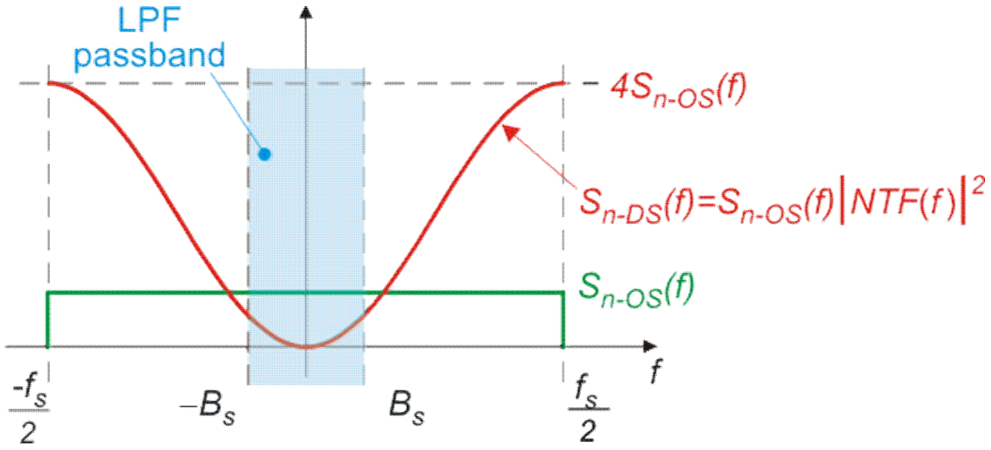

Fig. 1.16. Power spectral density of the quantization noise at the Δ-Σ modulator output.

In order to estimate the effective resolution increase, we have to calculate the total power of the quantization noise in the signal bandwidth, which is the noise power at the output of the LPF ($v_{n\text{-}out}$).

$$\left\langle v_{n-out}^2 \right\rangle = \int_{-B_S}^{B_S} S_{n-DS}(f)df = S_{n-OS}\int_{-B_S}^{B_S} 4\sin^2\left(\pi\frac{f}{f_S}\right)df \tag{1.30}$$

Since the integral involve only frequencies $f \le B_S$ and $f_S = 2r_{OS}B_S$ with $r_{OS}>>1$ (large oversampling ratio), then $f<B_S<<f_S$. For this reason the argument of the sine is << 1 and we can approximate the sine with its argument. As a result:

$$\left\langle v_{n-out}^2 \right\rangle \cong S_{n-OS}\int_{-B_S}^{B_S} 4\left(\pi\frac{f}{f_S}\right)^2 df = S_{n-OS}\frac{4\pi^2}{f_S^2}2\frac{B_S^3}{3} \tag{1.31}$$

Since $S_{n\text{-}OS}=S_{n\text{-}NR}/r_{OS}$, by means of simple algebraic transformations, we obtain the following expression:

$$\left\langle v_{n-out}^2 \right\rangle \cong \frac{\pi^2}{3}\left(\frac{2B_S}{f_S}\right)^2\left(\frac{2B_S S_{n-NR}}{r_{OS}}\right) = \frac{\pi^2}{3}\left(\frac{2B_S S_{n-NR}}{r_{OS}^3}\right) \tag{1.32}$$

Note that $2B_S S_{n\text{-}NR}$ is the total power of the quantization noise of the original ADC (which does not depend on the sampling frequency and is equal to $\Delta^2/12$). Then:

$$\frac{\left\langle v_{n-out}^2 \right\rangle}{\left\langle v_{nq-NR}^2 \right\rangle} \cong \frac{\pi^2}{3}\left(\frac{1}{r_{OS}^3}\right) \tag{1.33}$$

Applying (1.16), the resolution increase, *n'-n* is:

$$n'-n=\frac{1}{2}\log_2\left(\frac{\left\langle v_{nq-NR}^2\right\rangle}{\left\langle v_{n-out}^2\right\rangle}\right)\cong\frac{1}{2}\log_2\left(\frac{3}{\pi^2}r_{OS}^3\right)\cong\frac{3}{2}\log_2(r_{OS})-0.86 \tag{1.34}$$

where:

$$-0.86\cong\frac{1}{2}\log_2\left(\frac{3}{\pi^2}\right) \tag{1.35}$$

Equation (1.34) means that we have an increment of 1.5 bits every time the OSR (i.e. r_{OS}) is doubled. This is much more effective than the mere 0.5 bit of the simple oversampling converter. For example, with OSR=64 we gain 9 bit. If the ADC placed into the Δ-Σ modulator has 10 bit, then, with the architecture of Fig. 1.13 and OSR=64 we can virtually obtain a 19 bit converter, which is well beyond the possibilities of Nyquist-rate converters.

It is interesting to see what happens in the time-domain within the modulator loop when the input is a dc voltage. Fig. 1.17 shows the evolution of the main signal of the modulator. We will consider that the integrator *H(z)* starts with a zero output. Note that an integrator with the response indicated in (1.24) is simply an accumulator that, at each clock impulse, adds the input signal to the value accumulated at the output. Note that if the output of the integrator is zero, then also the ADC will produce initially a zero output.

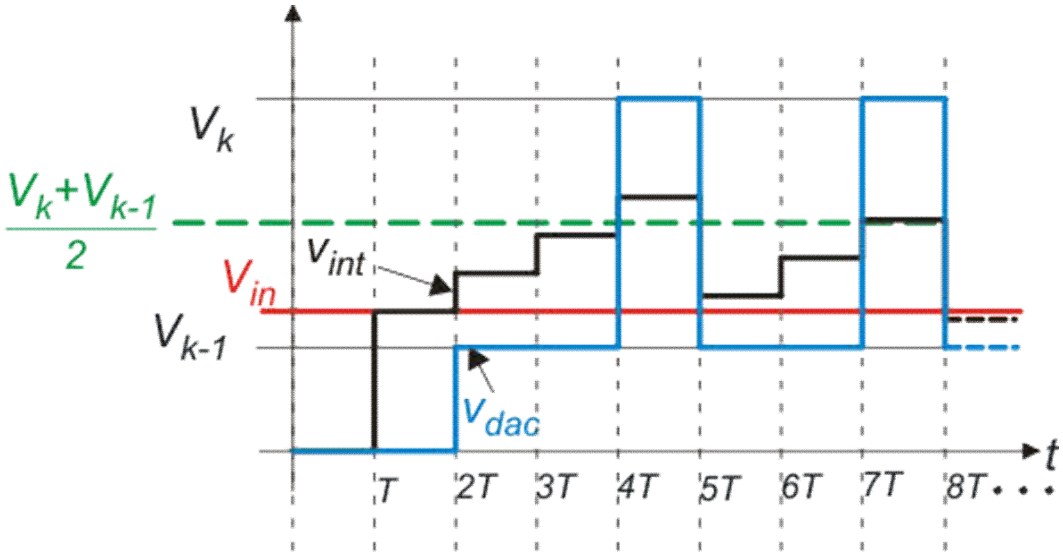

Fig. 1.17. Example of evolution of the main signals in the Δ-Σ modulator.

Therefore, at the first clock impulse (T), the integrator will add just v_{in} to its output, so that the integrator output jumps to v_{in}. At the next impulse ($2T$), the ADC approximates the integrator output to the closest available level (V_{k-1} in the figure) and the DAC will track the ADC output. The integrator will then integrate the difference between v_{in} and V_{k-1}. Since this difference is positive, the integrator output increases by a $v_{in}-V_{k-1}$ step. These steps are repeated until the integrator output stays below the threshold between the two adjacent levels (dashed line in the figure). When v_{int} passes the threshold, the

ADC and, consequently, the DAC will change their output to V_k (instant $4T$). At the next clock cycle ($5T$), since $V_k > v_{in}$, the integrator will decrease by $V_k - v_{in}$. From this instant on, voltage v_{int} will evolve with a sequence of positive increments followed by one (bigger) negative step and so on. As we have seen, the two steps are given by:

$$\begin{cases} v_{in} - V_{k-1} & \text{upward (positive) step} \\ -\left(V_k - v_{in}\right) & \text{downward (negative) step} \end{cases} \tag{1.36}$$

Over a long time-frame, negative and positive steps should balance, otherwise voltage v_{int} would diverge. The sequence is generally not periodical, unless the ratio of the two steps is a rational number. The output of the ADC (data-stream D_{st} in Fig. 1.13) is a sequence of V_{k-1} and V_k values, corresponding to positive and negative steps of the v_{int} voltage, respectively. The output filter extract the average of the D_{st} stream.

If v_{in} is perfectly halfway between the two adjacent levels, then the positive and negative steps are equal. In this case, it is possible to understand that, in a long time interval, the number of positive steps should equal the number of negative ones. Then, the occurrence of V_{k-1} values in the data stream is equal to the occurrence of V_k values. The average at the output of the LPF filter (D_{out}, representing V_{out}) is then $(V_k - V_{k-1})/2$, so that the input voltage is correctly represented.

If v_{in} is closer to V_{k-1}, the positive steps will be smaller than the negative ones. Therefore, we need a larger number of positive steps (i.e. V_{k-1} samples in the data-stream) to balance the negative ones (V_k values). The average V_{out} is than closer to V_{k-1}, matching the input voltage.

Since, as mentioned earlier, the sequence is not periodical then, in order to get a perfect average we would need an infinite time frame. In the frequency domain, this means that the LPF bandwidth tends to zero. Considering that the LPF bandwidth sets the signal bandwidth (see Fig.1.16), the OSR, given by $r_{OS} = f_S / 2B_S$ ratio would tend to infinity. This is clearly non-feasible and, with a finite OSR, the output D_{out} (V_{out}) will be marked by a certain amount of noise, visible as a fluctuation of the output codes. This fluctuation, which is just the residual quantization error, has a power given by (1.32), corresponding to the resolution increase expressed by (1.34).

It is interesting to observe that, in the asymptotic case of infinite time frame, the average of the data-stream value (v_{st}) equals v_{in}, indicating that, at least in principle, the delta-sigma approach enables the design of ADCs with an arbitrarily low INL. To understand this, let us consider the input of the $H(z)$ integrator in Fig. 1.13. If the integrator is ideal, the average of its input (i.e. the dc value of the input) should be zero, otherwise the output of the integrator would diverge. Then:

$$\left\langle v_{in} - v_{dac} \right\rangle = 0 \;\Rightarrow\; \left\langle v_{in} \right\rangle = \left\langle v_{dac} \right\rangle \tag{1.37}$$

Now, if the DAC is ideal, $v_{dac} = v_{st}$, then the average value of v_{st}, is equal to v_{in}, with no INL error. In this ideal case, the only source of error would be the imperfect average calculated by the LPF due to the finite OSR.

Another interesting point is that, even with a dc input, the signal applied to the internal ADC is not constant but varies continuously (see voltage v_{int} in Fig. 1.17). Then, the hypothesis of uniform spectral density for the quantization noise is more applicable even in the absence of dithering.

In practice, DACs are not ideal, as we have seen at the beginning of this chapter. Since the DAC is in the feedback path (see Fig. 1.13), the INL error of the DAC is turned into an error of the ADC. Fortunately, an intrinsically linear DAC exists. Such a DAC is the 1-bit DAC, consisting in a simple switch network that connects the output to V_{MIN} when the input bit is zero and to V_{MAX} when the bit is one. Such a DAC is linear since its output characteristic has only two points, which always lie on a straight-line. A delta sigma that uses internally a 1-bit ADC and, consequently, a 1-bit DAC, is shown in Fig. 1.18. The example of the figure, $V_{MIN}= V_{ref-}$ and $V_{MAX}=V_{ref+}$, where V_{ref+} and V_{ref-}-are the two analog levels produced by the DAC. The 1-bit ADC is simply a comparator that checks whether its input voltage is higher or lower than the middle of the input range.

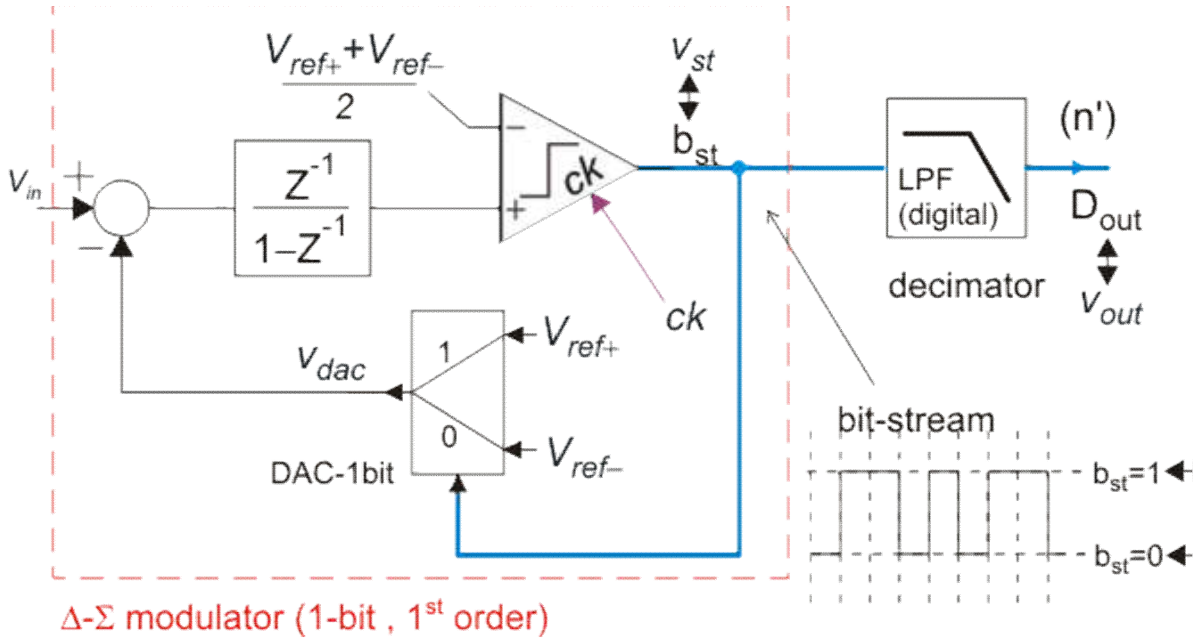

Fig. 1.18. Example of a single-bit first order Δ-Σ ADC.

In the example, the data-stream produced by the internal ADC is actually a bit stream, where the logic value "0" and "1" represent the analog values V_{ref-} and V_{ref+}, respectively. The final resolution (number of bits) is given by the initial resolution (1 bit) plus the resolution increment given by (1.34). To have an idea of the resolution that it is possible to achieve, we can consider, for example, an OSR of 512 (2^9). From (1.34) with n=1, we get n'=13.6. It is important to stress the fact that such a result can be achieved with no need of precise component matching. The bit-stream produced by an 1-bit Δ–Σ modulator is characterized by an average value that represents the input signal. Since the average of the is proportional to the number of "ones" present on a time-frame, the bit-stream is indicated as a "pulse density modulated" (PDM) signal.

The digital LPF is often implemented as a cascade of moving average filters. Since a single moving average filter have a frequency response that follows a sinc(f) behavior, the cascade of N blocks implements a so-called sinc^N filter. Sinc filters, which are generally implemented with a CIC (cascaded integrator-comb) architecture, have the great advantage of providing the required rejection of the out-

of-band quantization noise with no need of performing digital multiplications. This strongly reduce the complexity, area and power consumption of the digital circuits. The effect of the filter is limiting the output spectrum (both noise and signal) to $f_S/2r_{OS}=B_S$. Therefore, there is no need that D_{out} is updated at frequency f_S. For this reason, the LPF has also the function of reducing the data rate of the D_{out} stream to $2B_S$. This is done by dropping the samples in excess. Due to this function, the LPF is also indicated as "decimator filter".

Although the first order Δ-Σ converter can still represent an attractive solution for its simplicity, better performances can be obtained using higher order modulators. An n-order Δ–Σ ADC includes n integrators. The higher the order, the more bits are gained when the OSR is doubled. In a first order Δ-Σ converter, the resolution increases by 1 and ½ bits every-time the r_{OS} is doubled. For a single bit Δ–Σ ADC, achieving 16 bit if resolution would require an OSR of more than 1000. This means that, in order to convert an analog signal with 1 kHz bandwidth ($2B_S$=2 kHz), we need a sampling frequency (f_S) of nearly 2 MHz. A more efficient tradeoff between resolution and OSR can be obtained using a 2nd order Δ–Σ modulator, represented in the simplified diagram of Fig. 1.19.

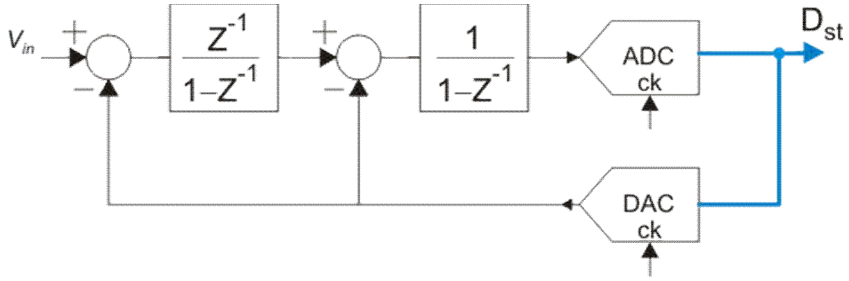

Fig. 1.19. Simplified block diagram of a 2nd order Δ-Σ ADC.

In a second order Δ–Σ ADC, the effective resolution increases by 5/2 bits every time the OSR is doubled. Therefore, a 16-bit ADC can be obtained from a 1-bit ADC (a comparator) with an OSR of only 128, strongly reducing the sampling frequency required for a given signal bandwidth. Second order delta-sigma converters, equipped with sinc^3 digital low pass filters, are widely used in integrated sensor interfaces. Signals in a 2nd order delta-sigma converters are much less intuitive than in the case of the first order converter, shown in Fig. 1.17.

Multi-bit Δ–Σ ADCs, i.e. delta-sigma converters where the internal ADC has a resolution (number of bits) greater than 1, are advantageous because we start from a greater resolution (n) and the increment (n'-n) needed to reach the target resolution (n') is smaller. A smaller increment means a smaller OSR, which, for a given signal bandwidth, means a smaller sampling frequency. This is particularly important in Δ–Σ ADCs to be used in telecommunication systems where signals with large bandwidth have to be converted with a relatively high resolution (to avoid dynamic range degradation). As mentioned earlier, the main problem of multi-bit converters is the DAC non-linearity. This problem is generally mitigated by using complex DEM (dynamic element matching) that continuously swap the elements of the DAC (i.e. the single resistors of the DAC in Fig. 1.3). In this way, on average, in a long time frame, all DAC elements tend to be balanced and the DAC behaves linearly.

Obviously, the analysis carried on so far is performed considering an ideal analog discrete-time integrator (block *H*). Imperfections of this block (e.g. input refereed offset, non-linearity, finite gain) introduce additional errors that should be taken into account while estimating the overall accuracy of the ADC.

References

[1] "Understand SINAD, ENOB, SNR, THD, THD + N, and SFDR", Analog Devices MT-003 Tutorial. Available online: http://www.analog.com/media/en/training-seminars/tutorials/MT-003.pdf

[2] R. Schreirer, G. C. Temes, "Understanding Delta-Sigma Data Converters", John Wiley & Sons, Inc, Hoboken, New Jersey, 2005.