1-0 WHAT IT IS

Today's world functions almost totally on decisions that are based on the gathering of data and its subsequent analysis. A child puts a foot into the water to determine the temperature before diving in. Life insurance firms gather huge volumes of actuarial data to determine risks and to establish premiums. There are nearly as many systems for acquiring data as there are types of data. This text, however, confines itself to a particular class of data acquisition system, defined as:

an electronic instrument, or group of interconnected electronic hardware items, dedicated to the measurement and quantization of analog signals for digital analysis or processing.

In this context, the data acquisition system (DAS) is the analog interface to the digital world. A graphic representation of where the data acquisition system "fits in" is found in Figure 1-1. Once the parameter to be measured is translated into the analog-electrical domain, the DAS performs the translation to the digital-electrical domain. In some cases the DAS simply records, or stores, the digital data, while more
sophisticated systems may be capable of analysis or further processing. For instance, a DAS may be as simple as a digital voltmeter (DVM), which displays its output as a decimal readout, or it may be complex enough to contain a large-scale computer as part of its hardware.

Now that we understand the function of this type of system, let's examine the components of a basic data acquisition system, as shown in the block diagram in Figure 1-2. Each of the blocks in this figure represents a particular data acquisition function, and each is defined in general terms as follows:

**Analog Multiplexer**

The analog multiplexer permits a number of signal sources to be automatically measured by the same data acquisition hardware. It consists of a series of switches whose inputs are tied to the various analog signals and whose outputs are tied to a common measuring point. Each input is individually connected to the measuring point in a predetermined sequence. The number of channels in a multiplexer may vary from two to several hundred.

**Signal Conditioning**

Very often the signals presented to the inputs of the data acquisition system are not in a form appropriate for conversion, and so they must be preconditioned. The required signal conditioning could consist of linear amplification, logarithmic amplification, filtering, peak detection, or sample-and-hold. Often more than one of these functions is required. For instance, it is not uncommon to combine amplification with filtering or to find a low-level amplifier before a sample-and-hold.

**Analog-to-Digital (A/D) Converter**

The analog-to-digital converter actually translates the analog signal into an encoded digital format. Of the numerous ways to perform this function, only about a half-dozen techniques have found wide acceptance. Most notable are the dual-slope integrating and the successive approximation converters. A/D converters are often referred to by the number of output digits they produce. In a binary system, the range is from 4 to 16 bits, while in a binary-coded decimal system, 3 to 4 digits are normal.

**Digital Clock**

The digital clock provides the master timing for the data acquisition system. It may be as simple as a multiphased crystal controlled oscillator, or it may provide the user with a wide selection of multiphased rates and modes of operation. Some systems also contain both time-of-day and day-of-year clocks.

**Manual Data Entry**

Many data acquisition systems provide users with a way to tag the data they are accumulating, through some sort of manual data entry. They may wish to note such things as the type of data, special conditions surrounding the measurements, and/or the run number. Such infor-
mation is generally provided in a “header” that precedes the actual measurement data.

Digital Buffer
The ability to record discrete events is often a requirement of a data acquisition system. Since these events are usually accompanied by the opening or closing of a switch, they represent a digital input. Out-of-tolerance conditions or other situations that might invalidate the data collection are most often designated as “discrete events.”

Output Buffer
The output buffer acts as the data collector for the DAS. In an ordered sequence, it gathers up such data as the multiplexer channel number, the signal conditioner gain, A/D converter data, manual data, clock information, and discrete events. The buffer combines the data with the proper format for entry into the recording or processing system. It also provides the proper buffering and control to interface with the recording or processing device.

If the processing device were a minicomputer, the output buffer might be called a “peripheral controller.”

Recording/Processing Device
A number of different equipment types can fill the role of the recording or processing device. Some of the equipment types commonly used are paper tape punches, teleprinters (TTY), magnetic tape units, line printers, cathode-ray tube displays, floppy disks, general-purpose digital computers, and special-purpose digital processors.

In recent years, all these components of the data acquisition system have become more automated; a typical block diagram can be seen in Figure 1-3. All the major functions of Figure 1-2 can be found with the exception of manual entry, which is accomplished directly by the computer through the keyboard display. Clock information is supplied by the computer, and timing is generated in the input/output (I/O) controller. This system configuration has two distinct advantages: first, on-line programmable processing is possible; and second, a host of storage media is available. The rapid expansion of the data acquisition field is directly attributable to the advent of low-cost computer hardware, which makes systems like that of Figure 1-3 a reality. Yet we should not overlook the dotted-line box in the lower right-hand corner of that figure: Software can easily become a more costly burden than the hardware.

The latest trend in test instrumentation is to embed a micro-
processor in the heart of the equipment, as shown in Figure 1-4. This places tremendous programming power and flexibility in the hands of the user.

Other data acquisition configurations are also worthy of note. Figure 1-5 shows a system that utilizes a variety of analog-to-digital converter types. Such a system would apply to a situation in which the test data comes from a variety of different sources. The output buffer must also perform the function of digital multiplexing.

![Figure 1-4. Microprocessor-Controlled DAS](image)

A totally different configuration is found in Figure 1-6. Here each channel has its own signal conditioning and A/D conversion. This system concept is usually employed when the analog signals come from a number of very diverse sources and/or when high calibration accuracy is required. Each channel may be separately aligned. A few years ago, this approach would have been an extremely expensive proposition. With the decrease in the cost of DAS hardware, however, it is rapidly becoming more and more viable.

![Figure 1-6. DAS with Conditioning and Conversion for Each Input](image)

Having carefully examined the data acquisition system itself, let's turn our attention to some of its potential uses. Although any hardware system that has both an analog and a digital nature will have a DAS embedded in it somewhere, we will restrict our discussion to four broad areas. They are:

1. data logging,
2. signal analysis,
3. automated factory testing, and
4. process control.

## 1-1 DATA LOGGING

The data logger is a data acquisition system that measures the analog inputs, translates the results into the digital domain, and stores the data for future processing or analysis. Figure 1-2 depicts the typical data logger system block diagram. From the earliest days of recorded
data, scientists have set up experiments with appropriate analog measuring devices at critical points in the operation. They have tabulated their data either by visual observation at set intervals or by recording it on another analog device, such as a pen recorder. In either case, they had to spend many hours analyzing the results. The data logger provides an automated method of making the measurements and recording the data. It can perform these functions at precise intervals and with a degree of accuracy beyond any person's physical capability. Because the data is stored in digital format, the data analysis is easily performed by an off-line computer. Since many data loggers are highly portable, data-taking can be automated even in very remote locations.

Most logging systems require very little attention. After the experiment is set up and the data logger starts, the operator may leave the measurement site. The data logger will carry out the measurement plan precisely as the operator set it up, and it will shut itself off when the experiment is complete. As long as we continue to explore the world around us, there seems to be no limitation for this type of hardware.

1-2 SIGNAL ANALYSIS

Many processes in our world today are characterized by time-varying signals from which much useful data may be extracted. Examples of such processes are signature analysis, radar tracking and warning, air traffic control, electronic countermeasures, seismic exploration, and patient monitoring. The most common way to process such information is to translate it into an appropriate Fourier series or transform for analysis in the frequency domain. Low-cost computer and processing hardware, coupled with computationally efficient fast Fourier transforms (FFT), has made this approach feasible. Since the FFT algorithms work with discrete signal samples represented in digital form, the front end of a system designed for signal analysis must be a typical data acquisition system. The growth of this area is bounded only by a few things: the cost of computer storage, along with the operating speed of computer and data acquisition hardware.

1-3 AUTOMATED FACTORY TEST EQUIPMENT

Large-volume manufacturing has provided another lucrative area for the data acquisition system. With an increase in production quantities, testing time—as well as the burden of training a large number of qualified test personnel—must be reduced. A desirable test system would require only inserting the product into a fixture, setting at most two or three simple switches, and depressing a test button. The test set would automatically exercise the product, quantize the results, compare the results with the desired answer, and activate a go or no-go signal as required. Hard copy should be available if necessary.

The heart of such a test system is, of course, our old friend the data acquisition system. Automated test systems often include a small, general-purpose digital computer. The computer provides inputs to exercise the product, stores or records the test information, and examines and analyzes the results. In some factories, the test system is so sophisticated that several test positions containing small computers are tied together by a larger central processor, which stores and analyzes data from and maintains the status of all test positions.

1-4 PROCESS CONTROL

Several years ago, process control systems were essentially human-controlled. Operators established the process parameters by means of analog set-point controllers, and they monitored results by means of visual recorders and annunciators. Simple, fixed control algorithms were generally used, and only a limited number of points could be monitored and controlled.

Today, the data acquisition system has become a way of life in most types of industrial process control. The first data acquisition system applications to process control came when a computer was added to monitor the process. Not only could the computer now monitor several hundred inputs, but it could also display the readings, make calculations, and provide correction data to the operator. This concept still required operators to take action, based on the computer output, but it gave users a comforting feeling that their process was not subject to the whim of a piece of “unreliable” computer hardware.

As computer hardware has become more efficient and reliable, the process control system has evolved into that pictured in Figure 1-7. The DAS and computer are clearly parts of the control loop as it measures the key parameters, makes reference comparisons, computes the control algorithm, and provides feedback correction to the process. The operator is now the monitor—interrogating the system through the operator console, checking the computer-displayed indicators, and answering alarms.

1-5 WHAT COMES NEXT?

The remainder of this text is designed to acquaint you with the principles of data acquisition systems. Chapters 2 and 3 cover the basics of sampling and quantization theory. Chapter 4 describes the basic DAS
Basic Sampling Concepts

2-0 INTRODUCTION

Fundamental to the operation of a data acquisition system is the concept of sampling, which may be defined as:

the act of measuring a continuous function at discrete time intervals.

Signals that represent some physical or analog parameter are converted to a series of discrete values. Unfortunately, many instrumentation engineers are content when they think they understand only the rudiments of the sampling process. As it turns out, since the data acquisition system is a sampled-data system, the overall system transfer function must be examined to achieve a proper solution.

A typical data acquisition/sampled-data system is shown in Figure 2-1. As the diagram indicates, four major processes are involved:

1. sampling,
2. quantization,
3. digital processing, and
4. recovery.
A properly designed system generally requires balancing the implementation of one of these processes with the characteristics of the others. For instance, the sample rate may be a compromise between the complexity of the sampler and the sophistication of the recovery filter. If the required accuracy and/or system dynamic range is high (a large number of quantizer levels), both the sample rate and the recovery process may be affected. The required number of quantization levels may also be dictated by the digital processing algorithm.

This chapter considers the fundamentals of sampling, while Chapter 3 is devoted to quantization and signal recovery. To round out the discussion, Chapter 3 ends with a practical example and a summary of design considerations.

Besides the required number of bits (or quantization levels), the only major problems in a data acquisition system involve first the proper conversion of analog signals to digital form, and then the subsequent reconstruction of the original signal, or its derivative, back to analog. (Consideration of the various digital processing algorithms, however, is beyond the scope of this text.)

### 2.1 Kinds of Signals

Before applying the concepts of sampling, quantization, and recovery, you must have a reasonable understanding of the limitations of the signal to be processed. Is the signal dc, quasi-dc, or dynamic? Is it deterministic or random? What are the dynamic range, the frequency range, and the noise characteristics? What is the shape of the frequency spectrum? These questions, along with the information to be acquired, are fundamental to the design of a data acquisition system.

We will now examine some of the signal types that might be encountered.

#### 2.1.1 Static or dc Signals

A static or dc signal is one that is stable or not changing. The first question you might ask is, "Why would you want to continuously measure an unchanging signal when one measurement would be suffi- cient?" In many systems the dc power source(s) may require monitoring. Generally, the measurement system records the same constant value over and over. Yet, should the voltage drop out of tolerance, an alarm is triggered so that corrective action may be taken. Often, this situation is compounded when an undesirable noise signal is superimposed on the dc signal. The results depend greatly on the type of measurement hardware and the occurrence of the noise, relative to the measurement point. Should the measurement be made coincident with a noise peak, the reading could deviate drastically from the expected result.

One solution to this problem is signal averaging. In this approach, a large number of measurements of the combined dc/noise signal are made, and their sum is divided by the number of such measurements. Thus the signal-to-noise ratio is improved by a factor equal to the square root of the number of measurements. If \( n \) measurements are made:

\[
\frac{S}{N}_{\text{new}} = \sqrt{n} \frac{S}{N}_{\text{original}} \tag{2-1}
\]

As the noise always has a finite bandwidth, each noise sample has some dependence on previous values. To realize the full advantage of averaging, allow enough time between samples to ensure that they are truly random. This waiting time between signals is inversely proportional to the system bandwidth.

Another alternative for noise removal is low-pass filtering. As we know, however, filters become very complicated as the noise rejection increases. Very often, the concepts of averaging and filtering are combined. Through trade-offs between filter bandwidth and sample rate, an optimum system solution can be achieved.

One thing becomes obvious at this point. Although we are trying to eliminate the noise rather than examine it, we must know the noise bandwidth and other system constraints and limitations.

#### 2.1.2 Quasi-dc Signals

A quasi-dc signal is a normally dynamic signal that does not change during the measuring period. Two examples of this type of signal are a flat-top pulse, whose amplitude is of interest, or possibly the output of an analog multiplexer. Figure 2-2 is one representation of this type of signal. Once the signal has reached the static state, all the concepts for measuring dc signals hold. Care must be exercised, however, when applying a low-pass filter to this type of measurement. System band-
2-1 KINDS OF SIGNALS

The constants \( a_n \) and \( b_n \) are found by the integrals

\[
a_n = \frac{2}{T} \int_{T/2}^{T} f(t) \cos nw_1 t \, dt, \quad n = 0, 1, 2, 3, \ldots
\]

and:

\[
b_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin nw_1 t \, dt, \quad n = 1, 2, 3, \ldots
\]

(2-3)

(2-4)

From these expressions, both an amplitude-versus-frequency and a phase-versus-frequency plot may be constructed. The amplitude plot will be a plot of \( \sqrt{a_n^2 + b_n^2} \), while the phase plot will be a plot of \( \tan^{-1}(-b_n/a_n) \).

Take the following exponential functions:

\[
\cos nw_1 t = \frac{e^{jnw_1 t} + e^{-jnw_1 t}}{2}
\]

and:

\[
\sin nw_1 t = \frac{e^{jnw_1 t} - e^{-jnw_1 t}}{2j}
\]

(2-5)

(2-6)

and substitute them into expression (2-2). You thus develop the compact form of the Fourier series:

\[
f(t) = \sum_{n=-\infty}^{\infty} C_n e^{jnwt}
\]

(2-7)

\[
C_n = \frac{1}{T} \int_{T/2}^{T/2} f(t) e^{-jnwt} dt
\]

(2-8)

The expressions of \( f(t) \) and \( C_n \) form a transform pair. The Fourier coefficient \( C_n \) is a complex number and is:

\[
C_n = a_n - jb_n = \sqrt{a_n^2 + b_n^2} e^{i\theta_n}
\]

(2-9)

where

\[
\theta_n = \tan^{-1} \frac{-b_n}{a_n}
\]

(2-10)

and the magnitude of \( C_n \) is:

\[
|C_n| = \sqrt{a_n^2 + b_n^2}
\]

(2-11)
Obviously, the plots of $|C_n|$-versus-frequency and $O_n$-versus-frequency are the same plots described by the less compact form of the Fourier series. Both plots are discrete line plots in the frequency domain.

The simplest periodic waveform is, of course, a single sinusoid. Both the time and frequency domain plots are shown in Figure 2-3. There is only one frequency domain plot, a single-amplitude spectral line at a frequency that represents the single sinusoid. Not terribly interesting, but it does point out that even the simplest of waveforms have representations in both domains.

![Figure 2-3. Time and Frequency Domain Representations for a Single-Frequency Signal](image)

Figure 2-4 shows a more interesting signal waveform that consists of several harmonically time-locked sinusoidal signals. The frequency-domain magnitude plot contains a dc component and three other discrete frequency components at $f_1$, $f_2$, and $f_3$. Their phase relationships are shown on the lower frequency domain plot. Note that the plots are in the form of discrete line spectra, which characterize periodic waveforms.

Perhaps the most important periodic waveform, from a data acquisition point of view, is that of Figure 2-5. The rectangular pulse train shown is typical of a real, finite-width sampling function. The resultant frequency domain function is:

$$C_n = \frac{4\tau \sin n\omega_1 \tau / 2}{T} \sin n\omega_1 \tau / 2$$  \hspace{1cm} (2-12)

As it turns out, $C_n$ is real and there is no need to plot $C_n$ and $\Theta_n$ separately. As Figure 2-5 indicates, the Fourier series transform produces a spectrum that is symmetrical about the $f = 0$ axis. Although the

![Figure 2-4. Typical Periodic Function](image)

"negative" frequencies are fictitious, they will prove useful when the effects of sampling are considered. Should the pulse train repetition rate increase ($T$ decreases), the number of spectral lines decreases. As the pulse width ($\tau$) decreases, the zero crossings move out, thus extending the frequency content over a wider frequency range.

Figure 2-6 carries this concept to its limit. In this case, the time-domain function is a sequence of unit impulse functions, where a unit impulse is defined as a pulse that has infinite height and zero width. The area of the pulse, however, is unity. With an infinitely small $\tau$, the first zero crossings in the frequency domain are at $\pm\infty$. The spectral lines are of equal amplitude as the energy is distributed evenly across the spectrum. An impulse pulse train in the time domain is then represented by an impulse spectra train in the frequency domain. This characteristic will be utilized later when explaining the effects of narrow pulse sampling.

As we know, the transient signal is an aperiodic signal that reduces to zero after some finite interval of time. If we consider the transform pair of (2-7) and (2-8) and allow $T$ to approach infinity (only one pulse or transient, ever), then the fundamental frequency, $\omega_1$, will approach zero and a new transform pair is created:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega \hspace{1cm} (2-13)$$

and:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \hspace{1cm} (2-14)$$

Now as a Fourier integral representation, the ability to transfer
2-1 KINDS OF SIGNALS

back and forth from the two domains is once again preserved. An example of the application of the Fourier integral is seen in Figure 2-7. The time domain function is a single square pulse centered about the y-axis. Although the result is still of the form (sin x)/x, something interesting has occurred. The plot is not a series of discrete line spectra as we found with the pulse train of Figure 2-5. As T has gone to infinity, 1/T goes to zero, and all possible line spectra are present. The result is a closed spectrum, with all frequencies contained. Once again, the "negative" frequencies are produced by the transformation.

The reverse situation occurs with a "box" filter in the frequency domain, where the frequency spectrum looks the same as the single-pulse time function of Figure 2-7. The resultant time plot is, of course, of the form (sin x)/x.

Figure 2-7. Single Pulse with Frequency Spectrum

Random Signals

Unlike deterministic signals, the exact value of a random signal may not be predicted in advance. Since most signals we must deal with fall into this category, the concepts of probability and statistics play a
major role in their analysis. Using statistical techniques, we can compute the average or dc value, the dc power and ac power, and the total power. We can also perform signal averaging, signal correlation, or a complete frequency analysis.

As the random signal is not repeatable, its time function is aperiodic. Its frequency plot is therefore a "closed" spectrum, as shown for the band-limited signal of Figure 2-8. The spectrum shown is fairly typical of signal spectra we might encounter in the everyday use of the data acquisition system.

A deterministic signal may sometimes appear random, such as if it were buried in noise. First, you would perform random analysis, utilizing such techniques as signal averaging or correlation, to extract the deterministic signal. When that process is complete, the signal can then be analyzed by normal deterministic analysis.

![Band-Limited Nonperiodic Signal](image)

**Figure 2-8.** Band-Limited Nonperiodic Signal

### 2-1.4 Convolution

One other concept is important to a basic understanding of sampling: the principal of convolution. The time domain convolution integral is expressed as:

\[
g(\tau) = \int_{-\infty}^{\infty} f(t)h(\tau - t)dt \quad (2-15)
\]

It is concerned with displacement, multiplication, and integration. The concept is most often used to analyze a circuit output response to a given input stimuli.

In the example found in Figure 2-9, the square-pulse input, \( f(t) \), is applied to a simple RC circuit whose impulse function is the reverse exponential, \( h(t) \). The convolution integral of (2-15) could also be written as:

\[
g(\tau) = \int_{-\infty}^{\infty} f(t)h(-t + \tau)dt \quad (2-16)
\]

The functions \( h(t) \) and \( h(-t) \) can be seen in Figures 2-9b and 2-9c. We see that \( h(t) \) is obtained by folding \( h(-t) \) about the zero-time axis. The function \( h(-t + \tau) \) represents the function \( h(-t) \) shifted by \( \tau \)-seconds along the time axis, as seen in Figure 2-9d. The resulting effect of convolution (see Figure 2-9e) is to move the reverse impulse response curve, \( h(-t + \tau) \), across the time function, \( f(t) \), while evaluating the shaded area for each new value of \( \tau \). The net result is the lower part of Figure 2-9e, the summation of all the shaded areas.

It is also important to realize that convolution in the time domain is equivalent to multiplication in the frequency domain, and:

\[
\int_{-\infty}^{\infty} f(t)h(t)dt \quad \leftrightarrow \quad F(\omega)H(\omega)
\]

**Convolution**

**Multiplication**

\[ (2-17) \]
The converse is also true, as:

\[ f(t)h(t) \leftrightarrow \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)H(W - \omega)d\omega \]

**Multiplication**  **Convolution**  

(2-18)

As one last example, let’s examine the convolution of the frequency spectrum of Figure 2-10a with the impulse function of Figure 2-10b. As we see, both frequency spectra are perfectly symmetrical about zero, and, therefore, \( H(W - \omega) \) is exactly \( H(\omega) \). Evaluation of the integral may be easily accomplished visually. As the impulse spectrum, \( H(\omega) \), passes across, or is convolved with, the \( f(\omega) \) spectrum, a completely new spectrum is imprinted, symmetrically about each impulse.

### 2-1.5 Important Things to Remember

Before proceeding to the concepts of sampling, note the following points:

1. There are three basic kinds of signals: dc, quasi-dc, and dynamic.
2. Dynamic signals have both time and frequency domain representations.
3. We may readily transfer from one domain representation to the other and vice versa.
4. Multiplication in the time domain is equivalent to convolution in the frequency domain and vice versa.

### 2-2 THE SAMPLING FUNCTION

In Section 2-0, we defined sampling as the act of measuring a continuous signal at discrete time intervals. So we could consider the sampler as a simple switch. When the switch is closed, it is equivalent to multiplying the input signal by unity for the duration of the closure period. The results of sampling with a unit-amplitude impulse train is shown in Figure 2-11, where the signal, \( f(t) \), is sampled by \( h(t) \). The output is the impulse train, \( f^*(t) \), whose amplitudes trace out the envelope of the original signal, \( f(t) \).

In actual practice, hardware constraints place severe limitations on our ability to achieve a true impulse sample function. For ease of understanding, however, we will utilize the impulse function to explain the basic sampling concepts, describing the added effect of finite-width sampling near the end of the chapter.

So the sampling function consists of multiplication in the time domain, as defined by the expression:

\[ f^*(t) = f(t)h(t) \]

(2-19)

We also know that multiplication in the time domain is convolution in the frequency domain. The effect of sampling in both domains is shown in Figure 2-12. From the information presented in the previous sections, we should not be surprised at the results. As seen in Figure 2-12a, the random time function translates to a closed spec-
BASIC SAMPLING CONCEPTS

\[ f(t) \xrightarrow{h(t)} f^*(t) = f(t)h(t) \]

\[ f(t) \]  
\[ h(t) \]  
\[ f^*(t) \]

**Figure 2-11.** The Impulse Sampler

pectrum in the frequency domain, while, as Figure 2-12b shows, the impulse train is an impulse train in both domains. Finally, in 2-12c, we find that the amplitude-modulated sampled time function, \( f^*(t) \), appears as a series of spectra, as the frequency spectrum, \( F(\omega) \), is convolved with the impulse function, \( H(\omega) \). Each of the spectra of \( F^*(\omega) \) is centered about multiples of the sample frequency, \( f_s = 1/T \). Note that these multiple spectra are present in the sampler output and can become a problem later if complete signal recovery and reconstruction are required.

We learned in Section 2-1.3 that periodic functions in the time domain become sampled spectra (Figures 2-4 and 2-5) in the frequency domain. Now we find that sampled functions in the time domain translate to periodic spectra in the frequency domain.

**2-3 THE SAMPLING THEOREM**

![Image showing the sampling theorem](image)

Our discussion of sampling would not be complete without reference to the sampling theorem, which states that:

Signals with a finite bandwidth of \( f \)-hertz can be completely described by sampling the time signals at instants separated by \( T = 1/f_s \) seconds.
In simple terms, the signal must be sampled at a rate at least twice as high as the highest frequency in the spectrum. This concept is shown in Figure 2-13, where the sinusoid is sampled at its positive and negative peaks. The theory then assumes that the resultant impulse sample train could be passed through an ideal "box" filter and the original sinusoid frequency reproduced. The amplitude of the filter output depends on where the original sinusoid is sampled.

\[ f(t) = A \sin \omega t \]

Figure 2-13. Basics of the Sampling Theorem

Since the sampling theorem defines the limit for an absolutely perfect situation, it has a number of serious problems. First, suppose we were so unlucky as to choose sample points that fall precisely on the zero crossings. This case, shown in Figure 2-14a, would, of course, produce zero output from the sampler. While the case is trivial, the problem is fundamental.

The second difficulty arises in trying to build a filter that cuts off perfectly above the highest frequency, \( f \). In contrast, consider what happens if the reconstruction filter is a simple digital-to-analog converter (zero-order hold). The result, shown in Figure 2-14b, is that the square wave has many more frequencies contained than the original single sine wave.

The last, and perhaps most serious, problem is that we tend to become attached to the "highest frequency of interest" and often also consider it to be \( f \), the "highest frequency contained." In actuality, no spectrum exists that cuts off perfectly at \( f \), and substantial energy exists in the frequencies beyond it. This brings us to the concept of aliasing.

### 2-4 ALIASING

Webster’s dictionary defines the word alias as "an assumed name" or "to be called by another name." The concept of aliasing, in a sampled data system, means the same thing when referred to frequencies. When a given frequency is sampled at too low a rate, it appears as a totally different lower frequency at the output of the sampler. So we say that the resultant frequency is the "alias" of the original.

An example of aliasing in the time domain is shown in Figure 2-15. The actual signal is being sampled at a rate that is somewhat less than one sample per cycle. The resultant frequency is in the vicinity of one-third the original signal. If we were able to accurately read the waveforms of Figure 2-15, we would find:

\[ f_{\text{alias}} = f_{\text{actual}} - f_{\text{sample}} \]  \hspace{1cm} (2-19)

In fact, \( f_{\text{alias}} \) is always the difference between the actual signal and some harmonic of the sample frequency (\( f_s \)). Consider an example where \( f_{\text{sample}}(f_s) = 100 \text{ kHz} \) and the signal contains the frequencies 75 kHz, 125 kHz, and 175 kHz. The sampler results are:

<table>
<thead>
<tr>
<th>( f )</th>
<th>Formula</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>75 kHz</td>
<td>( f_s - f )</td>
<td>100 - 75 = 25 kHz</td>
</tr>
<tr>
<td>125 kHz</td>
<td>( f - f_s )</td>
<td>125 - 100 = 25 kHz</td>
</tr>
<tr>
<td>175 kHz</td>
<td>( 2f_s - f )</td>
<td>200 - 175 = 25 kHz</td>
</tr>
</tbody>
</table>

Figure 2-15. Aliasing in the Time Domain
As we see, all three frequencies "look" like 25 kHz as they are aliased.

Other interesting observations can be made if we examine the effects of aliasing in the frequency domain. In Section 2-2, we discovered that, when we sampled a signal in the time domain, it appeared as a system of multiple spectra in the frequency domain with each spectrum centered about a multiple of the sample frequency, $f_s$ (Figure 2-12).

Consider what happens if we reduce the sample rate. In the time domain, the sample pulses spread apart. In the frequency domain, however, the spectra move closer together. Eventually, if we continue to reduce the sample rate, the spectra touch and finally overlap. Once the spectra have overlapped, aliasing has occurred. Such a situation is found in Figure 2-16. An important item to note is that the aliasing occurs about a point that is equal to one-half the sample frequency, or $f_s/2$. This point is often referred to as the folding frequency because all frequencies above $f_s/2$ are "folded" back into the spectrum below it. As an interesting side note, consider the rectangular spectra of Figure 2-17. The original spectrum and the first harmonic spectrum can be butted tightly together at the very point of folding, $f_s/2$.

The following expression may be written:

$$f_s - f_c = f_c$$

where $f_c$ = the cut-off frequency

or:

$$f_s = 2f_c$$

and:

$$f_c = f_s/2$$

This pictorial of the sampling theorem shows that, in the perfect situation, the sample frequency must be equal to or greater than twice the highest frequency contained in the spectrum. It also clearly indicates that aliasing or folding occurs about $f_s/2$.

The obvious cure for aliasing is, of course, to sample at a high enough rate. Yet doing so is not always feasible due to hardware constraints. Also recognize that signals usually contain noise, which is much wider in its bandwidth than the frequencies of interest. Unless some form of filtering is interjected, the noise folds over and over, back into the useful spectrum. So then, we might add a presampling or "aliasing" filter prior to sampling. The result might look like Figure 2-18. A "box" filter is chosen, and it is therefore impossible to construct. But it does illustrate the point. The spectra no longer overlap.

Like so many solutions, an aliasing filter is not a panacea, and certain problems come with it. First, complex filters introduce time delay, which must be considered if they are placed in the middle of a process control feedback system. They are also expensive. If a multichannel system requires fast analog multiplexing, each channel

must have a separate filter prior to the multiplexer. The cost per filter must then be multiplied by the number of channels. The solution to a given problem is usually a combination of sample rate selection and presample filtering.

Such a situation is found in Figure 2-19. The given signal spectrum has a peak or hump at frequencies well beyond the useful range. Sampling at a rate commensurate with the useful portion of the spectrum will fold the unwanted peak back into the middle of the information to be observed, as shown in Figure 2-19b. The problem could be solved by substantially increasing the sample rate, as Figure 2-19c
area without causing degradation to the wanted portion of the spectrum. It also precludes the salvaging of any information that lies in the folded area.

Most systems are based on a compromise of sample rate with pre-sample filtering and, as we will discover in the next chapter, the signal recovery or reconstruction technique. One thing to keep in mind is that you can never solve the problem completely. The noise bandwidth almost always exceeds the wanted signal bandwidth and so some is always folded back. The problem becomes not whether folding should occur but rather how much of it can be tolerated.

### 2-5 Band-Pass Sampling

So far, our discussion has considered video-type signals where the spectrum starts at dc, or zero frequency, and goes out to some frequency where it falls to zero. Another class of signals has no dc component but rather starts at some low frequency, \( f_l \), and goes to some high frequency, \( f_H \). An example of such a signal is found in Figure 2-21a. Just as we would expect, when the signal is sampled, multiple band-pass spectra appear, centered about multiples of the sample frequency. There is a band-pass sampling theorem that applies to this situation, as:

\[
\frac{2f_s}{K} > f_s > \frac{2f_H}{K + 1}
\]  

(2-20)
where
\[ f_L = \text{low 3-dB frequency} \]
\[ f_H = \text{high 3-dB frequency} \]
\[ f_s = \text{sample frequency} \]
\[ K_{\text{max}} < \frac{f_L}{f_H - f_L} = \frac{f_L}{\text{Bandwidth}} \]

The lowest sample frequency, \( f_s \), is found by using the largest integer, \( K \), which gives a valid equality for (2-20). It can be proven that at \( K_{\text{max}} \):

\[ f_s = \frac{2f_L}{K} = \frac{2f_H}{K + 1} = 2 \times \text{Bandwidth} \]

which is a corollary to the basic sampling theorem.

Let's consider what happens if the sample rate decreases. Looking at Figure 2-22a, we see that folding occurs when the spectra first touch, where \( f_s - f_H = f_L \), or \( f_s = 2f_H \). Now look at Figure 2-22b and observe what happens. The folding stops when \( f_s - f_L \) equals \( f_L \) and the sample rate is \( 2f_L \). This rate is lower than the rate where folding

![Figure 2-21. Band-Pass Signal Properly Sampled](image)

(a) Bandpass signal  
(b) After sampling

![Figure 2-22. Band-Pass Sampling with Reduced Rates](image)

(a) Folding starts, \( f_s = 2f_H \)  
(b) Folding stops, \( f_s = 2f_L \)

2.5 Band-Pass Sampling

started by two times the bandwidth. Should the bandwidth be small with respect to \( f_L \), folding occurs in many areas and then stops as sample frequency is reduced, while the unwanted spectra continue to fill in the gap between the zero axis and \( f_L \).

Consider a problem where:

\[ f_H = 250 \text{ kHz} \]
\[ f_L = 210 \text{ kHz} \]

and therefore:

\[ \text{Bandwidth} f_H - f_L = 40 \text{ kHz} \]

\( K_{\text{max}} \) is then:

\[ K_{\text{max}} = \frac{210}{40} = 5 \quad (K \text{ must be an integer}) \]

Then, using (2-20), \( f_s \) is:

\[ \frac{420}{5} > f_s > \frac{500}{6} \]

or:

\[ 84 \text{ kHz} > f_s > 83.3 \text{ kHz} \text{ (choose 83.5 kHz)} \]

To analyze this problem further, observe the chart in Fig. 2-23. The values of \( K \) from \( K = 5 \) to \( K = 0 \) have been plotted. You should note that the spectrum plots always lie totally on a side of one of the triangles. At the lowest sample rate, \( K = 5 \), the spectrum nearly fills the side, while at the high rate, \( K = 0 \), it occupies only a small portion of the \( K = 0 \) slope. Should any of the spectrum plots cross over and occupy space on more than one side of a triangle, it is a sign that aliasing is occurring. As the sample rate gets lower, joining up the spectra and ensuring a result that is free of folding both become more difficult. Sample rates higher than 600 kHz simply reduce the size of the spectrum plot relative to the triangle side and move it down the slope towards zero. Also note that we have been working with infinite steep-sided spectra, while more realistic ones are rounded and tail off as they move towards zero, from both \( f_L \) and \( f_H \). The concept is valid, however, and it may be utilized if you understand the content of your input signal and its resulting frequency domain spectrum.
2.6 FINITE-WIDTH SAMPLING AND APERTURE TIME

Unfortunately, impulse sampling is not realizable in a real system environment. Regardless of how narrow the effective pulse width or sampling window is with respect to the frequency component being observed, it is always finite. The effect in the frequency domain is shown in Figure 2-24. As the sample pulses start to have finite width in the time domain, the frequency domain plot starts to develop (sin x)/x characteristics, and the resultant multiple spectrum amplitudes change correspondingly. Obviously this produces an effective filtering action that acts on unwanted spectra. Before we get too enthusiastic and decide to always sample with wide pulses, let's recognize that the same filtering action deteriorates the original spectrum as well. This point brings us to the subject of aperture time.

*Aperture time* refers to the width of the sampling window or of the actual time the sampler requires to obtain an accurate reading of the analog input signal. Of concern is the fact that the signal to be measured is changing while the system is acquiring a reading. If the sample function could be truly an impulse and infinitely narrow, the sampler output would be precisely the input value at the point of sampling.

Consider the sine wave of Figure 2-25. If this waveform is to be measured accurately at any point, the sampler must successfully handle the point of the greatest rate of change or the zero crossing. The slope at that point is the derivative of $V(t)$ evaluated at the zero crossing, or:

$$
\left. \frac{dV(t)}{dt} \right|_{t=0} = 0.5V \cos \omega t
$$

As $\cos \omega t = 1$ at zero, and $\omega = 2\pi f$, the result is:

$$
\left. \frac{dV(t)}{dt} \right|_{t=0} = 0.5V \omega = (0.5V)(2\pi f) = \pi fV \text{ Volts/sec}
$$

Referring to the same figure, we can express the maximum slope with reference to the aperture time by:

$$
\frac{\Delta V}{\Delta T} = \pi fV \tag{2-21}
$$

where

$$
\Delta T = \text{aperture time} = T_a = \frac{\Delta V}{V} \times \frac{1}{\pi f}
$$

To make (2-21) track with the aperture time formulas that are common to the data acquisition industry, we have to jump ahead and introduce the concept of quantization (a more detailed discussion will be found in Chapter 3). In the data acquisition system, the output of the sampler is further translated into the digital world by means of a quantizer. The quantizer allows the sampled signal to have only d
crete amplitude values. For a binary system, the full-scale range is segmented into intervals defined by:

\[ \Delta V = \frac{V}{2^n} \]  \hspace{1cm} (2-22)

where

\[ n = \text{the number of bits} \]
\[ V = \text{full-scale voltage} \]

With reference to (2-21), we recognize that the parameter \( V \) is directly related to the \( \Delta V \) of (2-22) and, substituting, we find that:

\[ \tau_a = \frac{V/2^n \times 1}{V} \times \frac{1}{\pi f} \]

\[ = \frac{1}{2^n \pi f} \]  \hspace{1cm} (2-23)

Given the aperture time and number of bits, we find the maximum frequency that can be sampled to the accuracy implied:

\[ f = \frac{1}{2^n \pi \tau_a} \]  \hspace{1cm} (2-24)

These expressions are predicated on the input moving a full quantum level, or the least significant bit (LSB), during the aperture window. Should the system be specified on the basis of ±½LSB, the expressions become:

\[ \tau_a = \frac{1}{2^{n+1} \pi f} \]  \hspace{1cm} (2-25)

and:

\[ f = \frac{1}{2^{n+1} \pi \tau_a} \]  \hspace{1cm} (2-26)

Some waveforms are more readily examined directly in terms of their rate of change rather than of their frequency content. Take, for instance, the signal of Figure 2-26. The relationship here is:

\[ \text{Slope} = \frac{dv}{dt} \]

\[ \text{Expected Output} \]
\[ \text{Delayed Output} \]
\[ \text{Actual Output} \]
\[ \text{Amplitude} \]
\[ t \]

\[ \text{Input Signal} \]
\[ \text{Aperture Window} \]
\[ \text{Answer Here} \]

Figure 2-26. Trapezoidal Waveform
In actuality, the aperture time window consists of two parts. The first is the effective sample pulse width, as it manifests itself at the measurement point. The second is the uncertainty of the measurement window positioning from sample point to sample point. This latter effect, called aperture uncertainty, manifests itself as “jitter” on the sampling waveform. In a properly designed system, the aperture uncertainty is much smaller than the basic aperture time, with the difference ranging from three times up to an order of magnitude. If the aperture time is small compared to the period of the highest frequency being sampled—that is, \( \tau_a \ll 1/\omega_0 \)—the waveform is simply displaced in time after sampling, and aperture uncertainty may be used in place of aperture time. An example of aperture uncertainty can be found for the trapezoidal waveform in Figure 2-26b. The answer expected at the end of the aperture window is the value seen at point one. The actual output, however, is closer to the average signal value of point two, as the signal has been moving during the window. The effect of a series of samples is to simply displace, or delay, the output in time. That this uncertainty can be significant is shown by the following example.

Given a sampled data system with:

\[
\begin{align*}
\tau_a &= 50 \text{ nsec} \\
\tau_u &= \text{aperture uncertainty} = 4 \text{ nsec} \\
n &= \text{number of bits} = 10 \\
\text{accuracy} &= \pm 1/4 \text{LSB}
\end{align*}
\]

If we utilize the aperture time, \( \tau_a \), the highest frequency we could examine accurately at all points is:

\[
f = \frac{1}{(2^{11})\pi(50 \times 10^{-9})} = 3.1 \text{ kHz}
\]

Using aperture uncertainty in place of \( \tau_a \), we find:

\[
f = \frac{1}{(2^{11})\pi(4 \times 10^{-9})} = 38.9 \text{ kHz}
\]

an improvement of greater than 10 to 1.

This discussion considers situations where instantaneous aperture error is important. Very often, however, we are concerned with the average error introduced at the frequency of interest. In such cases, we would consider the aperture window as an aperiodic, rectangular pulse, in the time domain, and multiply its \((\sin x)/x\) frequency domain representation with the input signal spectrum. If the aperture width is too wide, the input spectrum is degraded by the \((\sin x)/x\) filter action. For instance, if the aperture window is 1 \( \mu \text{sec} \) in width, a 2-percent degradation of the input signal would take place around 40 kHz.

### 2-7 FREQUENCY RESOLUTION

Often the data acquisition system is required to determine which frequencies are present in a spectrum and how much energy each frequency contains, relative to the others. Such an application is called spectral or spectrum analysis. Since this application means the system must be able to separate one frequency from another, we are interested in frequency resolution. In a general sense, the frequency resolution postulate says that if a function is available for \( K \) seconds, \( 1/K \) cycle per second (cps) may be resolved. In other words, at least one complete cycle of the frequency must be seen. Although spectrum analysis requires adequate sample rate selection, like any other sampled system process, the resolution is a function of the viewing period and not the sampling frequency. Take two examples:

**Example 1:**

| Observe for: | 100 sec |
| Sample rate: | 1,000 samples/sec |
| Foldover frequency: | 500 cps |
| Resolution: | 1/100 cps |

**Example 2:**

| Observe for: | 100 sec |
| Sample rate: | 5,000 samples/sec |
| Foldover frequency: | 2,500 cps |
| Resolution: | 1/100 cps |

As we see, the higher sampling rate affects folding or aliasing, but does not affect resolution.

A factor that does have a substantial effect is the shape of the viewing window. The simplest of shapes, the rectangular window Figure 2-27, looks like an aperiodic pulse of width \( K \). So it translates a \((\sin x)/x\) function in the frequency domain. The single sine wave becomes a single spectral line. Application of the window function to the single frequency is a multiplicative process. So the two functi
Figure 2-27. Rectangular Window with a Single Sine Wave

are convolved in the frequency domain to produce a \((\sin x)/x\) spectrum, which is centered about the \(f_0\) spectral line. Although this function can be implemented by a simple switch, it is not an optimum window function. Figure 2-28 shows the frequency domain result for three frequencies. In this diagram, \(f_1\) clearly stands out. The other two, \(f_2\) and \(f_3\), are starting to show considerable overlap. If the composite were drawn showing the effects of all the \((\sin x)/x\) tails, they would be less separable. As more and more frequencies are present, \(f_2\) and \(f_3\) would become more and more difficult to separate.

Fortunately, several other very tailored windows have been developed to solve this problem. The most popular of these are found

Figure 2-28. Rectangular Window with Three Frequencies

PROBLEMS

2-1. Define the concept of sampling.

2-2. a. Name the three general kinds of signals.
   b. What is the basic difference between deterministic and random signals?

2-3. A data acquisition input has a signal-to-noise ratio of 10:1.
   a. What theoretical improvement can be realized by averaging over 1,024 samples?
   b. What is the effective signal-to-noise ratio?

2-4. Using expressions (2-7) and (2-8) as a basis, develop expressions of (2-13) and (2-14) for the Fourier integral.

2-5. a. The concept of convolution is concerned with which mathematical concepts?
   b. What three important concepts should you remember when working with the time and frequency domains?

2-6. What mathematical operation defines the act of sampling?
   a. in the time domain, and
   b. in the frequency domain?

2-7. The input to a data acquisition system contains the following...
discrete frequencies:

<table>
<thead>
<tr>
<th>Frequency (kHz)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>52.5</td>
<td>205</td>
</tr>
<tr>
<td>105</td>
<td>243</td>
</tr>
<tr>
<td>143</td>
<td></td>
</tr>
</tbody>
</table>

If the sample frequency is 100 kHz, what will each input frequency appear as in the output?

2-8. Consider a band-pass signal with a rectangular shape. Prove that \( f_s = 2B_w \) at \( K_{\text{max}} \).

2-9. A data acquisition system has an aperture time of 10 nsec and a resolution of 12 bits.
   a. What is the highest frequency that can be sampled to within \( \pm \frac{1}{4} \text{LSB} \)?
   b. If the system had an aperture uncertainty of 1 nsec, what would \( f_{\text{max}} \) be?

2-10. A data acquisition system has an aperture time of 2 nsec. To within what resolution accuracy can a 10-kHz signal be sampled at the point of maximum rate of change?

2-11. A signal is sampled for 120 sec at a 20-kHz sample rate, using a rectangular resolution window.
   a. What is the foldover frequency?
   b. What is the frequency resolution?

REFERENCES


3-0 INTRODUCTION

In the previous chapter, we discovered the many complication encountered when performing proper sampling of the various types of analog signals. This chapter completes the sampling picture. It discusses the conversion of the time samples to digital form (quantization and the subsequent reconstruction either of the original signals or some derivative (recovery). Like the pieces of an interlocking puzzle all three functions must fit together properly if the sampled-data system is to produce the desired results. Since, to a great extent, each depends on the other, the proper system solution is a simultaneous optimization of all three. The end of the chapter rounds out the total sampled-data system discussion by providing an example that highlights the trade-offs to be considered.

3-1 QUANTIZATION

The major premise of the data acquisition system is that signals are converted from the analog domain to the digital domain, so that a digital computer or some other digital process may operate on them. This process, called quantization, is defined as:
the conversion of an input function that has values in a continuous range to an output that has only discrete values.

Sometimes, quantization is combined with sampling and the two operations occur simultaneously. In other implementations, such as when a sample-and-hold is used, sampling and quantization are two very distinct and separate operations. Here, the sample-and-hold makes the basic measurement and holds the value while the quantizer converts it into a digital format.

The transfer function for a typical quantizer, or analog-to-digital converter, is found in Figure 3-1. While the analog input voltage may be any value between zero volts and the full-scale value, $V_{FS}$, the output can exist only as one of sixteen discrete values from zero to fifteen. So we find the transfer function to be a staircase. Each of the staircase "flats" and its associated digital output code are referred to as quantum levels. The distance between any two quantum levels is called a quantum interval. For a binary- (or power-of-2)-based quantizer, the quantum interval, $\alpha$, is defined as:

$$\alpha = \frac{V_{FS}}{2^n}$$

where $V_{FS} =$ full-scale voltage input
2 = number base for binary
$n =$ number of bits (binary digits)
$2^n =$ number of quantizing intervals

For the quantizer of Figure 3-1, we find that:

$$\alpha = \frac{V_{FS}}{2^4} = \frac{V_{FS}}{16}$$

Should 5 bits have been required, $\alpha$ would be:

$$\frac{V_{FS}}{2^5} \quad \text{or} \quad \frac{V_{FS}}{32}$$

and the transfer function staircase would contain twice as many steps as Figure 3-1, and they would be half-sized. The steps continue double in number and halve in size as the number of quantizer increase, as follows:

<table>
<thead>
<tr>
<th>$n$</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
</tr>
<tr>
<td>6</td>
<td>64</td>
</tr>
<tr>
<td>7</td>
<td>128</td>
</tr>
<tr>
<td>8</td>
<td>256</td>
</tr>
<tr>
<td>9</td>
<td>512</td>
</tr>
<tr>
<td>10</td>
<td>1,024</td>
</tr>
<tr>
<td>11</td>
<td>2,048</td>
</tr>
<tr>
<td>12</td>
<td>4,096</td>
</tr>
<tr>
<td>13</td>
<td>8,192</td>
</tr>
<tr>
<td>14</td>
<td>16,384</td>
</tr>
<tr>
<td>15</td>
<td>32,768</td>
</tr>
<tr>
<td>16</td>
<td>65,536</td>
</tr>
</tbody>
</table>

Referring to Figure 3-1, note that level 16 does not actually exist; that the highest binary number output available is level 15. It should be mentioned that the voltage range of 0 volts to $V_{FS}$ was selected.
simplicity. In actuality the quantizer input could range from any arbitrary low value to any arbitrary high value.

Several binary codes are commonly used. The most common are the straight binary and offset binary, which are found in Table 3-1. Straight binary is a unipolar code where 0000 is equivalent to level 0 (also analog zero), while 1111 is equivalent to level 15. On the other hand, offset binary is a bipolar code with 0000 equal to -7, and 1111 equal to +7. For this situation, the 2^3 bit (most significant) is the sign bit, with 0 = - and 1 = +. There are two "zero" levels in the offset binary code, each half a quantum interval from true analog zero.

Two other codes are more compatible with standard digital computers (Table 3-2). They are the one's and two's complement binary codes. Both bipolar, they differ primarily in the positioning of the true analog zeros. The one's complement code is identical with the offset binary code with the exception that the sign bit is inverted. In this case, 1 is negative and 0 is positive. Like offset binary, this system has two zero levels. The two's complement code, however, has only one zero level, and it is coincident with the 0000 code. There is also one more negative level (-8) than positive (+7). The sign is also the complement of that for offset binary. Careful examination shows that three bipolar codes discussed thus far are simple extrapolations from straight binary requiring either offsetting of analog zero with respect to the code levels and/or simple inversion of the sign bit.

Another binary-type code, which was popular in the late sixties, is the absolute-value-plus-sign code shown in Table 3-3. A unique feature of this code is that the lower three bits are exactly the binary representation of the associated level; for example, 111 = 7. A result, the lower three bits reflect about the zero level; that +3 has the same code as -3. Like the other bipolar codes, the most significant (2^3) bit is the sign designator.

A code used by some quantizers is the Gray code. This code shown in Table 3-4, has two unique features. First, like the absolute value-plus-sign code, the lower three bits reflect about the scale point. Second, as the code progresses from one level to the next, only one bit changes. The Gray code is used for most types of electronic mechanical quantizers and for cyclic-type A/D converters.

### Table 3-1 Straight and Offset Binary Codes

<table>
<thead>
<tr>
<th>Bit Position</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>2^3 2^2 2^1 2^0</td>
<td>Straight Binary</td>
</tr>
<tr>
<td><strong>Straight</strong> Binary Zero</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 1</td>
<td>1 -6</td>
</tr>
<tr>
<td>0 0 1 0</td>
<td>2 -5</td>
</tr>
<tr>
<td>0 0 1 1</td>
<td>3 -4</td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>4 -3</td>
</tr>
<tr>
<td>0 1 0 1</td>
<td>5 -2</td>
</tr>
<tr>
<td>0 1 1 0</td>
<td>6 -1</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>7 0</td>
</tr>
<tr>
<td><strong>Offset</strong> Binary Zero</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>9 +1</td>
</tr>
<tr>
<td>1 0 1 0</td>
<td>10 +2</td>
</tr>
<tr>
<td>1 0 1 1</td>
<td>11 +3</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>12 +4</td>
</tr>
<tr>
<td>1 1 0 1</td>
<td>13 +5</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>14 +6</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>15 +7</td>
</tr>
</tbody>
</table>

### Table 3-2 One's and Two's Complement Binary Codes

<table>
<thead>
<tr>
<th>Bit Position</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>2^3 2^2 2^1 2^0</td>
<td>Two's</td>
</tr>
<tr>
<td>1 0 0 0</td>
<td>-8</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>-7</td>
</tr>
<tr>
<td>1 0 1 0</td>
<td>-6</td>
</tr>
<tr>
<td>1 0 1 1</td>
<td>-5</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>-4</td>
</tr>
<tr>
<td>1 1 0 1</td>
<td>-3</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>-2</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>-1</td>
</tr>
</tbody>
</table>

One's Zero

Two's Zero
Table 3-3 Absolute-Value-Plus-Sign Code

<table>
<thead>
<tr>
<th>Bit Position</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^7$</td>
<td>$2^6$</td>
</tr>
<tr>
<td>0 1 1 1 1</td>
<td>-7</td>
</tr>
<tr>
<td>0 1 1 1 0</td>
<td>-6</td>
</tr>
<tr>
<td>0 1 0 1 1</td>
<td>-5</td>
</tr>
<tr>
<td>0 1 0 0 0</td>
<td>-4</td>
</tr>
<tr>
<td>0 0 1 1 1</td>
<td>-3</td>
</tr>
<tr>
<td>0 0 1 0 0</td>
<td>-2</td>
</tr>
<tr>
<td>0 0 0 1 0</td>
<td>-1</td>
</tr>
<tr>
<td>0 0 0 0 0</td>
<td>0</td>
</tr>
</tbody>
</table>

Zero

| 1 0 0 0 0 | +0 |
| 1 0 0 1 1 | +1 |
| 1 0 1 0 1 | +2 |
| 1 0 1 1 1 | +3 |
| 1 1 0 0 0 | +4 |
| 1 1 0 1 1 | +5 |
| 1 1 1 0 0 | +6 |
| 1 1 1 1 1 | +7 |

Table 3-4 The Gray Code

<table>
<thead>
<tr>
<th>Bit Position</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^7$</td>
<td>$2^6$</td>
</tr>
<tr>
<td>0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 0 1 1</td>
<td>1</td>
</tr>
<tr>
<td>0 0 1 1 1</td>
<td>2</td>
</tr>
<tr>
<td>0 0 1 0 0</td>
<td>3</td>
</tr>
<tr>
<td>0 1 1 1 1</td>
<td>4</td>
</tr>
<tr>
<td>0 1 1 1 1</td>
<td>5</td>
</tr>
<tr>
<td>0 1 0 1 1</td>
<td>6</td>
</tr>
<tr>
<td>0 1 0 0 0</td>
<td>7</td>
</tr>
</tbody>
</table>

Midpoint

| 1 1 0 0 0 | 8     |
| 1 1 0 1 0 | 9     |
| 1 1 1 1 0 | 10    |
| 1 1 1 1 1 | 11    |
| 1 0 1 1 1 | 12    |
| 1 0 0 1 1 | 13    |
| 1 0 0 0 1 | 14    |
| 1 0 0 0 0 | 15    |

One other commonly used code is the 8421 binary-coded decimal (BCD), as shown in Table 3-5. This code is utilized when direct decimal readouts are required. Although the codes are binary in form, they exist only in decades (0 to 9). When the limits of a decade are exceeded, the code carries into the next higher decade. Each successive decade is identical to that shown in the table, but it has a magnifier of 10 associated with it. The quantum interval for the 8421 BCD code is calculated as follows:

$$\alpha = \frac{V_{FS}}{10^m} \quad (3-2)$$

where $V_{FS} =$ full-scale voltage input

$2 =$ number base for decimal

$m =$ number of decades (or digits)

$10^m =$ number of quantizing intervals

Now let's get back to the quantization process itself. An expansion of a segment of Figure 3-1 is found in Figure 3-2. Here we see the transitions between the given levels $K - 1$ and $K$ and between levels $K$ and $K + 1$. Of interest here is the fact that, once the output code has made the transition from level $K - 1$ to level $K$, does not change again until the analog input has progressed from $V_K - \alpha/2$ to $V_K + \alpha/2$, or a value equivalent to a full quantum interval. So obviously, when the quantizer output level is $K$, all we can say about the input is that it is somewhere between $V_K - \alpha/2$ and $V_K + \alpha/2$. This element of uncertainty produces an irreducible error situation. The only time the output code is totally accurate is when it

Table 3-5 The 8421 BCD Code

<table>
<thead>
<tr>
<th>8 4 2 1</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 0 1</td>
<td>1</td>
</tr>
<tr>
<td>0 0 1 0</td>
<td>2</td>
</tr>
<tr>
<td>0 0 1 1</td>
<td>3</td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>4</td>
</tr>
<tr>
<td>0 1 0 1</td>
<td>5</td>
</tr>
<tr>
<td>0 1 1 0</td>
<td>6</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>7</td>
</tr>
<tr>
<td>1 0 0 0</td>
<td>8</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>9</td>
</tr>
</tbody>
</table>
Figure 3-2. Uncertainty at Quantizer Output

Input is exactly $V_K$. Having the output code remain constant, while the input ranges over all possible values $V_K \pm \alpha/2$, is the same as adding noise to the quantizer input. This effect is called quantization error or quantization noise. If we consider any given analog input, $f(t)$, to occur over a long period, it is probable that all values of $V_K \pm \alpha/2$ will be present and will cause the $K$th level to occur in a random fashion. The amplitude of the effective noise is established by the number of quantizer bits or quantum intervals. If the input deviation from the voltage value equivalent to the $K$th level, $V_K$, is designated as $\epsilon$ (Figure 3-2), the RMS quantization error is calculated as follows:

$$\sqrt{\frac{1}{\alpha} \int_{-\alpha/2}^{\alpha/2} \epsilon^2 \, d\epsilon} = \sqrt{\frac{\alpha^2}{12}} = \frac{\alpha}{2\sqrt{3}} \quad (3-3)$$

The character of the quantization error function is found in Figure 3-3. As you can see, the error is triangular in nature, and it cycles back and forth between $+\alpha/2$ and $-\alpha/2$ as the analog input proceeds up or down the scale. Quantization error has a probability function that is rectangular in nature as, over a large period, the input is as likely to occur at any one value of $\epsilon$ as any other, while the output remains fixed. This situation is true for the perfect quantizer. Yet, as Gordon reminds us (reference 5), quantizers add noise themselves from a number of internal sources. This noise, generally Gaussian in character, has a definite impact on the effective quantization error rate.

Figure 3-3. Expanded Section of Binary Quantizer Transfer Function with Error Curve

Figure 3-4 depicts this situation when the analog input is centered in the quantum interval. This figure shows that the noise peaks above $\pm 2.5\sigma$ extend into the adjacent quantum intervals and that 1.4 percent of the time the output code is in error. Unfortunately, as discussed, the input does not always occur at the center of the interval and it is just as likely to be at any other value.

Figure 3-5 gives an indication of what happens when the input is offset from the center. Although the error output favors the $K+$ side, it has also substantially increased in value. The quantizer output will be in error 6.7 percent of the time under these conditions. The type of result occurs when the RMS quantizer noise is only 20 percent of a quantum interval. You should therefore consider the noise figure of the analog-to-digital converter (quantizer) when making a selection.
3-2 QUANTIZATION AND RANDOM SIGNALS

In Chapter 2, we recognized that a certain class of dynamic signals is characterized by its statistics. These signals are defined by their probability density functions rather than by their absolute values at particular points in time. As it turns out, the requirements for quantizing resolution for this type of signal is much less stringent than it is for signals that we must precisely define at every point.

Consider the relationship shown in Figure 3-6. Figure 3-6a is the probability density function for the time/voltage signal of Figure 3-6b. The horizontal lines in 3-6b indicate the centers of the quantization intervals. As we see, the centers of the quantization intervals have been extended leftwards to the density function. That this representation is accurate is confirmed by the fact that the quantizer outputs can occur only at specific discrete values. The quantized density function is, then, a measure of the number of times that each discrete value occurred with respect to the total number of samples taken. Effectively, the original density function has been sampled in the voltage domain by a sample function that occurs at a rate equal to the quantum interval.

As this conclusion implies, such a situation may be treated like any other sampling problem. Consider the elements of Figure 3-7. Translation of the probability density function in the voltage domain to its complementary "spectrum" in the \( \nu \)-frequency domain is found in Figure 3-7a. Figure 3-7b shows the translation of the quantizing function from one domain to the other. Because these translations are
analogous to the sampling of a time signal with an impulse sample 

function (see Section 2-2), the results of Figure 3-7c and 3-7b are not 

surprising. The sampled probability density function in the voltage 
domain appears as multiple “spectra” in the $V$ domain. All the other 
sampling rules hold as well, and, if we have fewer quantization levels ($n$ increases), the “spectra” of 3-7d move closer together. If they should 

overlap, you would not be able to distinguish one level from another 

because the sample rate would be too low. So the principles of aliasing 

apply in this case as well. As you would expect, there is a quantizing 

thereom analogous to the sampling theorem (two samples per cycle …). 
The rule of thumb goes:

If a quantizing width is chosen such that the range of $V(t)$ 
is at least eight quantum intervals, then the first few statistical 
moments of the quantized signal may be used as moments of $V(t)$, providing Sheppard’s corrections are used (reference 1).

To see how this applies, consider the first two moments as shown:

$$m_1 = \text{Average value} = \int_{-\infty}^{\infty} Vp(V)dV \quad (3-3a)$$

$$m_2 = \text{Second moment} = \int_{-\infty}^{\infty} V^2p(V)dV \quad (3-3b)$$

The mean-squared and root-mean-squared (RMS) values of $V$ 

are further defined as:

$$\sigma^2 = m_2 - m_1^2 \quad (3-5)$$

and:

$$\sigma = \sqrt{m_2 - m_1^2} \quad (3-6)$$

Often very important are the statistical quantities, $m_1$, $\sigma$, and $\sigma^2$, when defining a random or nondeterministic signal. Recovery of these 

parameters by use of Sheppard’s corrections is:

$$\text{Average value} = \mu_1 = \int Vp(V^*)dV \quad (3-7)$$

where $p(V^*)$ = the quantized and sampled result

$$\text{Second moment} = \mu_2 - \frac{1}{12} \alpha^2 \quad (3-8)$$

3-3 SIGNAL RECOVERY

Mean-square value = $\mu_2 - \frac{1}{12} \alpha^2 - \mu_1^2 \quad (3-9)$

RMS value = $\sqrt{\mu_2 - \frac{1}{12} \alpha^2 - \mu_1^2} \quad (3-10)$

where $\mu_1$ and $\mu_2$ = the simple moments calculated from the 

quantized samples

$\alpha$ = the quantum interval

$\frac{1}{12} \alpha^2$ = the correction

Two observations can be made. First, the first moment or average 

value is the same whether quantized or not. Second, with a small 
increase in the number of quantum levels, no corrections are needed. 

One final comment on statistical quantization. Due to a corollary to 
sampling, the frequency content of the probability density function is 

important. Specifically, signals with favorite points or values that 

occur frequently must be handled carefully. This type of signal—such 
as single sinusoid—has a high “frequency” content, and the quantizing 

rule of thumb does not apply.

3-3 SIGNAL RECOVERY

In the first part of Chapter 2, it was implied that the sample data 

process was not complete until a function called recovery was effected. It 

was further suggested that this recovery function was necessary to 

reconstruct either the original time signal, or a derivative of it, after 

digital processing. Why is a recovery or reconstruction process necessary? There are two obvious reasons:

1. errors of omission, and

2. errors of commission.

For an error of omission, look at Figure 3–8. The dashed line represents 

the envelope of the original time function. The result of sampling, however, is a series of values equal to the sample points depicted 

by the arrow heads. At this point, a great deal of the signal trend information is missing. Once quantization has been added and the samples 
can have only discrete values, the difficulties are compounded.

The second reason for recovery—errors of commission—is found 
in Figure 2–12c of Chapter 2. After sampling, not only is the original
signal spectrum present, but any number of additional spectra are also present and centered about multiples of the sample frequency. If these are not eliminated, they may cause totally erroneous results within the digital process. They will surely prove confusing in a digital communications system.

A number of different recovery operators are used, and we will discuss some of the more common ones. But first, let’s consider the ideal recovery function, shown in Figure 3-9. The function is a rectangular or “box” filter in the frequency domain. When this function is translated to the time domain, it takes the form \((\sin x)/x\). Also, you should not be too surprised to find that recovery is the inverse function to sampling. Whereas sampling is multiplication in the time domain and therefore equivalent to convolution in the frequency domain, recovery is just the opposite. Utilizing the ideal recovery operator, then, is equivalent to convolving the \((\sin x)/x\) function of Figure 3-9 with the time samples of Figure 3-8.

The result is shown in Figure 3-10. If the parts of the various \((\sin x)/x\) functions are added together, the original waveform would be reconstructed. How closely the original function is reproduced would depend on the granularity of quantization. In Chapter 2, it was mentioned that the validity of the sampling theorem depended on the use of the ideal rectangular operator. To validate this statement, consider the two sample points from a single sinusoid. Now convolve the sample with the \((\sin x)/x\) function. The result is a reproduced sine wave.

Unfortunately, it is not physically possible to implement a filter with the ideal recovery function characteristics. In fact, the most commonly used reconstructing function has exactly the opposite characteristics. This is the zero-order hold function shown in Figure 3-11. As seen in Figure 3-11a, the time and frequency domain representations for the zero-order hold are the exact inverses of those for the ideal operator. The function is rectangular in the time domain and \((\sin x)/x\) in the frequency domain. Figure 3-11b provides the final convolved...
result in the time domain. This output is quickly recognized as that of a typical digital-to-analog (D/A) converter. Unlike the results shown in Figure 3-10, this reconstruction process results in substantial error. The worst case for this error would occur at the zero crossing of the sinusoid for the highest frequency contained in the original signal.

This error is diagrammed in Figure 3-12. Let's analyze this error. The maximum slope of the sinusoid, $V(t) = A \sin \omega t$, occurs at $t = 0$ and is:

$$\text{Maximum slope} = \left. \frac{dV(t)}{dt} \right|_{t=0} = A\omega$$

Using the terms of Figure 3-12, we also find that

$$\text{Slope} = A\omega = \frac{V_{\text{error}}}{T}$$

where $T$ = the sample period, and therefore

$$V_{\text{error}} = A\omega T$$

The percentage of error is:

$$\% \text{ error} = \frac{V_{\text{error}}}{A} \times 100 = \frac{A\omega T}{A} \times 100 \omega T$$

or, as $\omega = 2\pi f$:

$$\% \text{ error} = 200 \pi f T$$

(3-11)

Recognizing that $1/fT$ = samples/cycle, we can develop the expression:

$$\text{Samples/cycle} = \frac{200\pi}{\% \text{ error}}$$

(3-12)

Utilizing this expression, we find that you have to sample the original waveform by at least 628 samples/cycle to achieve an instantaneous reconstruction error of less than 1 percent. So now we see that the sample rate is directly affected by the type of reconstruction operator selected.

Another way to look at this situation is to consider that we have applied a narrow-band $(\sin x)/x$ filter to the original frequency spec-

\[ \text{Figure 3-12. Zero-Order Hold Error Function} \]

trum. One method of solving this problem with feasible hardware is to provide a narrow return-to-zero output from the D/A converter and follow it with sharp cutoff analog filter. This combination, shown in Figure 3-13, moves the first zero crossing, $1/T$, out in frequency so the original input spectrum is minimally affected by the $(\sin x)/x$ filtering action. The analog filter provides the storage to fill in the hole between the sample points.

In many situations, a digital computer is used to fill in or to interpolate between the sample points, as illustrated in Figure 3-14. This method of recovery is called the predictive first-order hold and the expression for the recovery operator is:

$$X(t) = \left[ \frac{X(n) - X(n-1)}{T} \right] (t - nT) + X(n)$$

(3-13)
Figure 3-14. Predictive First-Order Hold

At any given sample point, the computer calculates the slope between the previous sample and the present sample and assumes the slope between the next two points will be the same. This works well as long as the slope is not changing. When it is changing, such as at the top of the sinusoid, substantial errors start to creep in. In terms of the instantaneous error percentage, the samples per cycle for this recovery operator are:

\[
\text{Samples/cycle} = \frac{62.8}{\sqrt{\% \text{ error}}} \quad (3-14)
\]

The results are substantially better than that for the zero-order hold. Yet you must exercise care when using a predictive reconstructor in any closed-loop system. An erroneous assumption can cause the loop to become unstable.

Another first-order recovery operator, the linear first-order hold, can be seen in Figure 3-15. The expression for this recovery function is:

\[
X(t) = \frac{X(n) - X(n - 1)}{T} [t - nT] + X(n - 1) \quad (3-15)
\]

In this case, the computer waits until it knows the value of any two points and then simply draws a line between them. No prediction is involved in this process, but the output is always one sample period behind. The expression for samples/cycle versus the instantaneous error percentage is:

\[
\text{Samples/cycle} = \frac{22.3}{\sqrt{\% \text{ error}}} \quad (3-16)
\]

Often, to smooth out the recovered output, the recovery algo-

Figure 3-15. Linear First-Order Hold

rithm inserts intermediate points between the actual samples, example, where accurate recovery is very critical, second-order functions are sometimes used. This procedure uses not only the difference between the last two samples, but also the difference between that function and the difference between the previous two samples. Second-order computations may be effected for both the linear and predic functions.

Sample rates for the worst-case, instantaneous error result from expressions (3-12), (3-14), and (3-16). In many practical cases, root-mean-square (RMS) error is more useful. The RMS error is instantaneous error averaged over time. Table 3-6 provides a comparison of sample rates versus three orders of RMS recovery error. The recovery operators compared are zero-order hold (step), first-order linear, and 2-, 3-, and 4-pole Butterworth filters. This particular table is based on input data having the characteristics of a fourth-or

<table>
<thead>
<tr>
<th>Recovery Operator</th>
<th>Percentage of RMS Recovery Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.01</td>
</tr>
<tr>
<td>Zero-order (step)</td>
<td>12,000</td>
</tr>
<tr>
<td>Linear</td>
<td>160</td>
</tr>
<tr>
<td>Butterworth (n = 2)</td>
<td>1,920</td>
</tr>
<tr>
<td>Butterworth (n = 3)</td>
<td>280</td>
</tr>
<tr>
<td>Butterworth (n = 4)</td>
<td>108</td>
</tr>
</tbody>
</table>

*Note: Based on input having the characteristics of a fourth-order Butterworth spectrum.*
(4-pole) Butterworth spectrum, and the values on the chart have units in samples per cycle of a 3-dB bandwidth.

A similar chart could be constructed for other shapes and orders of possible input data spectra. This point highlights one of the great difficulties involved with data acquisition. There are nearly as many shapes and orders of input spectra as there are data acquisition systems. So only a few "cookbook" charts, like Table 3-6, are available, and they cover only a particular set of circumstances.

### 3-4 A DATA ACQUISITION EXAMPLE

At this point an example is appropriate, to tie the main concepts of Chapters 2 and 3 together. To do so, consider the digital communications system of Figure 3-16. The analog input has the characteristics of a fourth-order Butterworth spectrum, and for this exercise we assume that the highest frequency of interest is at 20 kHz. Requirements for the system are:

1. dynamic range >70 dB
2. aliasing error <0.1 percent
3. recovery error <0.1 percent RMS
4. aperture error <0.1 percent average

Things to be determined are:

1. number of quantization levels (bits),
2. sample rate,
3. sampling aperture, and
4. recovery operator.

Let's first consider the number of quantization levels or bits required. The system dynamic range has been specified as 70 dB. This is equivalent to the ratio 3\,162:1. The nearest greater binary number is 4,096, or 12 bits, which is equivalent to 72 dB. (Note: you might consider a 13-bit quantizer, which would provide a 78-dB dynamic range.)

Selection of the sampling rate is a bit more complicated because we have two requirements: one for aliasing and one for recovery error. Let's consider the aliasing problem first. Simply stated, it requires the error due to folding to be less than 0.1 percent of the expected value at the highest frequency of interest, 20 kHz. As we see in Figure 3-16, at 20 kHz, the signal levels are 4.4 percent of the dc value. The overlap of the first harmonic spectrum must not exceed 0.1 percent of the 4.4 percent value. This overlap occurs when the first harmonic spectrum

![Figure 3-16. Data Acquisition Example](image)

**Figure 3-16.** Data Acquisition Example

contribution of $f_s - 120 \, \text{kHz}$ coincides with the 20-kHz point on the original spectrum. This situation is shown in Figure 3-17. As expected, there is a 1,000:1, or 60-dB, difference between the two spectra. To satisfy this requirement must then be equal to or greater than:

$$120 \, \text{kHz} + 20 \, \text{kHz} = 140 \, \text{kHz}$$

Before we firmly establish 140 kHz as the sample rate, consider the impact of the recovery operator. To make things simple, let's choose the recovery filter to have the same characteristics as the input spectrum, namely a 4-pole Butterworth. Fortunately, Table 5 includes this type of filter for 0.1 percent recovery error. We quickly see that:

$$\frac{f_s}{f_{5\, \text{db}}} = 36$$

and, for our situation:

$$f_s = 36 \times 10 \, \text{kHz} = 360 \, \text{kHz}$$

So, for this particular case, the recovery requirements dictate the sample rate. At $f_s = 360 \, \text{kHz}$, aliasing is negligible. A good selection for this situation would be a 12-bit, 500-kHz, successive approximati
Figure 3-17. Effect of Aliasing, $f_s = 130$ kHz

A/D converter. With this choice established, the next decision concerns whether to use a sample-and-hold function, prior to the converter. The basic aperture of a 500-kHz A/D converter is:

$$\frac{1}{500 \text{ kHz}} = 2 \mu\text{sec}$$

Such an aperture would be equivalent to placing a $(\sin x)/x$ filter in the system, with the first zero crossing at 500 kHz. This insertion would cause between 1- and 2-percent attenuation at 20 kHz. To obtain a degradation of less than 0.1 percent, a sample-and-hold with an aperture less than 100 nsec would be necessary.

Note that we have been satisfied with RMS values of accuracy. Had a case been made for instantaneous accuracy at any point on the highest frequency of interest sinusoid, the aperture expressions of Section 2-6 would apply, and the required aperture uncertainty would be a few nanoseconds.

After the digital data has been transmitted from the acquisition point to the point of reception, recovery must be considered. We have already chosen the final recovery filter to be a 4-pole Butterworth, but first the data must be converted from digital binary numbers to analog values. Suppose we utilized a simple digital-to-analog converter (zero-order hold), with an update rate equal to the original conversion rate of 500 kHz, prior to the filter. We still have the problem of $(\sin x)/x$ filtering, which in this case would be more severe than the final filter. A way to solve the problem would be to utilize a D/A converter with return-to-zero outputs (Figure 3-18) of less than 100 nsec in width.

The final system configuration is shown in Figure 3-18. An
amplifier is utilized in the output to compensate for the loss incurred when low-energy pulses are applied to the complex filter.

3-5 SUMMARY OF DESIGN CONSIDERATIONS

There are, of course, many potential solutions for the example of Section 3-4. An arbitrary decision was made to utilize the 4-pole Butterworth recovery filter. Utilizing a more complex filter or adding interpolation points by computer could have substantially reduced the required sample rate. Increasing the aliasing error requirements, on the other hand, could have pushed the sample rate in the other direction. Each specific problem must be considered on its own merits, based on the available or feasible hardware. There are no hard-and-fast, safe rules of thumb. Designers are also fortunate if they can find charts, tables, or curves to let them "cookbook" their problems. Unfortunately, too many diverse problems need solving to allow for such design aids. Beware of the charts and curves that do exist, since their authors are most often constrained to select a perfect, classical situation; so such charts seldom have widespread usage. To evolve a successful design, you must know what the input signal spectrum looks like, what you wish to obtain from it, and how to apply the rules of sampling, quantization, and recovery on an individual basis.

PROBLEMS

3-1. A quantizer has a full-scale input range of +10 V and a resolution of 15 bits.
   a. What is the value of a quantum interval?
   b. Suppose the quantizer had a resolution of 4 BCD digits. What is the value of a quantum interval?

3-2. Consider Figure 3-1. Why do the transition points from one output code to the next occur at the midpoints of the quantization intervals?

3-3. Using the results of problem 3-1, and considering no internal quantizer noise, what is the RMS quantization noise?

3-4. A simple sinusoid has an average value, a mean-square value, and an RMS value. Does it lend itself to the quantization "rule of thumb" in Section 3-2? Why?

REFERENCES

3-5. The highest frequency in a signal being reconstructed is kHz. The recovery requirements dictate that the maximum instantaneous error be less than 0.1 percent. What minimum sample rates would be required using:
   a. the zero-order hold,
   b. the first-order predictive hold, and
   c. the first-order linear hold?

3-6. The 3-dB point of a fourth-order Butterworth spectrum being recovered is 20 kHz. If an RMS recovery error of 0.1 percent allowable, what would the minimum sample rates be if y used:
   a. the zero-order and
   b. the linear recovery operator?
   c. Compare the results of a and b with those of problem 3-

3-7. Develop a system solution for the data acquisition example Section 3-4. But utilize the following systems requirements:
   a. dynamic range >58 dB
   b. aliasing error <0.2 percent
   c. recovery error <1-percent RMS
   d. aperture error <0.5 percent instantaneous at 20 kHz

REFERENCES