BACKGROUND ELEMENTS

In this chapter we shall deal with the background knowledge necessary to properly understand and design data converters. Data converters are used in electronic circuits at the interface between the analog and the digital world. Conceptually a data converter procures a transformation of the signal representation: from continuous-time, continuous-amplitude to discrete-time, quantized amplitude. The transformation is inherently a non-linear operation that affects the spectrum of the signal and, possibly, modifies its information content. It is therefore important to be familiar with the theoretical implications of the data conversion process and to be aware of the limits of approximation used to analyse and design a data converter. Moreover, a proper knowledge on the mathematical tools used for the analysis and characterization of sampled-data systems is necessary.

1.1 THE IDEAL DATA CONVERTER

The basic functions of an analog-to-digital (A/D) or a digital-to-analog converter (D/A) can be separated into the sequence of elementary operations. Fig. 1.1 shows that an A/D system accomplishes the cascade of the following functions: continuous-time filtering, sampling, quantization and data coding. We shall study in the next sections why it is necessary to use a continuous-time fil-
ter and what is the effect of sampling and quantization on the signal [1]. Moreover, we shall discuss different coding schemes used to represent signals in the digital domain.

The D/A converter accomplishes the transcoding of the digital input into an equivalent analog signal and performs the reconstruction operation. Fig. 1.1 shows that the normal way to achieve reconstruction foresees the use of the cascade of two blocks: a sample and hold and a reconstruction filter.

1.2 THE SAMPLING OPERATION

The sampler transforms a continuous-time signal into its sampled-data representation. The output of an ideal sampler is a sequence of pulses that, for uniform sampling period, $T$, is represented by

$$x^*(t) = x^*(nT) = \sum_{-\infty}^{\infty} x(t)\delta(t-nT)$$  \hspace{1cm} (1.1)

where * denoted an ideal sampling operation.

Fig. 1.2 shows the waveform of a continuous-time signal and its sampled-data counterpart. The amplitude of the pulses representing the sampled data version equals the input at the sampling instances. Moreover, the sampled-data signal signifies at the sampling times, $nT$, only.

Equation (1.1) remarks the inherent non-linear nature of the sampling. It
The Sampling Operation

multiplies the input, $x$, with a sequence of delta functions. The right-hand side of the Fig. 1.3 represents this non-linear behavior by a mixer which modulating signal is the delta sequence. The reader can easily verify that the output of the modulator realizes equation (1.1). Describing the sampling process by using a mixer helps in understanding the operation of data converters in the undersampling mode that will be studied shortly.

The Laplace transform of an infinite sequence of delta functions is given by

$$\mathcal{L} \sum_{n=-\infty}^{\infty} \delta(t-nT) = \sum_{n=-\infty}^{\infty} e^{-nsT}$$

(1.2)

using the above equation and the definition of the Laplace transform in (1.1) results in

$$\mathcal{L} \left[ x(nT) \right] = \sum_{n=-\infty}^{\infty} X(s) e^{-jn\omega_s} = \sum_{n=-\infty}^{\infty} x(nT) e^{-nsT}$$

(1.3)

that provides two expressions of the Laplace transform of a signal sampled by an ideal sampler. The right-hand formula will be used to discuss the mapping between the $s$-plane and the $z$-plane. The left hand relationship tells us that the

Fig. 1.2 - Continuous time signal (a) and its sampled data representation (b).

Fig. 1.3 - Ideal sampler and its processing block-diagram.
spectrum of \( x^*(nT) \) is made by the infinite superposition of replicas of the \( x(t) \) spectrum. Each replica is shifted along the \( j\omega \) axis by \( n\omega_s = \frac{2\pi n}{T} \). The transformation of the input spectrum into its infinite replica is a consequence of the non-linearity of the sampling operation.

Observe that the spectrum of any real signal is band limited. Therefore, assuming that the signal bandwidth \( B \) is smaller than half of the sampling frequency, \( f_s/2 \), the replicas of the spectrum will not interfere one each other. Half of the sampling frequency, \( f_s/2 \), is often named the Nyquist limit. The frequency interval \( 0 - f_s/2 \) is also referred to as first Nyquist band (or first Nyquist zone). The next half-of-the-sampling frequency intervals, \( f_s/2 - f_s, f_s - 3f_s/2 \), are named second, third Nyquist zones, and so forth.

Fig. 1.4 a) and b) represents the above statement in a graphical manner. The spectrum spanning around the zero frequency in Fig. 1.4 a) represents the input before sampling. Fig. 1.4 b) provides the spectrum of the sampled signal, \( x(t) \). The part in the first Nyquist zone is the band-base repeats the input signal spectrum. Then we have the infinite image replicas.

Observe that, for the specific case of Fig. 1.4 a) and b), the spectrum within the band-base is limited enough so as the various replicas do not overlap one each other. Fig. 1.4 c) shows a critical situation: in addition to the input spectrum there is an interference signal located in between \( f_s/2 \) and \( f_s \). The sampling produces the replicas of the input spectrum as in Fig. 1.4 b), but, in addition the interference causes a component that falls inside the signal band (Fig. 1.4 d)), thereby corrupting the content of information of \( x(t) \). Therefore,
if we want to preserve the frequency content in the first Nyquist zone it is indispensable to reject all the frequency components (signal or interference terms) above \( f_s/2 \).

The above observation recalls us what is stated by the sampling theorem [1]: A band limited signal \( x(t) \) which Fourier spectrum \( X(j\omega) \) vanishes for \( |\omega| > \omega_s/2 \) is fully described by its sample values \( x(nT) \), if \( T < 2\pi/\omega_s \). The band limited signal \( x(t) \) is given by

\[
x(t) = \sum_{-\infty}^{\infty} x(nT) \frac{\sin(\omega_s(t-nT)/2)}{\omega_s(t-nT)/2}
\]

That represents the convolution of the input with the impulse response \( \sin(x)/x; \quad x = \omega_sT/2 \).

The input signal is band-limited. Therefore, knowing its bandwidth, it is possible to choose a sampling frequency that complies the sampling theorem. Nevertheless, as outlined above and depicted with Fig. 1.4 c) and d), the condition cannot be secured for noise and interferences: they may affect any frequency range, including the critical regions above the \( f_s/2 \) limit. It is therefore essential to ensure at the same time that the sampling frequency is suitably larger than twice the signal band and that no spur is folded back into the band of interest.

Because of the above needs any data conversion system uses before the sampler a low-pass continuous-time filter, as Fig. 1.1 shows. The filter is required to preserve the signal bandwidth and to reject all the frequency components out the band of interest. Since the folding of spurs into the band-base is also named aliasing, the continuous-time filter used is called anti-aliasing filter. The frequency response of the anti-aliasing filter depends on the required accuracy. Moreover, important specifications of the anti-aliasing filter are the harmonic distortions, the output swing, and noise performances.

Observe that the main purpose of the anti-aliasing filter is to preserve the signal spectrum only. Circumstances for which the signal band occupies only a fraction of the Nyquist interval and undesired signals corrupt other parts of the Nyquist interval are acceptable.

Assume that, as shown in Fig. 1.5, the input band is \( f_B \). Therefore, the frequency components from \( f_s/2 \) to \( f_s - f_B \) corrupt the frequency interval \( f_B - f_s/2 \)
Background Elements

(outside the band of interest), while the spurs above $f_s - f_B$ fold back into the frequency interval $DC - f_B$. Therefore, only the $(f_s - f_B) - f_s$ components threaten the signal spectrum and must be properly rejected. It turns out that there is a margin for the anti-aliasing filter: the transition from pass-band to stop-band spans from the signal band, $f_B$, to the frequency $f_s - f_B$.

The response of the anti-aliasing is often represented on a logarithmic axis. Each pole produces a roll-off by $20 \, dB$ per decade or $6 \, dB$ per octave. Therefore, we can estimate the complexity of the anti-aliasing filter by using the logarithmic margin $\log \left( \frac{f_s - 2f_B}{f_B} \right)$. If, for instance, the margin is only one octave achieving an attenuation of only $48 \, dB$ requires an $8-th$ order Butterworth filter. By contrast, a decade of logarithmic margin requires a third order filter for achieving $60 \, dB$ of attenuation. Therefore, high logarithmic margins reduce the filter complexity. However, the result is paid by a sampling frequency much higher than the minimum permitted and the request of a faster circuitry to achieve sampling and the following processing [2].

1.2.1 Undersampling

Some applications exploit the folding in the band-base of signals whose frequency components are higher than the Nyquist limit, $f_s/2$. They are named in various ways: harmonic sampling, band-base sampling, IF sampling, direct IF-to-digital conversion. Fig. 1.6 illustrated the method. The sampling of a signal which spectrum is in the second Nyquist zone, (Fig. 1.6 a), or the one of Fig. 1.6 b) which spectrum is in the third Nyquist zone produce the same sampled-data spectrum, as shown in Fig. 1.6 c): the original spectrum is folded in the band-base one time or twice respectively. The folding from the second band leads to a frequency reversal, while the one from the third band does not. Moreover, we have to assume that the spectrum of the continuous time signal

![Fig. 1.5 - Aliasing effect and anti-aliasing filter requirements.](image-url)
is band-limited and is included entirely within a Nyquist zone. The various replicas of the spectrum do not overlap and the information content of the original signal remains unchanged. The result is an extended consequence of the sampling theorem applied to higher Nyquist zones. Note that the sampling theorem is still verified: the band of the signal is smaller than half of the sampling frequency, although in an high frequency zone. Moreover, since the absolute location of the sampled signal band is completely within a given Nyquist zone, we avoid the overlap of the left and right tails of the signal spectrum [3].

Even systems using under sampling require the use of an anti-aliasing filter. The aliasing brings the signal in the band-base from a given higher Nyquist zone, but, also conveys interferences from other Nyquist zones. Moreover, spur signals can be in the band-base itself. Therefore, the anti-aliasing filter must remove all the possible interferences from all the Nyquist zones out of the one comprehending the signal. For the case of Fig. 1.6 a) the anti-aliasing-filter must be a band-pass from $f_s/2$ to $f_s$. For the case of Fig. 1.6 b) the band-pass filter must reject the spur outside the frequency interval $f_s - 3f_s/2$. Since we have a band-pass requirement the complexity of the filter depends on the margins on both sides of the Nyquist zone. If the band of interest is close to one Nyquist limit the order of the anti-aliasing filter is high even if we have an ample margin

![Fig. 1.6 - Undersampling of a signal in the second Nyquist and the third Nyquist band leading to the same sampled-data spectrum.](image)
on the other side. Therefore, the best situation occurs when the band of interest is median with respect to a given Nyquist zone.

Subsampling techniques have become popular in communication systems because they lead to an inherent demodulation of the input signal. The $IF$ mixer is no more necessary being frequency translation performed by the sampling itself. Therefore, a possible data conversion after the sampling enables a direct digital processing of the signal already folded in the band-base.

**Example 1.1**

Verify, with computer simulations, that the sampling at 1 MHz of two sinewave whose frequencies are 0.7 MHz and 1.3 MHz produces the same result of the sampling of a 300 kHz sinewave. Find the combination of sinewaves in the second Nyquist-zone that produce the same samples of the addition of two sinewaves, one at 300 kHz and amplitude 1, the second one at 400 kHz and amplitude 0.5.

**Solution:**

Fig. 1.7 shows a possible Simulink diagram. Multipliers achieve the sampling operation. The modulating signal is a pulse with unity amplitude and very small duty cycle (0.1%). The amplitude of the three sinewaves is the same. All the frequencies are normalized to the sampling frequency. Moreover, since the single folding of the 700 kHz sinewave causes a reversing of the phase, it is necessary to use for the 700 kHz sinewave a $\pi$ phase shift. Note that in order to obtain the result shown Fig. 1.8 it is necessary to set the following simulation parameters: fixed step and step size $= 0.002$. 

---

**Keep Note**

The frequency of operation of an undersampled circuit is typically much lower than the IF. However, the dynamic performances (bandwidth, slew-rate, jitter, ...) of the sampler must comply with high speed requirements.

As the $IF$ frequencies become higher, the dynamic performance requirements on the sampler become more critical. The input bandwidth and distortion performances must satisfy the specifications at the IF frequency, and not only in the base band. For undersampling applications the sampler must maintain good dynamic performance into higher order Nyquist zones. The resulting design problems (including the one associated to the anti-aliasing filter) can be comparable with the difficulty associated to the design of the $IF$ demodulator to be uses before the data converter.

---
1.2. The Sampling Operation

A sine wave at $f_a$ laying in the second Nyquist zone brings about an image in the band base at $f_s - f_a$. Therefore, images at 300 kHz and 400 kHz come from sine waves at 700 kHz and 600 kHz respectively. The amplitude of the 700 kHz component must be 1 while the one of the 600 kHz component is 0.5. Moreover, the frequency reversing requires a $\pi$ phase shift for both sine waves. It is easy to verify that superposition of the two above defined terms achieve the result: the sampling causes frequency translation but preserves amplitudes.

Fig. 1.7 - Waveforms of the sine wave used in Example 1.1.

Fig. 1.8 - Block diagram (top) and waveforms (bottom) that verify the mirrored folding from the second Nyquist zone.
1.3 AMPLITUDE QUANTIZATION

An continuous-time analog signal can assume any possible amplitudes within a given dynamic range. By contrast, a quantized signal takes only discrete amplitudes evenly spaced over the quantizer dynamic range. Therefore, the number possible amplitudes is finite and a discrete set of symbols is able to distinguish among them. capable to the signal. For example, the different symbols established by a binary $n$-bit word defines $2^n$ possible levels. In this case the spacing between the corresponding quantized amplitudes, $\Delta$, is

$$\Delta = \frac{V_{FS}}{2^n-1}$$

where $V_{FS}$ is the full scale amplitude. $\Delta$ is named the **quantization step**.

Consider, for example, a 10-bit quantizer with 1.023 V full-scale amplitude and 524.3 mV input. The quantization step, $\Delta$, is 1 mV. Moreover, all the inputs of the quantization interval 524 - 525 mV produce the same digital code. Conversely, a specific analog level represents a given digital code corresponding to the interval 524 - 525 mV. It is one of the edges of the quantization interval or (more often) the middle point. Therefore, using the same above example, the 10-bit quantizer transforms 524.3 mV input into a digital code represented by 524.5 mV. As a result, the quantization process alters the input by a 0.2 mV error.

The quantization process causes an error for all the input signal excluded the level representing the digital code. It is named **quantization error**, $\varepsilon_Q$, and is given by

$$\varepsilon_Q(nT) = x_{out}(nT) - x_{in}(nT)$$

The amplitude of $\varepsilon_Q$ ranges between $-\Delta/2$ and $\Delta/2$. Fig. 1.9 represents the quantization error for a 4-bit data converter. The figure displays 15 quantization intervals, whose amplitude is $\Delta = V_{FS}/15$. Outside the dynamic range of

![Fig. 1.9 - Quantization error for a 4-bit data converter.](image-url)
1.3. Amplitude Quantization

the converter \((0 - V_{FS})\) the error that is becomes larger, positive, or negative than \(\Delta/2\). If the digital code represents one of the extreme of the quantization step instead of the middle point the curve of Fig. 1.9 must be shifted up or down by \(\Delta/2\). The extent of one ending quantization interval become \(\Delta\) while the other goes to zero.

The quantization error is an inherent limit caused by the quantization process. It goes to zero only when the number of bits goes to infinite. In such a case the amplitude of the quantization step goes to zero and the system returns to use a continuous-amplitude representation of signals.

1.3.1 Quantization Noise

It is assumed that the reader is familiar with signal-to-noise ratio \((SNR)\) concept. It is widely used in analog system to estimate the dynamic range performances. The SNR is defined by

\[
\text{SNR}_{dB} = 20 \cdot \log\left(\frac{P_{\text{sign}}}{P_{\text{noise}}}\right)
\]

where \(P_{\text{sign}}\) and \(P_{\text{noise}}\) are the power of the signal and the power of the noise in the band of interest. A poor \(SNR\) denotes the limited capability of an analog system to discern signal from noise. A good \(SNR\) indicates that the system represents the signal quite accurately. Similar features depict a quantizer: the quantization error corrupt the input, thus limiting the accuracy with which the system can perceive a signal. Since the designer is familiar with the \(SNR\) it would be good to use the same concept for data converters. To make this possible it is necessary to perceive the quantization error as noise.

Practical situations do not always verify the above assumption. Obviously, a \(dc\) input does not produce a noise but a constant quantization error. Instead, with a busy signal produces a quantization error which value is different every clock cycle. Moreover, presuming a small quantization step, the frequent code transition makes unrelated successive samples of the quantization error. Therefore, under some operating conditions we can assume that the quantization error is an unpredictable fluctuation affecting the input signal.

**WARNING!**

Representing the quantization error by an additive noise is a critical approximation: it models a non-linear effect with a linear process.

Double check before relying on results achieved.
Four main conditions should be verified to ensure that the quantization error behaves like noise. They are:

- all the quantization levels are exercised with equal probability;
- a large number of quantization levels is used;
- the quantization steps are uniform;
- the quantization error is not correlated with the input.

Typically, a busy and large input signal fulfils the first condition. The second requisite holds for a medium or, better, for a high number of bits. Unfortunately, the condition is not verified for an important category of data converters. We will see shortly that sigma-delta modulators use quantizers with a minimum resolution, often 1-bit only. Therefore, for a sigma-delta converter the representation of the quantization error with an additive noise source is not well-founded. Nevertheless, the benefits resulting from the noise approximation are so relevant that, with the proper watchfulness, the designers utilize the noise approximation anyway.

Most quantizer conform to the third rule. Only a small fraction of data converter use a non-linear response (like, for instance a logarithmic response). Instead, the last requirement is not always verified. For example, Fig. 1.10 shows a 0.4 VFS sine-wave input sampled by a 10-bit quantizer which sampling period is 8 times the sine-wave period. The quantization error shows a repetitive pattern at the same frequency of the input signal. However, a change of the sampling frequency to 6.654 f_{sine} almost destroys the repetitive pattern.

1.3.2 - Properties of the Quantization Noise

Two key properties describe the features of any noise signal: the time average power and the noise power spectrum. Since the quantization noise is a sampled data signal, the power spectrum is meaningful in the Nyquist interval only: -f_s/2, f_s/2 or 0, f_s/2 for bilateral or for unilateral representations.

The estimation of the time average power assumes that the probability distribution function of the quantization error is constant in the interval -\Delta/2, \Delta/2 and zero outside. The assumption can be easily justified using the first condition given in the previous sub-section. Moreover, we have to remember that for an ideal quantizer the quantization error cannot be larger than ±\Delta/2. Since the integral of the probability distribution function over the entire \epsilon_Q range is equal to one, it results

\begin{equation}
\begin{align*}
p(\epsilon_Q) &= \frac{1}{\Delta} \quad \text{for} \quad \epsilon_Q \in \left[ -\frac{\Delta}{2}, \frac{\Delta}{2} \right] \\
p(\epsilon_Q) &= 0 \quad \text{otherwise}
\end{align*}
\end{equation}
The time average power of $\varepsilon_Q$ becomes

$$P_Q = \int_{-\infty}^{\infty} \varepsilon_Q^2 \cdot p(\varepsilon_Q) d\varepsilon_Q = \int_{-\Delta/2}^{\Delta/2} \varepsilon_Q^2 \cdot p(\varepsilon_Q) d\varepsilon_Q = \frac{\Delta^2}{12}$$

(1.9)

The above result permits us to calculate the SNR for specific inputs. The maximum amplitude of a sine-wave or a triangular-wave that a quantizer with

Fig. 1.10 - a) Quantization error for a 0.4 FS sine-wave sampled at 8 times $f_{\text{sine}}$.

b) Quantization error with sampling at 6.654 times $f_{\text{sine}}$.
dynamic range $V_{FS}$ can accommodate is $V_{FS}/2$. The power of a sine-wave with maximum amplitude is

$$P_{\text{sin}} = \int_0^T \frac{V_{FS}}{2} \sin^2 (2\pi fT) dt = \frac{V_{FS}^2}{8} = \frac{(\Delta(2^n - 1))^2}{8}. \quad (1.10)$$

The power of a triangular-wave is

$$P_{\text{triang}} = \frac{V_{FS}^2}{12} = \frac{[\Delta(2^n - 1)]^2}{12}. \quad (1.11)$$

Therefore using the above equations and (1.9), it results in

$$SNR_{\text{sin}}|_{dB} = (6.02 \cdot n + 1.78) \text{ dB} \quad (1.12)$$

$$SNR_{\text{triang}}|_{dB} = (6.02 \cdot n) \text{ dB} \quad (1.13)$$

that establish an important relationship between the maximum SNR achievable and the number of bits of the quantizer. Reminding the term $1.78$ dB for sine-wave input, equations (1.12) and (1.13) state that every bit of resolution improves the SNR by $6.02$ dB.

**Remember**

Every bit of resolution the SNR improves by 6 dB. Accordingly, the power of the quantization error decreases by a factor 4.

It is worth to recall that the above SNR estimation accounts for the quantization noise only. In a real circuits the electronic noise increases the noise level. Moreover, as we will study in a following chapter dealing with oversampling, part of the quantization noise can be filtered out. Therefore, the noise power affecting a sampled data system can be different (smaller or bigger) than $\Delta^2/12$. Nevertheless, equation (1.12) and (1.13) can be generalized and used to define the equivalent number of bits

$$n_{\text{eq, sine}} = \frac{SNR_{\text{eff}}|_{dB} + 1.78}{6.02} \quad (1.14)$$

$$n_{\text{eq, triang}} = \frac{SNR_{\text{eff}}|_{dB}}{6.02} \quad (1.15)$$
1.3. Amplitude Quantization

where $SNR_{eff}$ represents the effective signal-to-noise ratio influencing the system.

The second key property of the quantization noise is the power spectrum. It shows in which way the noise power is spread over the Nyquist interval. A mathematical principle states that the power spectrum is given by the Laplace transform of the auto-correlation function

$$P_\epsilon(f) = \int_{-\infty}^{+\infty} R_\epsilon(\tau)e^{-j2\pi ft}d\tau. \tag{1.16}$$

Unfortunately the above relationship does not help much. It is not easy to express the autocorrelation function of the quantization error. However, we suppose that two successive samples of the quantization error are not correlated. This assumption comes indirectly from the condition that the quantization noise is not correlated with the input and the observation that two successive input samples are correlated.

Since we assume that the autocorrelation of the quantization noise vanishes outside the $-T+T$ interval, the autocorrelation function looks like a sharp pulse that resembles a delta function. Since the Laplace transform of a delta is constant the power spectrum of the quantization error is almost white. Therefore, the power $P_Q = \Delta^2/12$ is spread uniformly over the Nyquist interval

$$P_\epsilon(f) = \frac{\Delta^2}{6f_s} \tag{1.17}$$

In summary, under the conditions discussed above it is possible to represent the quantization process with an additive noise. Therefore, as shown in Fig. 1.11, the effect of the quantizer can be modelled by a linear process.

Fig. 1.11 - Power spectrum in a quantizer which effect is modelled by an additive noise.
**Example 1.2**

Determine by a computer simulation the spectrum of the quantization error of a 10-bit data converter. Try different input amplitudes and frequencies of the input sinewave. Use, if the supporting software is available, the Exercise: “Spectral Analysis: Quantization Noise” of the package DCW (Data Converter Workbench).

**Solution:**

The quantization noise is the difference between the input and its quantization. An ideal quantizer is available as basic block in the Simulink environment. Alternatively, it is possible to quantize a signal by using the “round” function available in many packages for mathematical calculations and signal processing. The spectral analysis of the quantization error leads to results that critically depends on the sampling frequency used.

Simulations with different amplitude and frequency permits the reader to verify that in some cases the spectrum of the quantization error show a slightly coloured behaviour. The result show in

![Spectrum of signal](image)

**Fig. 1.12** - Spectrum of the quantization error determined by a 10-bit quantizer.

*Fig. 1.12 corresponds to an amplitude of 0.621, input frequency 4.12321, and sampling frequency 400. The spectrum looks like a white noise, as it is desired.*
1.4 CODING SCHEMES

A variety of coding schemes can represent the output of a quantizer. All of them are based on a binary representation [4].

**USB - Unipolar Straight Binary**: it is the simplest coding scheme. It is used for unipolar inputs. When using USB coding, the digital code made by all zero’s (0000) represents the first quantization level, $0V + 1/2 V_{LSB}$. As the digital code increments, the analog voltage increases one $V_{LSB}$ at a time, and when the digital code reaches the full scale (1111) the analog voltage becomes $V_{FS}-1/2 V_{LSB}$.

**CSB - Complementary Straight Binary**: is the opposite of the USB. CSB coding, like its opposite USB, is used for unipolar systems. The digital code (0000) represents the full scale while the code (1111) corresponds to the first quantization level $1/2 V_{LSB}$.

**BOB - Bipolar Offset Binary**: This coding scheme is suitable for bipolar systems (where the analog signal can be positive and negative). The bit in the most significant position denotes the sign of the input: 1 for positive signals and 0 for negative signals. Therefore, (0000) represents the full negative scale. The zero crossing occurs at (1000) and the digital code (1111) represents the full positive scale.

**COB - Complementary Offset Binary**: this coding scheme is complementary to the BOB scheme. All the bits are complemented and the meaning remains the same. Therefore, since (1000) denotes the zero crossing in the BOB scheme, in the COB, the zero crossing becomes (0111).

**BTC - Binary Two’s Complement**: it is one of the most used schemes. The bit in the MSB position indicates the sign in a complemented way: it is 0 for positive inputs and 1 for negative inputs. The zero crossing occurs at (0000). The counting of positive signals increases as the normal digital numbering. Thus, the positive full scale is (0111). The negative signals are the two complements of the positive counterpart. This leads (1000) to represent the negative full scale. The BTC coding system is suitable for microprocessor based systems or for the implementation of mathematical algorithms. Moreover it is the standard for digital audio.

**CTC - Complementary Two’s Complement**: it is the complementary code of BTC. All the bits are complemented and codes have the same meaning.

Table 1.1 summarizes the correspondence between input level and its coded representation for the different schemes discussed above. It refers to a $1,024 mV$ full scale and a 4-bit coding scheme. The extension to more bits is straightforward.
TABLE 1.1 - Unipolar and Bipolar Coding Schemes [1.024 VFS]

<table>
<thead>
<tr>
<th>$V_{th}$ [mV]</th>
<th>USB</th>
<th>CSB</th>
<th>$V_{th}$ [mV]</th>
<th>BOB</th>
<th>COB</th>
<th>BTC</th>
<th>CTC</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>0000</td>
<td>1111</td>
<td>-480</td>
<td>0000</td>
<td>1111</td>
<td>1000</td>
<td>0111</td>
</tr>
<tr>
<td>96</td>
<td>0001</td>
<td>1110</td>
<td>-416</td>
<td>0001</td>
<td>1110</td>
<td>1001</td>
<td>0110</td>
</tr>
<tr>
<td>160</td>
<td>0010</td>
<td>1101</td>
<td>-352</td>
<td>0010</td>
<td>1101</td>
<td>1010</td>
<td>0101</td>
</tr>
<tr>
<td>224</td>
<td>0011</td>
<td>1100</td>
<td>-288</td>
<td>0011</td>
<td>1100</td>
<td>1011</td>
<td>0100</td>
</tr>
<tr>
<td>288</td>
<td>0100</td>
<td>1011</td>
<td>-244</td>
<td>0100</td>
<td>1011</td>
<td>1100</td>
<td>0011</td>
</tr>
<tr>
<td>352</td>
<td>0101</td>
<td>1010</td>
<td>-160</td>
<td>0101</td>
<td>1010</td>
<td>1101</td>
<td>0010</td>
</tr>
<tr>
<td>416</td>
<td>0110</td>
<td>1001</td>
<td>-96</td>
<td>0110</td>
<td>1001</td>
<td>1110</td>
<td>0001</td>
</tr>
<tr>
<td>480</td>
<td>0111</td>
<td>1000</td>
<td>-32</td>
<td>0111</td>
<td>1000</td>
<td>1111</td>
<td>0000</td>
</tr>
<tr>
<td>544</td>
<td>1000</td>
<td>0111</td>
<td>32</td>
<td>1000</td>
<td>0111</td>
<td>0000</td>
<td>1111</td>
</tr>
<tr>
<td>608</td>
<td>1001</td>
<td>0110</td>
<td>96</td>
<td>1001</td>
<td>0110</td>
<td>0001</td>
<td>1110</td>
</tr>
<tr>
<td>672</td>
<td>1010</td>
<td>0101</td>
<td>160</td>
<td>1010</td>
<td>0101</td>
<td>0010</td>
<td>1101</td>
</tr>
<tr>
<td>736</td>
<td>1011</td>
<td>0100</td>
<td>224</td>
<td>1011</td>
<td>0100</td>
<td>0011</td>
<td>1100</td>
</tr>
<tr>
<td>800</td>
<td>1100</td>
<td>0011</td>
<td>288</td>
<td>1100</td>
<td>0011</td>
<td>0100</td>
<td>1011</td>
</tr>
<tr>
<td>864</td>
<td>1101</td>
<td>0010</td>
<td>352</td>
<td>1101</td>
<td>0010</td>
<td>0101</td>
<td>1010</td>
</tr>
<tr>
<td>928</td>
<td>1110</td>
<td>0001</td>
<td>416</td>
<td>1110</td>
<td>0001</td>
<td>0110</td>
<td>1001</td>
</tr>
<tr>
<td>992</td>
<td>1111</td>
<td>0000</td>
<td>480</td>
<td>1111</td>
<td>0000</td>
<td>0111</td>
<td>1000</td>
</tr>
</tbody>
</table>

### 1.5 THE TRANSCODER

The transcoding operation determines the transition between the digital and the analog world. We have seen that Table 1.1 defines, for different coding schemes, a relationship between an input voltage and its digital representation. The transcoder achieves the same relationship in the opposite direction.

Conceptually a transcoder is a switch matrix with $2^n$ analog voltages at the input. The switch matrix output is one of the inputs depending on the selection code (Fig. 1.13). We will see that the conceptual scheme is achieved in practice in flash DACs only. Other architecture perform the transcoding function in a more efficient way.

Since the digital input is a sampled-data signal the output of the transcoder
1.6 The reconstruction

1.6.1 A sampled-data signal as well. Therefore, the output has meaning only at the sampling times. Ideally, the output of a transcoder is made by a sequence of delta which amplitude is a discrete-level analog signal. However, that is what we have in an ideal case. A real circuit requires a finite time to generate the output analog voltage; moreover, the generated waveform is not a delta but it lasts for a finite time. Therefore, we have to account for a delay between the digital input and the transcoded output. Moreover the generated signal looks like a pulse.

We have to note that the digital-to-analog conversion does not strictly require to generate a sequence of deltas. The block after the transcoder that performs the reconstruction aims at the removal of the high frequency components produced by the deltas. We will see shortly that the reconstruction block includes a sample-and-hold as first element of the reconstruction system. Therefore, what is important is just to ensure that the transcoded generates the proper signal level at the sampling times.

1.6 THE RECONSTRUCTION

The spectrum of a sampled data signal is an infinite replica of the band-base spectrum. The task of the reconstructor is to remove the replicas and to extract a band-limited signal.

Ideally, a filter which transfers functions is

\[ H_{R, id} = 1 \quad \text{for} \quad -\frac{f_s}{2} > f > \frac{f_s}{2} \]

\[ H_{R, id} = 0 \quad \text{otherwise} \]  

\[ (1.18) \]
optimally achieves the result. Unfortunately the discrete-time impulse response of the ideal reconstruction filter is

\[ r(t) = \frac{\sin(\omega_s t / 2)}{\omega_s t / 2} \]  

(1.19)

that cannot be realized being non-recursive.

Therefore, it is possible to procure the reconstruction function in an approximated way only. The simplest form of the reconstruction filter is the sample and hold. However, the cascade of the two blocks shown in Fig 1.1 made by a sample and hold and a low pass filter, accomplishes a more satisfactory approximation of the reconstruction function. The transfer function of a sample-and-hold (show in Fig. 1.14) is

\[ H_{SH}(s) = \frac{1}{s} - e^{-sT} \]  

(1.20)

that on the \( j\omega \) axis becomes

\[ H_{SH}(j\omega) = T e^{-j\omega T/2} \left[ \frac{\sin(\omega T/2)}{\omega T/2} \right] \]  

(1.21)

Fig. 1.14 displays some attenuation in the Nyquist interval that, at the Nyquist limit \( f_s/2 \), becomes approximately 0.6 (-4.4 dB). By contrast, the ideal response is one until \( f_s/2 \) and drops to zero at higher frequency. Moreover, the sample and hold does not fully reject the signal for \( f > f_s/2 \). The response

![Amplitude response of the transfer function of the sample-and-hold.](image)
amplitude becomes lower than 0.1 (-20 dB) only for frequencies higher than 2.7 \(f_s\). In addition we have to account for some phase shift in the signal band. Any phase alteration is undesired in some applications, like data transmissions, where the phase is an important element of the information content.

The reconstruction filter placed after the sample and hold compensates for the sample-and-hold limitations. It possibly balances the in-band attenuation with a \(x/sin(x)\) emphasis. Moreover, it enhances the rejection in the stop-band.

Fig. 1.15 shows an example of frequency response of the reconstruction filter. The gain in the band-base goes up to 4.4 dB at \(f_s/2\) and compensates the attenuation of the S&H. After \(f_s/2\) the response rolls down quickly to improve the out-of-band rejection. The attenuation is larger at the frequencies for which the \(sinc\) function has its maxima.

1.7 DISCRETE AND FAST FOURIER TRANSFORMS

The second expression of equation (1.3), reported again below

\[
\mathcal{L} \{x^n(nT)\} = \sum_{-\infty}^{\infty} x(nT)e^{-nsT},
\]  

(1.22)
or its Fourier counterpart

\[
F[x^*(nT)] = X^*(j\omega) = \sum_{n=-\infty}^{\infty} x(nT)e^{-j\omega n T}
\] (1.23)

permits the designer to study the frequency behaviour of sampled data signals.

In practical cases the signal sequence is not infinite but is made by a finite number, \(N\), of samples (starting, for instance, from \(n = 0\)). If \(N\) is large enough a summation limited from \(0\) to \(N-1\) provides a good approximation of (1.23). In addition, a computer determines the Fourier transform \(X^*(j\omega)\) only at discrete values of \(\omega\). This is what the Discrete Fourier Transform (DFT) provides. It is defined by

\[
X(k) = \sum_{n=0}^{N-1} x(nT)e^{-j2\pi kn/(N-1)} \quad 0 \leq k \leq N-1
\] (1.24)

where the sequence \(x(nT), [n= 0, N-1]\) is a fraction of the periodic infinite sequence \(x(nT)\) which period is \(nT\).

Notice that although the sequence \(x(nT)\) is real, its DFT \(X(k)\), is complex: it permits us to compute, like the Fourier transform, the real and the complex components or the magnitude and the phase of the discrete frequency response.

The direct use of equation (1.24) involves \(N^2\) computations. The Fast Fourier Transform algorithm (FFT), proposed by Cooley-Tukey [5], leads to a more efficient way to calculate the DFT. The FFT algorithm can reduce the number of computation required from \(N^2\) down to \(N \log(N)\). The execution time for the FFT depends on the length of the transform [6]. It is fastest for \(N\) to be a power of two. Another possible algorithm is the FFTW proposed by Frigo and Johnson [7]. The FFTW algorithm is an improvement of the FFT in terms of speed and accuracy. The most popular packages for digital signal processing use one or both of the above mentioned algorithms.

The DFT or the FFT are typically used to estimate the SNR of an A/D converter and to determine its spurious free dynamic range (that will be defined in the next Chapter). For this type of estimations it is necessary to ensure that the quantization noise appears as white noise, spread uniformly over the Nyquist bandwidth. We have seen in Example 1.2 that the use of a sine-wave input can lead to a correlated quantization error. Moreover, the correlation between the quantization noise and the signal depends upon the ratio of the sampling frequency to the input signal. For example, the output of a 12-bit quantizer analysed using a 4096-point FFT leads to the tone affected left-hand spectrum of Fig. 1.16. The sampling frequency is 32 times the sine-wave used. By con-
1.7. Discrete and Fast Fourier Transforms © F. Maloberti

Contrast, the spectrum does not contain periodic tones (see the right-hand spectrum) if the ratio between sampling frequency and input frequency is changed into the non integer value $4096/127$.

It is worth remembering that the FFT expects that the input signal is periodic with period $nT$. Therefore, it is necessary to ensure that

$$x(NT) = x(0)$$ (1.25)

If the signal does not verify the above equation the spectrum is not accurate. It is therefore necessary to apply a windowing function that properly shapes the input sequence so that equation (1.25) is verified. In the cases of Fig. 1.16 windowing is not necessary. In the right-hand diagram the input sequence of 4096 samples accommodates 128 sine period. The left-hand case houses 127 (a prime number) sinewave periods of the input sequence. Therefore, in both cases the input sequence verify equation (1.25).

The FFT leads to a spectrum made by $N$ lines equally spaced in the frequency interval $0 - f_s$. Since the spectrum in the interval $f_s/2 - f_s$ is symmetrical with respect to the one in the base-band, we usually represent only the first half of the FFT result. Each line of the spectrum represents the power falling within a frequency interval given to the line spacing, $f_s/N$. The result is assumed centred around the line itself. Therefore, the FFT acts like an analog spectrum analyser which bandwidth is $f_s/N$. When the number of

---

**Remember**

When using the FFT make sure that the sequence of samples used is periodical and/or use windowing.

With sinewave inputs use a prime number as ratio between the sinewave period and the sampling period.

---

![Fig. 1.16 - Spectrum of a quantized signal (12-bit) for two different ratios $f_s/f_a$.](image)

Fig. 1.16 - Spectrum of a quantized signal (12-bit) for two different ratios $f_s/f_a$. 

Let us consider a signal made by a full scale sinewave which power is $P_s$ and a white noise which total power in the Nyquist interval is $P_n$. The FFT with 2048 points will spread the power $P_n$ over the 1024 lines of the Nyquist interval while the power $P_s$ will be concentrated over one line only. Therefore, as shown in Fig. 1.17, the floor, $S_{\text{noise}}$, of the lines representing the noise is below the noise power value

$$S_{\text{noise}} = P_n - 10 \log (N/2)$$

(1.26)

The term $10 \cdot \log (N/2)$ is called processing gain of the FFT.

If we want to observe spur components, like the tones produced by harmonic distortion, it is necessary to use an FFT with a suitable number of points so that the tones emerge over the FFT noise floor.

1.8 THE Z-TRANSFORM

The mathematical tool used to study a linear, continuous-time system is the Laplace (or the Fourier) transform [8]. Continuous-time linear operations are: addition, subtraction, derivative and integral. The use of the Laplace operator translates a set of equations containing integrals and derivatives into a set of linear algebraic equations in the $s$-domain. The solution of a linear system is much easier than the solution of a derivative-integral system of equations.
Moreover, the *s-domain* analysis provides useful information on the behaviour of the circuit.

The above benefit are quite attractive and can be extended to a linear, sampled-data system by the use of a transformation that is the time-discrete counterpart of the Laplace transform [9]. It is the *z*-transform, defined by

\[ \mathcal{Z}[x(nT)] = \sum_{n=-\infty}^{\infty} x(nT)z^n \] (1.27)

that, from a mathematical point of view, is a geometrical series that converges in a region of the *z-plane* for which \(|z| < 1\). Assuming that the series converges in some region of the *z-plane*, it is assumed that the *z*-transform is the analytical extension of the converged solution over the entire *z-plane*.

Discrete-time linear operations are: addition, subtraction and delay. It is easy to verify that the *z*-transform is a linear operator. Therefore, when applied to the addition or the subtraction of two signals, it results in

\[ \mathcal{Z}[X_1(nT) \pm X_2(nT)] = \mathcal{Z}[X_1(nT)] \pm \mathcal{Z}[X_2(nT)]. \] (1.28)

Moreover, the *z*-transform of a delayed signal is given by

\[ \mathcal{Z}[X(nT - T)] = \sum_{n=-\infty}^{\infty} x(nT - T)z^{-n} = z^{-l}x(z) \] (1.29)

therefore, the operator \(z^{-l}\) corresponds to a delay by one sampling period.

Comparing the second relationship of (1.3) and the definition of the Laplace transform one obtains the following relationship

\[ z \rightarrow e^{sT} \] (1.30)

that describes a mapping between two complex planes: *s* and *z*. The mapping establishes a link between the continuous-time and the corresponding discrete-time signals.

The use of (1.8) shows that the transformation is not biunivocal. As shown in Fig. 1.18, the *joω* axis in the *s*-plane corresponds to the unity circle in the *z*-domain. Moreover, the *dc* (\(\omega = 0\)) corresponds to the point \((1,0)\) in the *z*-plane, while the point \((-1,0)\) maps the frequencies \(n\pi/T\) (\(n = \pm 1, \pm 3, \pm 5, \ldots\)). Therefore, one point on the *joω* axis leads to one point on the unity circle in the *z*-plane. By contrast one point in the unity circle maps an infinite number of points on the *joω* axis. Moreover, the frequency response of a sampled-data system (*joω* axis) results from the behaviour of the *z*-transfer function on the unity
circle. The non-biunivocal mapping reflects the operation of an ideal sampler: the spectrum of the continuous-time signal is transformed into an infinite sequence of replicas.

1.9 KT/C NOISE

The practical implementation of any sampler foresee the measure of the input signal through a given resistance and a load capacitor. Fig. 1.19 a) shows a schematic representing a the basic operation of a practical sampler. It is made by an ideal switch and an $R_sC_s$ network. When the switch is closed the capacitor $C_s$ is charged at the input voltage $V_x$, thus producing $V_x^*$. In order to have a proper sampling it is necessary to assume that the time during the switch is closed is much larger than the time constant $R_sC_s$. Moreover, the bandwidth of the input signal is much lower than $1/(2\pi R_sC_s)$. The resistance $R_s$ represents the total resistance in series with the signal, including the on-resistance of the switch. The noise of the resistance controls the noise performances of the circuit. Assuming, as shown in Fig. 1.19 b), that the noise associated to the resistance $R_s$ is the thermal contribution only, we have to account for a power generator with white spectrum $4kTR_s$.

The $R_sC_s$ network establishes a low-pass filtering action over $V_{n,in}^2$. Therefore, noise spectrum at the output is given by the spectrum at the input multiplied by the square of the transfer function of $R_sC_s$ the network.
Therefore, the input white spectrum is shaped into a coloured spectrum by the single pole $R_s-C_s$ network (Fig. 1.19 c). As stated above the low-pass frequency of the $R_s-C_s$ network is higher than the sampling frequency; Therefore, the aliasing affects the operation. Namely, as shown in Fig. 1.19 d), the noise components from the higher Nyquist zones are folded in the band-base resulting into a piling up of the spectra of all the Nyquist zones. The spectrum form the even zones is reversed while the one form the odd zones is folded unchanged. The result will lead to a spectrum in the band-base that is almost white. The total noise power stored on the capacitor $C_s$ is calculated by

$$P_{n,C_s} = \int_0^{\infty} V_{n,\text{out}}^2(f) df = 4kTR_s \int_0^{\infty} \frac{I}{1 + (\omega R_s C_s)^2} df = \frac{kT}{C_s}$$  \hspace{1cm} (1.32)$$

that is independent on the value of $R_s$. The independency is explained by observing that an increases of $R_s$ produces two effects: an increase of the input white noise and a better filtering action. The two effects have opposite consequences on the output noise and, in the specific case, they exactly compensate one each other.
The \( kT/C \) noise is superposed with the input and degrades its SNR. Therefore, two noise terms affect a quantized signal: the quantization noise and the \( kT/C \) noise. In order to preserve the resolution of the quantizer it is necessary to keep the \( kT/C \) term negligible with respect to the quantization noise. Therefore, the sampling capacitance must be larger and larger as the resolution of the quantizer increases.

In some systems the sampling frequency used is higher than twice the signal bandwidth. As a result the \( kT/C \) power, spread uniformly over the Nyquist interval does not affect completely the band of interest. A digital filter possibly placed after the analog-to-digital converter permits the designer to remove part of the \( kT/C \) noise, thus improving the performances of the system.

Table 1.2 gives the noise voltage corresponding to \( kT/C \) and the (fractional) resolution of a quantizer that, for 1 VFS produces a quantization noise equal to \( kT/C \).

### TABLE 1.2 - \( kT/C \) voltages and corresponding resolutions (V _FS_ =1 V).

<table>
<thead>
<tr>
<th>( C ) [pF]</th>
<th>( V_n ) [( \mu V )]</th>
<th>Bit</th>
<th>( C ) [pF]</th>
<th>( V_n ) [( \mu V )]</th>
<th>Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>203.9</td>
<td>12.3</td>
<td>10</td>
<td>20.4</td>
<td>15.6</td>
</tr>
<tr>
<td>0.3</td>
<td>117.7</td>
<td>13</td>
<td>30</td>
<td>11.77</td>
<td>16.3</td>
</tr>
<tr>
<td>1</td>
<td>64.50</td>
<td>13.9</td>
<td>100</td>
<td>6.45</td>
<td>17.2</td>
</tr>
<tr>
<td>3</td>
<td>37.24</td>
<td>14.7</td>
<td>300</td>
<td>3.72</td>
<td>18</td>
</tr>
</tbody>
</table>

### 1.10 PROBLEMS

1. **Repeat Example 1.1 but perform the verification in the frequency domain by the use of the FFT. Verify that the spectrum of the signal folded from the second Nyquist zone does not depend on the phase of the two sinewave used.**

2. **Determine, by a computer simulation, the spectrum of the quantization noise at the output of a 2-bit quantizer. The input is a sinewave with**
1.10. Problems © F. Maloberti

1.10. Problems

1.10. Problems

1.3 A processing system is made by a sampler that samples a sinewave with frequency $39/2048 \, f_s$ and amplitude $0.46 \, V_{FS}$. Then the result is held for 1 sampling period; finally, there is a 10-bit quantizer. Using a a behavioural simulation tool and the output signal only determine the quantization noise and the spectrum component at the input signal.

1.4 Determine the USB coding of $0.367 \, V_{FS}$ and the BOB and the BTC coding of $-0.763 \, V_{FS}$. Assume that the quantizer has 10-bit of resolution.

1.5 Sample at $1\, MHz$ a $0.23456\, MHz, 0.67\, V_{FS}$ sinewave. Hold the samples for an entire sampling period and re-sample the result at $10\, MHz$. Estimate the spectrum of the resulting signal and compare it to the spectrum given in Fig. 1.14.

1.6 Determine, by a computer simulation, the quantization error at the output of a 8 bit quantizer. Assume at the input a 372.3 Hz sinewave sampled at 2 kHz. Estimate the distribution function of the quantization error (20 points in the $-\Delta/2, \Delta/2$ range). The accuracy of the distribution function must be better than 5%.

1.7 Determine (in dB) the gain required to compensate the sample-and-hold response at $0.36 \, f_s$ and the attenuation necessary to ensure a image rejection better than 75 dB at $1.5 \, f_s$.

1.8 Derive the two spectra given in Fig. 1.16. Compare the results with the spectrum coming out from the sampling of a sinewave for which $f_s/f_a = 4096/129$. Evaluate the spectrum modification for $f_s/f_a = 4096/127.5$. Repeat the simulations using 2048 and 8192 points.

1.9 Determine the point of the z-plane mapping the following points of the s-plane: $s = 2 + j2/T ; s = -2 + j2/T ; s = -2 - j2/T ; s = -1 + j\pi/2T ; s = -2 + j\pi/2T ; s = -3 + j\pi/2T$.

1.10 Calculate the processing gain with a $2^{14}$ point FFT. Assume to represent the spectrum of a $1 \, V_{peak}$ sinewave passed through a 10-bit quantizer. What is the minimum amplitude of harmonic components that can be observed?

1.11 What is the noise floor established by a sampling capacitor of 3 pF. The sampling frequency is 58 MHz. What is the equivalent resistance leading to the same noise floor?
1.11 REFERENCES


