Note: Printed Manuals are not in Color

Sampling

Objectives

This chapter explains the following:

- The principles of sampling, especially the benefits of coherent sampling
- How to apply sampling principles in a test situation to digitize an analog signal with a waveform digitizer or an ADC
- How analog signals can be created from a set of digitized samples
- How a continuous signal can be sampled
- The basic capabilities of different Fourier transform algorithms
- The results obtained from digitizing a sample set utilizing Fourier transforms
- Inverse Fourier transformation and how it can be used to create a sample set that can be used to create a complex waveform
- How to generate an analog signal by applying sampling principles
- How sampling a single frequency can represent more than one set of frequencies, which can corrupt sampled data
- How to prevent spectral replication/aliasing
- sin(x)/x amplitude errors
- The causes of and how to prevent spectral leakage
- How windowing functions can minimize leakage

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Terms and Definitions used in this Chapter

Alias	A false signal that is created as a by-product of sampling and Fourier transformations
Coherent Sampling	Sampling an integer number of samples from an integer number of cycles
Discrete Fourier Transform (DFT)	A mathematical algorithm that converts time domain signals to frequency domain
Fast Fourier Transform (FFT)	A mathematical algorithm that is substantially faster than DFT
Fourier Frequency	Spectral resolution
Frequency Bin	Spectral resolution in the frequency domain; also known as the Fourier Frequency
Inverse Fast Fourier Transform (IFFT)	Converts frequency domain signals to the time domain
Nyquist Frequency	The minimum frequency required to sample a signal containing a various frequencies of interest. The Nyquist Frequency must be greater than 2x the highest frequency of interest
Nyquist Limit	The highest frequency of interest that can be represented when sampling at a particular Nyquist Frequency. The Nyquist Limit is ½ the Nyquist Frequency
Quantization Error	Analog signal amplitude error determined by LSB size
Spectral Leakage	Frequency energy leakage into adjacent spectral bins
Unit Test Period	The time required to sample a specified number set of cycles

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Sampling Theory

Sampled signals behave differently than continuous signals. In order to create or analyze sampled signals correctly, we need to apply proper sampling theory. In this chapter we explain the basics of sampling theory and coherent sampling, and introduce the student to potential problems and issues when dealing with sampled waveforms.

Sampling Requirements

Consider a continuous time signal that must be digitized and stored as a sequence of digital numbers. Sampling must occur at specific intervals, referred to as the sample time, and be converted to a digital number. These digital numbers are stored as “sample points” that describe the waveform. If the samples are taken correctly, an infinite-valued signal can be completely represented by a set of finite sample points. Sample time is the inverse of the sample frequency $F_{s.}$

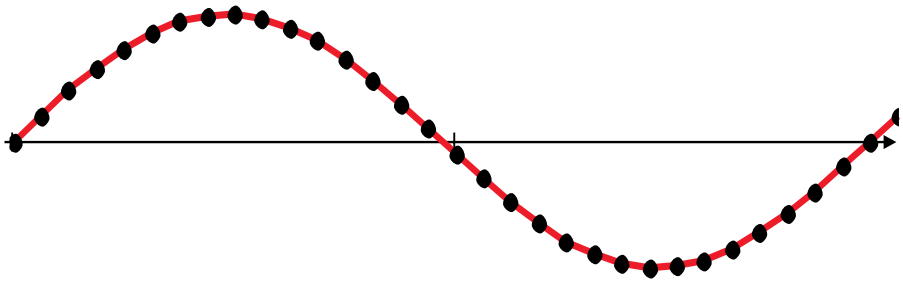

Figure 6.1: Sampling a Continuous Waveform. In this example, 31 samples were taken over one cycle of the waveform. The samples are equally spaced in time. Sample 32 occurs at the beginning of the second cycle.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

The Shannon Theorem

In 1949, Claude Shannon[1] proved that a continuous mathematical function can be completely determined by samples taken at twice the highest frequency component in the function. This applies to classic mathematical functions that go from minus to plus infinity. When a periodic signal such as a sine wave is sampled at exactly twice the signal frequency, the information is not quite enough to later reconstruct the signal, as seen in Figure 6.1.

The Nyquist Limit

A small set of samples do not cover a broad range, and have restrictive sample requirements; sampling only a small portion of an infinite series prevents complete information recovery. Harry Nyquist demonstrated that to recreate a signal from its samples, you must sample at a rate higher than two times the highest frequency of interest contained in the signal. The important point is that more than two samples per test signal period are required, not much more; “just one extra sample will do the trick.”[2] Naturally, the more samples per period, the more information that can be gleaned from the sample set.

The phrase “highest frequency of interest” does not always refer to the fundamental frequency being tested. Other frequency components of interest, such as harmonics may require consideration.

Periodicity

Sampling theory applies to signals that are considered by the mathematics to be periodic even if they are not. Therefore, errors are introduced if a signal is sampled that is not periodic.

Sample Sets

A sample set is an array of digital information that represents a sampled analog signal. A sample set can be created using a Waveform Digitizer, or it can be mathematically created from an algorithm. A sample set contains no information about time or frequency in the numbers. It is important to understand the relationship between sample time and signal frequency and how they relate to the index of each sample value in a set. In general, more samples are better from the perspective of digitizing or generating signals. In this chapter, the discussion of sampling applies to both digitizing and generating signals.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Converting a Time Sample Set to Frequency

A benchtop instrument known as a spectrum analyzer displays the frequency characteristics of signals. The basic operation of a spectrum analyzer is similar to putting the signal into a bank of bandpass filters as shown in Figure 6.2.

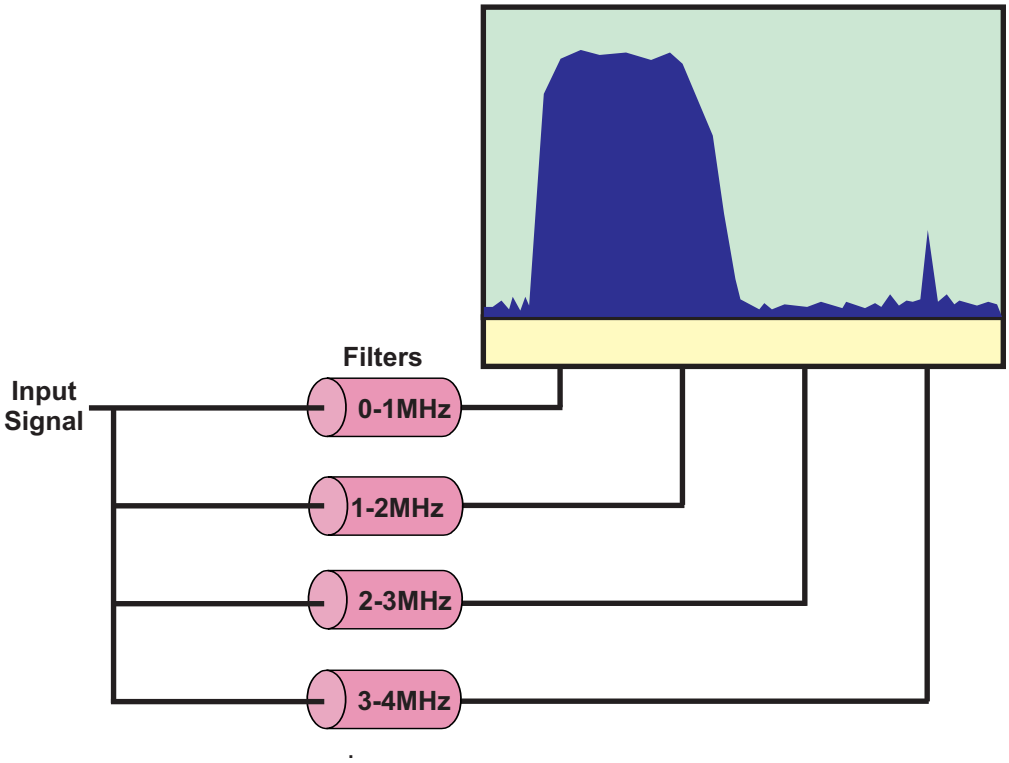

Figure 6.2: How a Spectrum Analyzer Computes Frequency Data. An input is passed through several filters simultaneously, and the resulting RMS voltage is plotted on a display.

Each filter extracts the sinusoidal components in its bandpass range and sets the amplitude to be plotted for that band. The first benchtop spectrum analysis instruments worked this way, and some purely continuous spectrum analyzers still work this way. However, because DSP-based spectral analysis offers many advantages over traditional techniques, this type of spectral analysis is fast becoming obsolete, .

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

The section, *Fourier Series* on page 2-23, describes how a complex waveform can be analyzed as a sum of pure sinusoids and discusses the mathematics required for the analysis. A set of discrete samples taken from a time waveform can also be analyzed and mathematically converted into a discrete set of frequency data that represents the sinusoidal frequency components of the waveform. The conversion of time to frequency data is performed with an algorithm known as the Discrete Fourier Transform.

Discrete Fourier Transform (DFT)

The DFT is a mathematical algorithm that translates amplitude data taken in time into amplitude data versus frequency. It is the mathematical equivalent of a spectrum analyzer and takes amplitude versus time data and returns amplitude versus frequency data. Mathematically, the algorithm is a series summation of the product of each sample times a complex number. It is the discrete equivalent of the Fourier integral in Equation (2.32) and is stated as:

$$X(b) = \sum_{n=0}^{N \angle 1} x(n)[cos(2\pi nb/N) \angle j sin(2\pi nb/N)] \quad \textbf{(6.1)}$$

where n = one of N samples, N = total number of samples, b = one of B frequency bins where each bin represents a frequency range of F_s/N and j is the imaginary operator. The DFT algorithm uses each sample point in the summation from 0 to N - 1 for each analyzed frequency. Note that all N sample points contain information about all B frequencies, thus each of the B frequencies for which information is desired requires a summation of N time sample products. Processing a DFT is slow, because N^2 calculations are necessary. For example, a 2000 point DFT requires 4 million calculations, often floating point calculations, that are slower than integer calculations.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Fast Fourier transform (FFT)

The FFT remedies the DFT speed problem by skipping over portions of the summations that produce redundant information. A mathematical description of an FFT algorithm is non–trivial and is not important to our use of it. Consider it as a tool that can be used without having to understand the details of how it works. Two facts are important to know for using the FFT:

- The number of sample points must be a power of 2
- The number of additions and multiplications is $\frac{N}{2} \cdot log_2 N$

For a 2048 point data set, an FFT requires 11,264 calculations; using a DFT requires over 4 million!

Using a Fast Fourier Transform Algorithm

The Fast Fourier Transform (FFT) algorithm is a mathematical tool used in mixed signal testing as well as other types of signal processing. An FFT passes time data into the algorithm and receives the returned frequency data.

The time data passed to an FFT algorithm is an array of real amplitude sample values, taken from a signal or created mathematically using the Soft Test DSP Lab software. Because an FFT operates in complex number space, it returns frequency data as a real and an imaginary frequency array.

Let the Test System Do the Processing

The DSP subsystem in mixed signal testers will have optimized routines for much of this “number crunching.” For example, there will be a built in routine that takes a set of time samples and returns a set of magnitude points, and another routine that takes a rectangular frequency array pair and returns a magnitude and phase array pair. Make sure you know and understand which built-in tester routines are available with your system, and then use them!

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Benefits of and Problems Caused by Sampling

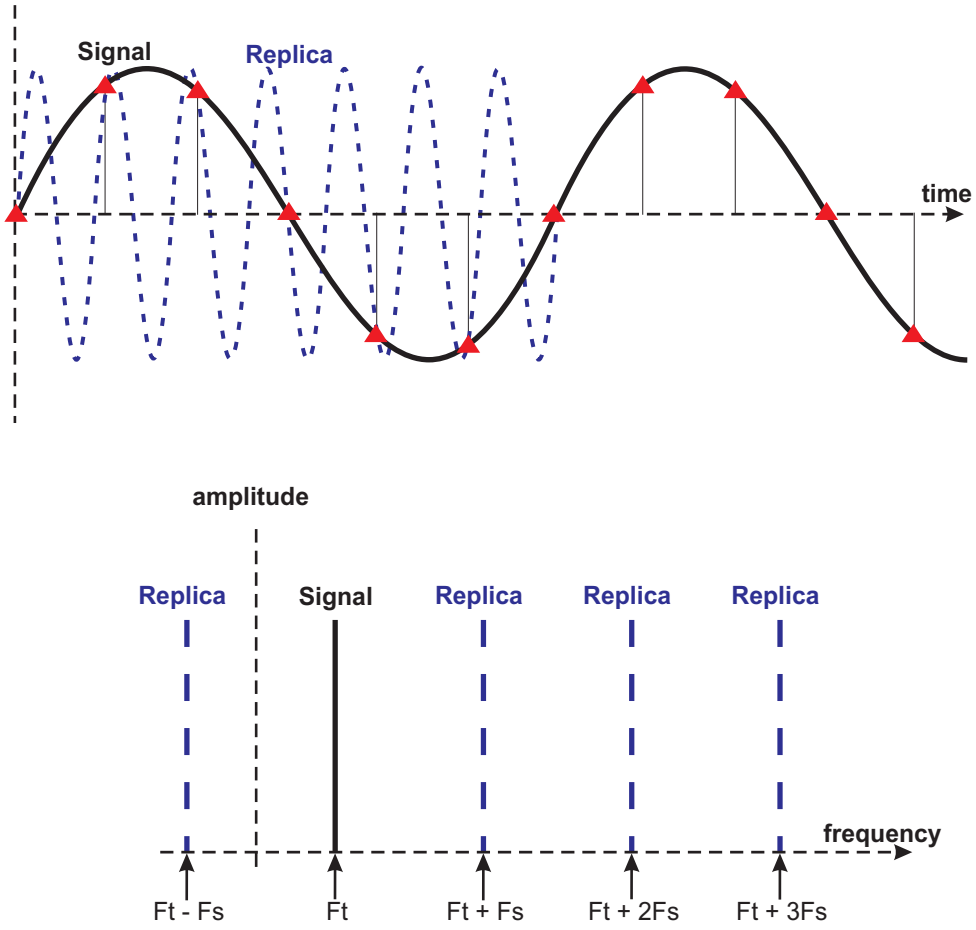

Figure 6.3: Ambiguity Inherent in Sampled Data. The sample points could represent any one of an infinite number of sinusoids.

Spectral Replication

Using DSP techniques, a set of amplitude samples over time can be converted into a set of samples in the frequency domain. The frequency information returned from a DSP operation such as an FFT can be ambiguous regarding frequency, because more than one sinusoid can fit the sample points.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

A sample set contains the original test signal frequency (F_t), replicated at frequencies every k times the sample frequency in both the positive and negative directions. Figure 6.3 on page 6-8 illustrates this phenomenon; mathematically, the frequency components in the test signal are replicated to other frequencies as:

$$F_{replica} = k \cdot F_s \pm F_t \text{ , } k = integer$$

Notice how the sampled waveform has identical components at additional frequencies; these replicas extend to infinity. The information returned by a DFT/FFT can be analyzed using the “low pass” data, that is, the band of frequencies out to $F_s/2$, ignoring the replicas.

Note that the replicas extend to infinity in both directions. A sampled signal with a component higher than those in the band of interest, but lower than the sample frequency F_s can end up *aliased* into the frequency band of interest. This can be a problem—if the input signal being sampled contains signal components greater than $F_s/2$, those signal components are “folded” into the frequency data for the original signal, distorting the spectral information.

In fact, the spectrum from $F_s/2$ to F_s is not a replica per se - it is a mirror image. So is the spectrum from $3F_s/2$ to $2F_s$, and each “even numbered” spectrum. The “odd numbered” spectra are replicas.

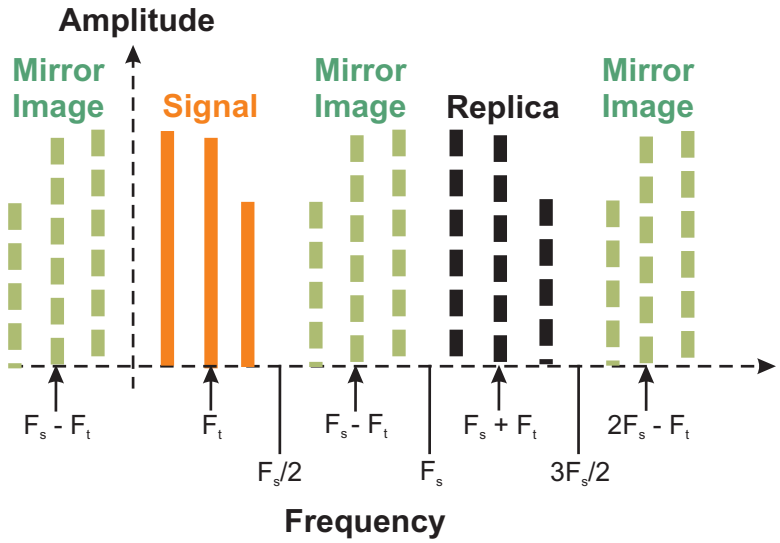

Figure 6.4: Spectral Replicas and Mirror Images. A set of discrete time domain samples results in a continuous frequency domain representation.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Aliasing

Although replication and aliasing can mean the same thing, we prefer to define “aliasing” as the distortion of frequency results due to signal components that are folded into the frequency band of interest from outside the Nyquist band.

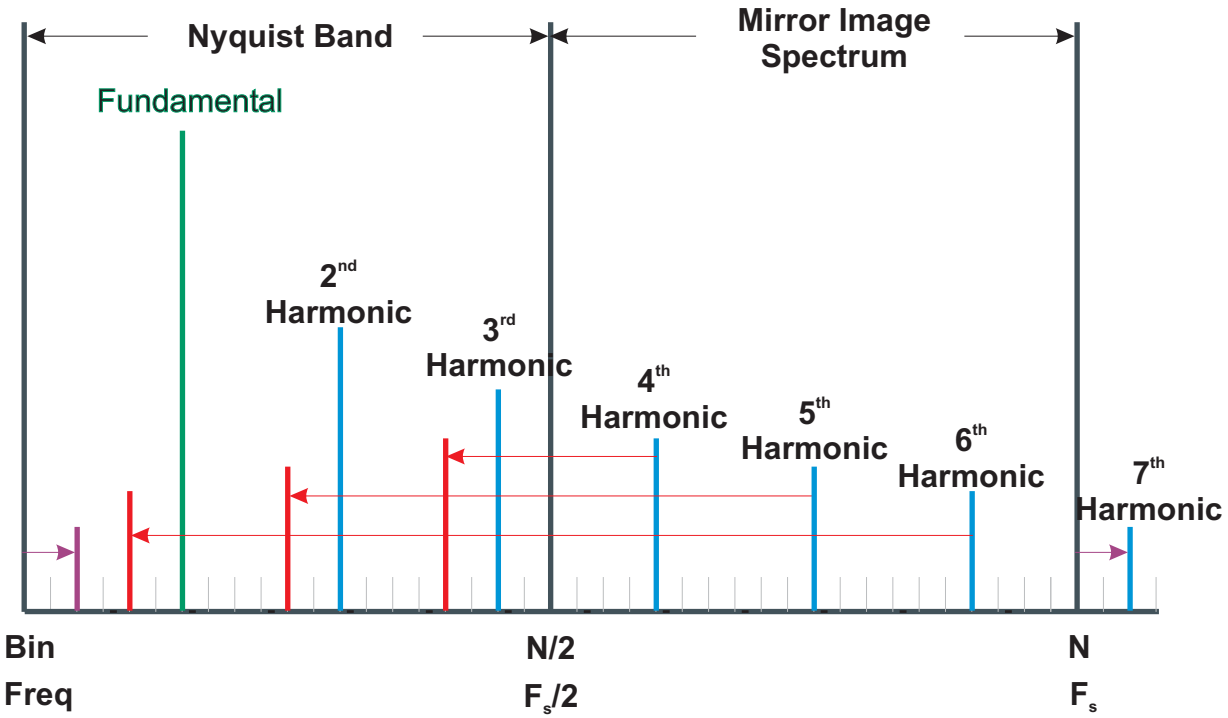

Figure 6.5: Effects of Aliased Signal on the Nyquist Band.

An alias adds non–harmonically related frequency components to the sampled signal which cannot be distinguished nor removed from the signal of interest. In other words, a frequency component that is aliased into the frequency band of interest destroys the integrity of the information returned by the Fourier Transform. Notice in Figure 6.5 that the spectral band of interest is from 0 to $F_s/2$. Unless removed, replicas from other areas in the spectrum will be aliased into the low pass band of interest, corrupting the frequency data.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

An Aliasing Example

A common set of test parameters for a telecom device uses a test frequency of 1020Hz, sampled at 8KHz. The fourth harmonic and any higher frequencies will cause aliases to appear in the frequency spectrum.

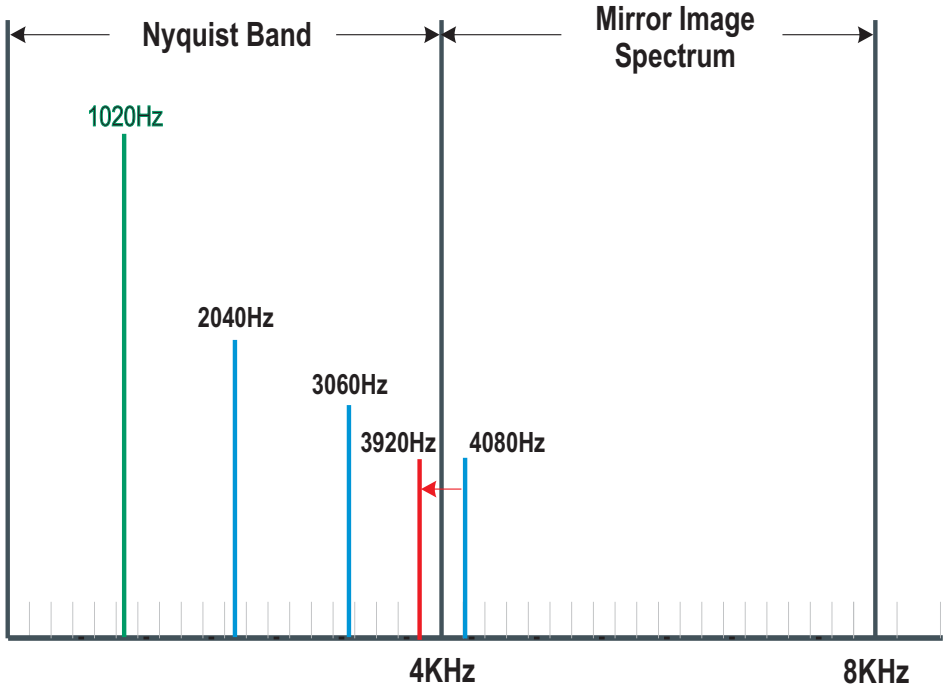

Figure 6.6: Aliasing Example. The fourth harmonic appears at the wrong frequency. Instead of appearing at 4080Hz, it “wraps back” to 3920Hz.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Prevention of Aliasing Errors

There is only one way to avoid the aliasing problem; filter the signal to remove frequency components greater than $F_s/2$ from the signal *before* it is either used as an input to an ADC, or it is digitized. Filtering must be done with a continuous analog filter as shown in Figure 6.7; any sort of switched capacitor filter will introduce its own high frequency noise, and will not solve the problem. The distorted data exists in and corrupts the time samples so it cannot be removed with mathematical “digital filtering” via a DSP algorithm after it has been digitized or converted to binary information.

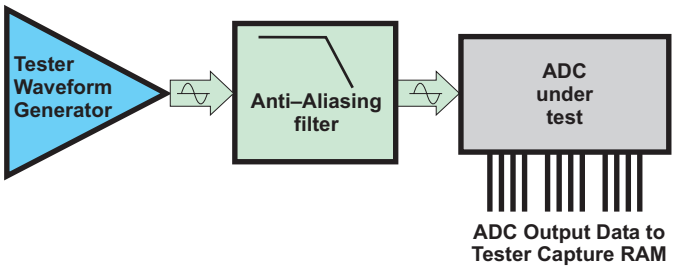

Figure 6.7: Use of an Anti-Aliasing Filter. The filter removes frequencies above the Nyquist Limit prior to digitizing.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Spectral Leakage

Recall that Fourier transform algorithms assume that time data passed to them is periodic. Because the DFT/FFT returns a discrete set of amplitude points in the frequency domain, its results contain information only about frequencies that are integer multiples of the fundamental frequency F_s/N. The result of this and the assumption of periodicity is that frequency components contained in the sample set which are not integer multiples of F_s/N "leak" into the frequency points which are returned by the DFT/FFT.

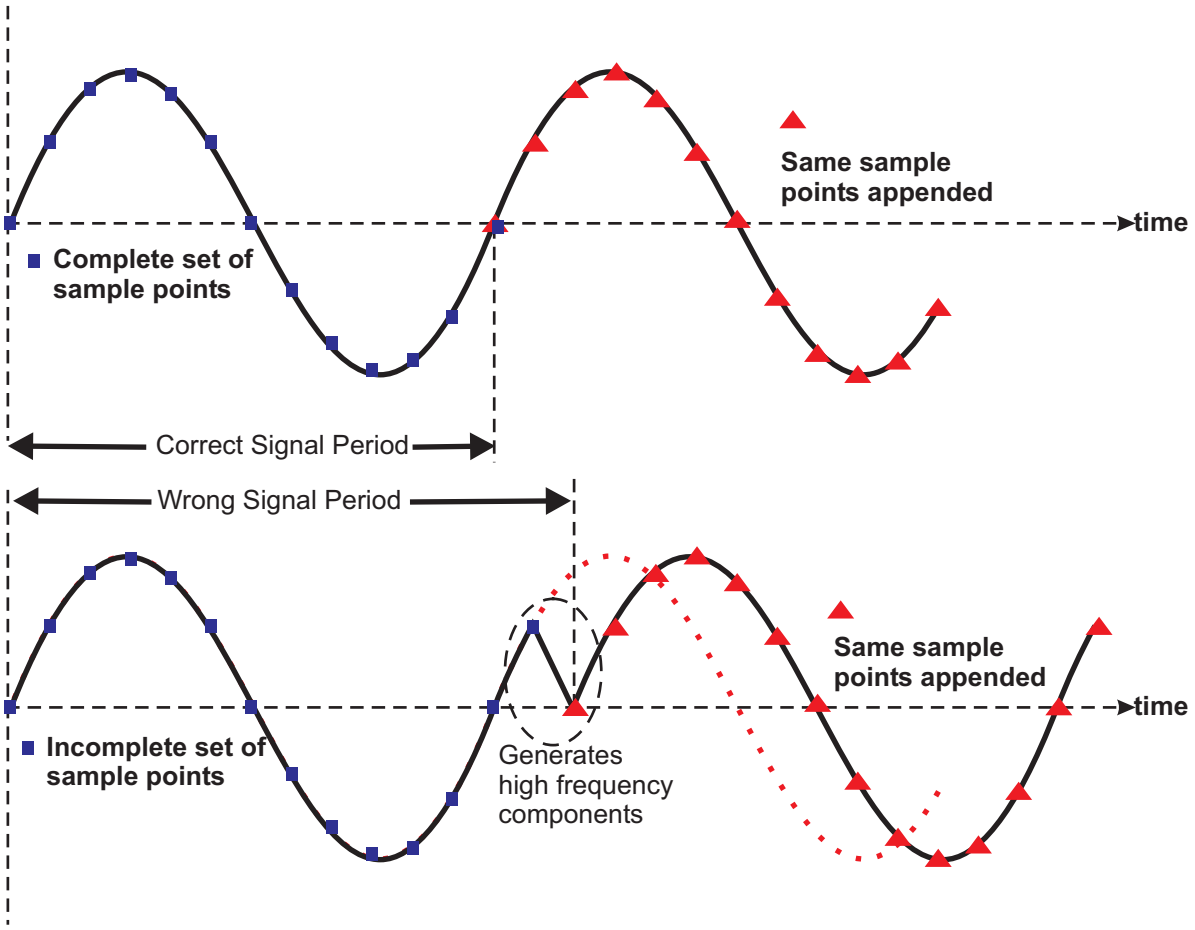

Figure 6.8: Sample Set with Spectral Leakage. Too many samples were taken, causing the non-overlapping effect.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Incomplete Sample Set

If a time sample set does not contain a precise integer number of cycles, spectral leakage will occur. If a longer sample set is created by appending a second set to the first set, leakage occurs because the pair do not form a continuous waveform. A discontinuity occurs where the two sample sets join, as shown in Figure 6.8 on page 6-13, causing a sharp edge within a sinusoid. As discussed earlier, a sharp edge in time contains high frequency components. If a sine wave is sampled, but the sample set does not include an exact number of test signal periods, then a sharp jump occurs at the roll–over point between two samples. The discontinuity shown in Figure 6.8 is exaggerated to illustrate the effect. Less extreme cases will cause less obvious errors, but will still distort test results.

In the general case of sampling, leakage is unavoidable, although time window functions can minimize the effects of leakage. However, in the ATE arena where signals and samples can be much more controlled, leakage can be eliminated by using a *coherent sample set*. First we will discuss the use of time window functions for minimizing leakage; *coherent sampling* to eliminate the problem is discussed later.

.

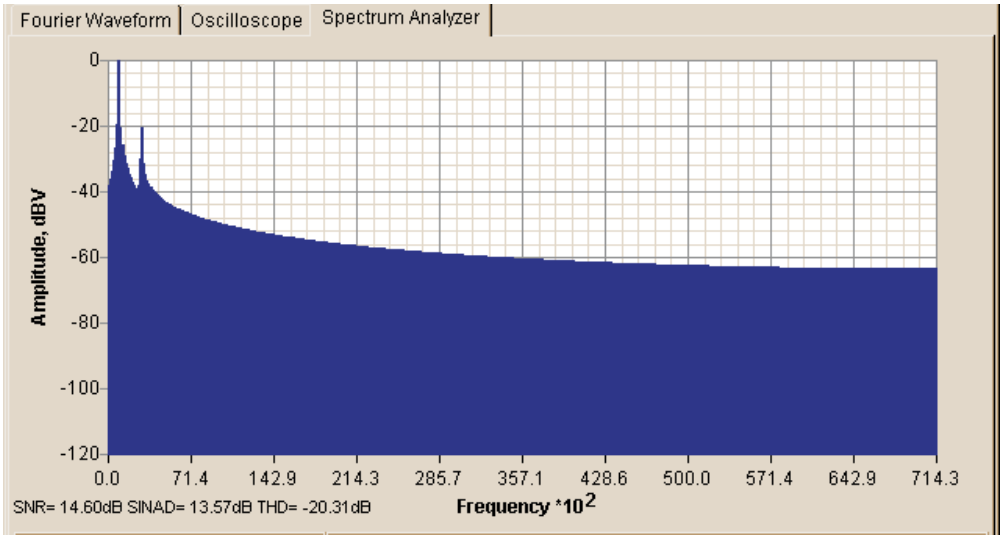

Figure 6.9: Spectral Leakage in the Frequency Domain. The leakage will interfere with signal-to-noise ratio testing.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Time Sample Windowing to Reduce Leakage

DFT and FFT algorithms treat sample data as though it were one period of a periodic sequence. If the data for a waveform is not periodic, then distortion may occur because the periodic waveform created by the DFT/FFT may have sharp discontinuities at the boundaries of the sample data set.

Name	Window Function
Rectangular	$coefficient_n = 1$
Triangular (Bartlett)	$coefficient_n = \frac{n}{N/2}$, for $0 \le n \le \frac{N}{2}$ and $coefficient_n = 2 \angle \frac{n}{N/2}$, for $\frac{N}{2} < n \le (N \angle 1$
Hann	$coefficient_n = \frac{1}{2}\left[1 \angle \cos\left(\frac{2\pi n}{N \angle 1}\right)\right]$
Hamming	$coefficient_n = 0.54 \angle 0.46\cos\left(\frac{2\pi n}{N \angle 1}\right)$
Blackman	$coefficient_n = 0.42 \angle 0.5\cos\left(\frac{2\pi n}{N \angle 1}\right) + 0.08\cos\left(\frac{4\pi n}{N \angle 1}\right)$
Blackman-Harris	$coefficient_n = 0.36 \angle 0.49\cos\left(\frac{2\pi n}{N \angle 1}\right) + 0.14\cos\left(\frac{4\pi n}{N \angle 1}\right) \angle 0.01\cos\left(\frac{6\pi}{N \angle}\right.$

Table 6.1: Window Functions.

The primary trade–off when using windowing functions is the width of the main lobe versus the height of the side lobes.[3] To prevent windowing functions from dominating the spectral information generated from the time samples, the main lobe must surround the test signal fundamental frequency and the amplitude of the side lobes must be less than the test signal's noise and harmonics amplitudes.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Using Window Functions

Window functions are normally associated with digitizing a waveform. The data collected by the digitizer is processed with a window function prior to passing the sample set array to the DFT/FFT algorithm. To apply a window function, create a loop that contains the window function and compute the value of the function for each n in the sample set array. Then multiply the value at that n^{th} array location by the computed value.

Time sample windowing is a mathematical trick to modify time samples to reduce distortion from this cause. Distortion is minimized by windowing the sample data set so the ends of the data set are smoothly tapered to zero. Some window functions work better in specific situations than others and the choice of the window type is somewhat of an empirical science. The important things to know about windowing are:

- A window equation is used to process time samples **before** performing an FFT or DFT.
- Unless special care is taken, windowing reduces spectral leakage to about -80dB, so if you must test a high resolution device (> 12 or 13 bits is a general rule) windowing may not help.
- Windowing functions are frequently part of the standard software of a mixed signal test system.

The windowing functions available in a mixed signal ATE system will accept a time sample array and return a windowed sample set array automatically.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Windowing Effects on Signals in the Time and Frequency Domains

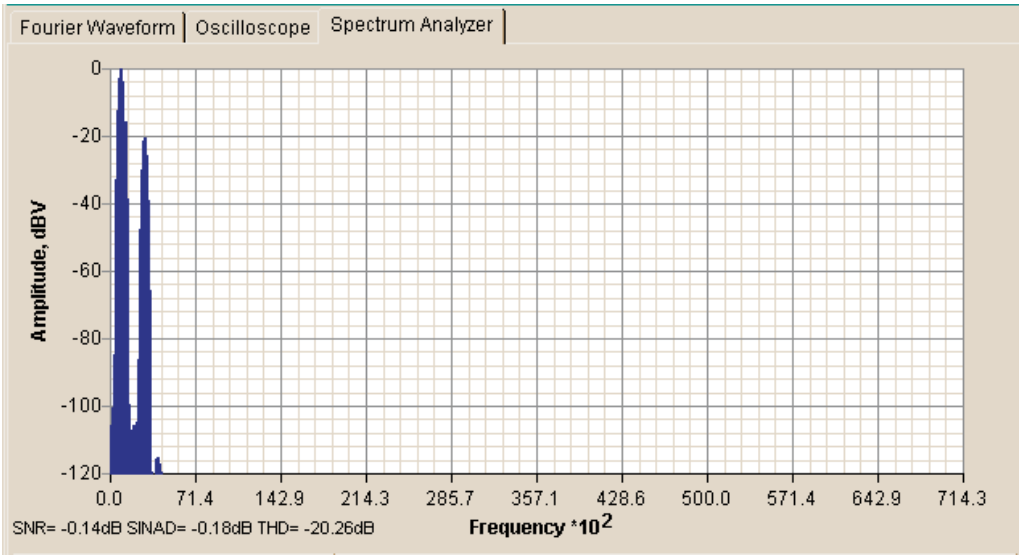

Figure 6.10: Windowing in the Frequency Domain. A Blackman-Harris window has been applied to the spectrum shown in Figure 6.9.

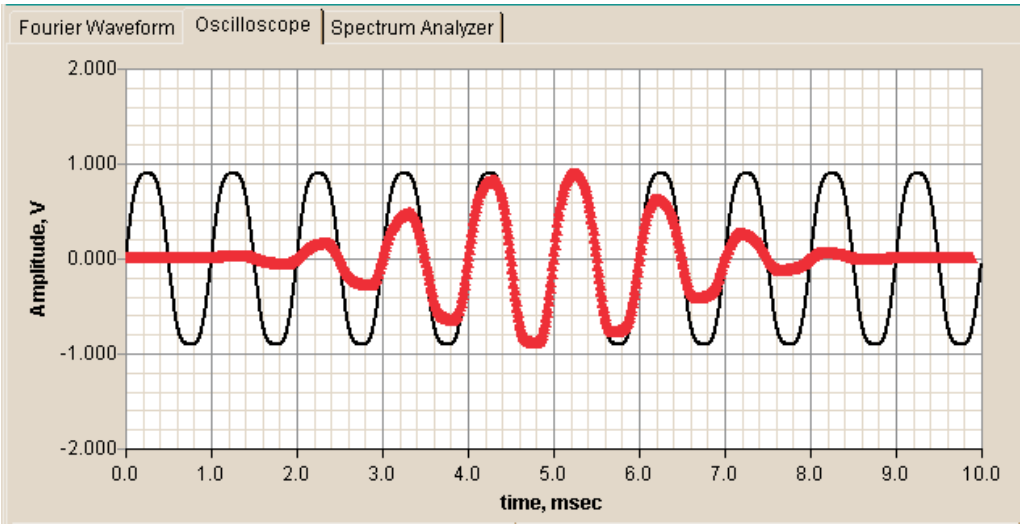

Figure 6.11: Windowing in the Time Domain. Note the reduction in amplitude at the end points, as well as in the middle of the waveform.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Sin(x) / x Amplitude Error

Sampling theory assumes that a given sample in time represents a finite amplitude with zero time width. In general, the output from a DAC or AWG remains constant between output value changes. This creates a waveform that can be modeled as a series of rectangular pulses of varying amplitude as shown in Figure 6.12.

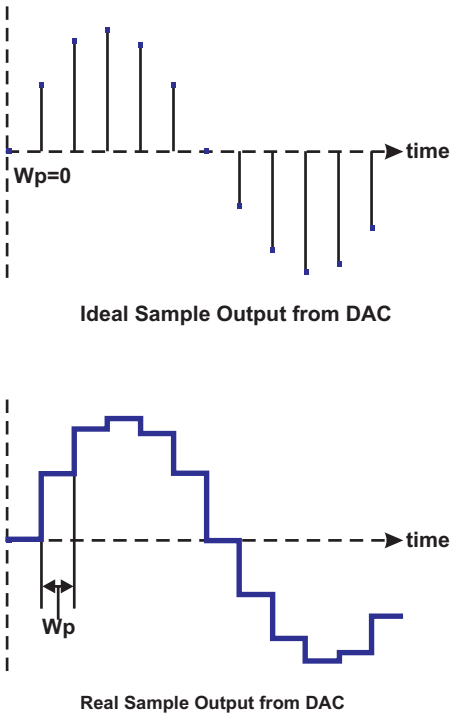

Figure 6.12: Sin(x) / x Amplitude Error from a DAC. The DAC provides a “zero order hold” function on the set of impulses. This time domain convolution results in frequency domain multiplication.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

When the frequency conversion math is performed for this series of rectangles, the frequency spectrum of a sine wave is multiplied by the spectrum of a rectangle, that has a *sin(x) / x* characteristic. The magnitude spectrum of a rectangle is shown at the top of Figure 6.13. The amplitude error versus frequency for the generated sine wave can be seen in the lower diagram. The error can be compensated with a filter or pre–compensated by modifying the data going to the waveform generator.

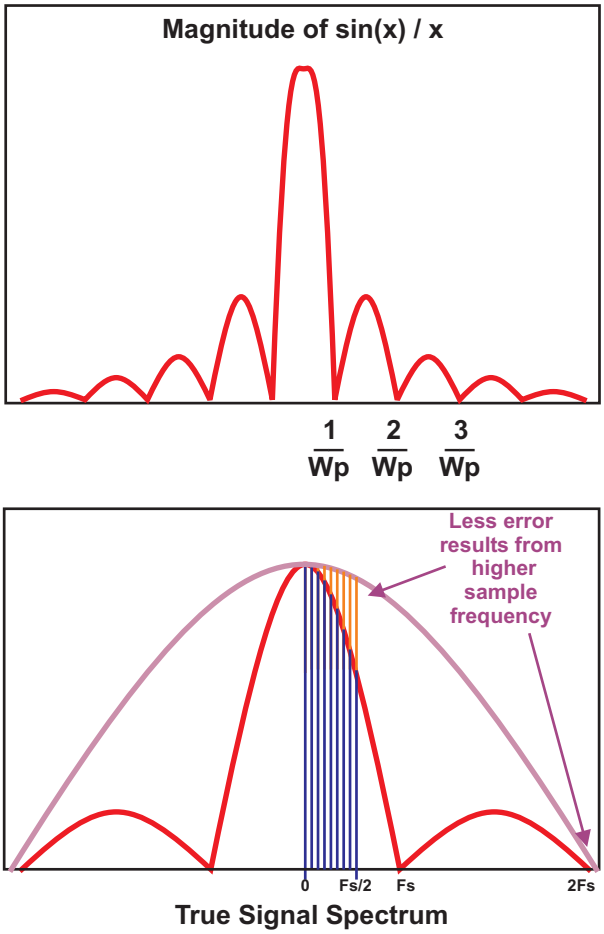

Figure 6.13: Sin(x) / x Amplitude Error. The top figure shows the FFT of a rectangular pulse. The bottom figure shows the results of multiplying pulses of various widths by a frequency spectrum of interest.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Notice that the pulse width in Figure 6.12 on page 6-18 is equal to the sample period or $1/F_s$, causing the frequency characteristic of the *sin(x)/x* curve to be zero at integer multiples of F_s. In a test situation, all frequencies of interest will be between DC and $F_s/2$. The error is an amplitude multiplication factor with bounds of zero and one, which is related to the ratio of the signal and sample frequencies. The higher the ratio F_s/F_t the less amplitude error as seen in the large diagram in Figure 6.13.

The multiplication factor is given by the *sin(x)/x* equation as related to F_s and F_t:

$$Multiplier = \frac{sin\left(\frac{\pi F_t}{F_s}\right)}{\frac{\pi F_t}{F_s}} \quad \textbf{(6.2)}$$

For any frequency component equal to the Nyquist frequency, the ratio of F_s/F_t will be two, and the multiplier value will be 0.6366 or (-3.9dB). The multiplier will be one at DC and zero for any frequency that is an integer multiple of F_s.

Figure 6.13 shows that higher sample frequencies yield less amplitude error. The effect is more clear in the time domain, when the samples are closer together; the stepped DAC output more closely approximates a true sinusoid. As F_s goes to infinity, the slope of the frequency domain *sin(x)/x* curve in Figure 6.13 near the frequencies of interest goes to horizontal, and the amplitude error goes to zero.

Truncation error

Digitizers do not have infinite resolution. If data is stored in a 16-bit word, it allows integer storage of quantities from 0 to 65535 (2^{16}). To calculate signal power, it is necessary to square the voltage, which will require more bits than the original value. When stored back into a 16-bit word, all bits that extend past the end of the 16-bit register are truncated. Although this is a simplified example of truncation error, it exists in any digitized signal. Digitized values are often represented using floating point values, which reduces truncation errors but does not eliminate them. Truncation errors are minimized by using double length variables and registers.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Quantization Error

Waveform digitizers, and ADCs in general, may be used to digitize signals. This process consists of two pieces: sampling and quantization.

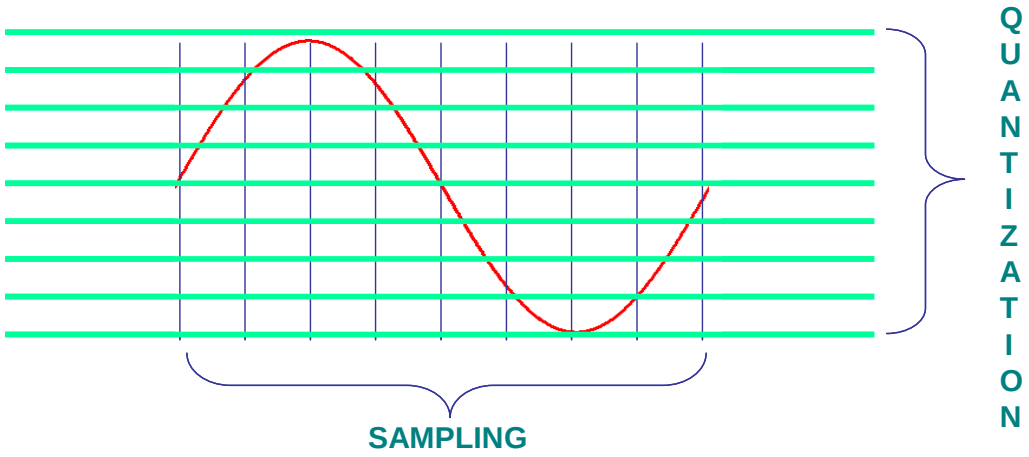

Figure 6.14: Digitization Process

The power of two to which an ADC can resolve a signal is the resolution of the digitizer, while the minimum value to which it resolves is one LSB. For example, an ADC which converts an analog signal to a 12-bit wide data word resolves that signal into one of 4096 values. A 12-bit ADC, with a ±5V full scale input range, has an LSB size of $10V/2^{12}$, or 2.44mV. This digitizer could not be used to directly measure an offset voltage of 100µV because it cannot resolve any voltage smaller than 2.44mV.

In this example, any analog signal change less than 2.44mV cannot be stored, or quantized, no matter how good the ADC. This results in an error known as quantization error, or Q. Q is bounded by ±1/2 LSB. It is equally likely to be any value between ±1/2 LSB.

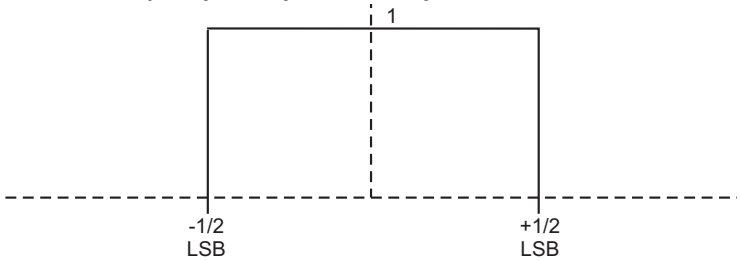

Figure 6.15: Probability Density Function of Quantization Error.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

The maximum signal to noise ratio of an ADC depends directly on the quantization error:[4]

$$6.02 \times bits + 4.77 + 20log\left(\frac{V_{in}}{V_{FS}}\right) dB = SNR \quad \textbf{(6.3)}$$

where V_{in} and V_{FS} are the ADC input signal and full scale input range as RMS values. If a sine wave analog input signal has a peak value equal to the ADC full scale input range, then the maximum (ideal) SNR of the digitizer is:

$$SNR = (6.02 \times bits + 1.76)\ dB \quad \textbf{(6.4)}$$

Thus you can expect to measure an incoming analog signal's SNR to no better than this ideal value for the digitizer. This also means that, to see the best test results, you must condition your analog signal such that it matches the full scale range of the ADC in the digitizer.

The concept of quantization error and the derivation of Equation 6.4 are continued in Chapter 8.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Slew Rate Error

Information can be lost in the digitizing process if the analog signal is moving too fast for the digitizer. The maximum rate of change of a sine wave occurs when t=0; the rate of change is given by its derivative. The derivative of $sin(\omega t) = \omega cos(\omega t)$; and when evaluated at t=0, $cos(\omega t) = 1$. Therefore, the maximum rate of change is (1 * ω), where $\omega = 2\pi f$. The maximum rate of change of a 10KHz signal is approximately 62.8V/ms. A digitizer with 12-bit resolution and a 5V full scale range has an LSB size of 1.22mV. To sample the sine wave at its fastest point, the conversion must be faster than 1.22mV/62.8V/ms or less than 20 ns; this exceeds 50MHz. Since track and hold circuits are much faster, they are used to sample signals, which moves the sampling task from the digitizer to the faster device.

Jitter Error

When digitized information is examined with a Fourier transform, the DFT/FFT algorithm assumes that the data was sampled at precise intervals with no time jitter. Any time deviation in the digitizer clock from its expected value causes an error. Jitter in the clock will distort the data the same way as if the signal had jittered. When a signal is digitized, only its amplitude information is preserved.

The data in the time sample array appears to the math algorithm as amplitude error, distorting the time waveform. Time jitter in a sampler distorts the frequency analysis results, which may appear as noise.

Suppose the sample timing is perfect and the input sine wave has no amplitude distortion, but jitters in time; the result is the same. In other words, ***jitter in either the sample timing or the analog signal timing can result in frequency distortion.*** If there is sample and signal jitter that is statistically independent, the problem is compounded even further. If there is sample and signal jitter that always moves exactly the same, the problem is hidden; this occurs when both the analog signal and the sample timing are created via a master clock.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Coherent sampling

Coherent sampling defines a way to guarantee that a sample set has a fixed, well defined relationship between the sample frequency F_s, the number of samples N, the test signal frequency F_t and the number of test signal periods sampled M. Coherent sampling restricts the timing relationships that exist between the test signal period and the sample period, making the timing setup for testing a particular device somewhat more difficult than with non–coherent sampling. It also requires a known waveform type, usually sinusoidal, as the input to a digitizer (ADC or WD) and the output of a generator (DAC or WG). If certain constraints are met, coherency also guarantees that the maximum amount of information about a particular waveform exists in the sample set (i.e. there are no duplicate samples).

Why Sample Coherently?

Coherent sampling is a luxury afforded to those in the ATE industry because we have complete control over the analog signal to be generated or digitized. Those in the communications industry do not have the option of coherent sampling and must wrestle with problems involving leakage, non-periodic signals, windowing and more. Thanks to coherent sampling, mixed signal ATE can have faster test times, less computation and better results without any real drawbacks. The only complication is that timing synchronization of the test signal and sample clock frequencies is required, increasing test development effort.

When digitizing a waveform, coherent sampling eliminates the need for any time windowing by guaranteeing that the sample set contains a complete, periodic waveform representation. When generating a waveform, there is no leakage because coherent sampling guarantees that there will be an exact integer number of sets of samples of signal cycles.

Frequency Bins

A FFT operation on a set of coherent samples taken in the time domain produces frequency amplitudes in the spectrum being tested; and puts all relevant information about the fundamental, its harmonics, and noise into specific, well defined frequency ranges of equal width. These frequency ranges are called bins. The width of a bin is determined by the relationship of the sample frequency to the number of samples taken. It is also related to the relationship of the test signal frequency and the number of test frequency cycles. This subject will be covered in greater detail later on in this chapter.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Coherency Formula Relationships

F_s, N, F_t and M

The relationship that creates a coherent sample set is that the number of samples N and the number of test cycles M must be whole positive integer values, and the sample parameters mentioned above must conform to the relationship given by Equation (6.5). In mixed signal testing, coherency means that a sample set contains an integer number of samples of an integer number of test signal cycles. No partial signal means no windowing is required. Coherent sampling also allows the most information to be collected about the test signal in the least amount of time.

The equation for coherent sampling is one of the most important relationships in DSP-based testing, and it is pretty simple:

$$\frac{F_s}{N} = \frac{F_t}{M} \qquad \textbf{(6.5)}$$

where

- F_s = Sample frequency.
- F_t = Test signal frequency.
- N = Number of samples taken. **N must be an integer.**
- M = Number of signal cycles over which samples are taken. **M must be an integer.**

Referring to Equation (6.5), note that N samples may be taken from a waveform over M test signal periods. If M = 1, all samples are taken in a single test signal period. If M > 1, the samples are spread over more than one test signal period. The total time required to take all samples is called the *Unit Test Period* (UTP) and requires M cycles of the test signal, which has frequency F_t.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Unit Test Period

A Unit Test Period (UTP) is defined as the time required to take all samples and is given by

$$UTP = \frac{M}{F_t} = \frac{N}{F_s} \quad \textbf{(6.6)}$$

Fourier Frequency

The resolution of the spectrum (the width of each frequency bin) is determined by the UTP of the sample set and is called the Fourier frequency (FF) or frequency resolution (F_{res}):

$$FF = F_{res} = \frac{1}{UTP} = \frac{F_s}{N} = \frac{F_t}{M} \quad \textbf{(6.7)}$$

Frequency Bins

Since the total number of frequency bins used for spectral analysis is *N*/2 and the resolution of the frequency bins is the Fourier frequency, we can state that the maximum frequency in the spectral data is $F_{tmax} = F_{res}*(N/2)$. With $F_{res} = F_t/M$, the fundamental test frequency F_t will appear in the M^{th} frequency bin.

Mutually prime N and M Assures Unique Sample Points

When taking samples of a sine wave over more than one period, it is possible to sample at the same place on the wave only in a different signal period. When duplicate samples are taken, no harm is done but it requires more test time and no additional information is gained.

To avoid duplicate samples, make M and N mutually prime. Mutually prime numbers means that the two numbers have no common factors other than 1. Neither can be divided by a common value other than 1. 16 and 27 are examples; 16 has 2 as its only factor and 27 has 3 as its only factor; note that neither 16 nor 27 is a prime number.

When using the FFT, N, as a power of 2, is only divisible by 2. If M is chosen as an odd number, it is not divisible by 2. **If an FFT algorithm is used, then N must be a power of 2, and choosing M as any odd integer will guarantee a coherent sample set with no duplicate points.**

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

A Coherent Sampling Example

Using the previous example for telecom, a 1020Hz signal is to be sampled at 8000Hz. Since the ratio of F_s and F_t is 400:51, the ratio for N and M must be the same:

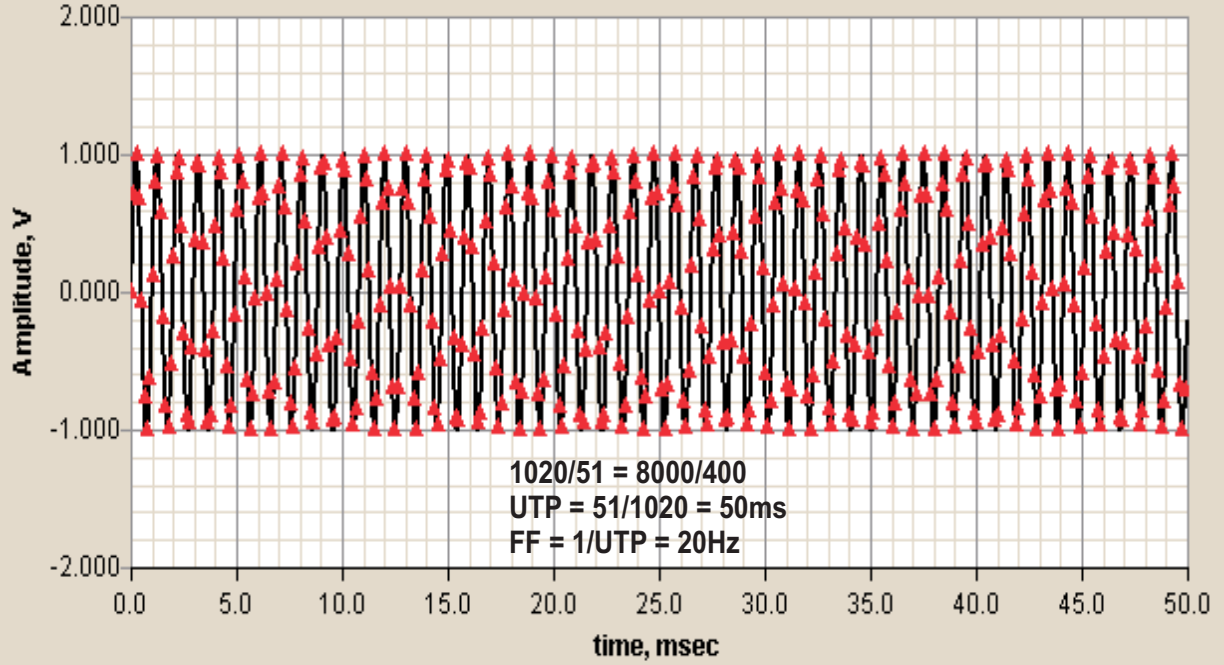

Figure 6.16: Coherent Sampled Waveform in the Time Domain.

The sample placement on the time domain waveform looks “funny”. This is because there are only about 8 samples per cycle of the waveform. This does, however, meet the Nyquist criteria for sampling.

Is this a good set of sampling parameters? For capture, it has one major drawback - the number of samples N is not a power of two. For waveform creation, this is not required, since an FFT is not executed.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

In the frequency domain, the spectral components are spaced equally at intervals of FF.

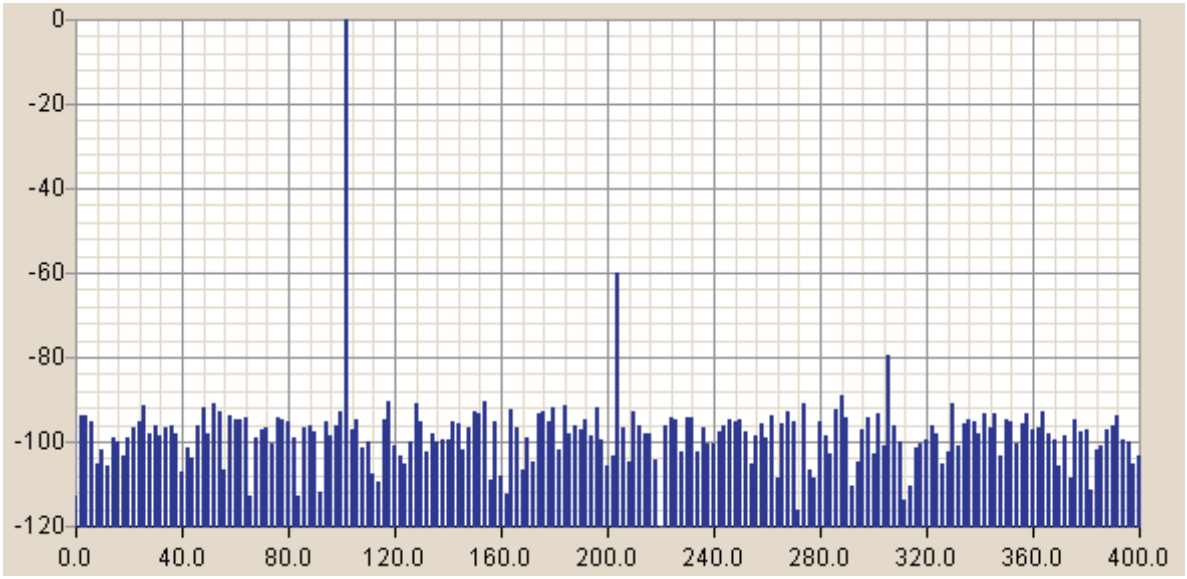

Figure 6.17: Coherently Sampled Waveform in the Frequency Domain.

You can see that the fundamental of 1020Hz is in the M^{th} bin. Since the FF is 20Hz, this is correct (51 * 20 = 1020Hz). Each frequency component represents the energy across a small bandwidth of 20Hz. If greater resolution is required, the number of bins can be increased, thereby decreasing FF.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Spectral Parameters

DFT and FFT Transforms

Figure 6.18 graphically displays a Fourier transform performed on a coherent sample set that will produce $N + 1$ spectral components called frequency bins. The first half of the spectrum contains $N/2 + 1$ points. A mirror duplicate of the first $N/2$ points was created in the second half of the spectrum, but it was discarded. Only the first $N/2 + 1$ bins are used. The first bin (Bin 0) is the DC component, which is not used in spectral analysis.

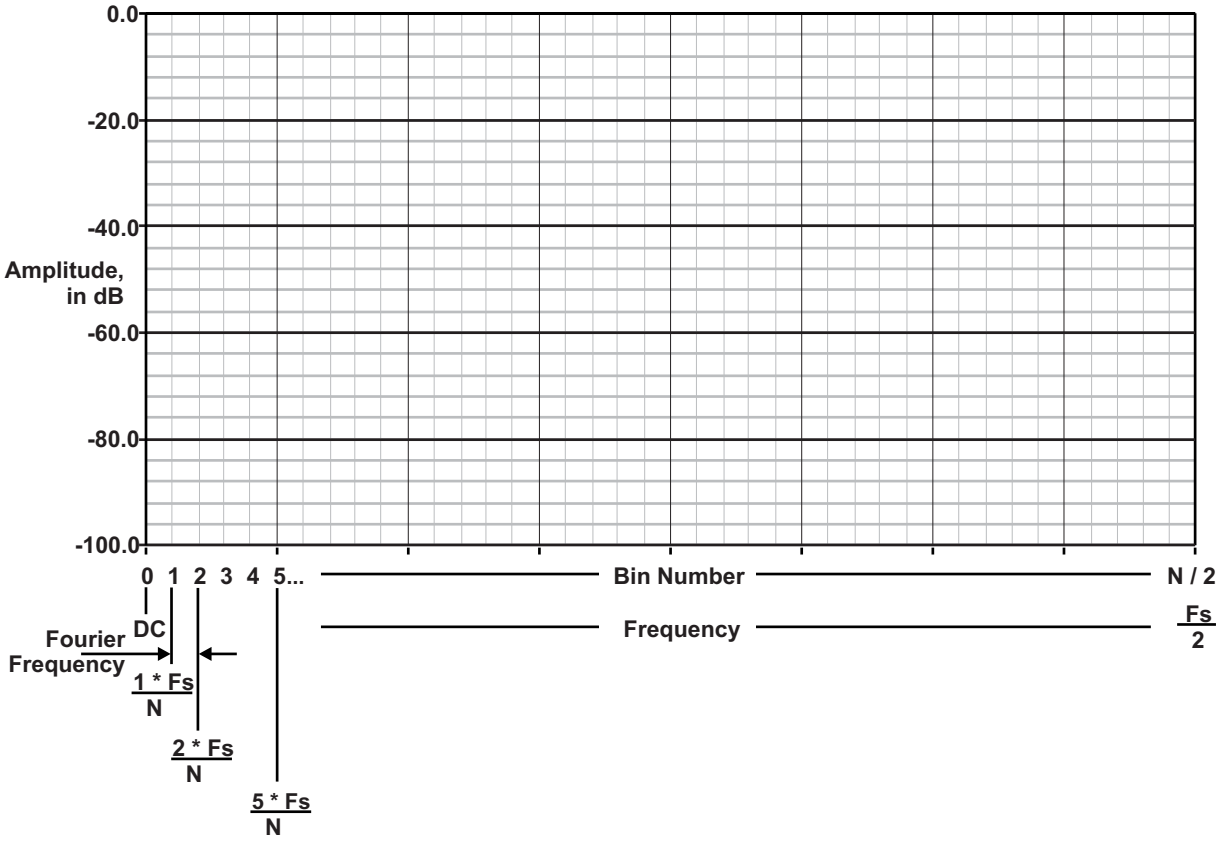

Figure 6.18: Frequency Axis Data from FFT. This figure shows a generic frequency spectrum, and how the Fourier Frequency (FF) is derived.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Digitizing samples

Sampling is not a complicated process. Only four things must be determined:

1. How many samples of a waveform are required (N)
2. How often the samples should be taken ($1 / F_s$)
3. Over how many cycles of the test waveform they should be taken (M)
4. Tester timing resolution

With these decisions made, put the signal into the digitizer, take the samples and store them.

Generating Time Samples

Often it is necessary to create data points for a waveform to be sent to a DUT. There are algorithms built into mixed signal ATE system's AWG for many of the common waveforms such as a sinusoid, square, triangle and other waveforms. If the required waveform is not available from the tester, it must be created. Two techniques that can be used to create waveform points are a software algorithm and an Inverse FFT algorithm.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Using a Software Algorithm

A computer algorithm to calculate the points and store them in an array is a fairly straightforward approach to creating a waveform. For example,

```
void MakeSinePoints(double Freq, SamplePeriod)
{
int N = 4096, i;
double WavePts[4096];
PI = 4 * arctan(1);
for (i = 0; i < N; i++)
    WavePts[i] = sin(2 * PI * Freq * (i * SamplePeriod)); // sin(2πft)
}
```

This is a good approach to generating waveform points when a single sinusoid or a tone that is the sum of two or more sinusoids is the desired waveform. The “sin” function returns values from -1 to +1 in amplitude, so it may be necessary to add some sort of scaling factor that each point is multiplied by. That factor depends on the required DUT input range and the WD output range.

This technique for generating a signal is also useful when testing a DAC, and is discussed in detail in *Using a Sine Wave Formula* on page 7-16.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Inverse Fourier Transform

The mathematics of the DFT and FFT algorithms are reversible; that is, a set of frequency data can be passed to an Inverse FFT routine, processed, and then a set of time samples are returned. This can be useful in some test situations, especially when generation of a complex waveform is required.

When the IFFT is most useful

If a circuit needs to be tested for bandwidth the input signal can have several forms:

Input	Requirements	Problems	Benefits
Many individual sine waves	Sine waves must be sent to the DUT one at a time, each at a different frequency, to test the entire filter bandwidth.	Each sine wave must be generated, the filter must be allowed to settle. Output must be digitized. Slow test	Easy to program Outputs are easy to digitize.
A single swept wave	A frequency modulated waveform must be created that tests the entire filter bandwidth.	The FM wave may be difficult to create and coherently sample.	Fast test.
A single signal that contains many sine wave components	A tone waveform must be created that contains all the necessary frequency components to test the entire filter bandwidth.	Coherent sampling of output waveform may be complicated depending on how many sine terms are summed.	Fast test Easy to create filter input signal using Inverse FFT

Table 6.2: Complex Waveform Generation.

As noted in the last column of the last row in Table 6.2, it is easy to create a tone containing multiple frequency components using an Inverse FFT. Fill an array with frequency data having very small values (e.g. -160dB) for the frequency points with no amplitude, and large values (e.g. 0dB) for frequency points at the tone components.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

How to use an Inverse FFT (IFFT) algorithm

The use of an IFFT algorithm is virtually identical to using a forward FFT. You fill an array with frequency data, pass it to the IFFT and an array of time samples is returned. Consider a filter under test that requires a bandwidth test for -3dB points at 1KHz, 2KHz and 4KHz. If we choose a 16-point IFFT that covers the frequency range of 0 to 10KHz, then each frequency bin will cover 10K / 16 = 625Hz. Table 6.3 on page 6-34 shows a frequency array that could represent the data points for a 16 point IFFT for this filter. Recall that the Frequency Bin values represent the array subscript for the values in the frequency array. The Amplitude Value is what we put into the array to pass to an IFFT function.

Limitations of using the Inverse Fourier Transform

Both the IFFT and the IDFT have limitations. One is that the amplitude of the returned data must be scaled; it does not equal anything in particular, and the peak value depends on the specific algorithm used.

Another limitation relates to the frequency points sent to the IDFT/IFFT. The only frequency points that can be represented are those of a specific frequency bin. By using the inverse transform when it makes the most sense, this limitation is normally not a problem.

Finally, recall that the Fourier transform assumes that the time data is periodic. Thus when passing in a set of frequency points, an inverse transform will return points for a complete waveform period; no partial period data can be created with an IDFT or IFFT.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Inverse Fourier Transform Bin Array

Notice that the fundamental contained in Bin 1 is given an Amplitude Value = 1. This will establish the 0dB level and all other values are considered to be relative to this level.

Frequency Bin (Array Position)	Frequency (Hz)	Amplitude Value
0	DC	10^{-8}
1	625	1
2	1250	10^{-8}
3	1875	1
4	2500	10^{-8}
5	3125	10^{-8}
6	3750	1
7	4375	10^{-8}
8	5000	10^{-8}
9	5625	10^{-8}
10	6250	10^{-8}
11	6875	10^{-8}
12	7500	10^{-8}
13	8125	10^{-8}
14	8750	10^{-8}
15	9375	10^{-8}

Table 6.3: Frequency Bins.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

Key points of this chapter

- Sample frequency must be more than twice the maximum frequency of interest
- Use an FFT to process samples if possible; it is much faster than a DFT
- Using an FFT requires that the number of samples N be a power of 2
- Coherent sampling requires M be an integer
- Coherent sampling with mutually prime N and M gives the fastest possible test time with the most amount of information, and does not produce leakage
- Leakage is caused by a sample set with an incomplete period, which can be reduced using a windowing function
- When generating signals with a DAC, a *sin(x)/x* amplitude error occurs. It can be compensated by calculating the error.
- An inverse Fourier transform can generate a set of time samples from a set of frequency data
- Using IFFT to generate time samples is best when multi–tone signals are required

References

1. C. E. Shannon, “Communication in the Presence of Noise,” *Proceedings of the Institute of Radio Engineers*, Vol. 37, 1949, pp 10.

2. Matthew Mahoney, *DSP Based Testing of Analog and Mixed Signal Circuits*, IEEE Computer Society Press, 1987, pp 37.

3. *Understanding Digital Signal Processing*, Richard G. Lyons, Addison Wesley Longman, Inc., 1997, pp 86.

4. Ibid, pp 362.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.

NOTES:

Copyright© Soft Test, Inc. All Rights Reserved.