

Sistemas de Comunicaciones basados en Radio Definida por Software (SDR)

Dr. Ing. Alejandro José Uriz

Repaso sobre Tratamiento Digital de Señales

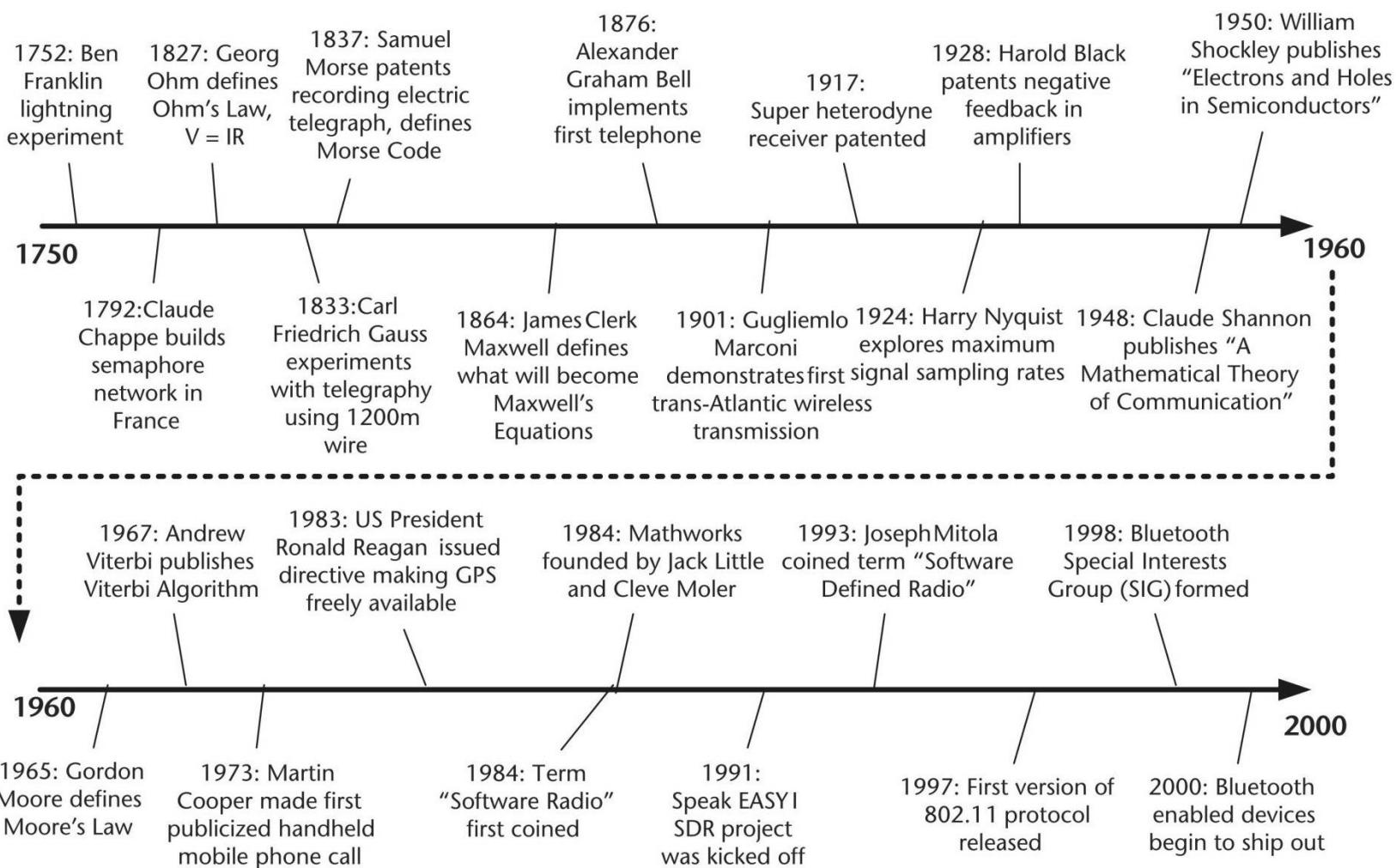
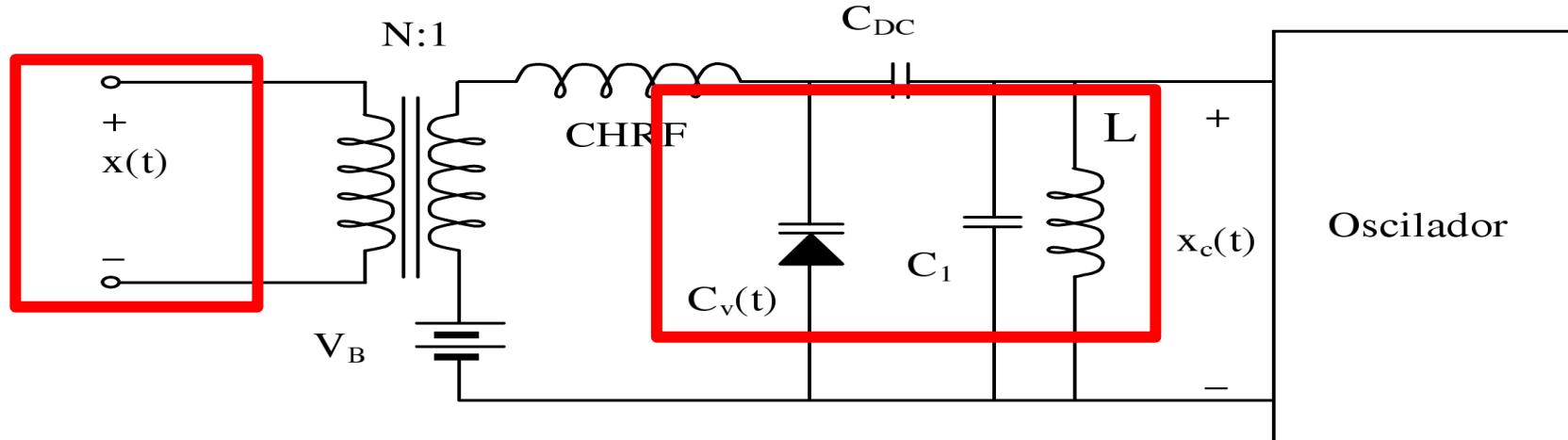


Figure 1.1 Timeline of several key milestones in communications.

Método directo de generación de FM



Este circuito consiste en un VCO. Su frecuencia de resonancia depende de L y C . La capacidad total del sistema es controlada mediante $x(t)$.

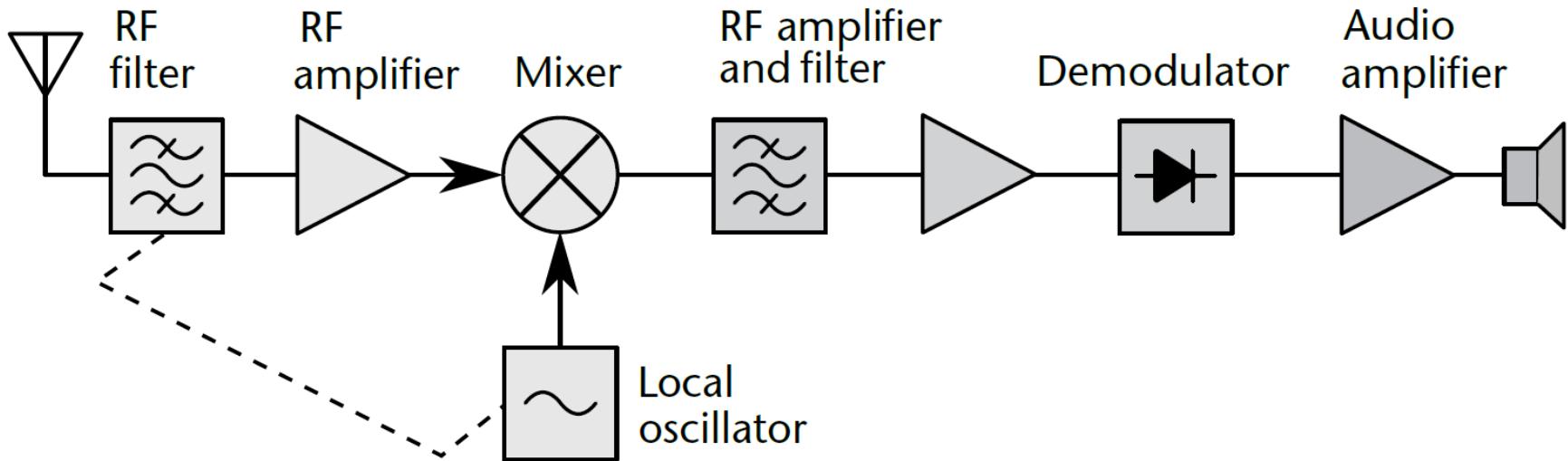


Figure A.3 Single-conversion superheterodyne radio receiver. The incoming radio signal from the antenna (left) is passed through an RF filter to attenuate some undesired signals, amplified in a radio frequency (RF) amplifier, and mixed with an unmodulated sine wave from a local oscillator. The result is a beat frequency or heterodyne at the difference between the input signal and local oscillator frequencies, a lower frequency called the IF. The IF signal is selected and strengthened by several IF stages that bandpass filter and amplify the signal. The IF signal is then applied to a demodulator that extracts the modulated audio signal. An audio amplifier further amplifies the signal, and the speaker makes it audible.

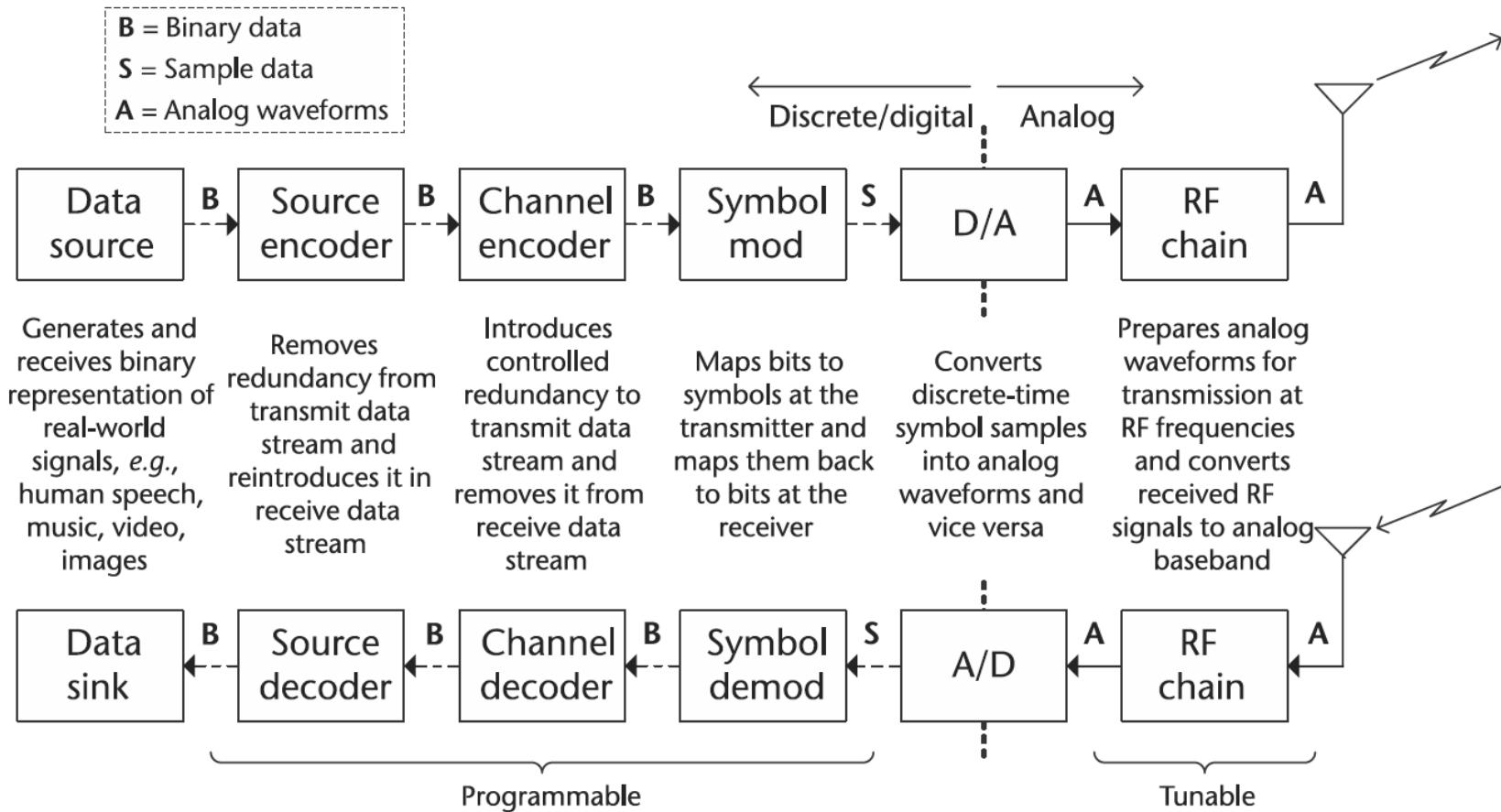


Figure 1.3 An illustration describing some of the important components that constitute a modern digital communications system. Note that for a SDR-based implementation, those components indicated as programmable can be realized in either programmable logic or software.

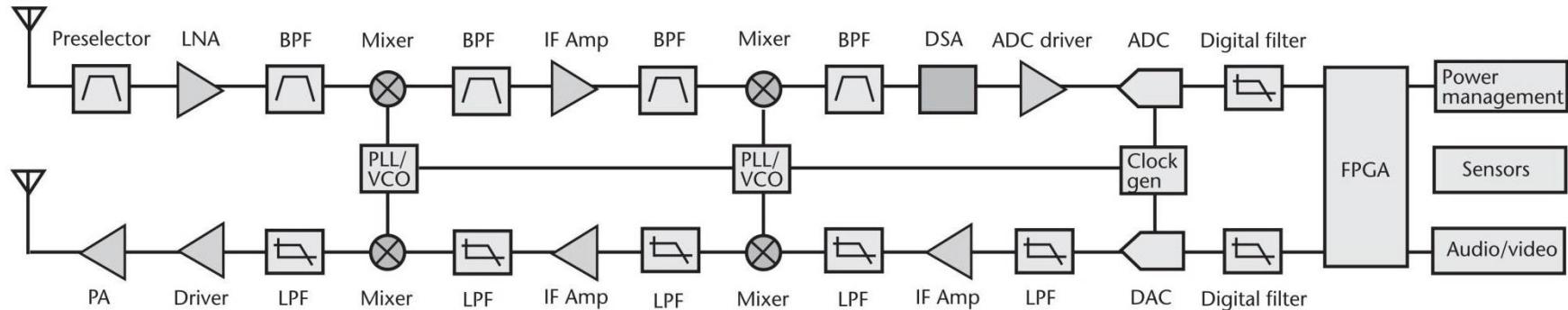


Figure 1.6 Multistage superheterodyne receive and transmit signal chains [4].

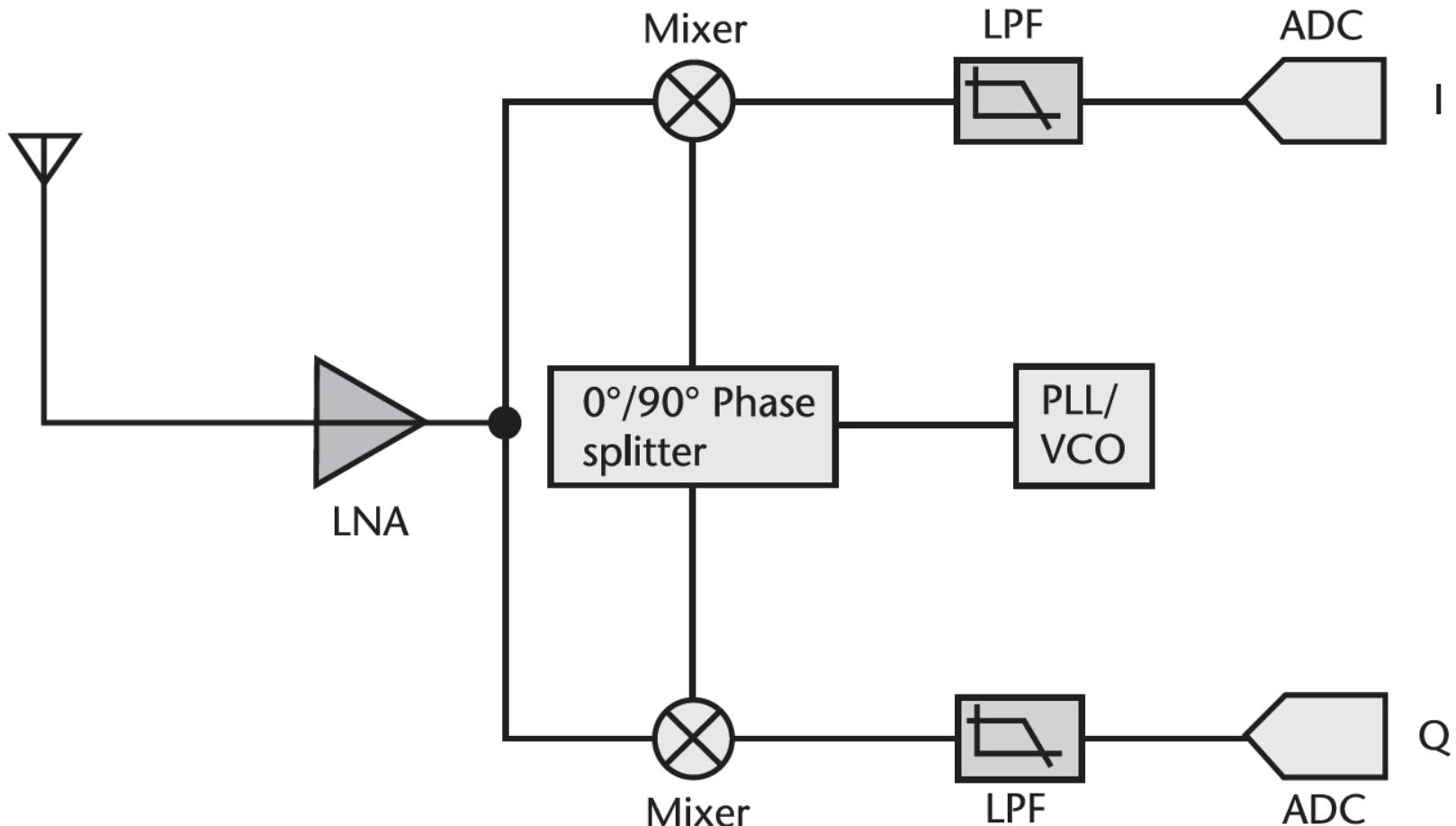


Figure 1.7 Zero IF architecture [4].

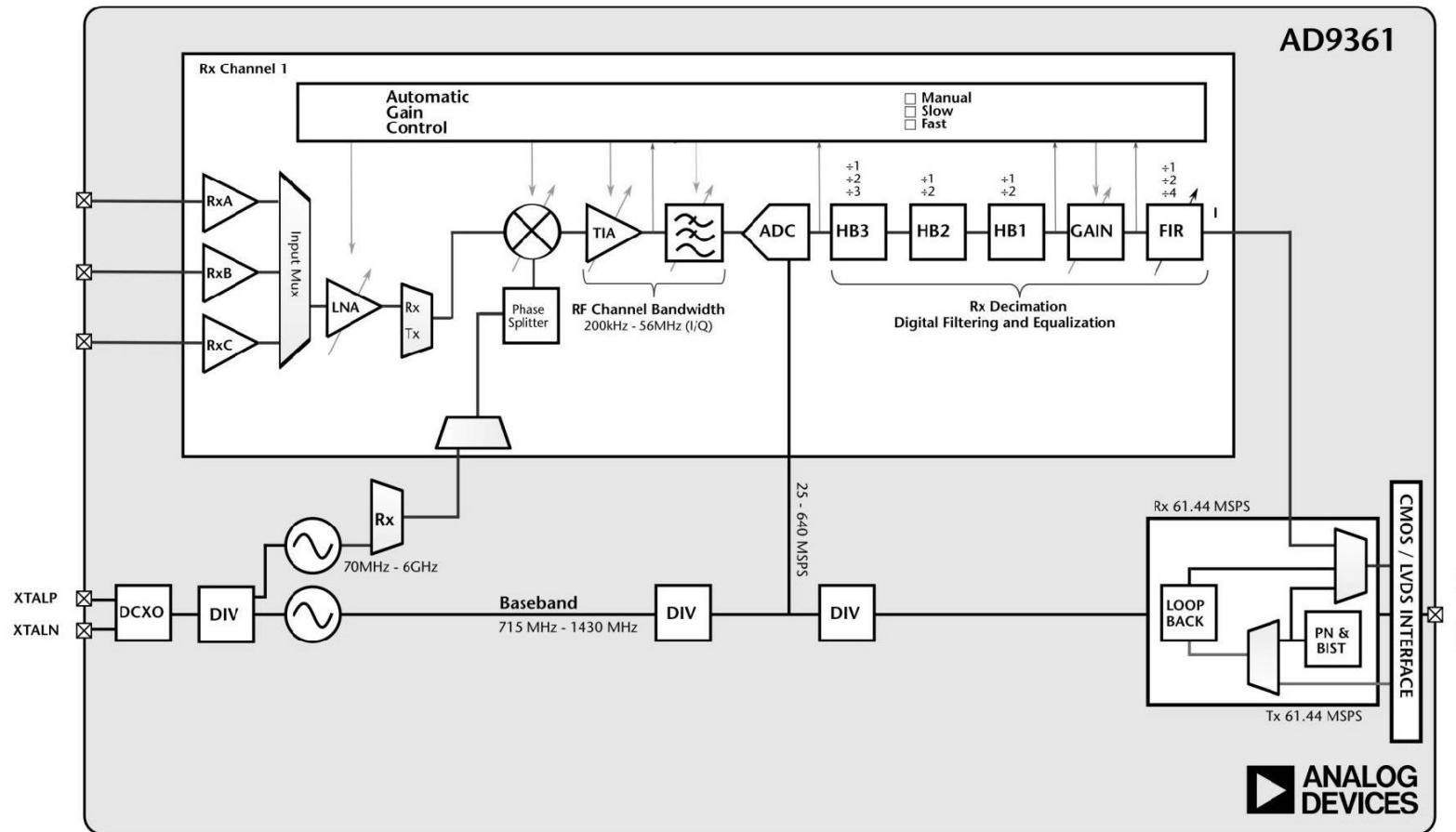


Figure 2.32 Simplified AD9361 receive block diagram.

| | Units | Protocol | OSI 7-Layer Model | TCP/IP 5-Layer Model |
|---------|----------|---|--|--|
| Level 7 | Data | Resource sharing, file access, SMTP, HTTP | Application Network process to application | |
| Level 6 | Data | Compression, JPEG, ASCII, TIFF, GIF | Presentation Data representation and encryption | Application (FTP, SMTP, HTTP, etc) |
| Level 5 | Data | Security, name recognition, RPC, SQL, NFS | Session Interhost communication | |
| Level 4 | Segments | Segmentation and traffic, TCP, SPX, UDP | Transport End to End connections and reliability | TCP |
| Level 3 | Packets | Routing, fragmentation, IP, IPX, ICMP | Network Path determination and logical addressing (IP) | Network |
| Level 2 | Frames | Sequencing, traffic control, Media Access | Data Link Physical addressing (MAC & LLC) | Data Link |
| Level 1 | Bits | Bits and volts data encoding | Physical Media, signal and binary transmission | Physical |

Figure 1.4 Seven-layer OSI model compared to five-layer TCP/IP model.

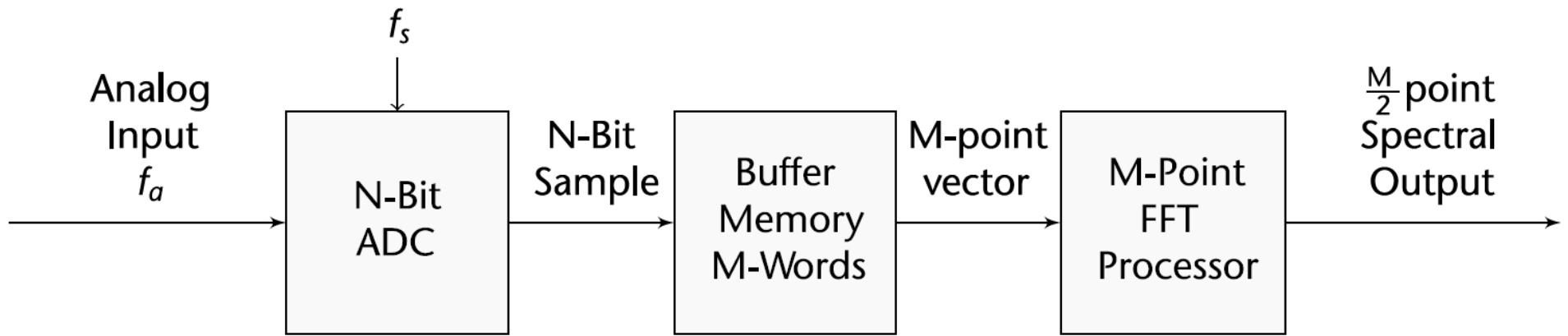


Figure 2.3 Generalized test set up for FFT analysis of ADC output [6].

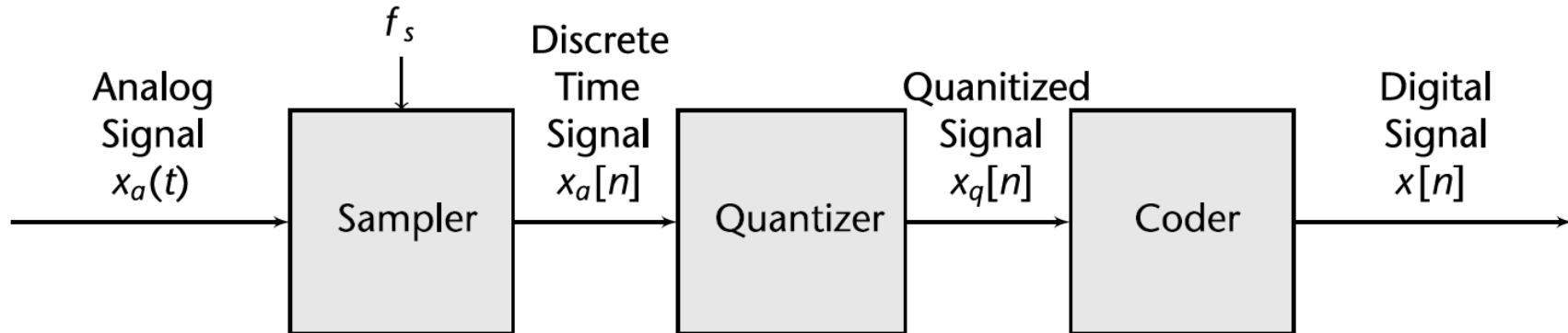


Figure 2.5 Basic parts of an analog-to-digital converter (ADC) [1]. Sampling takes place in the sampler block. $x_a(t)$ = analog continuous time signal; f_s is the digital sample rate; $x_a[n]$ is the discrete time continuous analog signal; $x_q[n]$ is the discrete time, discrete digital signal, which may come out as grey code; and $x[n]$ is the output of the coder in 2s complement form [7].

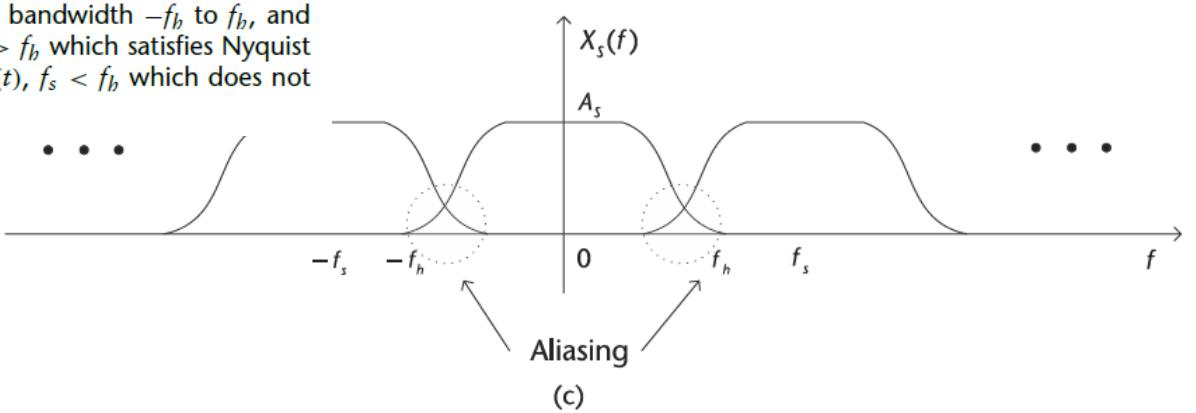
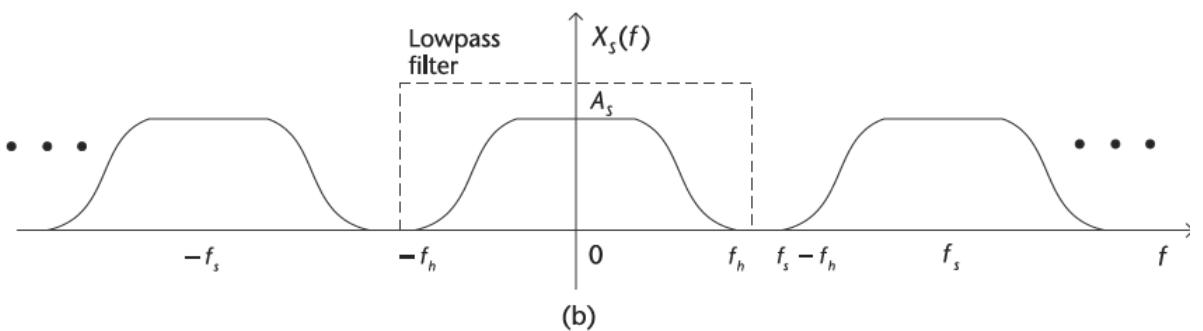
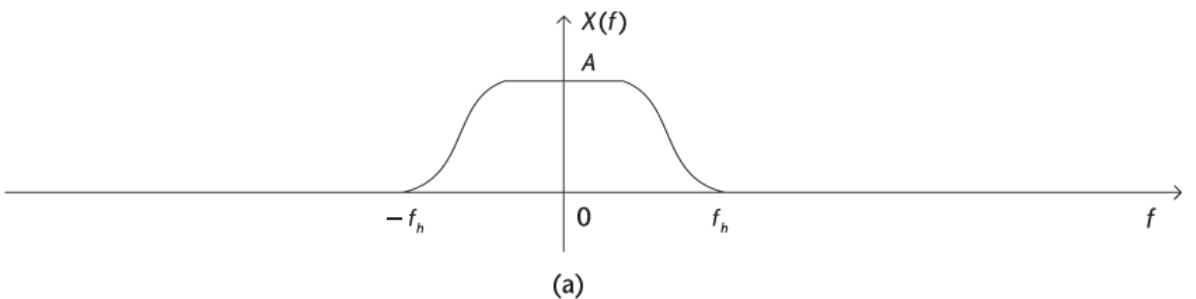


Figure 2.7 The spectrum of original signal $x(t)$ and the sampled signal $x_s(t)$ in the frequency domain. (a) The spectrum of original continuous-time signal $x(t)$, with bandwidth $-f_h$ to f_h , and amplitude A . (b) The spectrum of the digitally sampled signal $x_s(t)$, $f_s > f_h$, which satisfies Nyquist sampling theorem. (c) The spectrum of the digitally sampled signal $x_s(t)$, $f_s < f_h$ which does not satisfy Nyquist sampling theorem and has aliasing.

2.2.4 Nyquist Zones

The Nyquist bandwidth itself is defined to be the frequency spectrum from DC to $\frac{f_s}{2}$. However, the frequency spectrum is divided into an infinite number of Nyquist zones, each having a width equal to $0.5 f_s$ as shown in Figure 2.9. The frequency spectrum does not just end because you are not interested in those frequencies.

This implies that some filtering ahead of the sampler (or ADC) is required to remove frequency components that are outside the Nyquist bandwidth, but whose aliased components fall inside it. The filter performance will depend on how close the out-of-band signal is to $\frac{f_s}{2}$ and the amount of attenuation required. It is important to note that with no input filtering at the input of the ideal sampler (or ADC), any frequency component (either signal or noise) that falls outside the Nyquist bandwidth in any Nyquist zone will be aliased back into the first Nyquist zone. For this reason, an analog antialiasing filter is used in almost all sampling ADC applications to remove these unwanted signals.

2.2.5 Sample Rate Conversion

In real-world applications, we often would like to lower the sampling rate because it reduces storage and computation requirements. In many cases we prefer a higher sampling rate because it preserves fidelity. Sampling rate conversion is a general term for the process of changing the time interval between the adjacent elements in a sequence consisting of samples of a continuous-time function [10].

Decimation: The process of lowering the sampling rate is called *decimation*, which is achieved by ignoring all but every D th sample. In time domain, it can be defined as

$$y[n] = x[nD], \quad D = 1, 2, 3, \dots, \quad (2.13)$$

where $x[n]$ is the original signal, $y[n]$ is the decimated signal, and D is the decimation rate. According to (2.13), the sampling rates of the original signal and the decimated

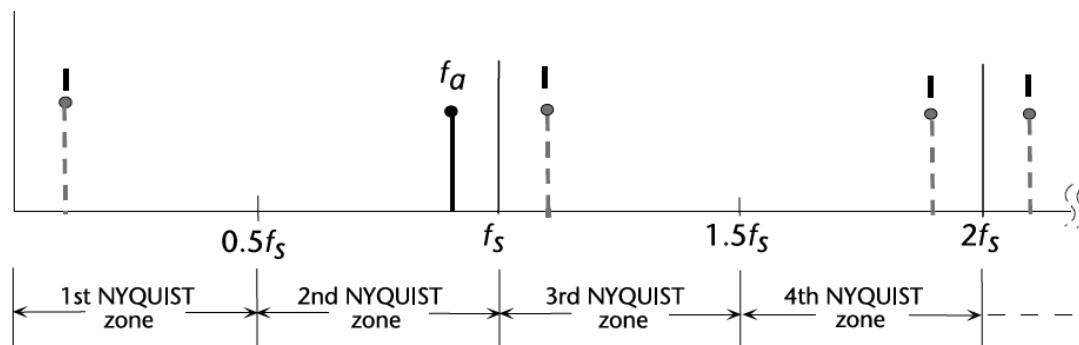


Figure 2.9 Analog signal f_a sampled at f_s has images (aliases) at $\pm kF_s \pm F_a$, $k = 1, 2, 3, \dots$

signal can be expressed as

$$F_y = \frac{F_x}{D}, \quad (2.14)$$

where F_x is the sampling rates of the original signal, and F_y is the sampling rates of the decimated signal.

Since the frequency variables in radians, ω_x and ω_y , can be related to sampling rate, F_x and F_y , by

$$\omega_x = 2\pi F T_x = \frac{2\pi F}{F_x}, \quad (2.15)$$

and

$$\omega_y = 2\pi F T_y = \frac{2\pi F}{F_y}, \quad (2.16)$$

it follows from the distributive property that ω_x and ω_y are related by

$$\omega_y = D\omega_x, \quad (2.17)$$

which means that the frequency range of ω_x is stretched into the corresponding frequency range of ω_y by a factor of D .

In order to avoid aliasing of the decimated sequence $y[n]$, it is required that $0 \leq |\omega_y| \leq \pi$. Based on (2.17), it implies that the spectrum of the original sequence should satisfy $0 \leq |\omega_x| \leq \frac{\pi}{D}$. Therefore, in reality, decimation is usually a two-step process, consisting of a lowpass antialiasing filter and a downampler, as shown in Figure 2.10. The lowpass antialiasing filter is used to constrain the bandwidth of the input signal to the downampler $x[n]$ to be $0 \leq |\omega_x| \leq \frac{\pi}{D}$.

In frequency domain, the spectrum of the decimated signal, $y[n]$, can be expressed as [1]

$$Y(\omega_y) = \frac{1}{D} \sum_{k=0}^{D-1} H_D\left(\frac{\omega_y - 2\pi k}{D}\right) S\left(\frac{\omega_y - 2\pi k}{D}\right), \quad (2.18)$$

where $S(\omega)$ is the spectrum of the input signal $s[n]$, and $H_D(\omega)$ is the frequency response of the lowpass filter $h_D[n]$. With a properly designed filter $H_D(\omega)$, the aliasing is eliminated, and consequently, all but the first $k = 0$ term in (2.18) vanish [1]. Hence, (2.18) becomes

$$Y(\omega_y) = \frac{1}{D} H_D\left(\frac{\omega_y}{D}\right) S\left(\frac{\omega_y}{D}\right) = \frac{1}{D} S\left(\frac{\omega_y}{D}\right), \quad (2.19)$$

for $0 \leq |\omega_y| \leq \pi$. The spectra for the sequence $x[n]$ and $y[n]$ are illustrated in Figure 2.11, where the frequency range of the intermediate signal is $0 \leq |\omega_x| \leq \frac{\pi}{D}$, and the frequency range of the decimated signal is $0 \leq |\omega_y| \leq \pi$.



$$Y(\omega_y) = \frac{1}{D} \sum_{k=0}^{D-1} H_D\left(\frac{\omega_y - 2\pi k}{D}\right) S\left(\frac{\omega_y - 2\pi k}{D}\right), \quad (2.18)$$

where $S(\omega)$ is the spectrum of the input signal $s[n]$, and $H_D(\omega)$ is the frequency response of the lowpass filter $h_D[n]$. With a properly designed filter $H_D(\omega)$, the aliasing is eliminated, and consequently, all but the first $k = 0$ term in (2.18) vanish [1]. Hence, (2.18) becomes

$$Y(\omega_y) = \frac{1}{D} H_D\left(\frac{\omega_y}{D}\right) S\left(\frac{\omega_y}{D}\right) = \frac{1}{D} S\left(\frac{\omega_y}{D}\right), \quad (2.19)$$

for $0 \leq |\omega_y| \leq \pi$. The spectra for the sequence $x[n]$ and $y[n]$ are illustrated in Figure 2.11, where the frequency range of the intermediate signal is $0 \leq |\omega_x| \leq \frac{\pi}{D}$, and the frequency range of the decimated signal is $0 \leq |\omega_y| \leq \pi$.

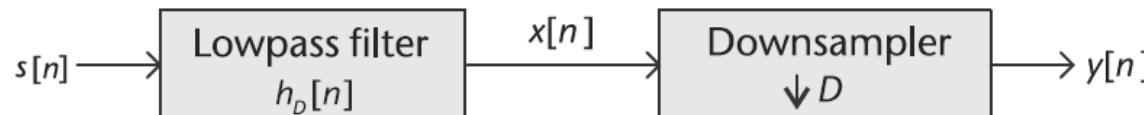
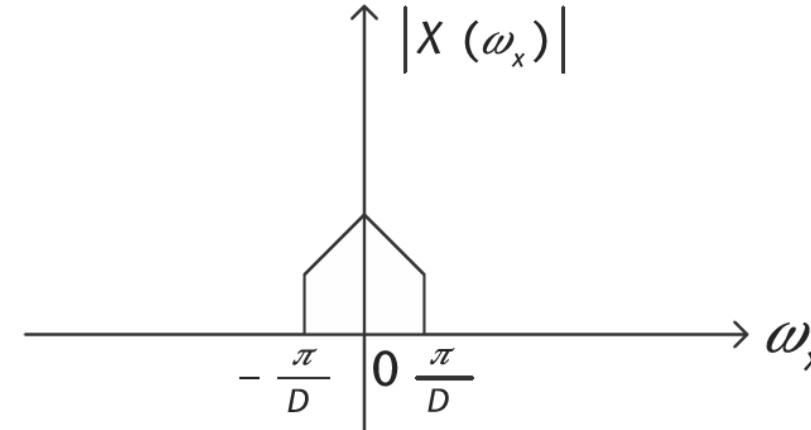
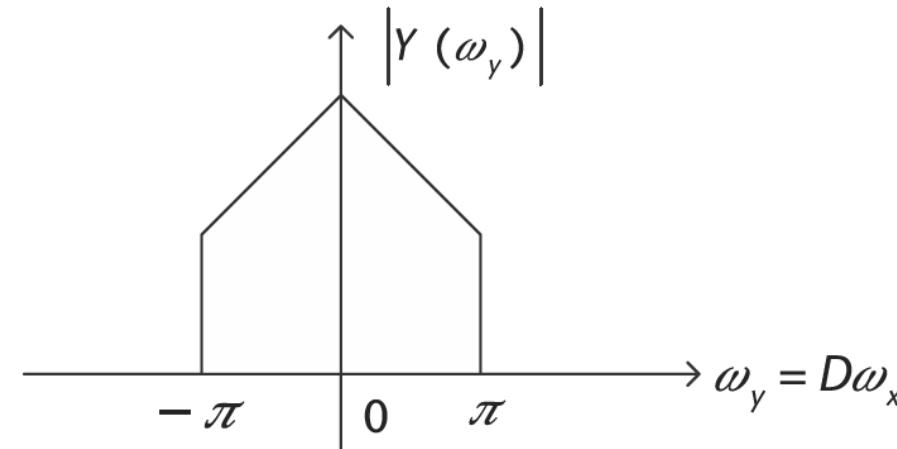


Figure 2.10 The structure of decimation, consisting of a lowpass antialiasing filter and a downsampler.

Figure 2.11 The spectra for the sequence $x[n]$ and $y[n]$, where the frequency range of ω_x is stretched into the corresponding frequency range of ω_y by a factor of D . (a) Spectrum of the intermediate sequence, and (b) spectrum of the decimated sequence.



(a)



Extraido de: <https://www.analog.com/en/education/education-library/software-defined-radio-for-engineers.html>

Interpolation: The process of increasing the sampling rate is called *interpolation*, which can be accomplished by interpolating (stuffing zeros) $I - 1$ new samples between successive values of signal. In time domain, it can be defined as

$$y[n] = \begin{cases} x[n/I] & n = 0, \pm I, \pm 2I, \dots \\ 0 & \text{otherwise} \end{cases}, \quad I = 1, 2, 3, \dots, \quad (2.20)$$

where $x[n]$ is the original signal, $y[n]$ is the interpolated signal, and I is the interpolation rate.

According to (2.20), the sampling rates of the original signal and the interpolated signal can be expressed as

$$F_y = IF_x, \quad (2.21)$$

where F_x is the sampling rates of the original signal, and F_y is the sampling rates of the interpolated signal. Since (2.15) and (2.16) also hold here, it follows that ω_x and ω_y are related by

$$\omega_y = \frac{\omega_x}{I}, \quad (2.22)$$

which means that the frequency range of ω_x is compressed into the corresponding frequency range of ω_y by a factor of I . Therefore, after the interpolation, there will be I replicas of the spectrum of $x[n]$, where each replica occupies a bandwidth of $\frac{\pi}{I}$.



Since only the frequency components of $y[n]$ in the range $0 \leq |\omega_y| \leq \frac{\pi}{I}$ are unique (i.e., all the other replicas are the same as this one), the images of $Y(\omega)$ above $\omega_y = \frac{\pi}{I}$ should be rejected by passing it through a lowpass filter with the following frequency response:

$$H_I(\omega_y) = \begin{cases} C & 0 \leq |\omega_y| \leq \frac{\pi}{I} \\ 0 & \text{otherwise} \end{cases}, \quad (2.23)$$

where C is a scale factor.

Therefore, in reality, interpolation is also a two-step process, consisting of an upsampler and a lowpass filter, as shown in Figure 2.12. The spectrum of the output signal $z[n]$ is

$$Z(\omega_z) = \begin{cases} CX(\omega_z I) & 0 \leq |\omega_z| \leq \frac{\pi}{I} \\ 0 & \text{otherwise} \end{cases}, \quad (2.24)$$

where $X(\omega)$ is the spectrum of the output signal $x[n]$.

The spectra for the sequence $x[n]$, $y[n]$ and $z[n]$ are illustrated in Figure 2.13, where the frequency range of the original signal is $0 \leq |\omega_x| \leq \pi$, and the frequency range of the decimated signal is $0 \leq |\omega_z| \leq \frac{\pi}{I}$.

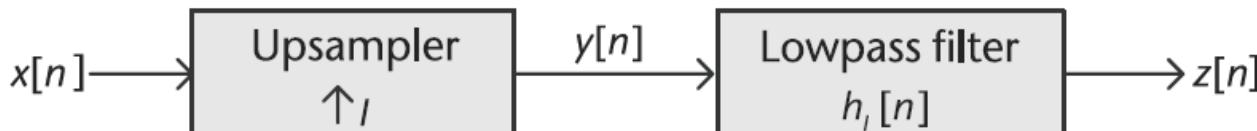


Figure 2.12 The structure of interpolation, consisting of an upsampler and a lowpass filter.

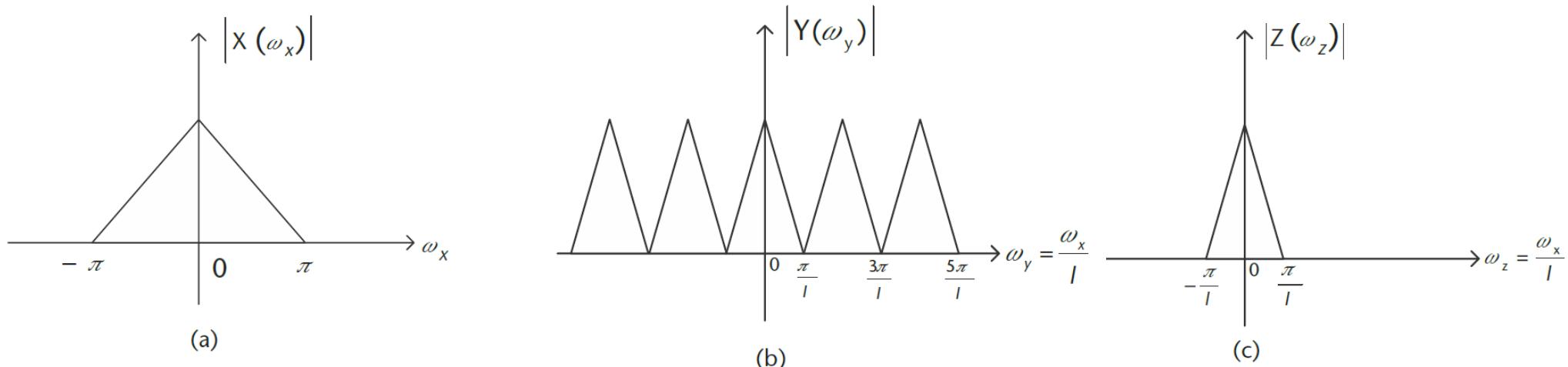


Figure 2.13 The spectra for the sequence $x[n]$, $y[n]$ and $z[n]$, where the frequency range of ω_x is compressed into the corresponding frequency range of ω_y by a factor of I . (a) Spectrum of the original sequence, (b) spectrum of the intermediate sequence, and (c) spectrum of the interpolated sequence.

2.3 Signal Representation

Understanding how a signal is represented can greatly enhance one's ability to analyze and design digital communication systems. We need multiple convenient numeric mathematical frameworks to represent actual RF, baseband, and noise signals. We usually have two: envelope/phase and in-phase/quadrature, and both can be expressed in the time and Fourier domains.

2.3.1 Frequency Conversion

To understand how we can move signals from baseband to RF and from RF to baseband, let us look more closely at modulators and demodulators. For example, in Figure 2.18 we see a very classical quadrature modulator. The ADL5375 accepts two differential baseband inputs and a single-ended LO, which generates a single-ended output. The LO interface generates two internal LO signals in quadrature (90° out of phase) and these signals are used to drive the mixers, which simply multiply the LO signals with the input.

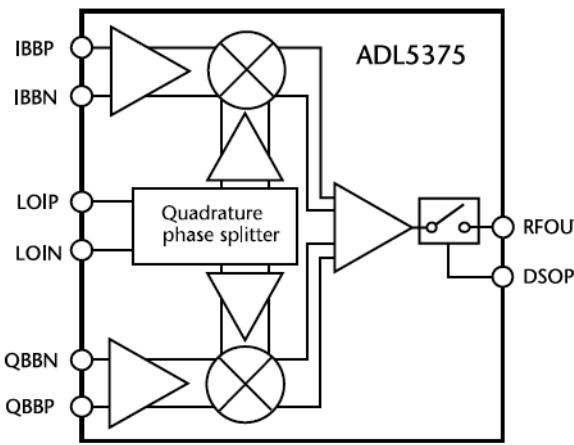


Figure 2.18 ADL5375 broadband quadrature modular with range from 400 MHz to 6 GHz.

Mathematically, this mixing process will take the two inputs IBB and QBB , which we will denote by $I(t)$ and $Q(t)$, multiply them by our LO at frequency ω_c , and add the resulting signal to form our transmitted signal $r(t)$. The LO used to multiply $Q(t)$ is phase shifted by 90° degree to make it orthogonal with the multiplication of the $I(t)$ signal. Consequently, this yields in the following equation:

$$r(t) = I(t)\cos(\omega_c t) - Q(t)\sin(\omega_c t). \quad (2.25)$$

The LO frequency is denote as ω_c since it will be typically called the carrier frequency, which exploits the phase relationship between the in-phase ($I(t)\cos(\omega_c t)$) and quadrature ($Q(t)\sin(\omega_c t)$) components. Therefore, the transmitted signal will contain both components but will appear as a single sinusoid.

At the receiver, we will translate or down mix $r(t)$ back into our in-phase and quadrature baseband signals through a similar process but in reverse. By applying the same LO with a second phase-shifted component to $r(t)$ with a lowpass filter,

we arrive at

$$I_r(t) = LPF\{r(t)\cos(\omega_c t)\} = LPF\{(I(t)\cos(\omega_c t) - Q(t)\sin(\omega_c t))\cos(\omega_c t)\} = \frac{I(t)}{2}, \quad (2.26)$$

$$Q_r(t) = LPF\{r(t)\sin(\omega_c t)\} = LPF\{(-I(t)\cos(\omega_c t) + Q(t)\sin(\omega_c t))\sin(\omega_c t)\} = \frac{Q(t)}{2}. \quad (2.27)$$

In practice, there will be some phase difference between the transmitter LO and receiver LO, which can cause rotation to $r(t)$. However, the phase relation between $I(t)$ and $Q(t)$ will always be maintained.

2.3.2 Imaginary Signals

Discussing signals as *quadrature* or *complex* signal is taking advantage of the mathematical constructs that Euler and others have created to make analysis easier. We try to refer to signals as in-phase (I) and quadrature (Q) since that is actually pedantically correct. As described in Section 2.3.1, the in-phase (I) refers to the signal that is in the same phase as the local oscillator, and the quadrature (Q) refers to the part of the signal that is in phase with the LO shifted by 90°.

It is convenient to describe this as I being *real* and Q being *imaginary* since it enables many mathematical techniques but at the end of the day is just a construct. A prime example of this convenience is frequency translation, which is performed by the mixer. We start from the Euler relation of

$$e^{jx} = \cos(x) + j \sin(x), \quad (2.28)$$

where we can define x as target frequency plus time. Taking the conventions from Section 2.3.1 but redefining $I(t)$ and $Q(t)$ as real and imaginary, we arrive at

$$y(t) = I(t) + jQ(t). \quad (2.29)$$

Now, if we assume $y(t)$ is a CW tone at frequency ω_a for illustration purposes, $y(t)$ becomes

$$y(t) = \cos(\omega_a t) + j \sin(\omega_a t). \quad (2.30)$$

Now applying (2.28) we can frequency shift $y(t)$ by the desired frequency ω_c :

$$\begin{aligned} y(t)e^{j\omega_ct} &= (I(t)\cos(\omega_ct) - Q(t)\sin(\omega_ct)) + j(Q(t)\cos(\omega_ct) + I(t)\sin(\omega_ct)) \\ &= \cos((\omega_c + \omega_a)t) + j \sin((\omega_c + \omega_a)t). \end{aligned} \quad (2.31)$$

Now our resulting signal will exist at frequency $\omega_a + \omega_c$ through a simple application of Euler's identity.

Extraído de: <https://www.analog.com/en/education/education-library/software-defined-radio-for-engineers.html>



Let us motivate the usefulness of a complex number representation further from the perspective of hardware. If we consider the mixer itself as in Figure 2.18, the IQ mixer will transmit signals with arbitrary phase and amplitude (within power constraints). This is an example of how we encode information into the data we transmit through differences in phase and amplitude. This is actually accomplished through the relation of our in-phase and quadrature signal, which can be used to create a single sinusoid with arbitrary phase and amplitude. Mathematically, we can produce a sinusoid with a specific envelope and phase (A, ϕ) with two orthogonal components sine and cosine. This relationship is written as

$$A \sin(\omega t + \phi) = (A \cos \phi) \sin(\omega t) + (A \sin \phi) \cos(\omega t). \quad (2.32)$$

Therefore, by just modifying the amplitude of our sine and cosine components over time, we can create the desired waveform from a fixed frequency and phase LO. Alternatively, we can consider others coordinate systems to visualize complex values. Expanding these complex numbers (rectangular coordinates) can be translated in order to be represent a magnitude and angle (polar) or even plotted as a vector.

The in-phase and quadrature sine waves plotted in Figure 2.19 show how things look via a phasor plot as time increases, with the vector indicating magnitude rotating around the axis. One can clearly see the phase shift between I and Q. In the time domain, you can also see the differences between in-phase and magnitude, although phase differences can be a little more subtle to notice. In a Cartesian plane, the signal appears as a rotating circle over time. The phasor plot will always rotate counterclockwise with time, and the Cartesian plot can rotate in either direction depending on if the phase difference between I and Q is positive or negative. While the time domain plot shows things moving as time changes, the phasor plot and Cartesian plane are snapshots in time (at $t = 0$ on the time domain plot). Finally, the frequency domain plot provides the spectrum of the phasor but only communicates magnitude and loses phase information. This happens because we are only plotting the real component of the spectrum. However, comparing the two waveforms in Figures 2.19 and 2.20, changes in the I and Q components (amplitude and phase relationship), will effect the other domains as well.

2.4 Signal Metrics and Visualization

Before an engineer can decide whether a project is done, he or she needs to conduct some form of verification. However, communications systems are complex to evaluate given their integrated nature and depth of transmit and receive chains. As referenced in Section 1.4, communication systems may have a variety of metrics beyond size, weight, power, and cost (SWaP-C). Performance metrics such as bit error rate (BER), data throughput, and distance are also only top-level system

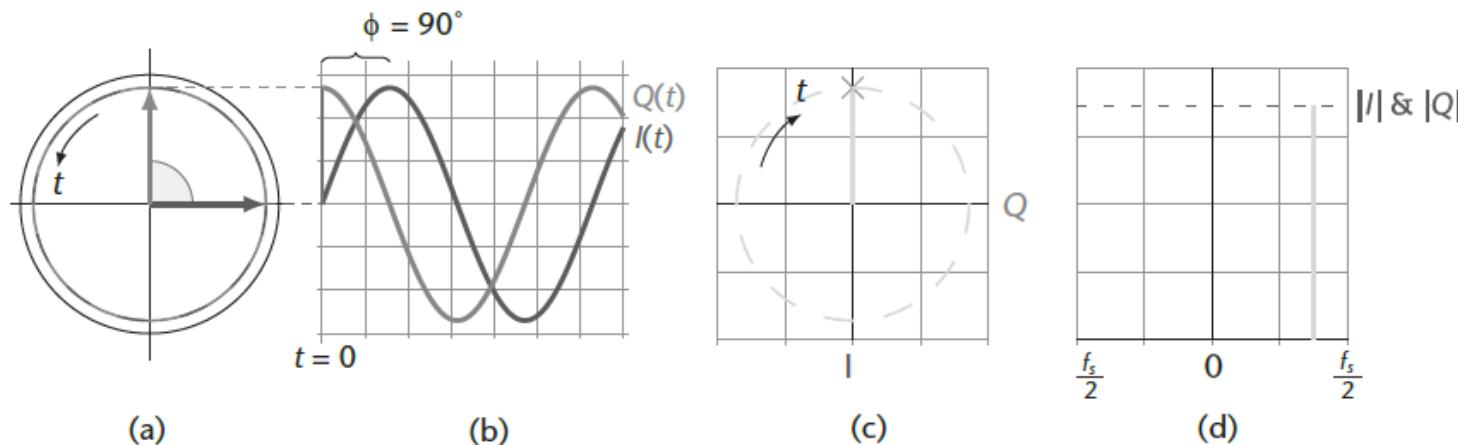


Figure 2.19 Same continuous wave signal, plotted in multiple domains. (a) Phasor $\text{rad}(t)$, (b) time $x(t) \rightarrow$, (c) Cartesian $(I, Q)(t)$, and (d) frequency $X(\omega)$.

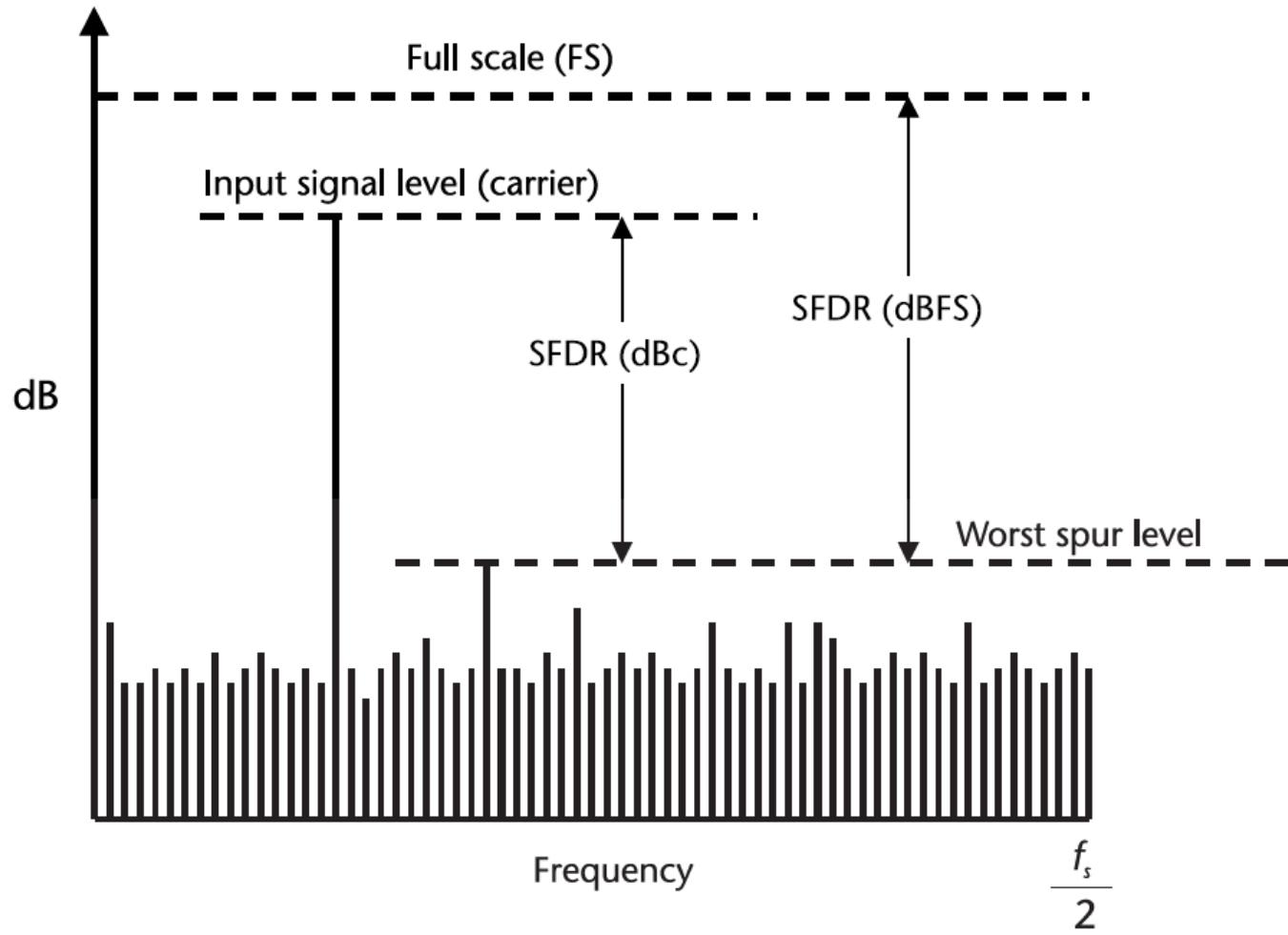


Figure 2.21 Spurious free dynamic range (SFDR) for BW DC to $f_s/2$.

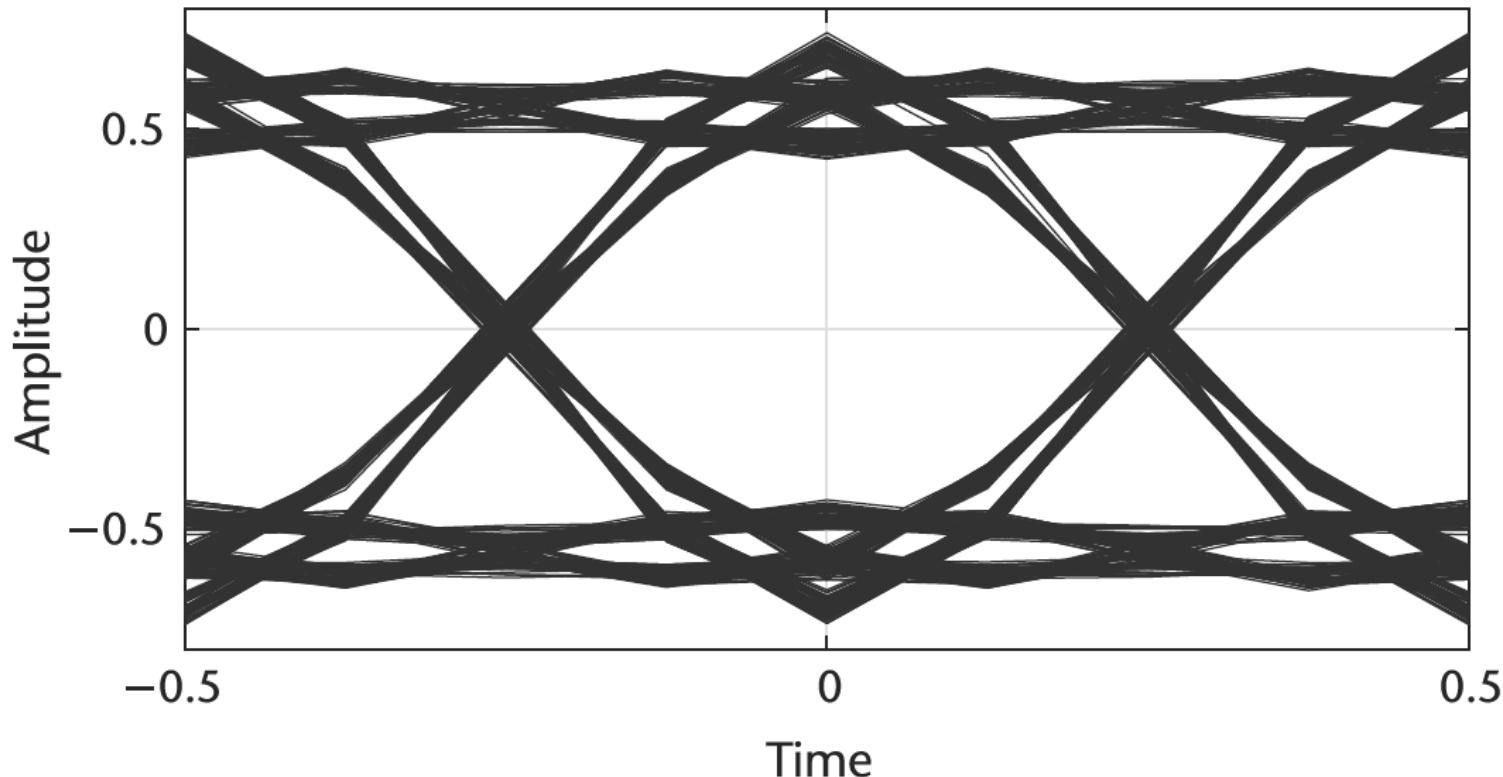


Figure 2.22 A typical eye pattern for the BPSK signal. The width of the opening indicates the time over which sampling for detection might be performed. The optimum sampling time corresponds to the maximum eye opening, yielding the greatest protection against noise.

2.5.2 Fixed Point Quantization

The only errors (DC or AC) associated with an ideal N -bit data converter are those related to the sampling and quantization processes. The maximum error an ideal converter makes when digitizing a signal is $\pm \frac{1}{2}$ LSB, directly in between two digital values. This is obvious from the transfer function of an ideal N -bit ADC, which is

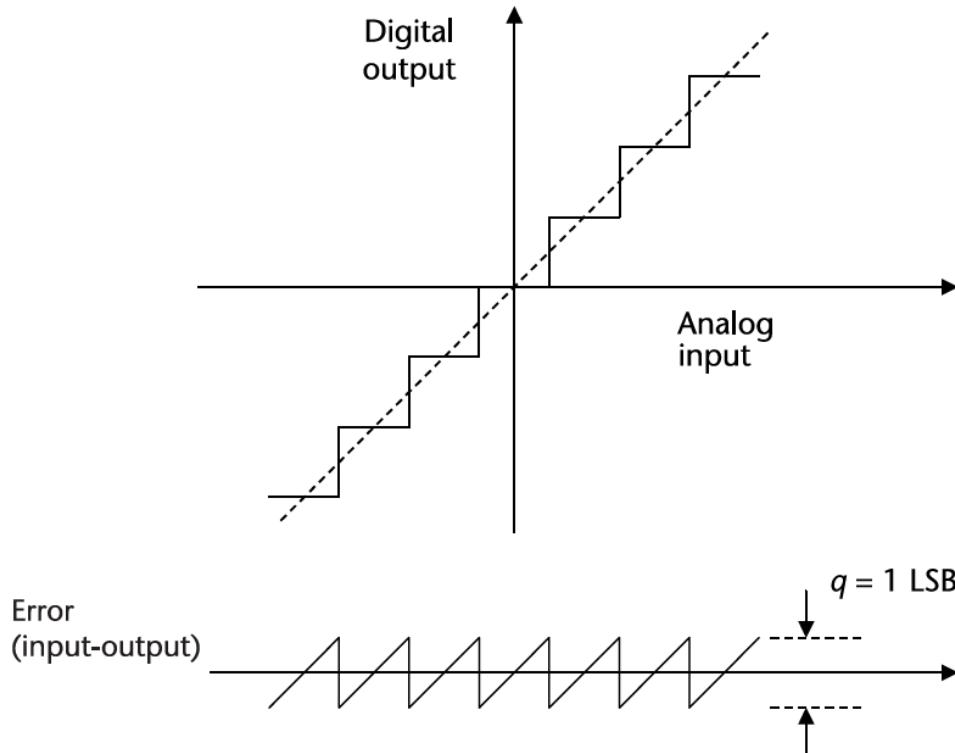


Figure 2.26 Ideal N -bit ADC quantization noise.

shown in Figure 2.26. The quantization error for any AC signal, which spans more than a few LSBs, can be approximated by an uncorrelated sawtooth waveform having a peak-to-peak amplitude of q , the weight of an LSB. Although this analysis is not precise it is accurate enough for most applications.

The quantization error as a function of time is shown in Figure 2.27. Again, a simple sawtooth waveform provides a sufficiently accurate model for analysis. The equation of the sawtooth error is given by

$$e(t) = st, \quad \frac{-q}{2s} < t < \frac{q}{2s}, \quad (2.33)$$

where s is the slope of the quantized noise. The mean-square value of $e(t)$ can be written:

$$\bar{e}^2(t) = \frac{q}{s} \int_{\frac{q}{2s}}^{\frac{-q}{2s}} (st)^2 dt = \frac{q^2}{12}. \quad (2.34)$$

The square root of (2.34), the root mean squared (RMS) noise quantization error, is approximately Gaussian and spread more or less uniformly over the Nyquist bandwidth of DC to $\frac{f_s}{2}$.

The theoretical SNR can now be calculated assuming a full-scale input sine wave $v(t)$

$$v(t) = \frac{q2^N}{2} \sin(\omega t), \quad (2.35)$$

Extraido de: <https://www.analog.com/en/education/education-library/software-defined-radio-for-engineers.html>



The theoretical SNR can now be calculated assuming a full-scale input sine wave $v(t)$

$$v(t) = \frac{q2^N}{2} \sin(\omega t), \quad (2.35)$$

by first calculating the RMS value of the input signal defined as

$$\sqrt{\bar{v}(t)^2} = \frac{q2^N}{2\sqrt{2}}. \quad (2.36)$$

Therefore, the RMS signal-to-noise ratio for an ideal N -bit converter is

$$SNR = 20 \log_{10} \left(\frac{RMS \text{ of full scale input}}{RMS \text{ of quantization noise}} \right) = 20 \log_{10} \left(\frac{\frac{q2^N}{2\sqrt{2}}}{\frac{q}{\sqrt{12}}} \right). \quad (2.37)$$

After some simplification of (2.37) we arrive at our SNR in dB:

$$SNR = 20 \log_{10} \left(\sqrt{\frac{3}{2}} 2^N \right) = 6.02N + 1.76. \quad (2.38)$$

$$SNR = 20 \log_{10} \left(\sqrt{\frac{3}{2}} 2^N \right) = 6.02N + 1.76. \quad (2.38)$$

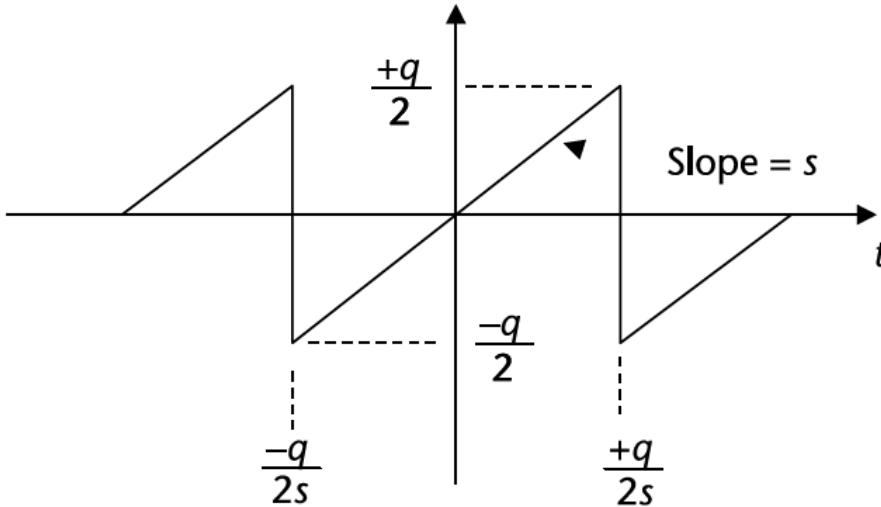


Figure 2.27 Ideal N-bit ADC quantization noise as a function of time.

2.5.4 Sigma-Delta Analog-Digital Converters

Sigma-delta (Σ - Δ) analog-digital converters (ADCs) have been known for over 50 years, but only recently has the technology (high-density digital VLSI) existed to manufacture them as inexpensive monolithic integrated circuits. They are now used in many applications where a low-cost, medium-bandwidth, low-power, high-resolution ADC is required. There have been innumerable descriptions of the architecture and theory of Σ - Δ ADCs, but most commence with a deep description of the math, starting at the integrals and go on from there. Since this is not an ADC textbook, we will try to refrain from the mathematical development and explore things based on the previous topics covered in this chapter.

There is nothing particularly difficult to understand about Σ - Δ ADCs. The Σ - Δ ADC contains very simple analog electronics (a comparator, voltage reference, a switch, and one or more integrators and analog summing circuits), and digital computational circuitry. This circuitry consists of a filter, which is generally, but not invariably, a lowpass filter. It is not necessary to know precisely how the filter works to appreciate what it does. To understand how a Σ - Δ ADC works, familiarity with the concepts of oversampling, quantization noise shaping, digital filtering, and decimation is required, all topics covered earlier in this chapter.

Let us consider the technique of oversampling with an analysis in the frequency domain. Where a DC conversion has a quantization error of up to $\frac{1}{2}$ LSB, a sampled data system has quantization noise. A perfect classical N-bit sampling ADC has an RMS quantization noise of $\frac{q}{\sqrt{12}}$ uniformly distributed within the Nyquist band of DC to $\frac{f_s}{2}$, where q is the value of an LSB, as shown in Figure 2.33(a). Therefore, its SNR with a full-scale sine wave input will be $(6.02N + 1.76)$ dB. If the ADC is less than perfect and its noise is greater than its theoretical minimum quantization noise, then its effective resolution will be less than N -bits. Its actual resolution, often known as its effective number of bits (ENOB), will be defined by

$$ENOB = \frac{SNR - 1.76dB}{6.02dB} \quad (2.39)$$

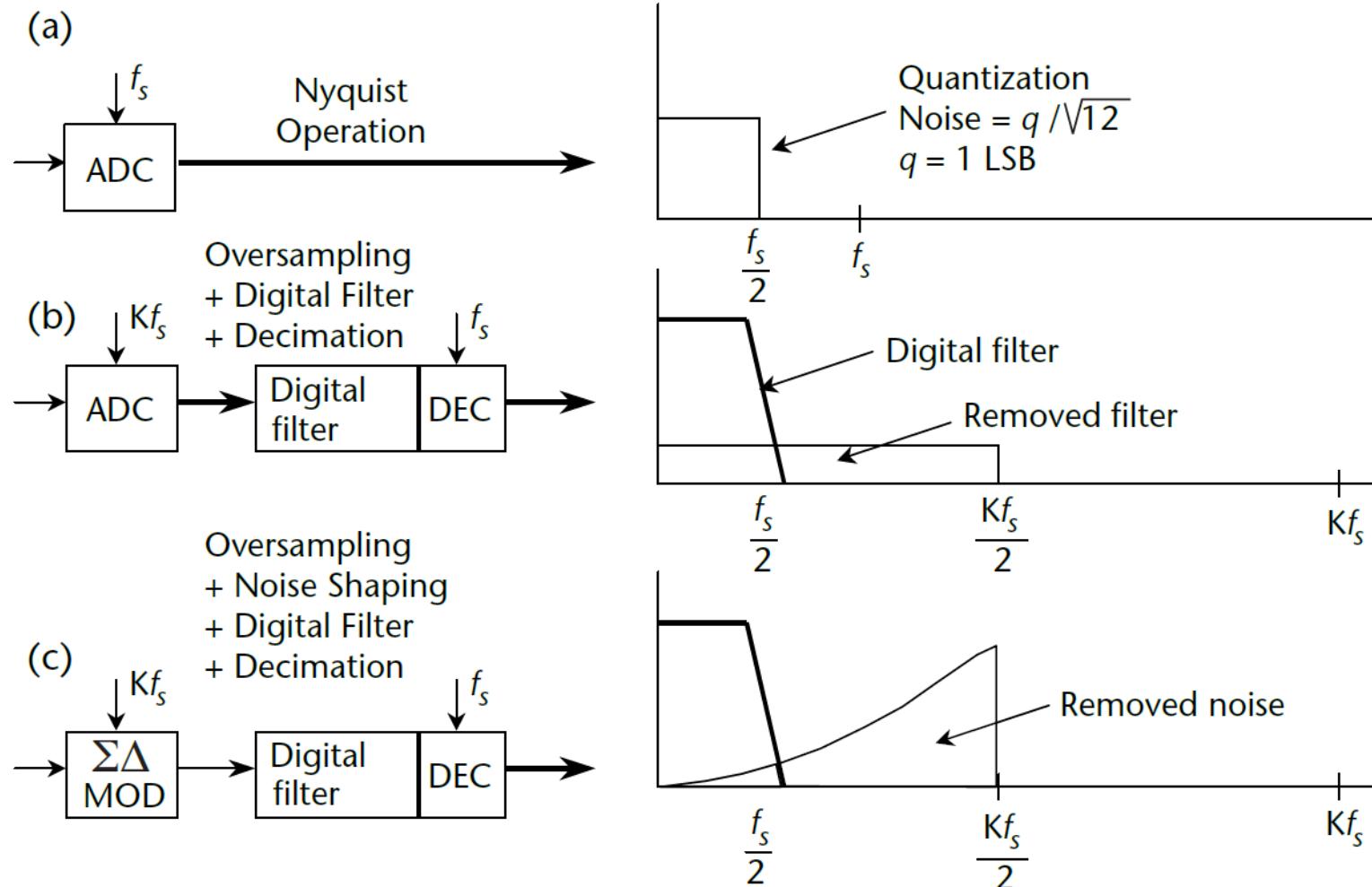


Figure 2.33 Oversampling, digital filtering, noise shaping, and decimation in a $\Sigma\Delta$ -ADC.

Practically, ENOB is calculated from measuring signal-to-noise-and-distortion (SINAD, or $S/(N + D)$), which is the ratio of the RMS signal amplitude to the mean value of the root-sum-square (RSS) of all other spectral components, including harmonics but excluding DC, and correcting for a nonfull-scale input signal [6]. We can modify (2.39) to take into account the full-scale amplitude A_{FS} and the true input amplitude A_{IN} as

$$ENOB = \frac{SINAD - 1.76dB + 20\log_{10}\frac{A_{FS}}{A_{IN}}}{6.02dB}. \quad (2.40)$$

If we choose a much higher sampling rate, Kf_s (see Figure 2.33[b]), the RMS quantization noise remains $\frac{q}{\sqrt{12}}$ but the noise is now distributed over a wider bandwidth DC to $\frac{Kf_s}{2}$. If we then apply a digital lowpass filter (LPF) to the output, we can remove much of the quantization noise but do not affect the wanted signal, resulting in an improved ENOB. Therefore, we can accomplish a high-resolution A/D conversion with a low-resolution ADC. The factor K is generally referred to as the oversampling ratio. It should be noted at this point that oversampling has an added benefit in that it relaxes the requirements on the analog antialiasing filter.

Since the bandwidth is reduced by the digital output filter, the output data rate may be lower than the original sampling rate (Kf_s) and still satisfy the Nyquist criterion. This may be achieved by passing every M^{th} result to the output and discarding the remainder. The process is known as decimation by a factor of M . Decimation does not cause any loss of information (see Figure 2.33[b]) as long as the decimation does not violate the Nyquist criterion. For a given input frequency, higher-order analog filters offer more attenuation. The same is true of $\Sigma-\Delta$ modulators, provided certain precautions are taken. By using more than one integration and summing stage in the $\Sigma-\Delta$ modulator, we can achieve higher orders of quantization noise shaping and even better ENOB for a given oversampling ratio as is shown in Figure 2.34.

The actual $\Sigma-\Delta$ ADC found in the AD9363 used in the Pluto SDR is a fourth order, as shown in Figure 2.35 and described in the *Analog Devices Transceiver Support* Simulink model. As can be seen, reality is always a little more complicated than theory or first-order approximations.

2.6 Digital Signal Processing Techniques for SDR

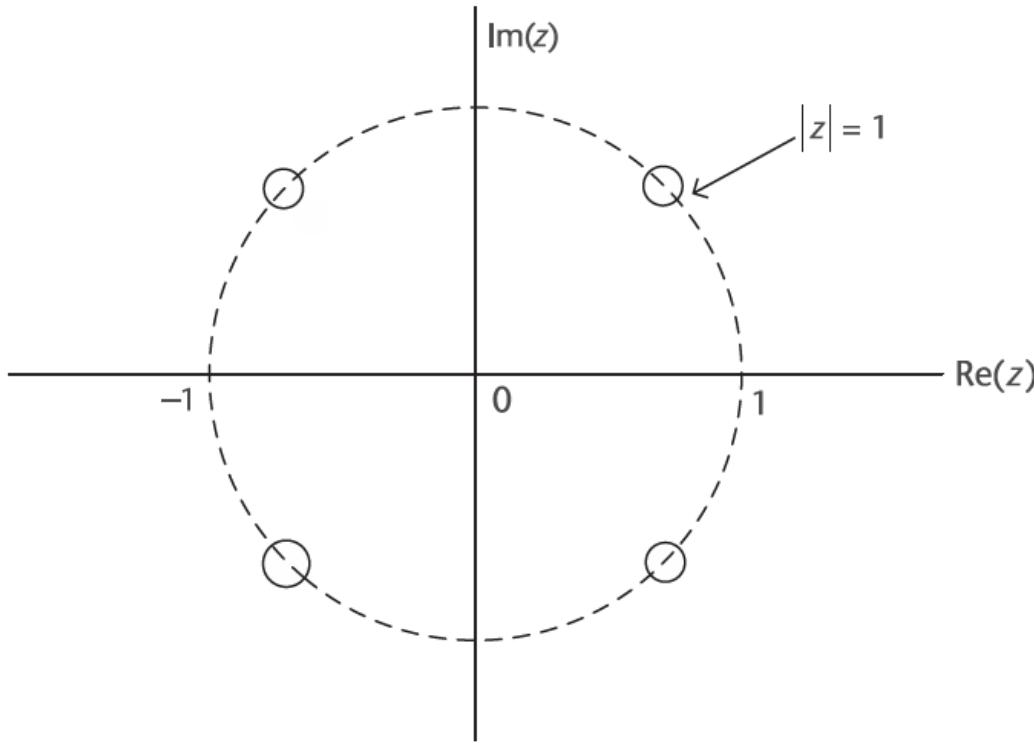


Figure 2.43 The zeros of a CIC filter defined in (2.65), where all the zeros are on the unit circle.

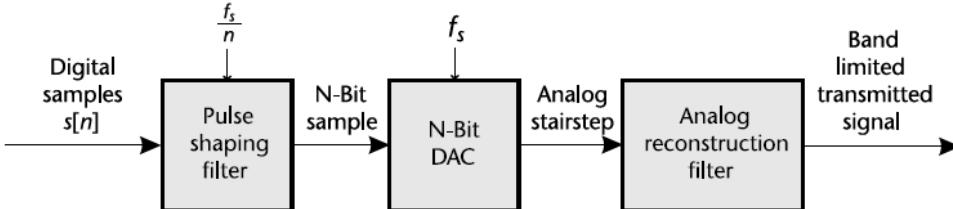


Figure 2.46 On the transmitter side, the DAC converts the digital symbols into an analog signal for transmission.

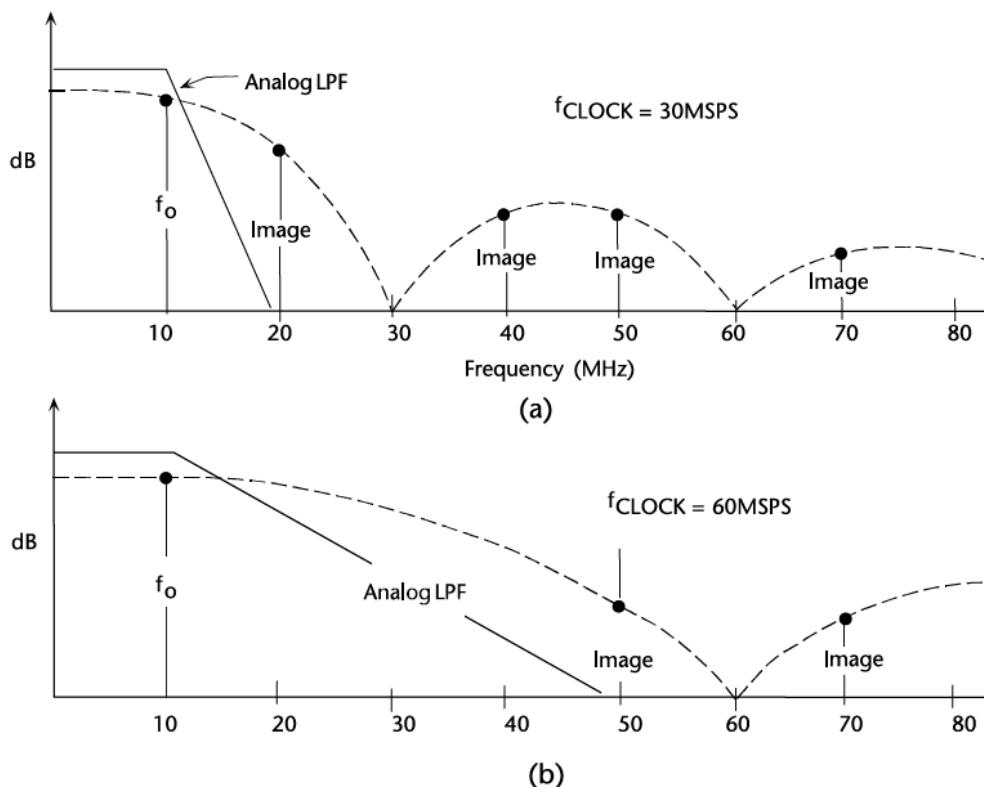


Figure 2.47 Analog reconstruction filter requirements for $f_o = 10$ MHz, with $f_s = 30$ MSPS, and $f_s = 60$ MSPS [18].

Extraido de: <https://www.analog.com/en/education/education-library/software-defined-radio-for-engineers.html>

2.7.4 Nyquist Pulse-Shaping Theory

In a communication system, there are normally two pulse-shaping filters, one on the transmitter side, and the other on the receiver side, as shown in Figure 2.49, where

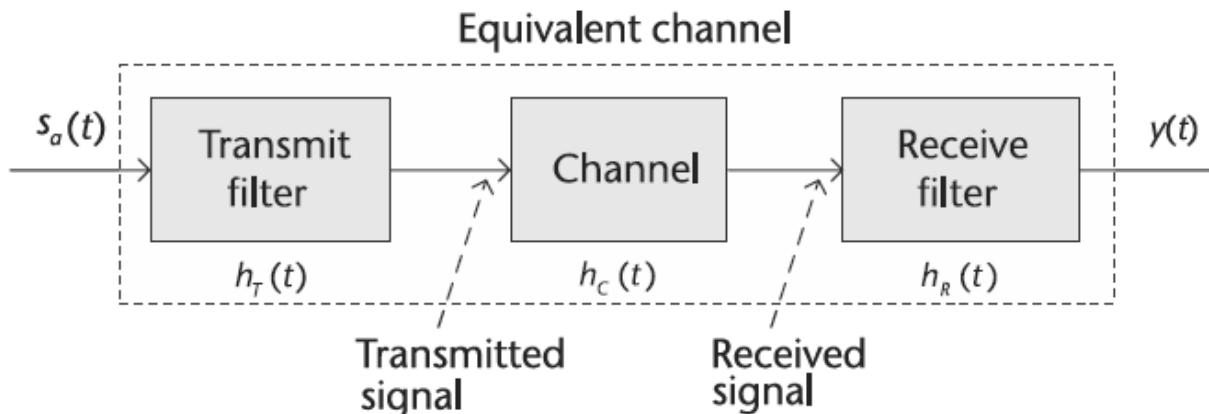


Figure 2.49 The equivalent channel of a communication system, which consists of the transmit filter, the channel, and the receive filter.

we use $h_T(t)$ and $h_R(t)$ to denote the impulse response of the transmit filter and receive filter. For simplicity, when considering the Nyquist pulse-shaping theory, we usually use the concept of equivalent channel, which not only includes the channel itself, but also the two pulse-shaping filters. Therefore, the impulse response of the equivalent channel is

$$h(t) = h_T(t) * h_C(t) * h_R(t). \quad (2.71)$$

Extraído de: <https://www.analog.com/en/education/education-library/software-defined-radio-for-engineers.html>

$$h(t) = h_T(t) * h_C(t) * h_R(t). \quad (2.71)$$

Now, let us consider under which condition we can assure that there is no intersymbol interference between symbols. The input to the equivalent channel, $s_a(t)$, has been defined in (2.70). As mentioned before, each analog pulse is scaled by the value of the symbol, so we can express $s_a(t)$ in another way:

$$s_a(t) = \sum a_k \delta(t - kT), \quad (2.72)$$

where a_k is the value of the k th symbol. It yields the output to the equivalent channel, $y(t)$, which is

$$y(t) = s_a(t) * h(t) = \sum a_k [\delta(t - kT) * h(t)] = \sum a_k h(t - kT). \quad (2.73)$$

Therefore, given a specific time instant, for example, $t = mT$, where m is a constant, the input $s_a(t)$ is

$$s_a(mT) = \sum a_k \delta(mT - kT) = a_m. \quad (2.74)$$

Consequently, the output $y(t)$ becomes

$$y(mT) = \sum a_k h(mT - kT) = a_0 h(mT) + a_1 h(mT - T) + \dots + a_m h(mT - mT). \quad (2.75)$$

Since we do not want the interference from the other symbols, we would like the output to contain only the a_m term, which is

$$y(mT) = a_m h(mT - mT). \quad (2.76)$$

Extraido de: <https://www.analog.com/en/education/education-library/software-defined-radio-for-engineers.html>

$$y(mT) = a_m b(mT - mT). \quad (2.76)$$

Moreover, it means at a time instant $t = mT$, we need to have

$$h(mt - kT) = \begin{cases} C & k = m \\ 0 & k \neq m \end{cases}, \quad (2.77)$$

where C is some nonzero constant.

If we generalize (2.77) to any time instant t , we can get the Nyquist pulse-shaping theory as below. The condition that one pulse does not interfere with other

pulses at subsequent T -spaced sample instants is formalized by saying that $b(t)$ is a *Nyquist pulse* if it satisfies

$$h(t) = h(kT) = \begin{cases} C & k = 0 \\ 0 & k \neq 0 \end{cases}, \quad (2.78)$$

for all integers k .



2.7.5 Two Nyquist Pulses

In this section, we will introduce two important Nyquist pulses; namely, *sinc pulse* and *raised cosine pulse*. When considering (2.78), the most straightforward pulse we can think of is a rectangular pulse with time width less than T , or any pulse shape that is less than T wide. However, the bandwidth of the rectangular pulse (and other narrow pulse shapes) may be too wide. Narrow pulse shapes do not utilize the spectrum efficiently, but wide pulse shapes may cause ISI, so what is needed is a signal that is wide in time (and narrow in frequency) that also fulfills the Nyquist condition in (2.78) [9].

In mathematics, the sinc function is defined as

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}, \quad (2.79)$$

and is shown in Figure 2.50, when variable x takes an integer value k , the value of the sinc function will be

$$\text{sinc}(k) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}. \quad (2.80)$$

$$\text{sinc}(k) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases} . \quad (2.80)$$

In other words, zero crossings of the normalized sinc function occur at nonzero integer values.

Another important property of sinc function is that sinc pulse is the inverse Fourier transform of a rectangular signal, as shown in Figure 2.51(a). Suppose the

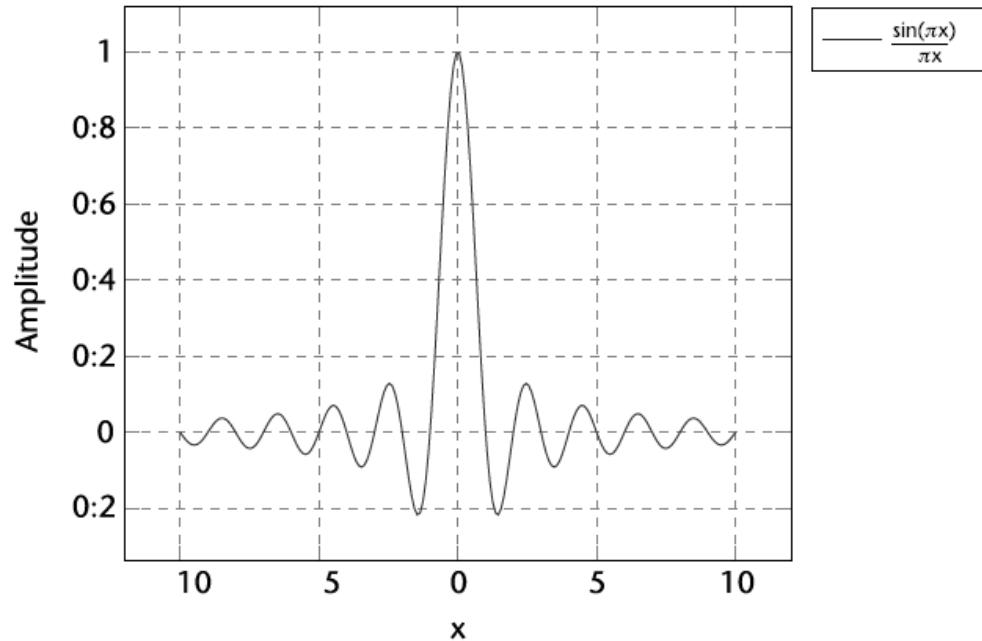
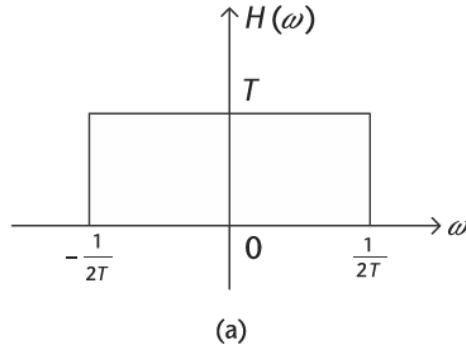
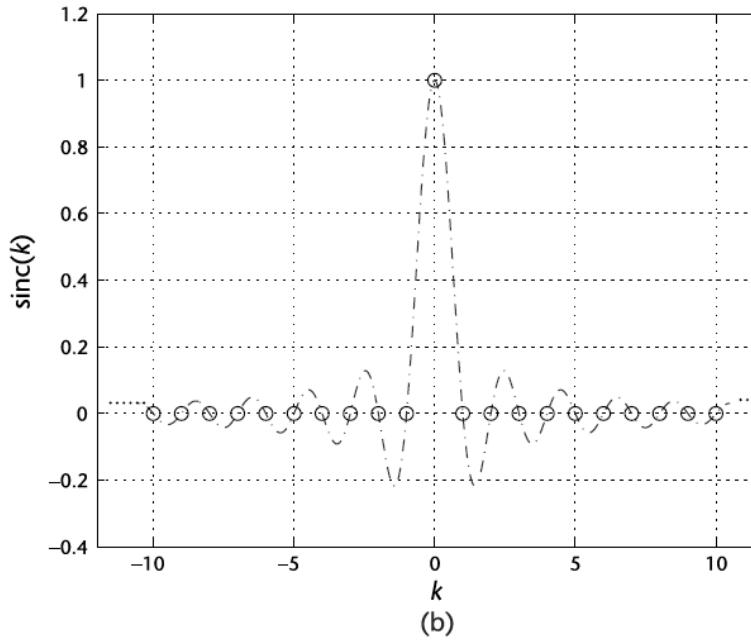


Figure 2.50 The plot of sinc function as defined in (2.79). The x -axis is x , and the y -axis is $\text{sinc}(x)$.



(a)



(b)

Figure 2.51 The sinc pulse on time domain and its Fourier transform, rectangular pulse, on frequency domain. (a) The rectangular pulse on frequency domain, defined in (2.81), and (b) the sinc pulse defined in (2.84). The x-axis is k , where $k = \frac{t}{T}$, and the y-axis is $\text{sinc}(k)$.

Extraido de: <https://www.analog.com/en/education/education-library/software-defined-radio-for-engineers.html>

rectangular signal is defined as [19]:

$$H(\omega) = \begin{cases} T & |\omega| < \frac{1}{2T} \\ 0 & \text{otherwise} \end{cases}. \quad (2.81)$$

Taking the inverse Fourier transform of the rectangular signal will yield the sinc signal as

$$h(t) = \operatorname{sinc}\left(\frac{t}{T}\right). \quad (2.82)$$

Change the variable $t = kT$ in (2.82) yields

$$h(t) = h(kT) = \operatorname{sinc}\left(\frac{kT}{T}\right) = \operatorname{sinc}(k). \quad (2.83)$$

Since k is an integer here, according to (2.80), we can continue writing (2.83) as

$$h(t) = h(kT) = \text{sinc}\left(\frac{kT}{T}\right) = \text{sinc}(k) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}. \quad (2.84)$$

Comparing (2.84) with (2.78), we can easily find that if we make $t = kT$, the sinc pulse exactly satisfies Nyquist pulse-shaping theory in Section 2.7.4. In other words, by choosing sampling time at kT (sampling frequency equals $\frac{1}{T}$), our sampling instants are located at the equally spaced zero crossings, as shown in Figure 2.51(b), so there will be no intersymbol interference.

Recall the Nyquist sampling theorem in Section 2.2.3 states that a real signal, $x(t)$, which is bandlimited to B Hz can be reconstructed without error using a minimum sampling frequency of $F_s = 2B$ Hz. In this case, the minimum sampling frequency is $F_s = \frac{1}{T}$ Hz. Therefore, the corresponding minimum bandwidth is

$$B = \frac{F_s}{2} = \frac{1}{2T}, \quad (2.85)$$

which is exactly the bandwidth of the rectangular signal defined in (2.81). Based on the discussion above, this choice of sinc pulse $h(t)$ yields the minimum bandwidth $B = B_{\min} = \frac{1}{2T}$, so it is called the Nyquist-I Pulse [20].

Sinc pulses are a very attractive option because they are wide in time and narrow in frequency, which means they have the advantages of both spectrum efficiency and no ISI. However, sinc pulses are not practical since they have ISI sensitivity due to timing errors. For instance, for large t , the sinc pulse defined in (2.82) has the following approximation:

$$h(t) \sim \frac{1}{t}, \quad (2.86)$$

so it is obvious that timing error can cause large ISI. We must also note that sinc pulse are infinite in time, making them unrealizable.

Consequentially, we need to introduce Nyquist-II pulses, which have a larger bandwidth $B > B_{\min}$, but with less ISI sensitivity. Since this type of pulse is more practical, it is much more widely used in practice.

The raised cosine pulse is one of the most important type of Nyquist-II pulses, which has the frequency transfer function defined as

$$H_{RC}(f) = \begin{cases} T & 0 \leq |f| \leq \frac{1-\beta}{2T} \\ \frac{T}{2} \left(1 + \cos \left(\frac{\pi T}{\beta} \left(|f| - \frac{1-\beta}{2T}\right)\right)\right) & \frac{1-\beta}{2T} \leq |f| \leq \frac{1+\beta}{2T} \\ 0 & |f| \geq \frac{1+\beta}{2T} \end{cases}, \quad (2.87)$$

where β is the rolloff factor, which takes value from 0 to 1, and $\frac{\beta}{2T}$ is the excess bandwidth.

The spectrum of raised cosine pulse is shown in Figure 2.52. In general, it has the bandwidth $B \geq 1/(2T)$. When $\beta = 0$, the bandwidth $B = 1/(2T)$, and there is no excess bandwidth, which is actually a rectangular pulse. On the other end, when $\beta = 1$, it reaches the maximum bandwidth $B = 1/T$.



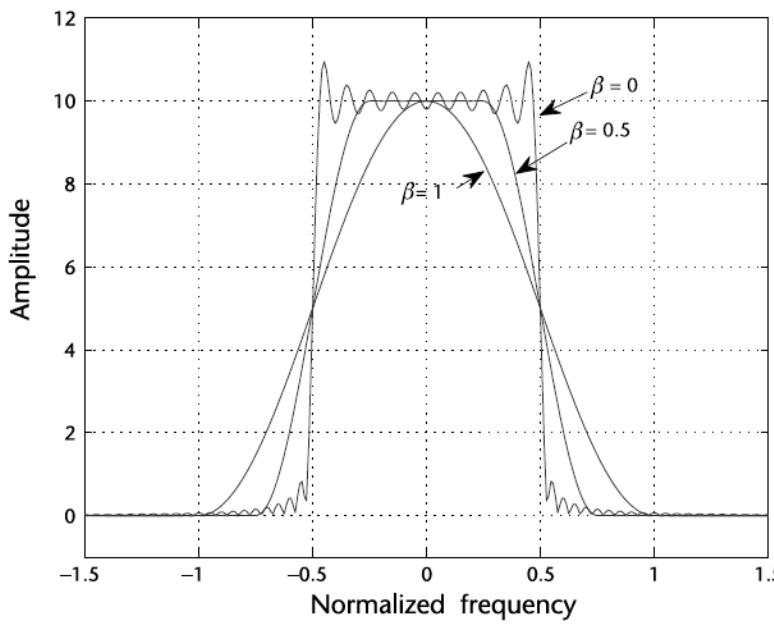


Figure 2.52 Spectrum of a raised cosine pulse defined in (2.87), which varies by the rolloff factor β . The x -axis is the normalized frequency f_0 . The actual frequency can be obtained by f_0/T .

By taking the inverse Fourier transform of $H_{RC}(f)$, we can obtain the impulse response of raised cosine pulse, defined as

$$h_{RC}(t) = \frac{\cos\left(\pi \frac{\beta t}{T}\right)}{1 - \left(2 \frac{\beta t}{T}\right)^2} \text{sinc}\left(\frac{\pi t}{T}\right). \quad (2.88)$$

Nyquist-II pulses do not have an ISI sensitivity because its peak distortion, the tail of $h_{RC}(t)$, converges quickly, which can be expressed as

$$h_{RC}(t) = \frac{\cos\left(\pi \frac{\beta t}{T}\right)}{1 - \left(2\frac{\beta t}{T}\right)^2} \operatorname{sinc}\left(\frac{\pi t}{T}\right). \quad (2.88)$$

Nyquist-II pulses do not have an ISI sensitivity because its peak distortion, the tail of $h_{RC}(t)$, converges quickly, which can be expressed as

$$D_p = \sum_{n=-\infty}^{\infty} |h_{RC}(\epsilon' + (n - k))| \sim \frac{1}{n^3}. \quad (2.89)$$

Therefore, when timing error occurs, the distortion will not accumulate to infinity in a similar fashion related to Nyquist-I pulses [20].

Actually, in many practical communications systems, *root raised cosine filters* are usually used [21]. If we consider the communication channel as an ideal channel and we assume that the transmit filter and the receive filter are identical, we can use root raised cosine filters for both of them, and their net response must equal to $H_{RC}(f)$ defined in (2.87). Since the impulse response of the equivalent channel can be expressed as

$$h(t) = h_T(t) * h_C(t) * h_R(t), \quad (2.90)$$

where $h_C(t)$ is the impulse response of the communication channel, and $h_T(t)$ and $h_R(t)$ are the impulse responses of the transmit filter and the receive filter, it means on frequency domain, we have

$$H_{RC}(f) = H_T(f)H_R(f). \quad (2.91)$$

Extraido de: <https://www.analog.com/en/education/education-library/software-defined-radio-for-engineers.html>



Actually, in many practical communications systems, *root raised cosine filters* are usually used [21]. If we consider the communication channel as an ideal channel and we assume that the transmit filter and the receive filter are identical, we can use root raised cosine filters for both of them, and their net response must equal to $H_{RC}(f)$ defined in (2.87). Since the impulse response of the equivalent channel can be expressed as

$$h(t) = h_T(t) * h_C(t) * h_R(t), \quad (2.90)$$

where $h_C(t)$ is the impulse response of the communication channel, and $h_T(t)$ and $h_R(t)$ are the impulse responses of the transmit filter and the receive filter, it means on frequency domain, we have

$$H_{RC}(f) = H_T(f)H_R(f). \quad (2.91)$$

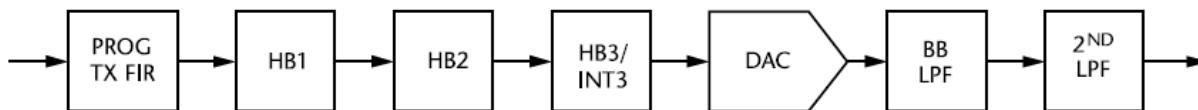


Figure 2.53 Internal AD9361 Tx signal path.

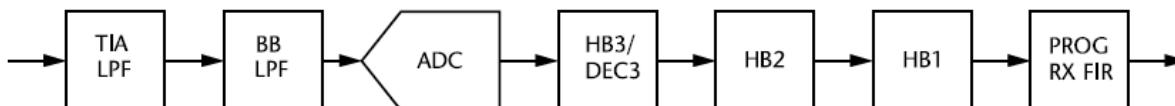


Figure 2.54 Internal AD9361 Rx signal path.

Therefore, the frequency response of root raised cosine filter must satisfy

$$|H_T(f)| = |H_R(f)| = \sqrt{|H_{RC}(f)|}. \quad (2.92)$$

Extraído de: <https://www.analog.com/en/education/education-library/software-defined-radio-for-engineers.html>