Signals and Systems

Chapter 2

Introduction

Everyone involved in SDR development needs to not only have a solid background in signals and systems, but also RF and analog baseband processing.

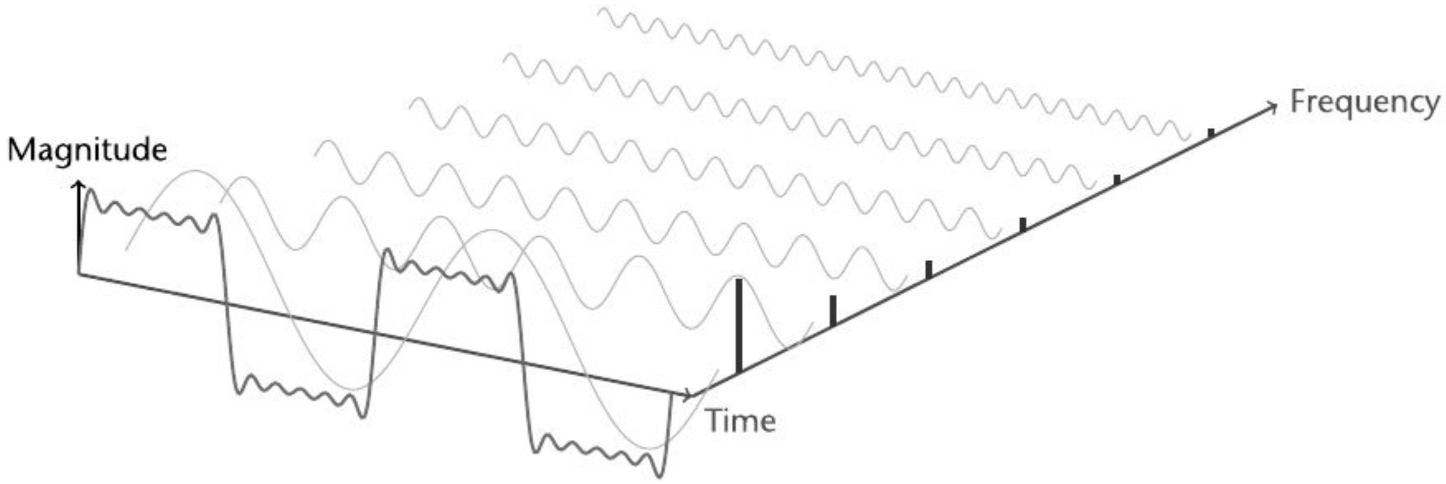
2.1 Time and Frequency Domains

The Fourier transform is just a different way to describe a signal. For example, investigating the Gibbs phenomenon, which states if you add sine waves at specific frequency/phase/amplitude combinations you can approximate a square wave, as follows:

$$x(t) = \sin(t) + \frac{\sin(3t)}{3} + \frac{\sin(5t)}{5} + \ldots = \sum_{n=1}^{n=\infty} \frac{\sin(n \times t)}{n}; \quad n = odd$$

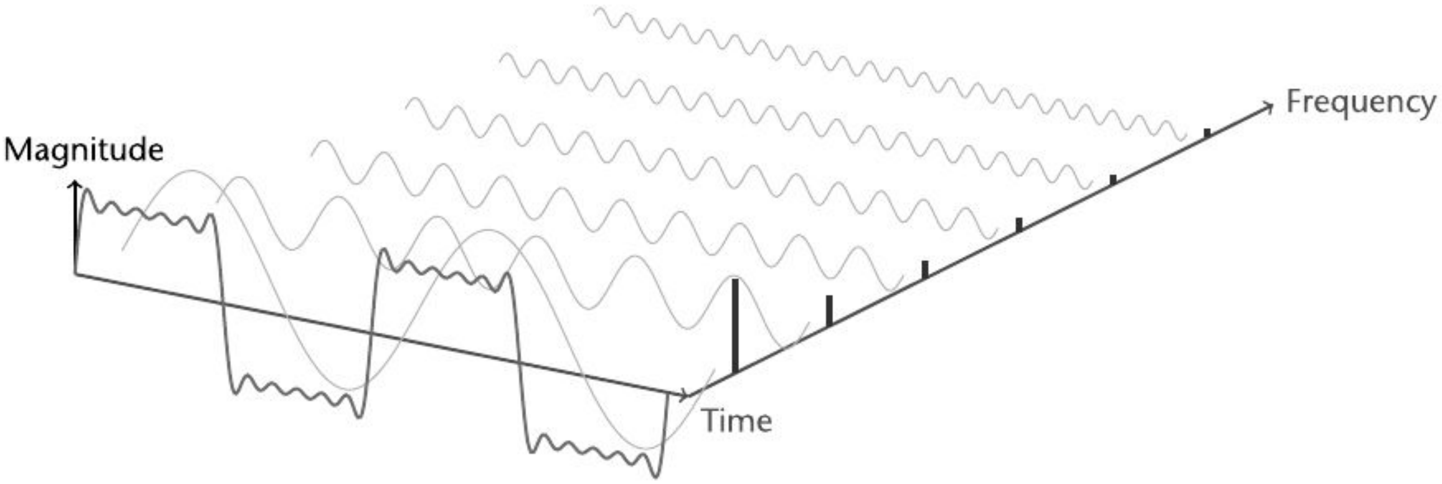
2.1 Time and Frequency Domains

When we look at the signal across the time axis that is perpendicular to the frequency axis, we observe the time domain. We cannot see the frequency of the sine waves easily since we are perpendicular to the frequency axis.



2.1 Time and Frequency Domains

We can easily make out the signal magnitude and frequency, but have lost that time aspect. Both views represents the same signal such that, they are just being observed things from different domains via transforms.



2.1 Time and Frequency Domains

Let’s look at some MATLAB code on *gibss.m* file.

2.1.1 Fourier Transform

The Fourier transform of x(t) is defined as:

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$$

The inverse Fourier transform is defined as:

$$x(t) = \int_{-\infty}^{\infty} X(\omega) e^{j2\pi\omega t} d\omega$$

There are several commonly used properties of Fourier transform that are useful when studying SDR Fourier domain, which have been summarized in Table 2.1 for your reference.

2.1.1 Fourier Transform

Table 2.1 Fourier Transform Properties*

Property	*Time Signal*	*Fourier Transform Signal*
Definition	$x(t)$	$\int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$
Inversion formula	$\int_{-\infty}^{\infty} X(\omega)e^{j2\pi\omega t}d\omega$	$X(\omega)$
Linearity	$\sum_{n=1}^{N} a_n x_n(t)$	$\sum_{n=1}^{N} a_n X_n(\omega)$
Symmetry	$x(-t)$	$X(-\omega)$
Time shift	$x(t-t_0)$	$X(\omega)e^{-j\omega t_0}$
Frequency shift	$x(t)e^{j\omega_0 t}$	$X(\omega-\omega_0)$
Scaling	$x(\alpha t)$	$\frac{1}{\lvert\alpha\rvert}X(\frac{\omega}{\alpha})$
Derivative	$\frac{d^n}{dt^n}x(t)$	$(j\omega)^n X(\omega)$
Integration	$\int_{-\infty}^{\infty} x(\tau)d\tau$	$\frac{X(\omega)}{j\omega}+\pi X(0)\delta(\omega)$
Time convolution	$x(t)*h(t)$	$X(\omega)H(\omega)$
Frequency convolution	$x(t)h(t)$	$\frac{1}{2\pi}X(\omega)*H(\omega)$

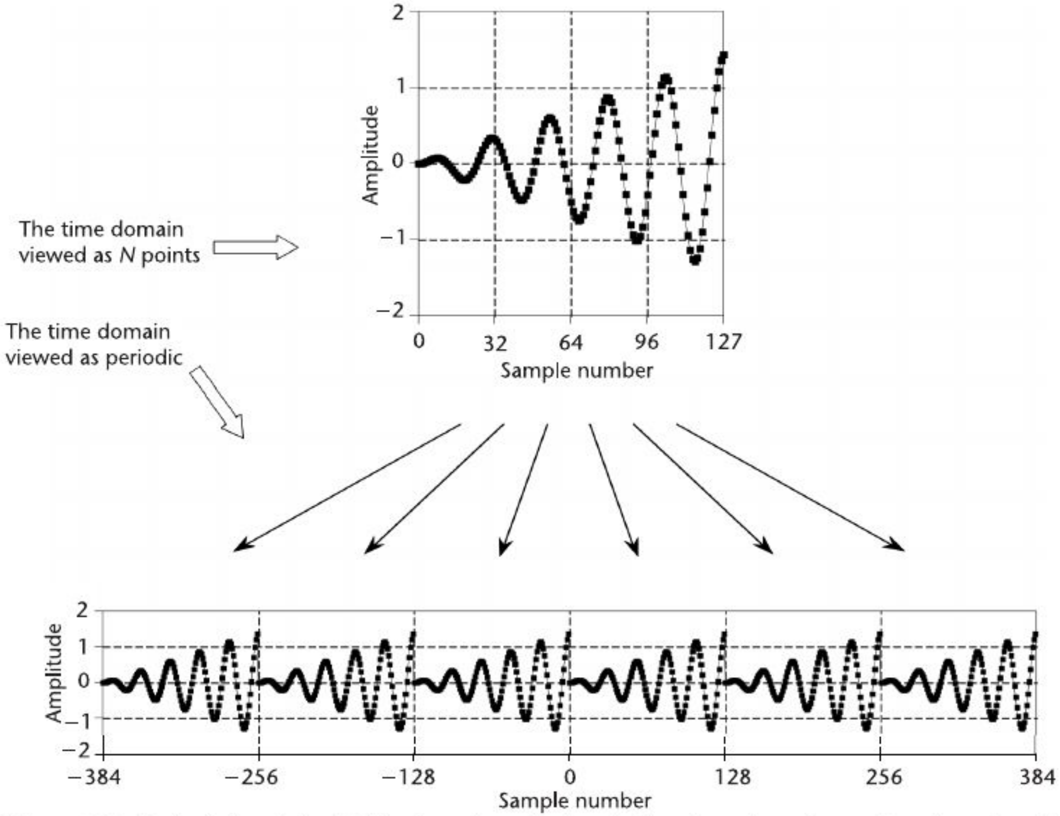
* based on [2]. Suppose the time signal is $x(t)$, and its Fourier transform signal is $X(\omega)$

2.1.2 Periodic Nature of the DFT

The DFT views both the time domain and the frequency domain signals as periodic. If you want to use the DFT (and its fast implementation, the FFT), you must conform with the DFT’s periodic view of the world.

This is the reason that window functions need to be preapplied to signal captures before applying an FFT function, which is multiplied by the signal and removes the discontinuities by forcing them to zero.

2.1.2 Periodic Nature of the DFT



2.1.3 Fast Fourier Transform

- The spectral output of the FFT is a series of M/2 points in the frequency domain.
- The spacing between the points is fs/M (resolution of FFT).
- Total frequency range covered is DC to fs/2.

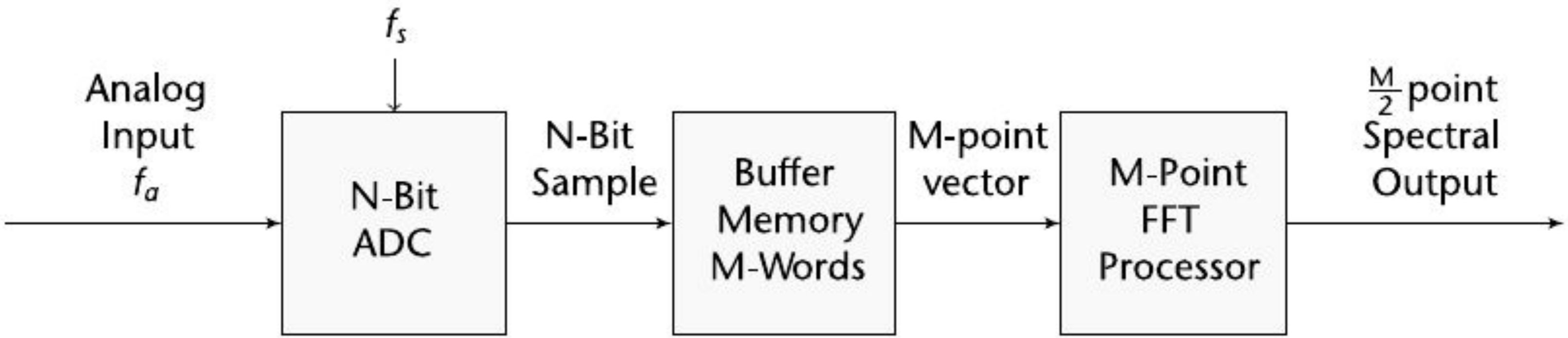


Figure 2.3 Generalized test set up for FFT analysis of ADC output [6].

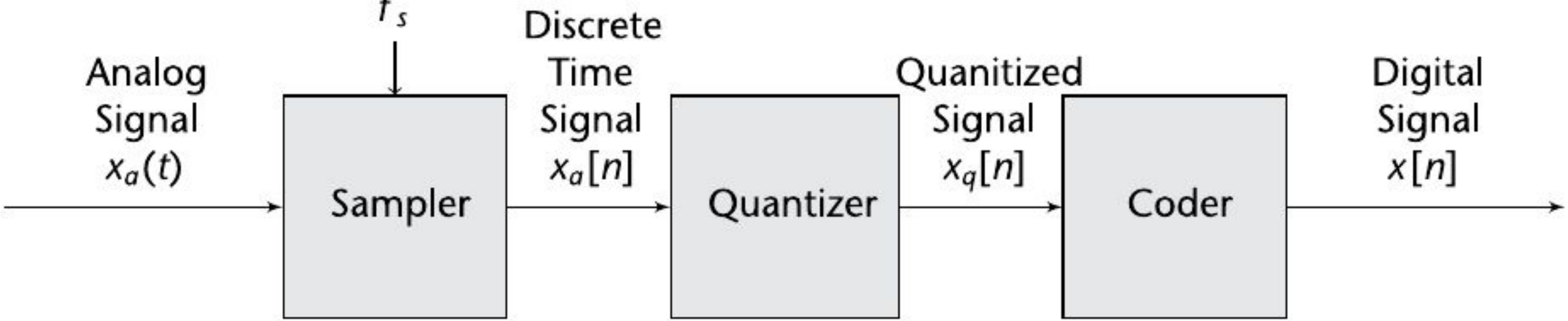
2.2 Sampling Theory

A continuous-time analog signal can be converted to a discrete-time digital signal using sampling and quantization.

Sampling is the conversion of a continuous-time signal into a discrete-time signal obtained by taking the samples of the continuous-time signal at discrete-time instants.

Quantization is the process of converting the sample amplitude into a digital format.

2.2 Sampling Theory



2.2 Sampling Theory

A discrete-time signal can also be converted to a continuous-time signal using reconstruction.

Sometimes, the reconstructed signal is not the same as the original signal. If the sampling satisfies the Nyquist sampling theorem, the signal can be reconstructed without losing information.

2.2.1 Uniform Sampling

Specify the sampling interval as a constant number.

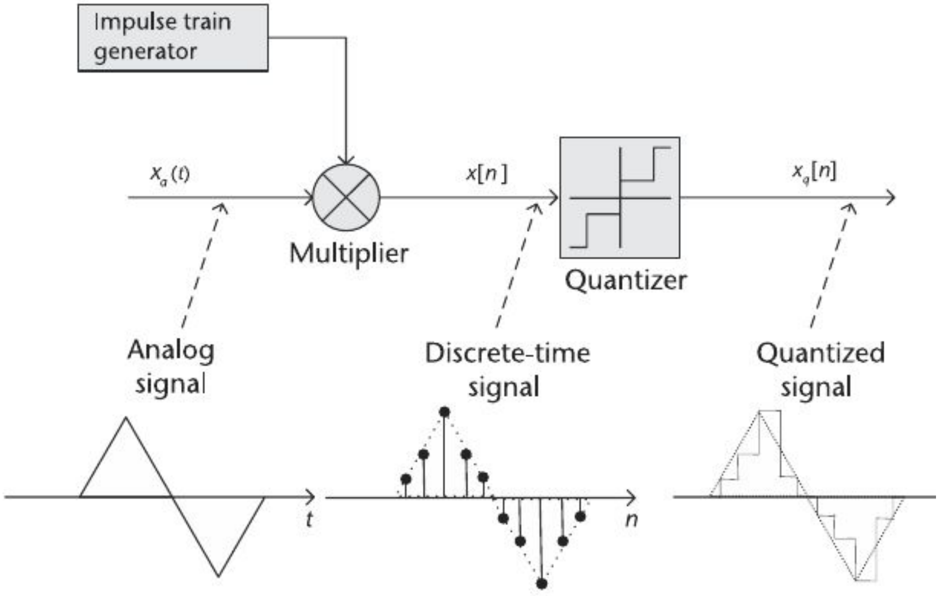
Using this method, we are taking samples of the continuous-time signal every Ts seconds, which can be defined as

$$x[n] = x(nT_s), \quad -\infty < n < \infty$$

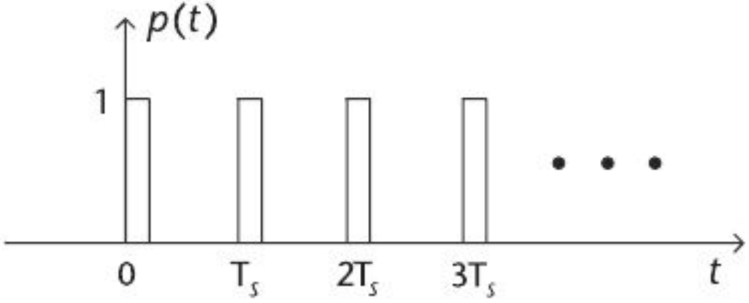
An equivalent model for the uniform sampling operation can be described by using an impulse train p(t)

$$x_s(t) = x(t)p(t)$$

2.2.1 Uniform Sampling



(a)



(b)

2.2.2 Frequency Domain Representation of Uniform Sampling

If we can express p(t) as

$$p(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT_s), \quad k = 0, 1, 2, ...,$$

Defining the uniform sampling

$$x_s(t) = x(t)p(t) = x(t) \sum_{k=-\infty}^{\infty} \delta(t - kT_s).$$

2.2.2 Frequency Domain Representation of Uniform Sampling

Multiplication of x(t) and p(t) will yield the convolution of X(ω) and P(ω):

$$X_s(\omega) = \frac{1}{2\pi} X(\omega) * P(\omega)$$

The Fourier transform of p(t) is

$$P(\omega) = \frac{\sqrt{2\pi}}{T_s} \sum_{k=-\infty}^{\infty} \delta(\omega - k\frac{2\pi}{T_s}) = \frac{\sqrt{2\pi}}{T_s} \sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_s)$$

*Errata: I think it should be 2π, see https://class.ece.uw.edu/235dl/EE235/Project/lesson19/lesson19.html

2.2.2 Frequency Domain Representation of Uniform Sampling

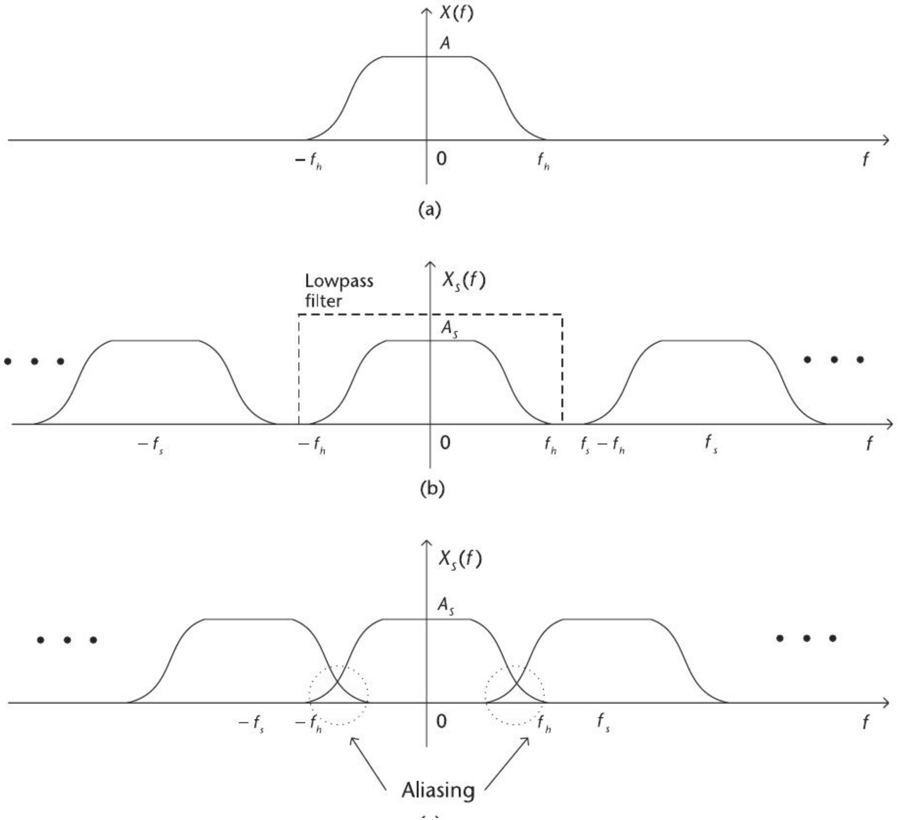
Performing convolution with a collection of delta function pulses at the pulse location, we get

$$X_s(\omega) = \frac{1}{\sqrt{2\pi}\,T_s} \sum_{k=-\infty}^{\infty} X(\omega - k\omega_s).$$

2.2.3 Nyquist Sampling Theorem

The Nyquist sampling theorem states that a real signal, $x(t)$, which is bandlimited to B Hz can be reconstructed without error from samples taken uniformly at a rate $R > 2B$ samples per second.

2.2.3 Nyquist Sampling Theorem



2.2.4 Nyquist Zones

The Nyquist bandwidth itself is defined to be the frequency spectrum from DC to fs/2. However, the frequency spectrum is divided into an infinite number of Nyquist zones, each having a width equal to fs/2 as shown in Figure 2.9.

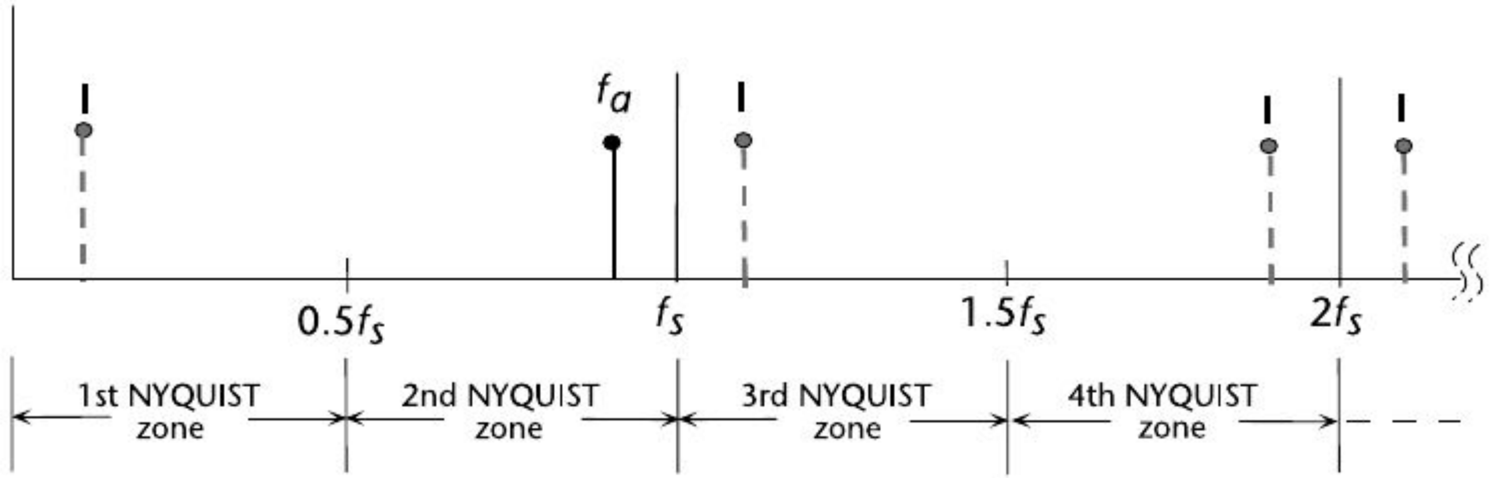


Figure 2.9 Analog signal f_a sampled at f_s has images (aliases) at $\pm kF_s \pm F_a, k = 1, 2, 3, ...$

2.2.4 Nyquist Zones

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.

2.2.4 Nyquist Zones

Let’s look at some MATLAB code on *nyquist.m* file.

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.

2.2.5 Sample Rate Conversion

Decimation: The process of lowering the sampling rate is called decimation, which is achieved by ignoring all but every Dth sample.

In time domain, it can be defined as

$$y[n] = x[nD], \quad D = 1, 2, 3, \ldots$$

The sampling rates of the original signal and the decimated can be expressed as:

$$F_y = \frac{F_x}{D}$$

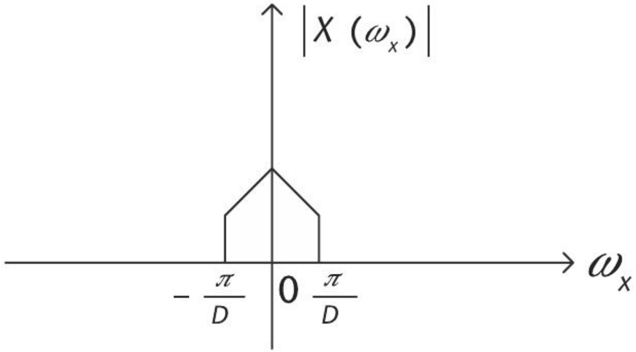
Where Fx is the sampling rates of the original signal, and Fy is the sampling rates of the decimated signal.

2.2.5 Sample Rate Conversion

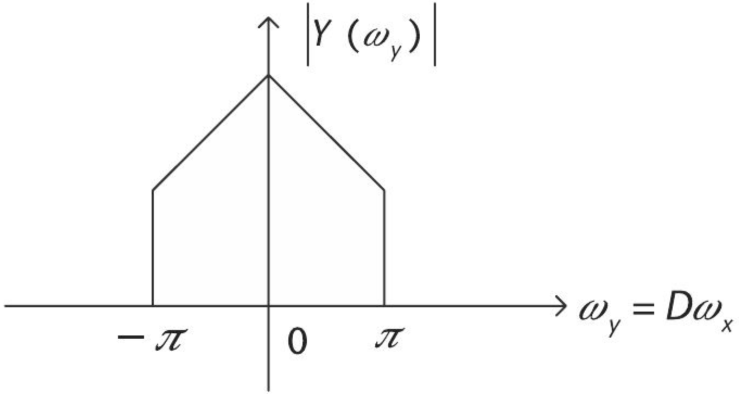
It can be proved that the following relationship is also valid

$$\omega_y = D\omega_x$$

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



(a)



(b)

2.2.5 Sample Rate Conversion

In reality, decimation is usually a two-step process, consisting of a lowpass antialiasing filter and a downsampler, as shown in Figure 2.10.

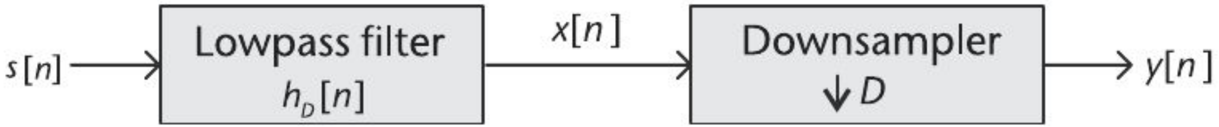


Figure 2.10 The structure of decimation, consisting of a lowpass antialiasing filter and a downsampler.

2.2.5 Sample Rate Conversion

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

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

where S(ω) is the spectrum of the input signal s[n], and H_D (ω) is the frequency response of the lowpass filter h_D [n].

2.2.5 Sample Rate Conversion

With a properly designed filter HD (ω), the aliasing is eliminated, and consequently, all but the first k = 0 term vanish

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

2.2.5 Sample Rate Conversion

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

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

The sampling rates of the original signal and the interpolated signal can be expressed as:

$$F_y = IF_x$$

Where Fx is the sampling rates of the original signal, and Fy is the sampling rates of the interpolated signal.

2.2.5 Sample Rate Conversion

It can be proved that the following relationship is also valid

$$\omega_y = \frac{\omega_x}{I}$$

Which means that the frequency range of ωx is compressed into the corresponding frequency range of ωy 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 I/π.

$\omega_y = \frac{\pi}{I}$ should be rejected by passing it through a lowpass filter with the following frequency response:

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

2.2.5 Sample Rate Conversion

In reality, interpolation is also a two-step process, consisting of an upsampler and a lowpass filter

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

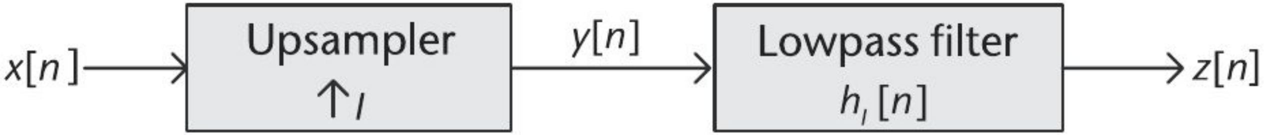
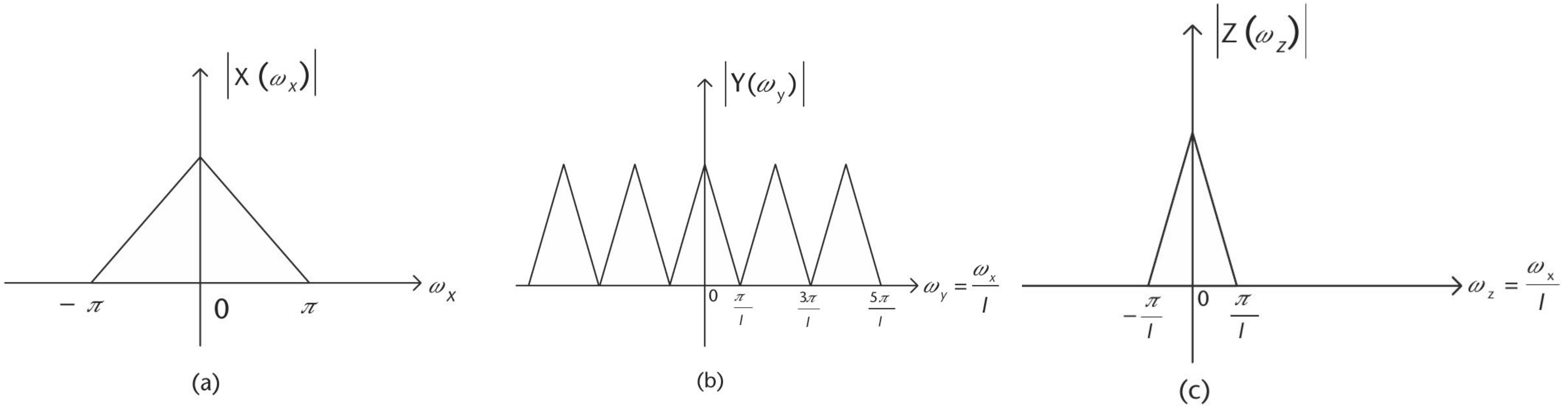


Figure 2.12 The structure of interpolation, consisting of an upsampler and a lowpass filter.

2.2.5 Sample Rate Conversion



2.2.5 Sample Rate Conversion

Let’s look at some MATLAB code on *updownsample.m* file.

2.3 Signal Representation

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

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 and these signals are used to drive the mixers, which simply multiply the LO signals with the input.

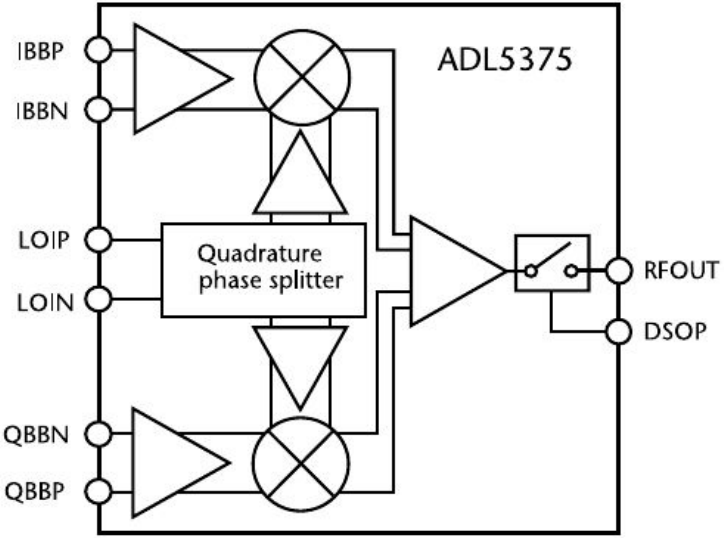


Figure 2.18 ADL5375 broadband quadrature modular with range from 400 MHz to 6 GHz.

2.3.1 Frequency Conversion

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 ωc , 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.

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

The transmitted signal will contain both components but will appear as a single sinusoid.

2.3.1 Frequency Conversion

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

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

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

2.3.2 Imaginary Signals

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.

We start from the Euler relation of

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

Redefining I(t) and Q(t) as real and imaginary, we arrive at

$$y(t) = I(t) + jQ(t)$$

We can easily make a frequency shift as follows

$$y(t)e^{j\omega_c t} = \big(I(t)cos(\omega_c t) - Q(t)sin(\omega_c t)\big) + j\big(Q(t)cos(\omega_c t) + I(t)sin(\omega_c t)\big)$$

2.3.2 Imaginary Signals

For example, if a signal y(t) is a CW tone at frequency ω_a

$$y(t) = cos(\omega_a t) + jsin(\omega_a t)$$

We can apply a frequency shift

$$y(t)e^{j\omega_c t} = cos((\omega_c + \omega_a)t) + jsin((\omega_c + \omega_a)t)$$

2.3.2 Imaginary Signals

If we consider the mixer itself as in Figure 2.18, the IQ mixer will transmit signals 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

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

$$\mathbf{A}\sin(\omega t + \phi) = (\mathbf{Acos}\phi)\sin(\omega t) + (\mathbf{Asin}\phi)\cos(\omega t)$$

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.

2.3.2 Imaginary Signals

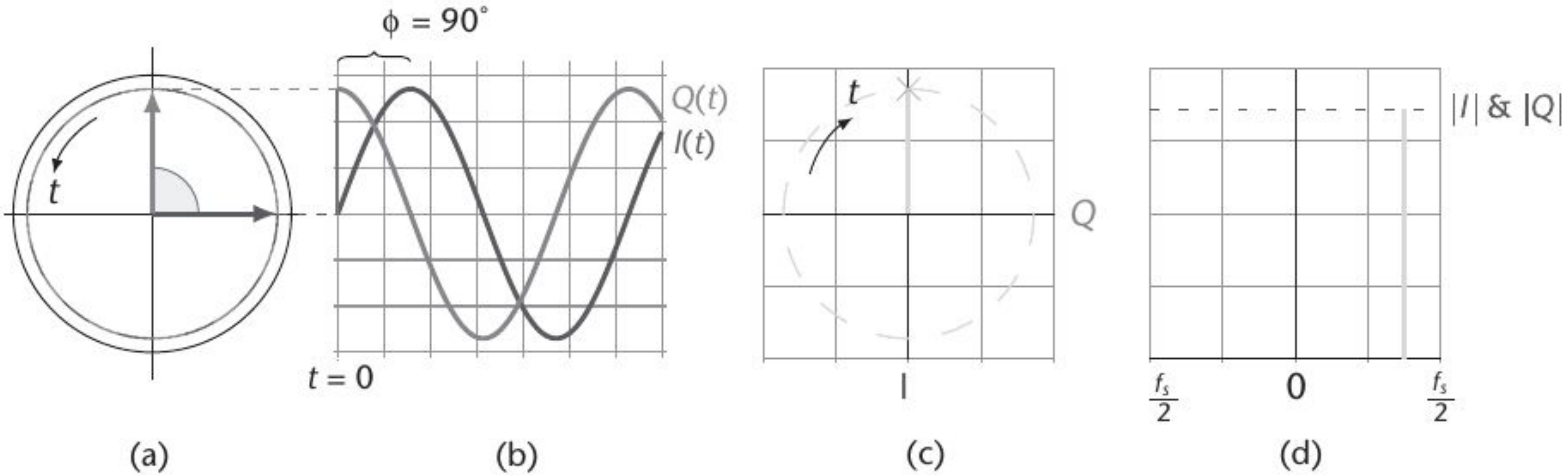


Figure 2.19 Same continuous wave signal, plotted in multiple domains. (a) Phasor *rad*(t), (b) time $x(t) \rightarrow$, (c) Cartesian $(I, Q)(t)$, and (d) frequency $X(\omega)$.

2.4 Signal Metrics and Visualization

- Communications systems are complex to evaluate given their integrated nature and depth of transmit and receive chains.
- Just as we have system-level specifications for the entire system, we have specifications and measurement techniques for other subsystems and SDR building blocks. In this way, we know we are not over- or underdesigning specific components.
- System-level specifications are met by ensuring each block in the system will allow those specifications to be met.
- A system is only as good as the weakest link.
- There are numerous techniques and tools to measure almost everything. Studying communications is not for those who do not want to be rigorous.

2.4.1 SINAD, ENOB, SNR, THD, THD + N, and SFDR

Using and understanding these measurements will help you analyze your designs and make sure you are designing something to be the most robust.

Table 2.2 Six Popular Specifications

Property	*Definition*	*MATLAB Function*
SFDR	Spurious free dynamic range	`sfdr`
SINAD	Signal-to-noise-and-distortion ratio	`sinad`
ENOB	Effective number of bits	
SNR	Signal-to-noise ratio	`snr`
THD	Total harmonic distortion	`thd`
THD + N	Total harmonic distortion plus noise	

2.4.1 SINAD, ENOB, SNR, THD, THD + N, and SFDR

Spurious Free Dynamic Range (SFDR) is the ratio of the root mean squared (RMS) value of the signal to the rms value of the worst spurious signal regardless of where it falls in the frequency spectrum.

SFDR is generally plotted as a function of signal amplitude and may be expressed relative to the signal amplitude (dBc) or the ADC full-scale (dBFS) as shown in Figure 2.21.

SFDR considers all sources of distortion regardless of their origin, and is a useful tool in evaluating various communication systems.

2.4.1 SINAD, ENOB, SNR, THD, THD + N, and SFDR

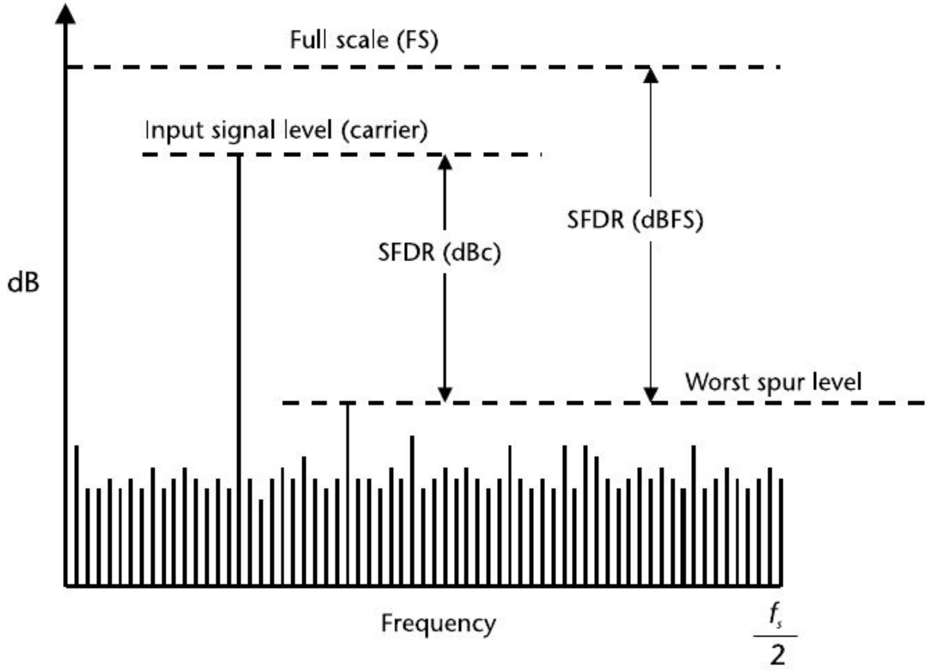


Figure 2.21 Spurious free dynamic range (SFDR) for BW DC to $f_s/2$.

2.4.1 SINAD, ENOB, SNR, THD, THD + N, and SFDR

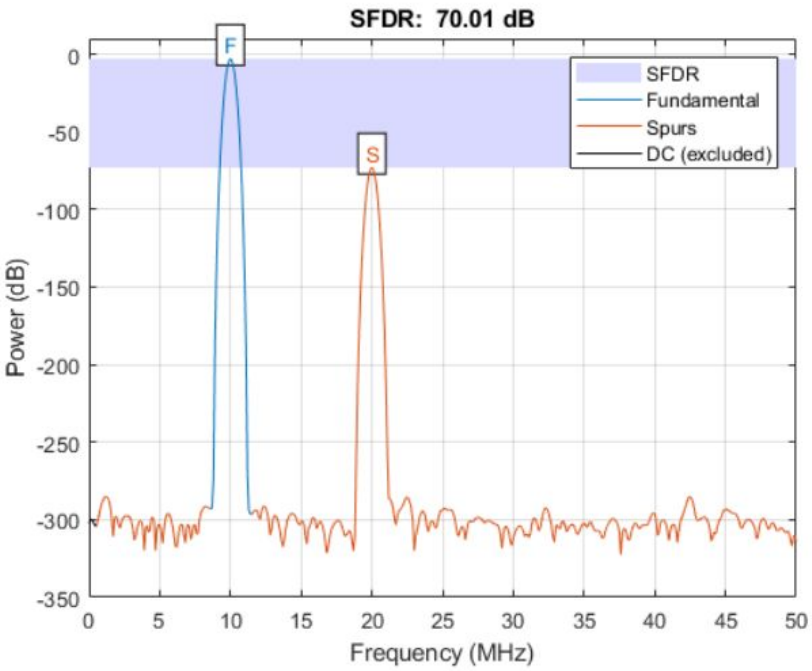
```
deltat = 1e-8;
fs = 1/deltat;
t = 0:deltat:1e-5-deltat;
x = cos(2*pi*10e6*t)+3.16e-4*cos(2*pi*20e6*t);
r = sfdr(x,fs)
```

```
r = 70.0063
```

Display the spectrum of the signal. Annotate the fundamental, the DC value, the spur, and the SFDR.

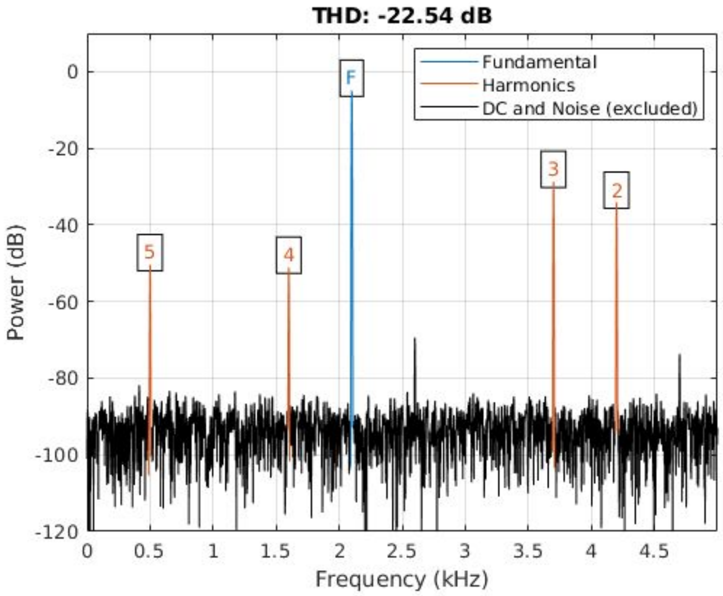
```
sfdr(x,fs);
```

MATLAB Code: openExample('signal/SFDROfSinusoidExample')



2.4.1 SINAD, ENOB, SNR, THD, THD + N, and SFDR

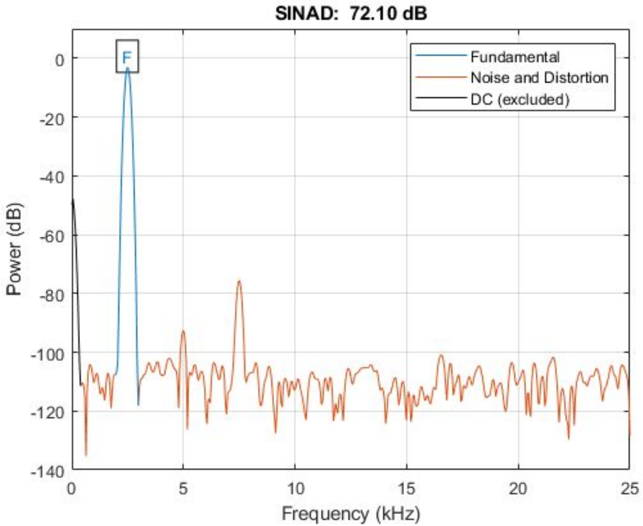
Total Harmonic Distortion (THD) is the ratio of the rms value of the fundamental signal to the mean value of the root-sum-square of its harmonics (generally, only the first five harmonics are significant).



MATLAB Code: openExample('signal/THDWithAndWithoutAliasedHarmonicsExample')

2.4.1 SINAD, ENOB, SNR, THD, THD + N, and SFDR

Signal-to-noise-and-distortion (SINAD, or **S/(N + D)** 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. SINAD is a good indication of the overall dynamic performance of an analog system because it includes all components that make up noise and distortion.



MATLAB Code: openExample('signal/SINADOfAmplifiedSignalExample')

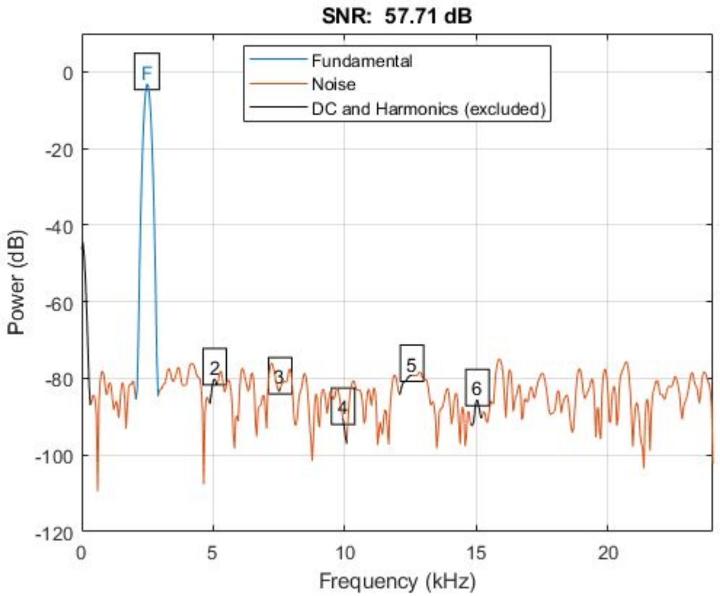
2.4.1 SINAD, ENOB, SNR, THD, THD + N, and SFDR

Total Harmonic Distortion Plus Noise (THD + N) is the ratio of the rms value of the fundamental signal to the mean value of the root-sum-square of its harmonics plus all noise components (excluding DC).

For a given input frequency and amplitude, SINAD is equal to THD + N, provided the bandwidth for the noise measurement is the same for both (the Nyquist bandwidth).

2.4.1 SINAD, ENOB, SNR, THD, THD + N, and SFDR

Signal-to-Noise Ratio (**SNR**, or sometimes called **SNR-without-harmonics**) is calculated from the FFT data the same as SINAD, except that the signal harmonics are excluded from the calculation, leaving only the noise terms.



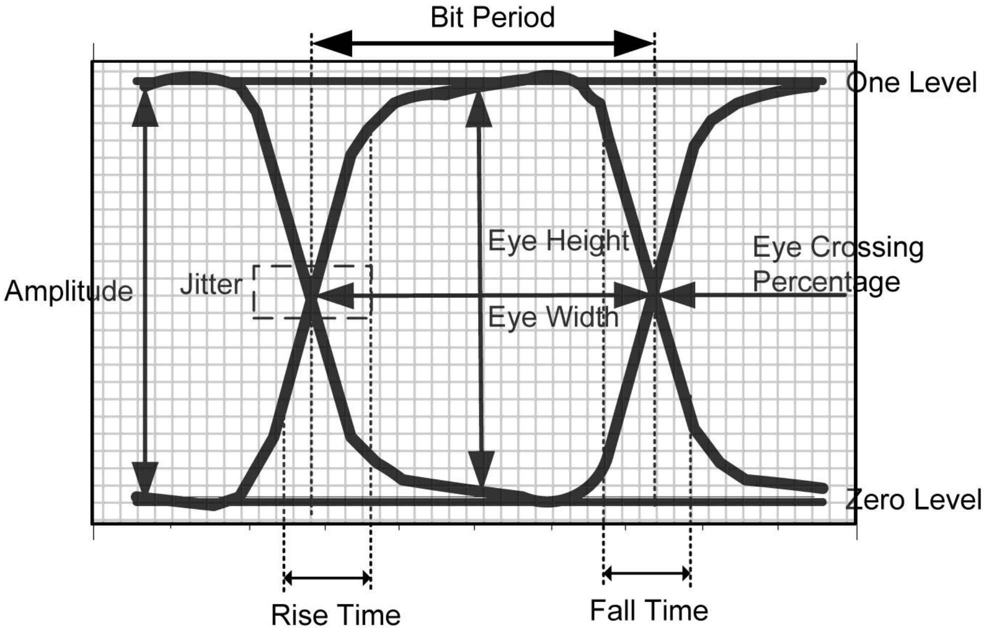
2.4.1 SINAD, ENOB, SNR, THD, THD + N, and SFDR

Effective Number of Bits (ENOB) is a measure of the dynamic range of an analog-to-digital converter (ADC), digital-to-analog converter, or their associated circuitry. The resolution of an ADC is specified by the number of bits used to represent the analog value. Ideally, a 12-bit ADC will have an effective number of bits of almost 12. However, real signals have noise, and real circuits are imperfect and introduce additional noise and distortion. Those imperfections reduce the number of bits of accuracy in the ADC. The ENOB describes the effective resolution of the system in bits. An ADC may have 12-bit resolution, but the effective number of bits when used in a system may be 9.5.

$$ENOB = \frac{SNR - 1.76dB}{6.02dB}$$

2.4.2 Eye Diagram

In telecommunication, an eye diagram, also known as an eye pattern, is an time domain display in which a digital data signal from a receiver is repetitively sampled and applied to the vertical input, while the data rate is used to trigger the horizontal sweep.



2.4.2 Eye Diagram

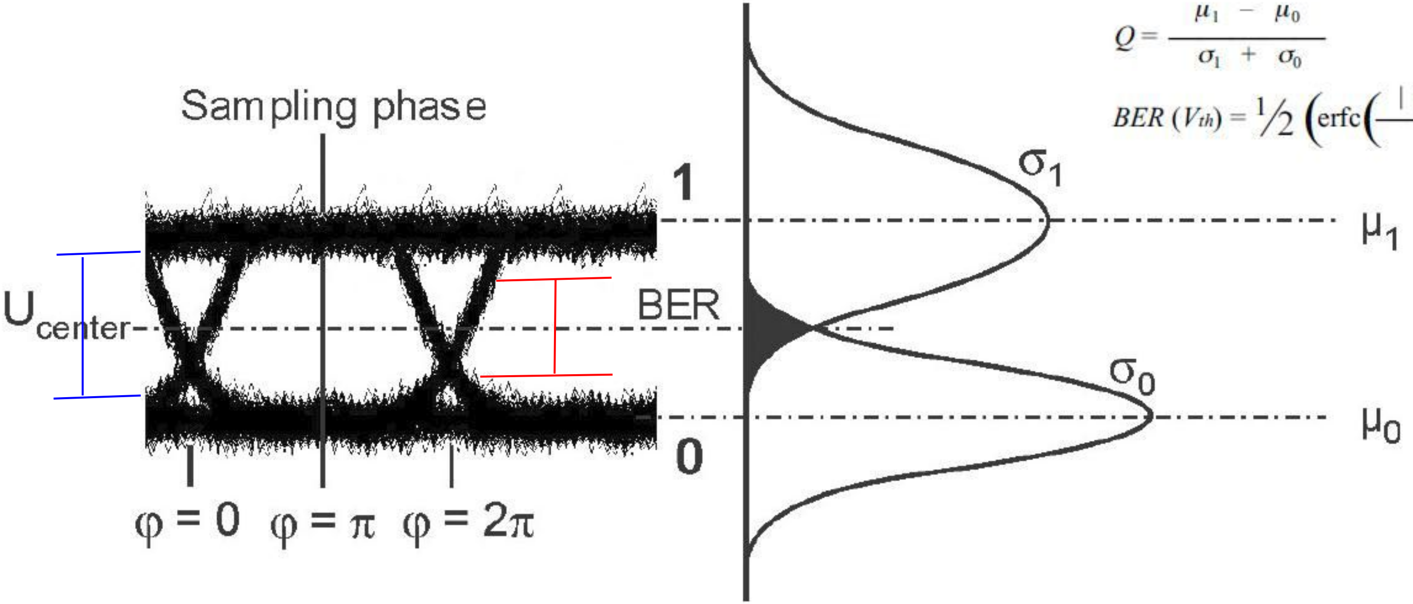
Several system performance measures can be derived by analyzing the display, especially the extent of the intersymbol-interference (ISI). As the eye closes, the ISI increases; as the eye opens, the ISI decreases.

The two key measurements are the **vertical opening**, which is the **distance between BER threshold points**, and the **eye height**, which is the **minimum distance between eye levels**. Larger vertical and horizontal openings in the eye are always better.

2.4.2 Eye Diagram

$$Q = \frac{\mu_1 - \mu_0}{\sigma_1 + \sigma_0}$$

$$BER\,(V_{th}) = \tfrac{1}{2}\left(\mathrm{erfc}\left(\frac{|\mu_1 - V_{th}|}{\sigma_1}\right) + erfc\left(\frac{|V_{th} - \mu_0|}{\sigma_0}\right)\right)$$



As illustrated above, the width of the curves represents the noise in the signal. The lower the noise, the narrower the distribution curve and therefore the smaller the overlap. Indeed, the overlap area on the distribution curves is what determines the BER, for that is where the receiver has a greater chance of interpreting a ‘1’ as a ‘0’ and vice versa - misinterpretations that produce bit errors.

2.4.2 Eye Diagram

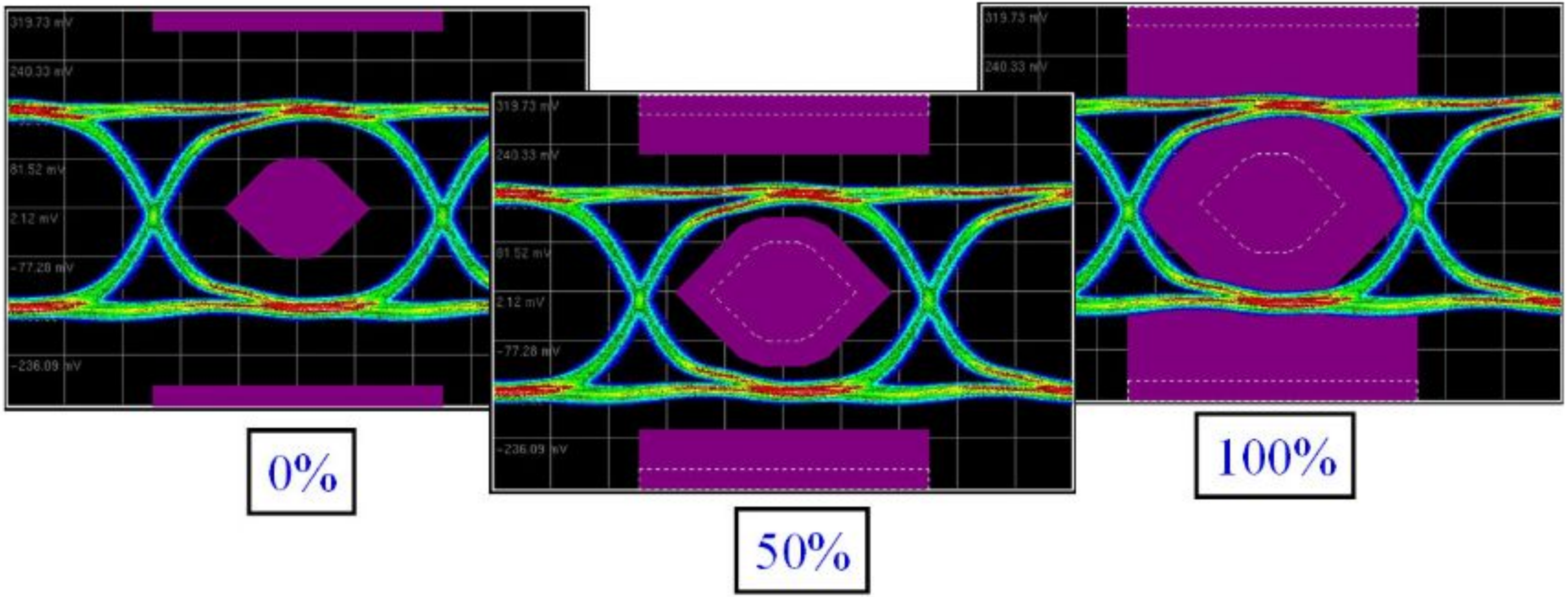


Figure 31. A typical mask margin test is shown for 0, 50, and 100 percent. The instrument automatically and linearly increases the mask line (shown by the dashed white line) until a failure is caused in the eye pattern measurement. The failed mask margin value represents with how much margin the current eye pattern measurement passes the compliance mask. This is a useful tool for monitoring the quality of the production process.

2.4.2 Eye Diagram

Let’s look at some MATLAB code on *eye_example.m* file and *eyeDiagram.m

2.5 Receive Techniques for SDR

Signals physically recovered by a SDR start out as a time-varying electric field, which induces a current in the receiving antenna and resulting in a detectable voltage at the receiver.

Transmission by the SDR, on the other hand, are time-varying voltages being applied to an antenna, which causes movement of electrons as an outwardly radiating electric field.

2.5.1 Nyquist Zones

A signal must be sampled at a rate equal to or greater than twice its bandwidth in order to preserve all the signal information.

There is no mention of the absolute location of the band of sampled signals within the frequency spectrum relative to the sampling frequency.

The only constraint is that the band of sampled signals be restricted to a single Nyquist zone.

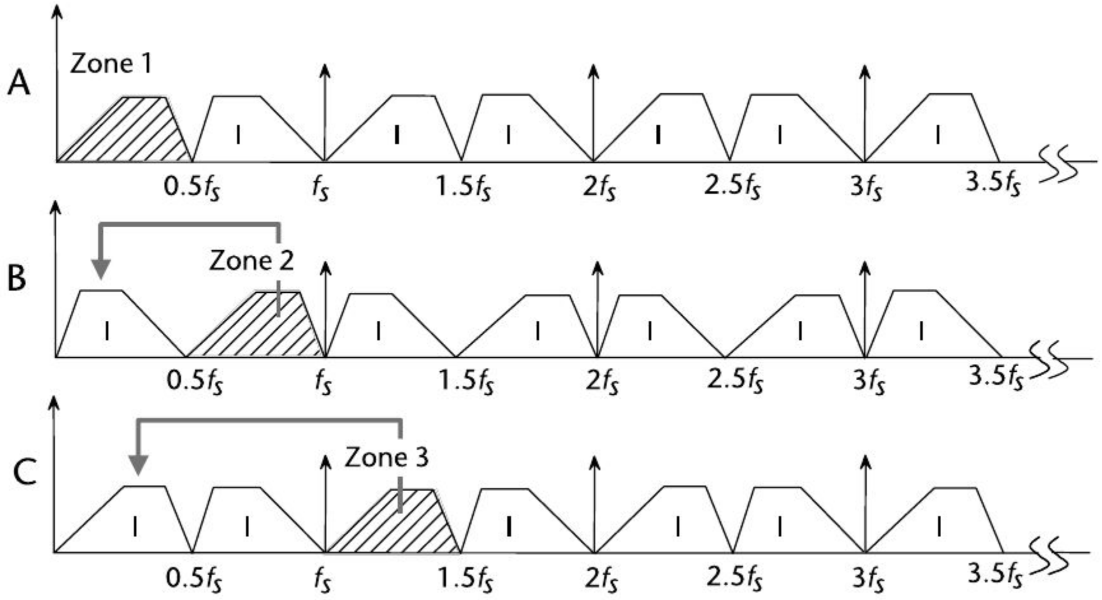


Figure 2.25 Nyquist region folding.

2.5.1 Nyquist Zones

The primary function of the antialiasing filter is to ensure the use of only one Nyquist zone.

Sampling signals above the first Nyquist zone has become popular in communications because the process is equivalent to analog demodulation.

It is becoming common practice to sample IF signals directly and then use digital techniques to process the signal, thereby eliminating the need for an IF demodulator and filters.

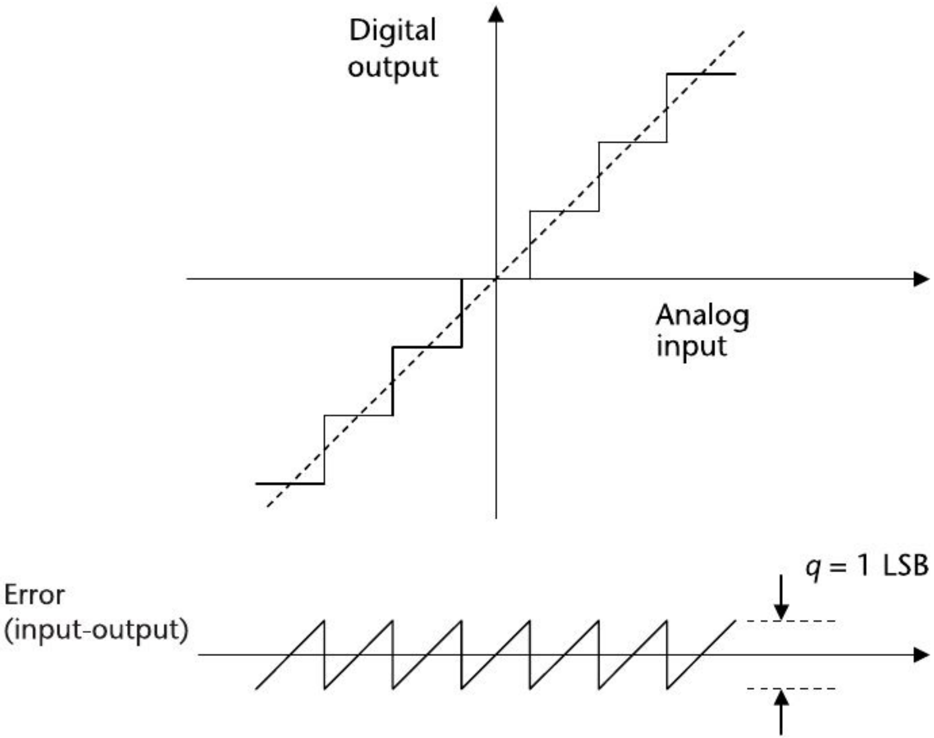
As the IF frequencies become higher, the dynamic performance requirements (bandwidth, linearity, distortion, etc.) on the ADC become more critical as performance must be adequate at the second or third Nyquist zone, rather than only baseband. This presents a problem for many ADCs designed to process signals in the first Nyquist zone.

For modulation schemes that pass information at DC, this can be very difficult to work around without using an undersampling technique.

2.5.2 Fixed Point Quantization

The maximum error an ideal converter makes when digitizing a signal is ±1/2 LSB, directly in between two digital values.

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



2.5.2 Fixed Point Quantization

The mean-square value of e(t) can be written:

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

The root mean squared (RMS) noise quantization error, is approximately Gaussian and spread more or less uniformly over the Nyquist bandwidth of DC to fs/2.

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

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

2.5.2 Fixed Point Quantization

The RMS value of the input signal defined as

$$\sqrt{\overline{v}(t)^2} = \frac{q2^N}{2\sqrt{2}}$$

The RMS signal-to-noise ratio for an ideal N-bit converter is

$$SNR = 20\,log_{10}\left(\frac{RMS\ of\ full\ scale\ input}{RMS\ of\ quatization\ noise}\right) = 20\,log_{10}\left(\frac{\frac{q2^N}{2\sqrt{2}}}{\frac{q}{\sqrt{12}}}\right)$$

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

(This is for over DC to bandwidth fs/2.)

2.5.2 Fixed Point Quantization

If digital filtering is used to filter out noise components outside BW, then a correction factor (called process gain) must be included in the equation to account for the resulting increase in SNR.

Be cautious because we assumed that the quantization noise is uncorrelated to the input signal. Under certain conditions where the sampling clock and the signal are harmonically related, the quantization noise becomes correlated and the energy is concentrated at the harmonics of the signal.

In a practical ADC application, the quantization error generally appears as random noise because of the random nature of the wideband input signal and the additional fact that there is a usually a small amount of system noise that acts as a dither signal to further randomize the quantization error spectrum.

2.5.3 Design Trade-offs for Number of Bits, Cost, Power, and So Forth

The most important aspect to remember about both receive chains (I/Q) is the effect of quantization from the ADC itself. That is, an N-bit ADC (with a fixed reference) have only 2^N possible digital outputs.

The resolution of data converters may be expressed in several different ways: the weight of the least significant bit (LSB), parts per million of full-scale (ppm FS), and millivolts (mV).

2.5.3 Design Trade-offs for Number of Bits, Cost, Power, and So Forth

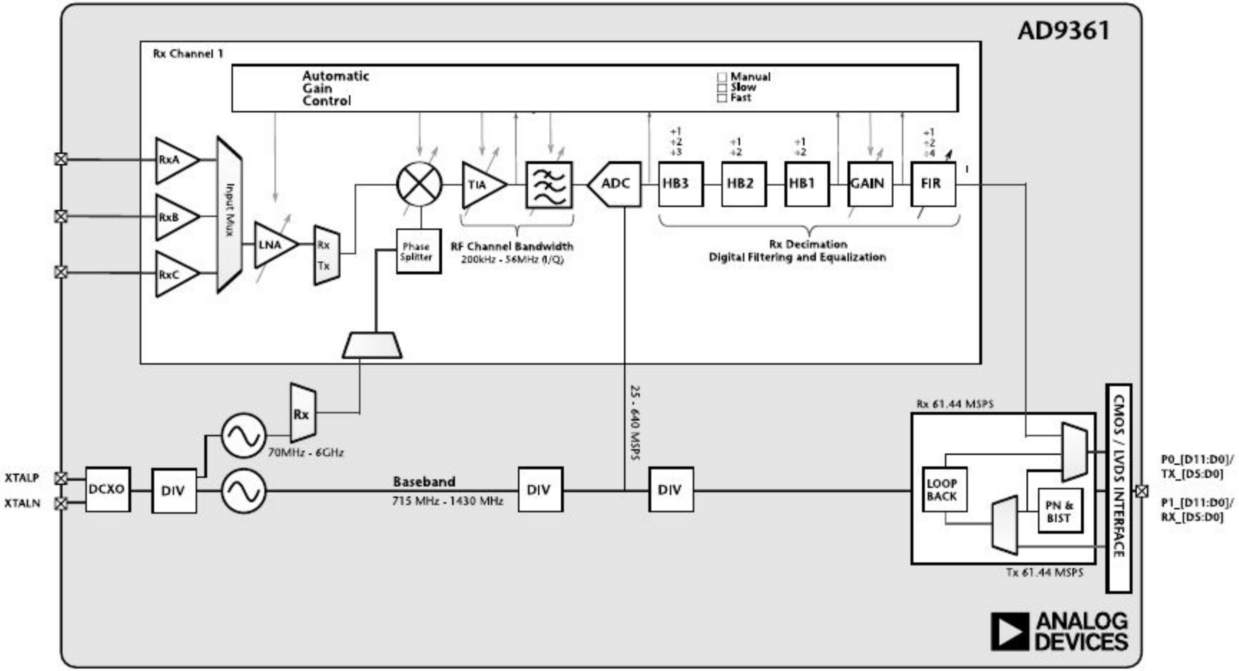
Table 2.3 Quantization: The Size of a Least Significant Bit

Resolution (N)	2^N	*Voltage (20 Vpp)*[1]	*PPM FS*	*%FS*	*dBFS*
2-bit	4	5.00 V	250,000	25	−12
4-bit	16	1.25 V	62,500	6.25	−24
6-bit	64	313 mV	15,625	1.56	−36
8-bit	256	78.1 mV	3,906	.391	−48
10-bit	1,024	19.5 mV	977	.097	−60
12-bit	4,096	4.88 mV	244	.024	−72
14-bit	16,384	1.22 mV	61.0	.0061	−84
16-bit	65,536	305 μV	15.2	.0015	−96
18-bit	262,144	76.2 μV	3.81	.00038	−108
20-bit	1,048,576	19.0 μV	.953	.000095	−120
22-bit	4,194,304	4.77 μV	.238	.000024	−132
24-bit	16,777,216	1.19 μV	.0596	.0000060	−144
26-bit	67,108,864	298 nV [1]	.0149	.0000015	−156

[1] 600 nV is the Johnson (thermal) noise in a 10-kHz BW of a 2.2 $k\Omega$ resistor at 25°C.

2.5.3 Design Trade-offs for Number of Bits, Cost, Power, and So Forth

In a brief recap from operational amplifier theory, two types of gain are associated with amplifiers: signal gain and noise gain. We want to increase the signal but at the same time keep the noise as as low as possible. This is accomplished by increasing the signal in the analog domain before digitizing it with the ADC.



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.

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.

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

$$ENOB = \frac{SNR - 1.76dB}{6.02dB}$$

2.5.4 Sigma-Delta Analog-Digital Converters

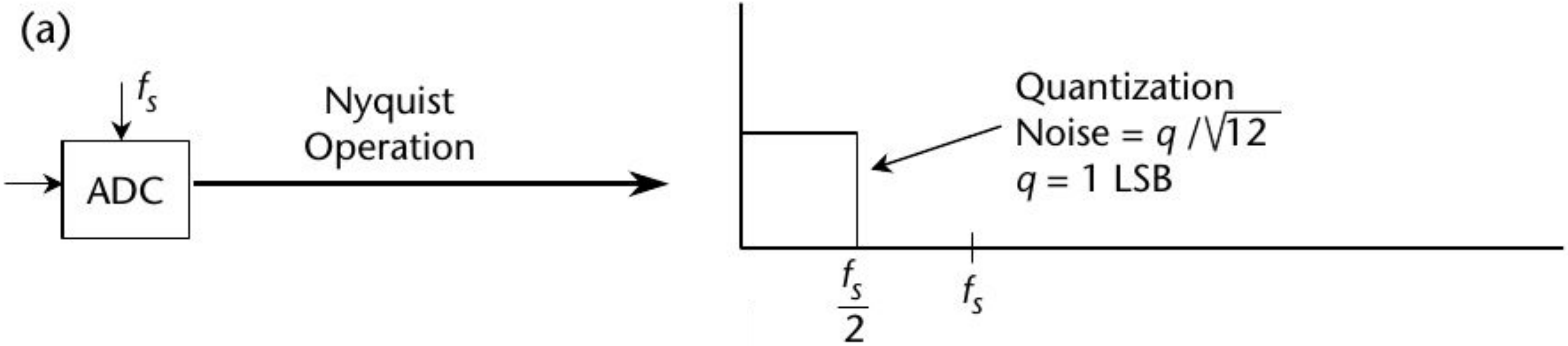
We can modify the previous equation to take into account the full-scale amplitude A_{FS} and the true input amplitude A_{IN} as

$$ENOB = \frac{SINAD - 1.76dB + 20log_{10}\frac{A_{FS}}{A_{IN}}}{6.02dB}$$

2.5.4 Sigma-Delta Analog-Digital Converters

How Sigma-delta (Σ- Δ) ADCs work:

First think of a typical ADC operation. At Nyquist operation noise is distributed from DC to fs/2.

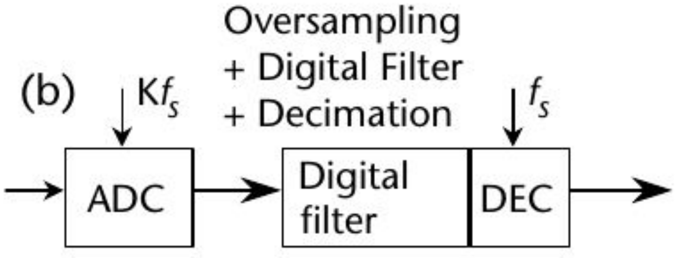


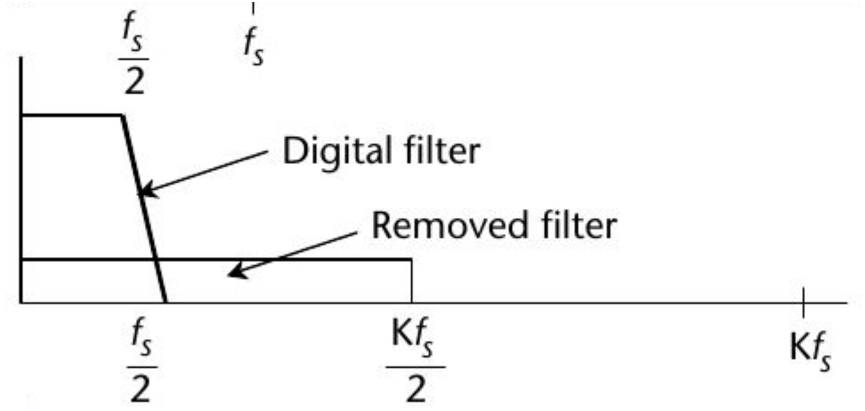
2.5.4 Sigma-Delta Analog-Digital Converters

How Sigma-delta (Σ- Δ) ADCs work:

Second think about this:

If we can oversample (interpolate) at Kfs, noise is now distributed from DC to Kfs/2. If digital LPF is applied, ENOB can be improved. To return our signal, we can then decimate to reduce our sampling rate.

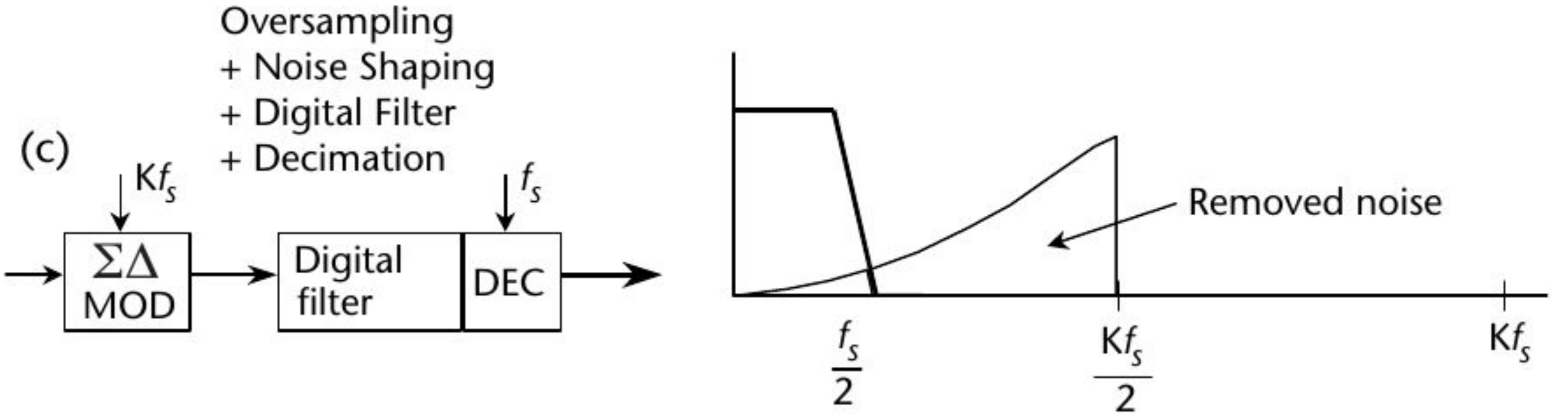




2.5.4 Sigma-Delta Analog-Digital Converters

How Sigma-delta (Σ- Δ) ADCs work:

By using more than one integration and summing stage in the Σ- Δ modulator, we can achieve higher orders of quantization noise shaping and even better ENOB for a given oversampling ratio.



2.5.4 Sigma-Delta Analog-Digital Converters

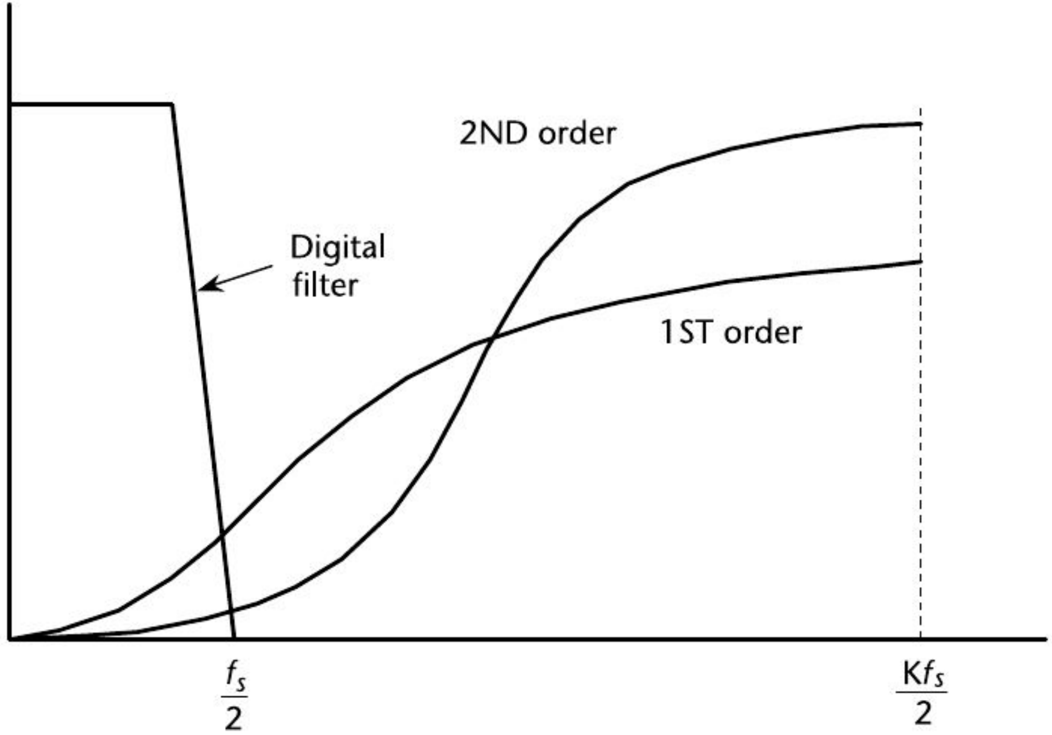


Figure 2.34 Σ-Δ modulators shape quantization noise.

2.5.4 Sigma-Delta Analog-Digital Converters

Let's look at some MATLAB code on *sfdr_test.m* file

2.6 Digital Signal Processing Techniques for SDR

- DSP relies on the concept of translating analog information into digital representations and by some mechanisms processing that data.
- Engineers and scientists rely on common tools and languages such as C and Verilog.
- Many important DSP software issues are specific to hardware, such as truncation error, bit patterns, and computational speed and efficiency of processors.

2.6.1 Discrete Convolution

Convolution is a mathematical tool of combining two signals to form a third signal, and forms the foundation for all DSP. Using the strategy of impulse decomposition, systems are described by a signal called the impulse response. Convolution is important because it relates the three signals of interest: the input signal, the output signal, and the impulse response.

$$y[i] = \sum_{j=0}^{M-1} h[j] \times x[i-j] = h[i] * x[i],$$

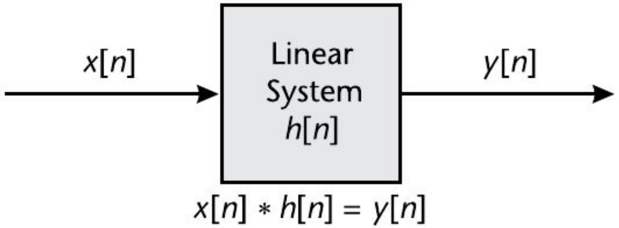


Figure 2.36 How convolution is used in DSP. The output signal from a linear system is equal to the input signal convolved with the system's impulse response.

2.6.2 Correlation

Cross-correlation and autocorrelation are two important concepts in SDR. Cross-correlation is a measure of similarity of two series as a function of the displacement of one relative to the other.

$$(f \star g)[n] = \sum_{j=-\infty}^{\infty} f^*[m] \times g[m+n]$$

Correlation is a mathematical operation that is very similar to convolution. It uses two signals to produce a third signal. This third signal is called the cross-correlation of the two input signals. If a signal is correlated with itself, the resulting signal is instead called the autocorrelation.

The value of the cross-correlation is maximized when the target signal is aligned with the same features in the received signal.

Using correlation to detect a known waveform is frequently called matched filtering.

2.6.3 Z-Transform

The z-transform of a discrete-time signal x[n] is defined as the power series:

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$

The z-transform is used to analyze discrete-time systems. Its continuous-time counterpart is the Laplace transform, defined as following:

$$X(s) = \int_{-\infty}^{\infty} x(t) e^{-st} dt$$

The z-transform and the Laplace transform can be connected by

$$z = e^{sT} \qquad s = \frac{1}{T} \ln(z)$$

2.6.3 Z-Transform

The z-transform has no meaning unless the series converge. The set of all the z values that makes its z-transform converge is called region of convergence (ROC). Based on the theory of series, these z-values must satisfy

$$\sum_{n=-\infty}^{\infty} \left| x[n] z^{-n} \right| < \infty$$

Table 2.4 Z-Transform Table: Selected Pairs[1]

$x[n]$	$X(z)$	*Region of Convergence*
$\delta[n]$	1	all z
$a^n u[n]$	$\frac{z}{z-a}$	$\lvert z\rvert > \lvert a\rvert$
$na^n u[n]$	$\frac{az}{(z-a)^2}$	$\lvert z\rvert > \lvert a\rvert > 0$
$n^2 a^n u[n]$	$\frac{az(z+a)}{(z-a)^3}$	$\lvert z\rvert > a > 0$
$\left(\frac{1}{a^n} + \frac{1}{b^n}\right) u[n]$	$\frac{az}{az-1} + \frac{bz}{bz-1}$	$\lvert z\rvert > \max(\frac{1}{\lvert a\rvert}, \frac{1}{\lvert b\rvert})$
$a^n u[n] \sin(\omega_0 n)$	$\frac{az \sin\omega_0}{z^2 - 2az\cos\omega_0 + a^2}$	$\lvert z\rvert > a > 0$
$a^n u[n] \cos(\omega_0 n)$	$\frac{z(z - a\cos\omega_0)}{z^2 - 2az\cos\omega_0 + a^2}$	$\lvert z\rvert > a > 0$
$e^{an} u[n]$	$\frac{z}{z-e^a}$	$\lvert z\rvert > e^{-a}$
$e^{-an} u[n] \sin(\omega_0 n)$	$\frac{ze^a \sin\omega_0}{z^2 e^{2a} - 2ze^a \cos\omega_0 + 1}$	$\lvert z\rvert > e^{-a}$
$e^{-an} u[n] \cos(\omega_0 n)$	$\frac{ze^a (ze^a - \cos\omega_0)}{z^2 e^{2a} - 2ze^a \cos\omega_0 + 1}$	$\lvert z\rvert > e^{-a}$

2.6.4 Digital Filtering

When doing signal processing, we usually need to get rid of the noise and extract the useful signal by the use of filters, digital filters in our case.

Since ideal brick wall filters are not achievable in practice, we limit our attention to the class of linear time-invariant systems specified by the difference equation

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k]$$

This system has the following frequency response:

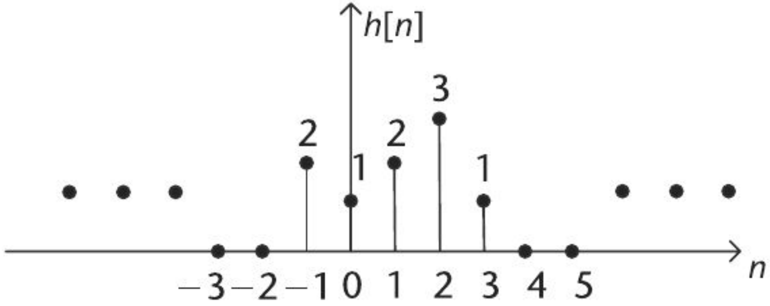
$$H(z) = \frac{\sum\limits_{k=0}^{M} b_k e^{-z}}{1 + \sum\limits_{k=1}^{N} a_k e^{-z}}$$

2.6.4 Digital Filtering

There are two basic types of digital filters, finite impulse response (FIR) and infinite impulse response (IIR) filters.

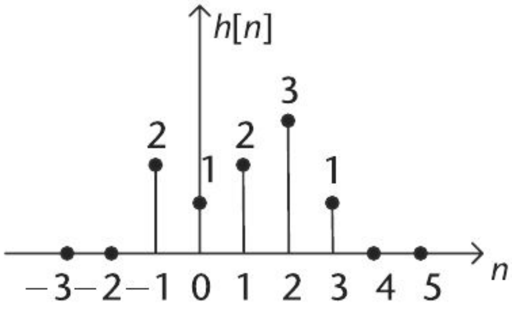
IIR Filter

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k]$$



FIR Filter

$$y[n] = \sum_{k=0}^{M-1} b_k x[n-k]$$



2.6.4.1 Case Study: Cascaded Integrator-Comb Filters

Cascaded integrator-comb filters (CIC filters) play an important role in the SDR hardware. They were invented by Eugene B. Hogenauer and are a class of FIR filters used in multirate signal processing. The CIC filter finds applications in interpolation and decimation.

A CIC interpolator filter consists of one or more comb filter, an upsampler, and integrator pairs. The decimator is just a reverse structure of integrator and comb filters, and a downsampler instead of a upsampler.

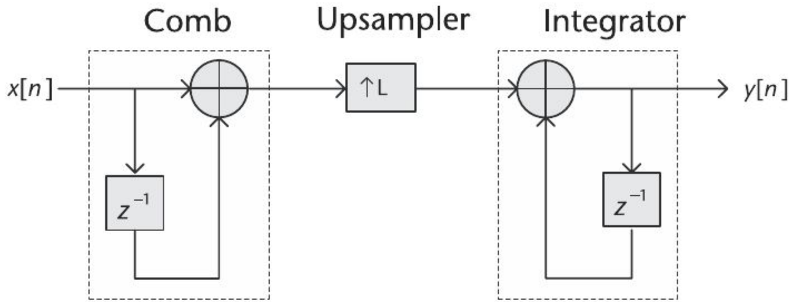


Figure 2.42 The structure of an interpolating cascaded integrator-comb filter [17], with input signal $x[n]$ and output signal $y[n]$. This filter consists of a comb and integrator filter pair, and an upsampler with interpolation ratio L.

2.6.4.1 Case Study: Cascaded Integrator-Comb Filters

The comb filter has the system function:

$$H_L(z) = \frac{1}{M+1} \frac{[1 - z^{-L(M+1)}]}{(1 - z^{-L})}$$

where L is decimation or interpolation ratio, M is number of samples per stage, usually 1 or 2.

The common form of the CIC filter usually consists of several stages, then the system function for the composite CIC filter is

$$H(z) = H_L(z)^N = \left(\frac{1}{M+1} \frac{1 - z^{-L(M+1)}}{1 - z^{-L}} \right)^N$$

where N is number of stages in the composite filter.

2.6.4.2 Case Study: FIR Halfband Filter

2.7 Transmit Techniques for SDR

An analog signal converted to a digital signal using an ADC cannot be directly used for transmission. These signals must be first conditioned then converted back into an analog signal DAC.

Extra decimation will allow bit growth and extra fidelity to gain in the system.

When the time domain data is consider beyond the digital limits (Nyquist band), aliases of signals are still visible.

This is why we traditionally have two filters in the transmit portion of a SDR, a digital transmit or pulse-shaping filter, which changes each symbol in the digital message into a digital pulse stream for the DAC. This is followed by an analog reconstruction filter, which removes aliasing caused by the DAC

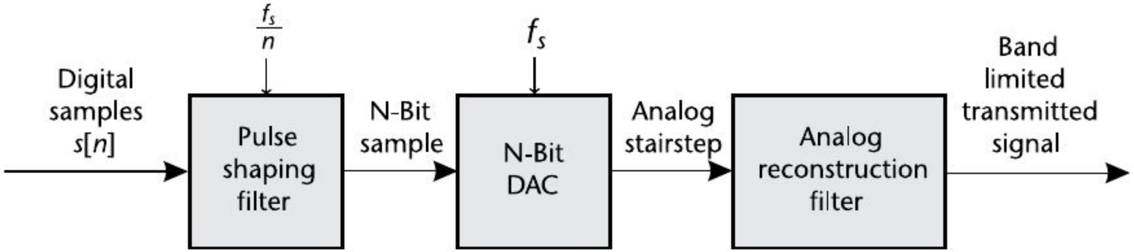


Figure 2.46 On the transmitter side, the DAC converts the digital symbols into an analog signal for transmission.

2.7.1 Analog Reconstruction Filters

Increasing data rate on de DAC and the interpolating allows the use of simpler analog reconstruction filters.

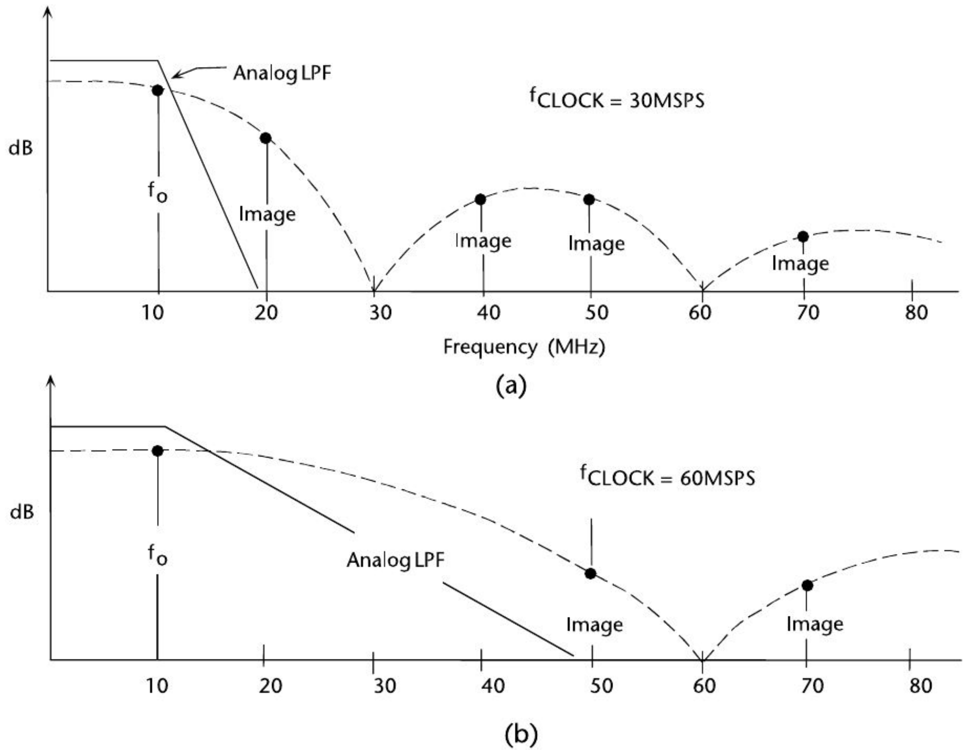


Figure 2.47 Analog reconstruction filter requirements for $f_o = 10$ MHz, with $f_s = 30$ MSPS, and $f_s = 60$ MSPS [18].

2.7.2 DACs

Nearly all DACs operate by holding the last value until another sample is received. This is called a zeroth-order hold.

The zeroth-order hold can be understood as the convolution of the impulse train with a rectangular pulse, having a width equal to the sampling period.

Unfortunately the sinc function attenuates signals in the passband so some correction needs to be performed.

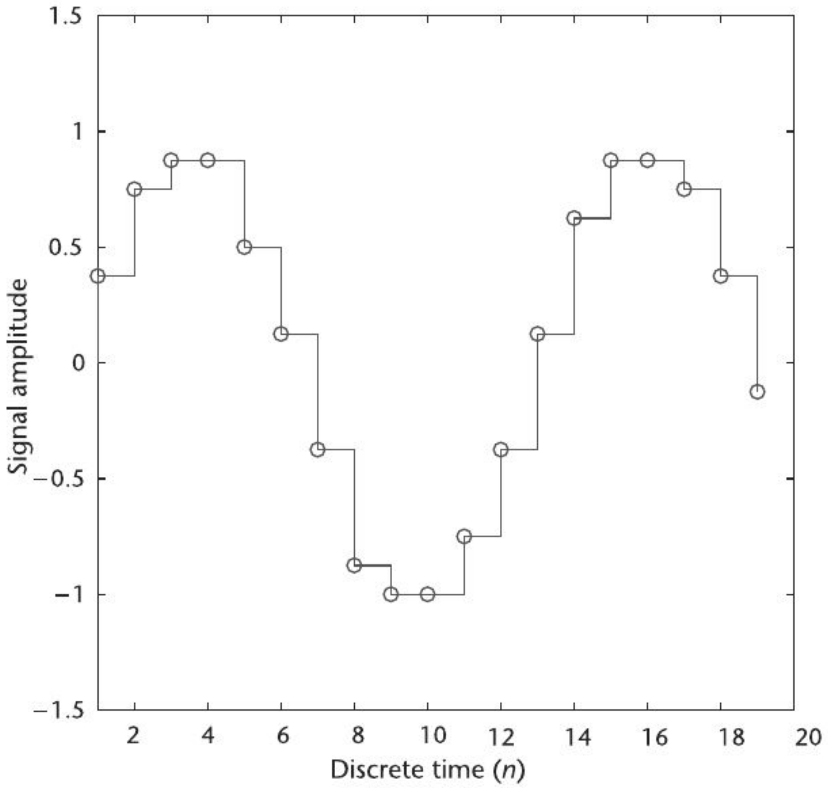


Figure: zeroth-order hold

2.7.2 DACs

The analog pulse train at the DACs output is

$$s_a(t) = s[n]p(t) = \sum_{k=-\infty}^{\infty} s(kT)\delta(t - kT) = \begin{cases} s(kT) & t = kT \\ 0 & t \neq kT \end{cases}$$

The zeroth-order hold results in the spectrum of the impulse train being multiplied by sinc function, given by the equation

$$H(f) = \left| \frac{\sin(\pi f / f_s)}{\pi f / f_s} \right|$$

2.7.3 Digital Pulse-Shaping Filters

When transmitting an alternating bitstream of ones and zeros in a digital modulation scheme, the generated the square wave has infinite frequency information.

We cannot recover such a signal due to finite bandwidth and interference between adjacent symbols.

There are two situations when adjacent symbols may interfere with each other:

- When the pulse shape is wider than a single symbol interval T.
- When there is a nonunity channel that smears nearby pulses, causing them to overlap.

In order to solve these problems, pulse-shaping filters are introduced to bandlimit the transmit waveform.

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.

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.

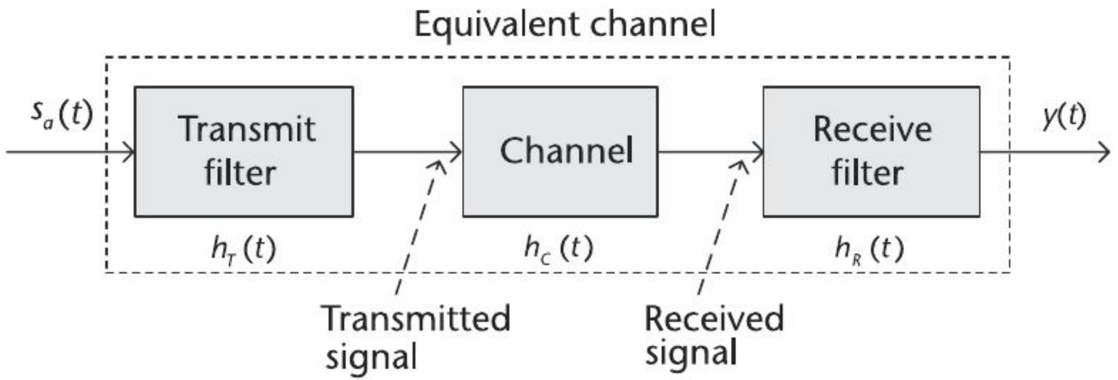


Figure 2.49 The equivalent channel of a communication system, which consists of the transmit filter, the channel, and the receive filter.

2.7.4 Nyquist Pulse-Shaping Theory

The impulse response of the equivalent channel is

$$h(t) = h_T(t) * h_C(t) * h_R(t)$$

We say that h(t) is a Nyquist pulse if it satisfies

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

Which is a condition that one pulse does not interfere with other pulses at subsequent T-spaced sample instants.

2.7.5 Two Nyquist Pulses: Sinc Pulse

Consider the rectangular pulse

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

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

$$h(t) = \text{sinc}\left(\frac{t}{T}\right)$$

Change the variable t = kT yields

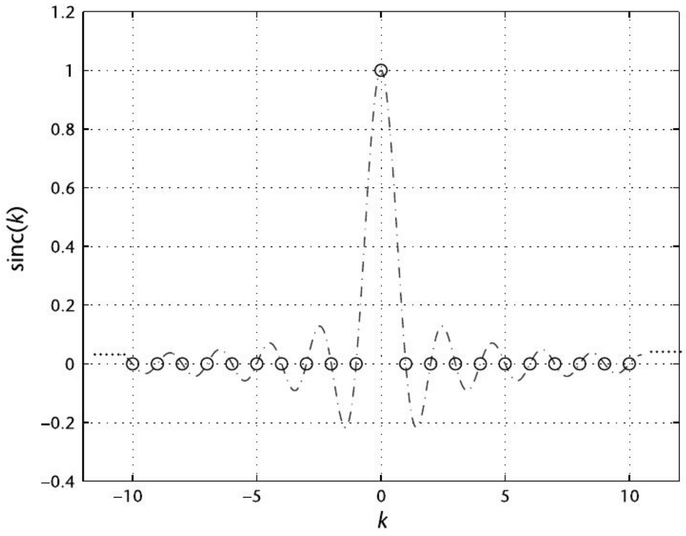
$$h(t) = h(kT) = \text{sinc}\left(\frac{kT}{T}\right) = \text{sinc}(k)$$

2.7.5 Two Nyquist Pulses: Sinc Pulse

We can continue writing

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

Therefore the sinc pulse exactly satisfies Nyquist pulse-shaping theory.



Our T sampling instants are located at the equally spaced zero crossings, so there will be no intersymbol interference.

2.7.5 Two Nyquist Pulses: Sinc Pulse

Sinc pulse h(t) yields the minimum bandwidth

$$B = B_{\min} = \frac{1}{2T}$$

so it is called the Nyquist-I Pulse

2.7.5 Two Nyquist Pulses: Sinc Pulse

Sinc pulses are a very attractive option because they are:

- Wide in time (no ISI)
- Narrow in frequency (spectrum efficient)

Sinc pulses are not practical because they:

- Have ISI sensitivity due to timing errors. For large t, they have the following approximation

$$h(t) \sim \frac{1}{t}$$

- Are infinite in time, making them unrealizable.

2.7.5 Two Nyquist Pulses: Raised Cosine

Is a Nyquist-II pulse, which has a larger bandwidth B > Bmin , but with less ISI sensitivity.

The raised cosine pulse has the frequency transfer function defined as

$$H_{RC}(f) = \begin{cases} T & 0 \le |f| \le \frac{1-\beta}{2T} \\ \frac{T}{2}\left(1+\cos\left(\frac{\pi T}{\beta}(|f| - \frac{1-\beta}{2T})\right)\right) & \frac{1-\beta}{2T} \le |f| \le \frac{1+\beta}{2T} \\ 0 & |f| \ge \frac{1+\beta}{2T} \end{cases}$$

β is the rolloff factor, which takes value from 0 to 1, and β/2T is the excess bandwidth.

2.7.5 Two Nyquist Pulses: Raised Cosine

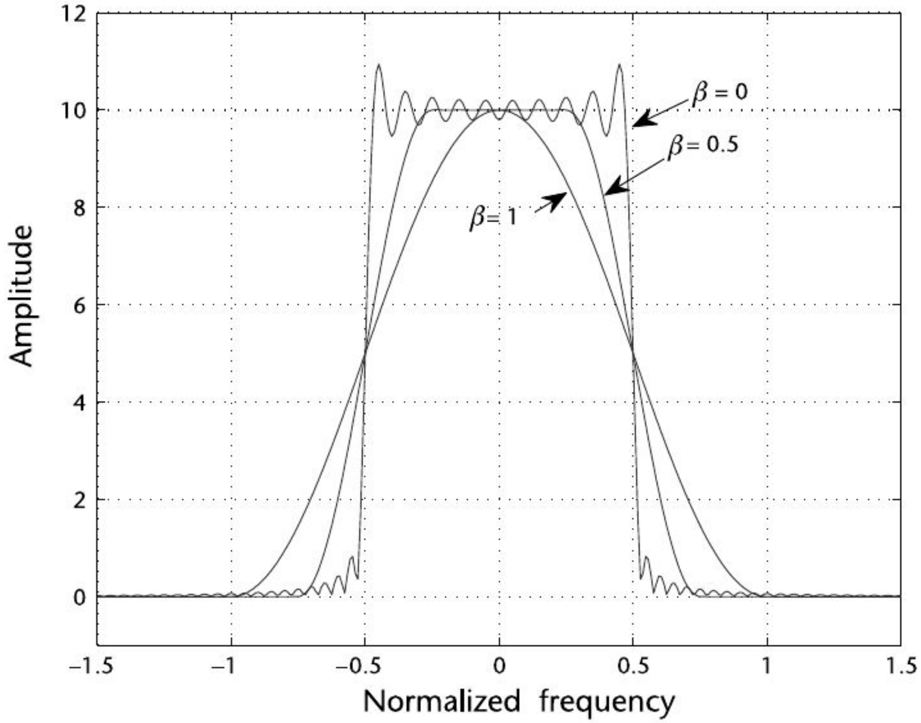


Figure 2.52 Spectrum of a raised cosine pulse defined in (2.87), which varies by the rolloff factor β. The x-axis is the normalized frequency f_0. The actual frequency can be obtained by f_0/T.

2.7.5 Two Nyquist Pulses: Raised Cosine

The impulse response of raised cosine pulse, defined as

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

The tail of $h_{RC}(t)$, converges quickly, therefore the distortion will not accumulate to infinity.

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

2.7.5 Two Nyquist Pulses: Root Raised Cosine

In many practical communications systems, root raised cosine filters are usually used

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 $H_{RC}(f)$

The impulse response of the equivalent channel can be expressed as

$$h(t) = h_T(t) * h_C(t) * h_R(t)$$

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

$$H_{RC}(f) = H_T(f)H_R(f)$$

2.7.5 Two Nyquist Pulses: Root Raised Cosine

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

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