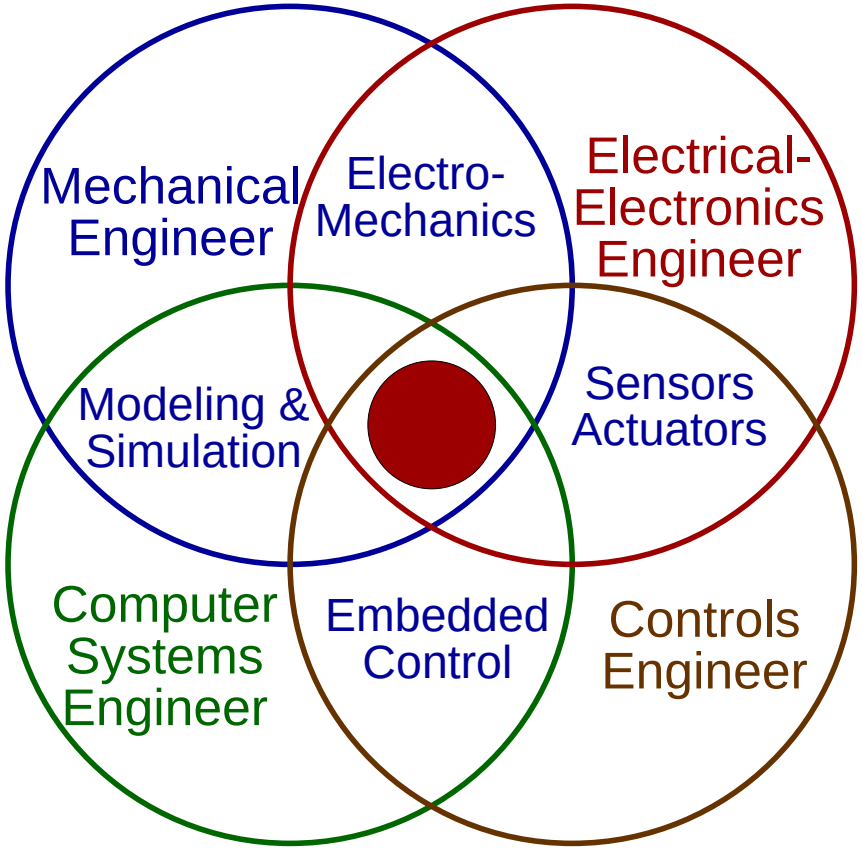
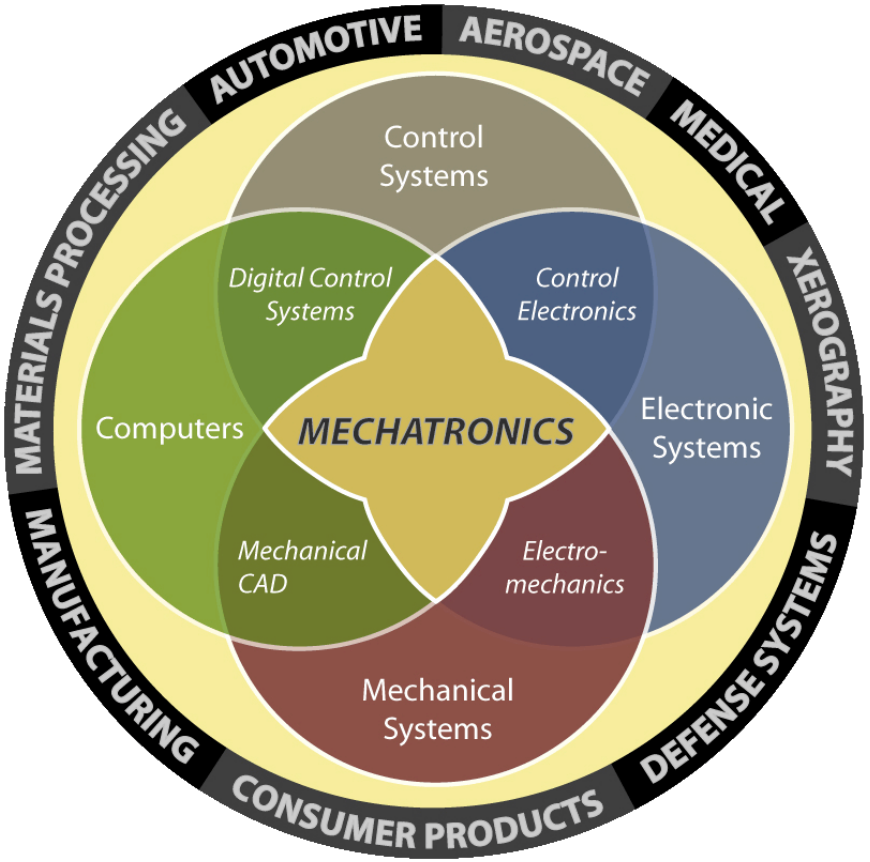
Digitization: Sampling & Quantization

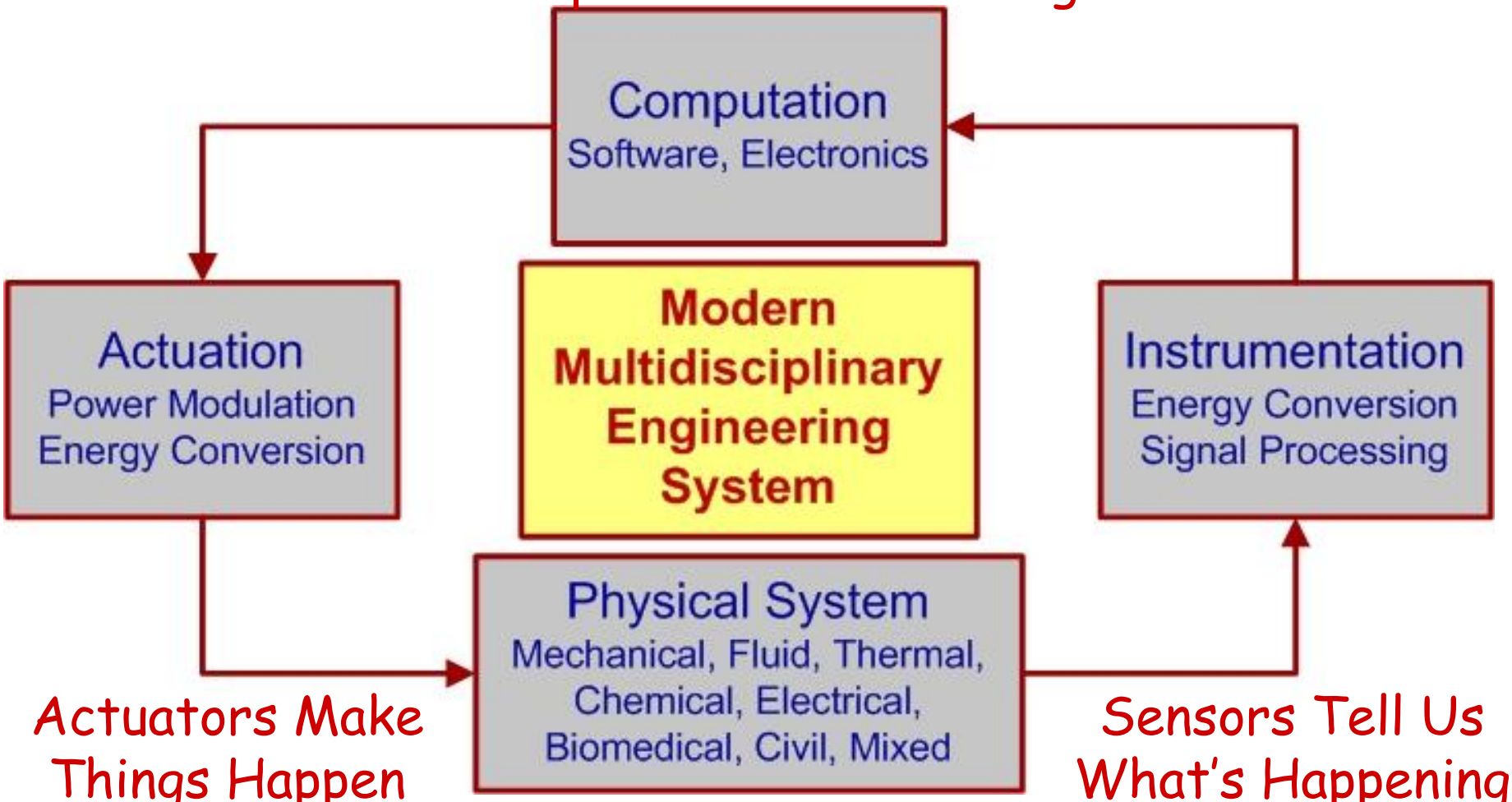




Motivation

- Why does every engineer need to know about digitization – sampling, aliasing, bits and bytes, and quantization?
- First, as an educated person, one should know how all the digital devices we use every day – computers, cell phones, TVs and cameras, CD and DVD players – work!
- Second, almost every engineering system is computer-controlled and the power domain (sensors, process, actuators) communicates with the information domain (computer, D/A and A/D converters) through digitization.
- All engineers need to be able to design and implement a digital control system and predict its behavior!

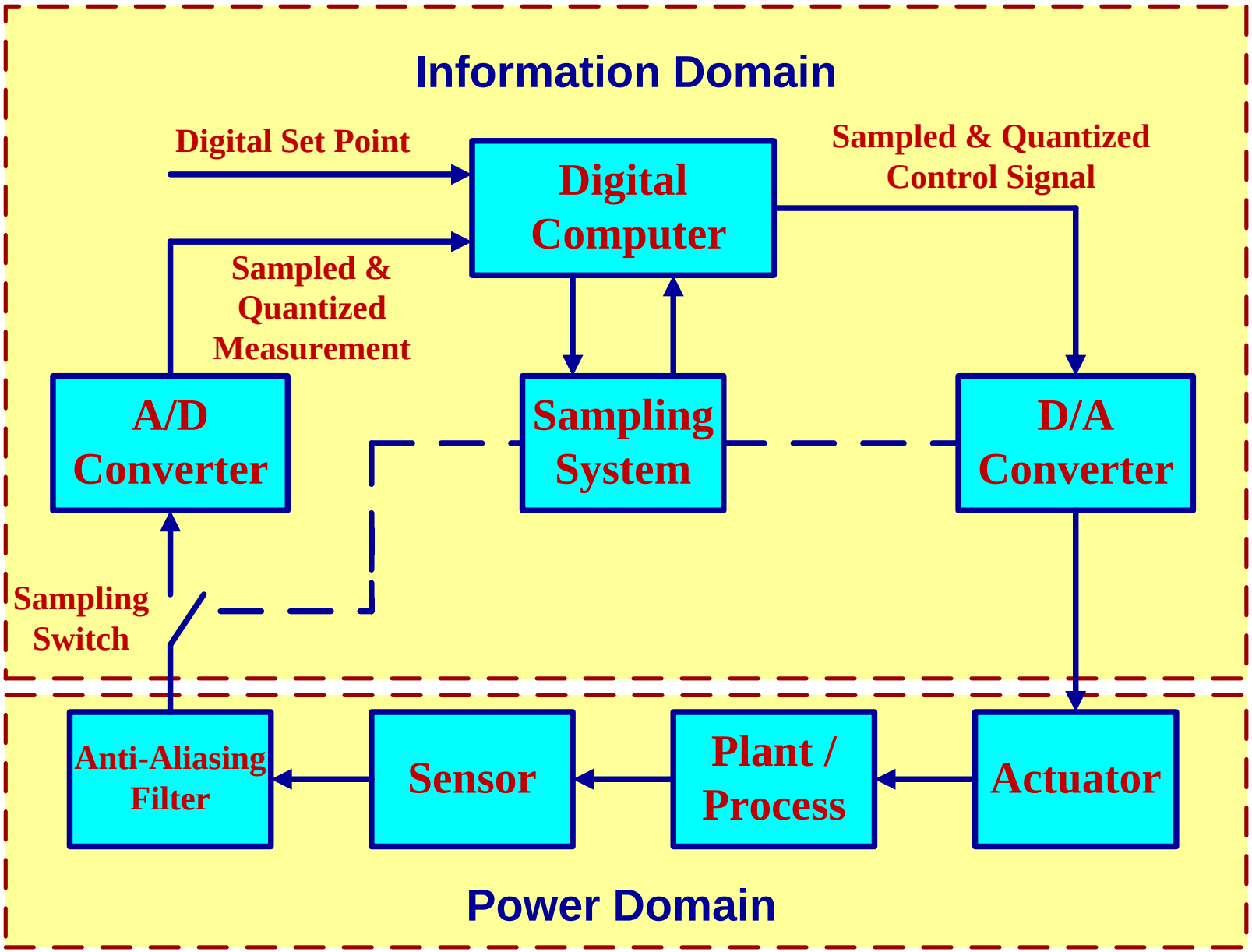
Computer Real-Time
Complex Decision Making



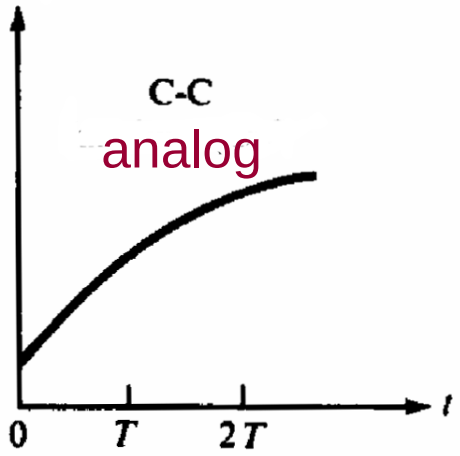
Actuators Make
Things Happen

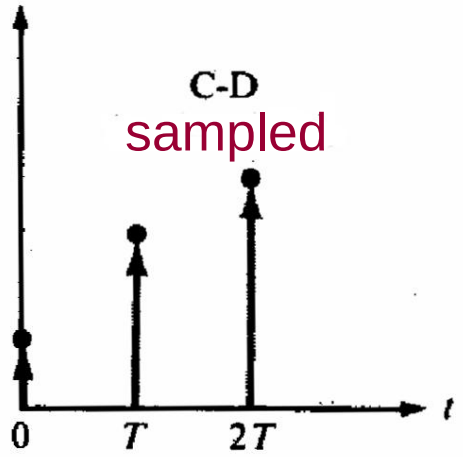
Sensors Tell Us
What's Happening

Plant / Process is the
Subject of Our Attention

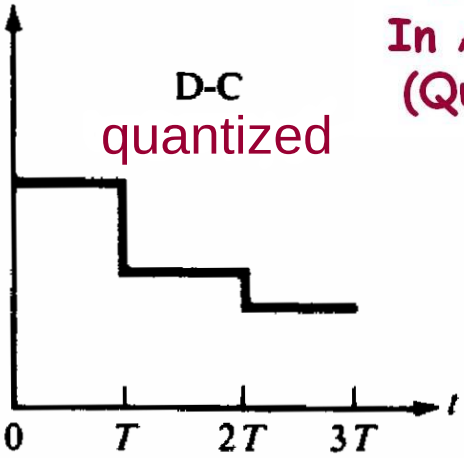

Information Domain
Digital Set Point
Digital Computer
Sampled & Quantized Control Signal
Sampled & Quantized Measurement
A/D Converter
Sampling System
D/A Converter
Sampling Switch
Anti-Aliasing Filter
Sensor
Plant / Process
Actuator
Power Domain

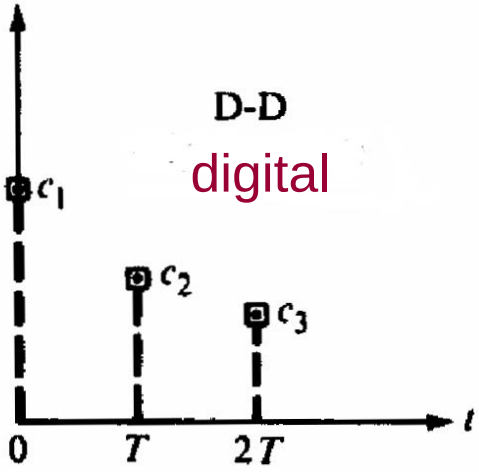
Computer-Controlled System Signals



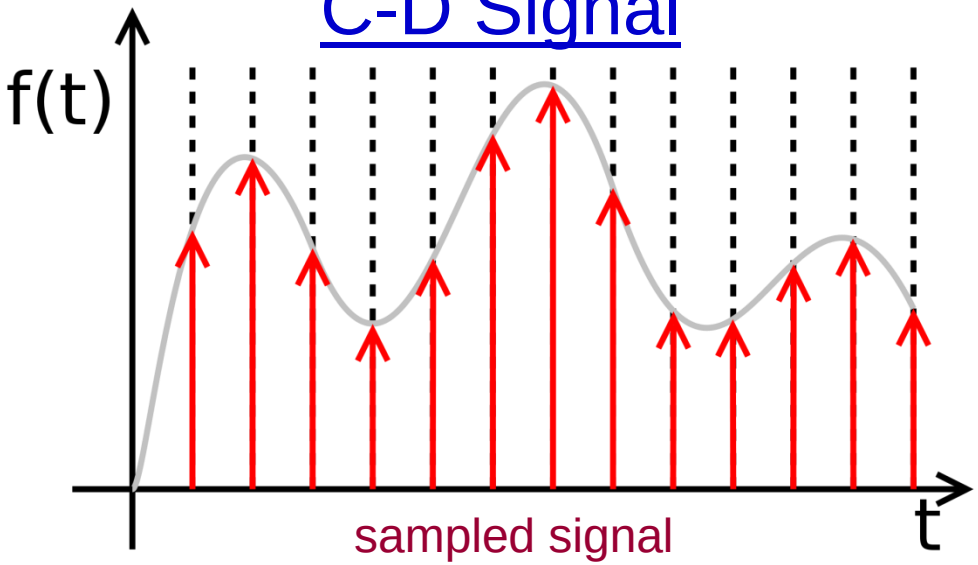


	Continuous In Time	Discrete In Time (Sampled)
Continuous In Amplitude	C-C	C-D
Discrete In Amplitude (Quantized)	D-C	D-D





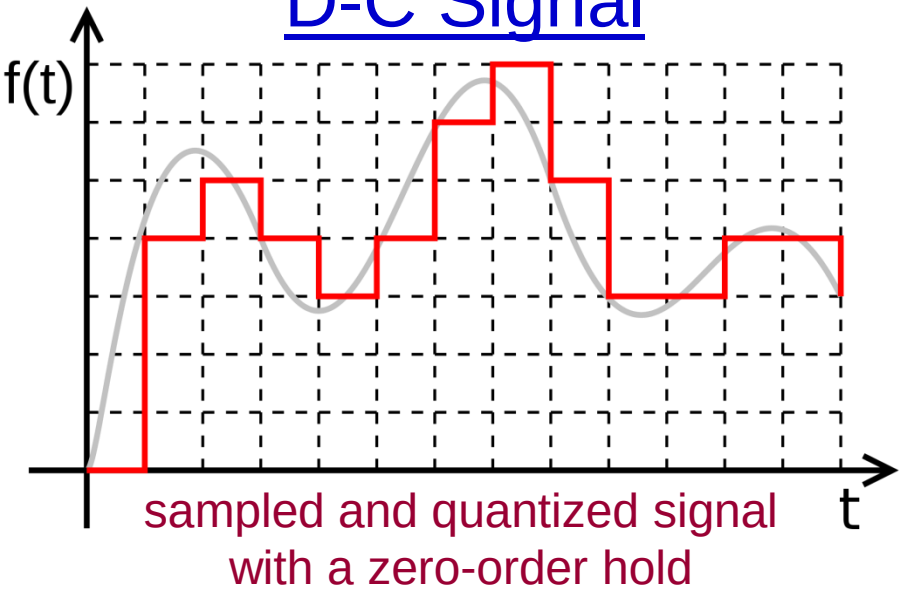
C-D Signal



Zero-Order Hold

Sampled and quantized values are held constant between sampling instants, which makes the signal discrete in amplitude, but continuous in time.

D-C Signal



Concept Questions

- What are the common elements in a modern multidisciplinary engineering system? Draw a block diagram of that system.
- In a computer-controlled engineering system, what is the power domain? What is the information domain?
- Using the descriptors continuous / discrete in amplitude and continuous / discrete in time, explain the following signals found in computer-controlled systems: analog, sampled, quantized, and digital.

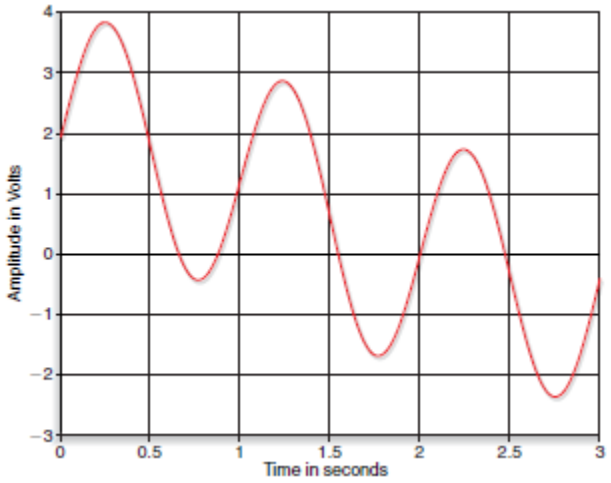
Everything is Going Digital!

- How do engineers take information such as sounds, pictures, and text and turn them into numbers that can be processed by our ever-increasing computing capabilities?
- Digitization is the process of representing various types of information in a form that can be stored and processed by a digital device. It is the combined operations of sampling and quantization, also called analog-to-digital (A/D) conversion.
- A sample is a numerical value representing the height of a waveform at a particular time, or the brightness of an image at a particular point.
- A digital signal is a set of sampled values represented in binary form as bits (binary digits) that can be turned back into the original form.

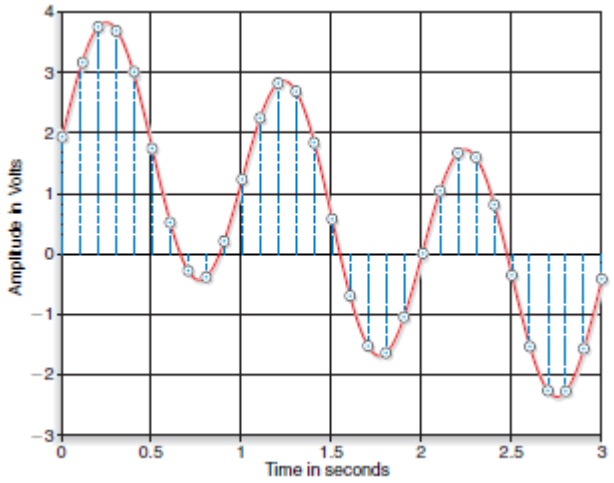
From Information to Numbers

- We see and hear continuous variations in color, light, and sound. Digital devices can record and process only sequences of numbers, not the original sounds, pictures, and movies.
- How is it possible that information can be turned into a list of numbers without losing any quality? Engineers need to deal with both technology and the limits of human perception.
- Sampling is the process of recording values (samples) of a signal at distinct points in time or space. Waveforms are sampled in time. Images are sampled in space. Time-varying scenes (video) are sampled in both time and space.
- How often should one take measurements to ensure that the numbers allow us to faithfully recreate the original information?

- We need to understand the basic concept of sampling. For simplicity, we will focus on sampling of a waveform in time.
- Sampling of a waveform refers to the process of recording values at distinct time points along a signal. Usually, samples of a signal are observed at uniformly-spaced time instants, e.g., every 0.1 sec, as shown below. The heights of the dots above or below the time axis (horizontal axis) represent the samples.



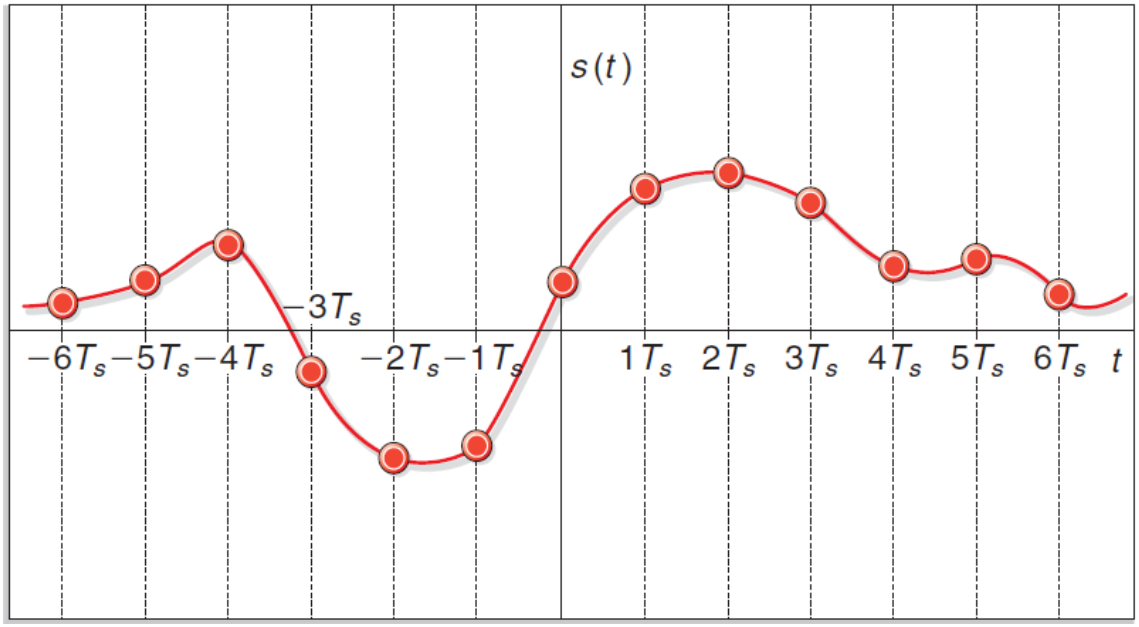
analog signal



samples of the analog signal

- An analog signal that varies quickly must be sampled more frequently than an analog signal that varies slowly.
- The sampling period T_s is the spacing between two adjacent samples, i.e., seconds per sample.
- The sampling rate or frequency f_s is the number of samples per second (Hz).

$$f_s = \frac{1}{T_s}$$

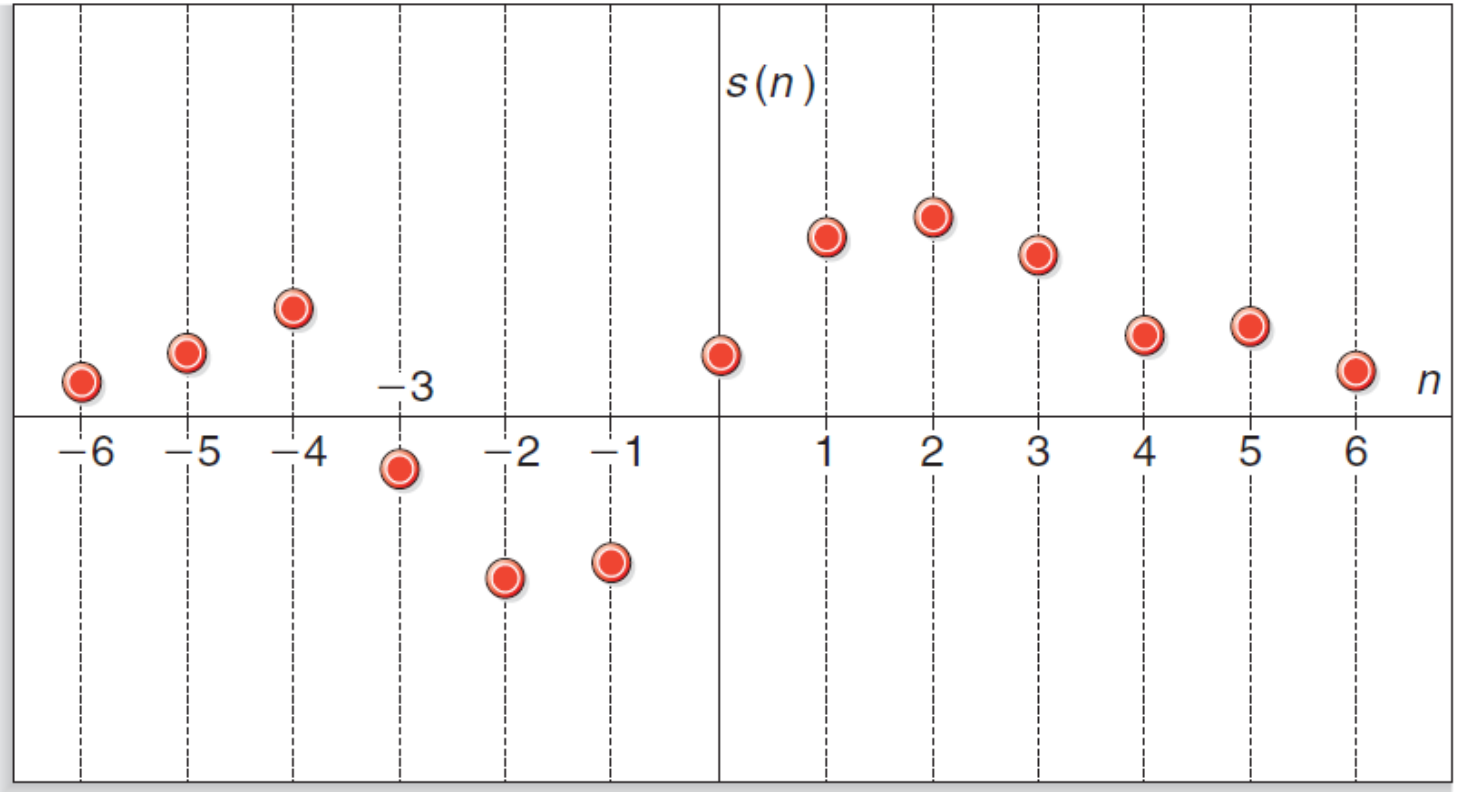


sampled values of an analog signal s(t)

- The samples of a signal are a sequence of numbers s[n].

$$s[n] = s(nT_s) \qquad n = 0, \pm 1, \pm 2, \ldots$$

- A sequence is an ordered set of numbers. A sequence of samples is the set of numerical sample values of a signal.

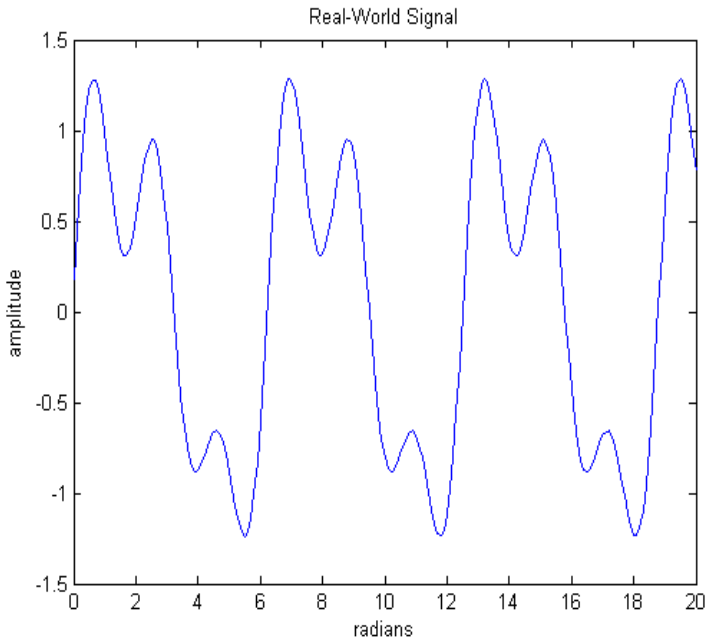


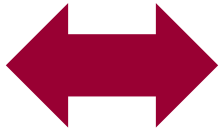
samples s[n] of the analog signal s(t)

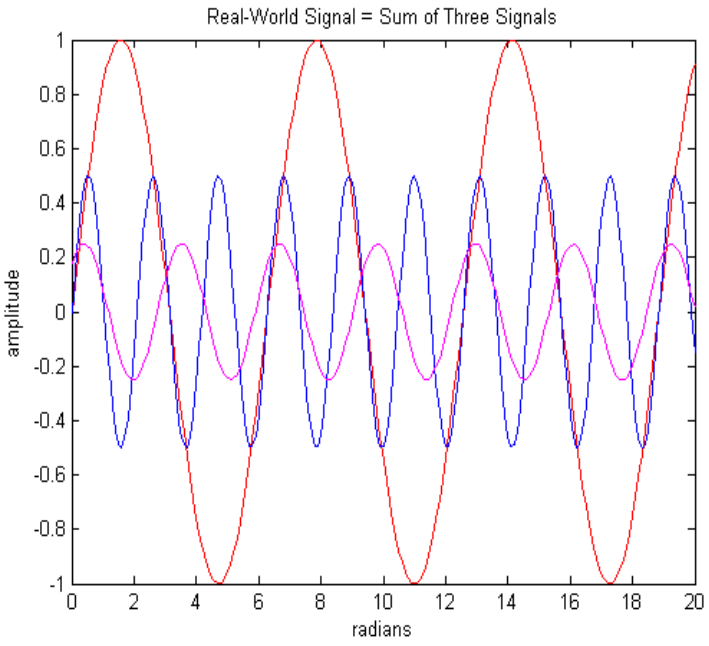
Converting Numbers Back into Analog Information

- How fast do we need to take measurements so that the digital signal, or list of numbers, becomes an accurate reflection of the original signal?
- We need to determine the appropriate sampling frequency.
- Let's consider the sampling of a sinusoidal signal that has a frequency f (cycles per second) and a period T (seconds per cycle). Remember $f = 1/T$.
- Complicated signals can be constructed as sums of sinusoids (see next slide). If we can figure out the necessary sampling rate for a sinusoid, we also can determine the necessary rate for other signals made with sinusoids.

Over one hundred years ago, Jean Baptiste Fourier showed that any waveform that exists in the real world can be generated by adding up sine waves. By picking the amplitudes, frequencies, and phases of these sine waves, one can generate a waveform identical to the desired signal.





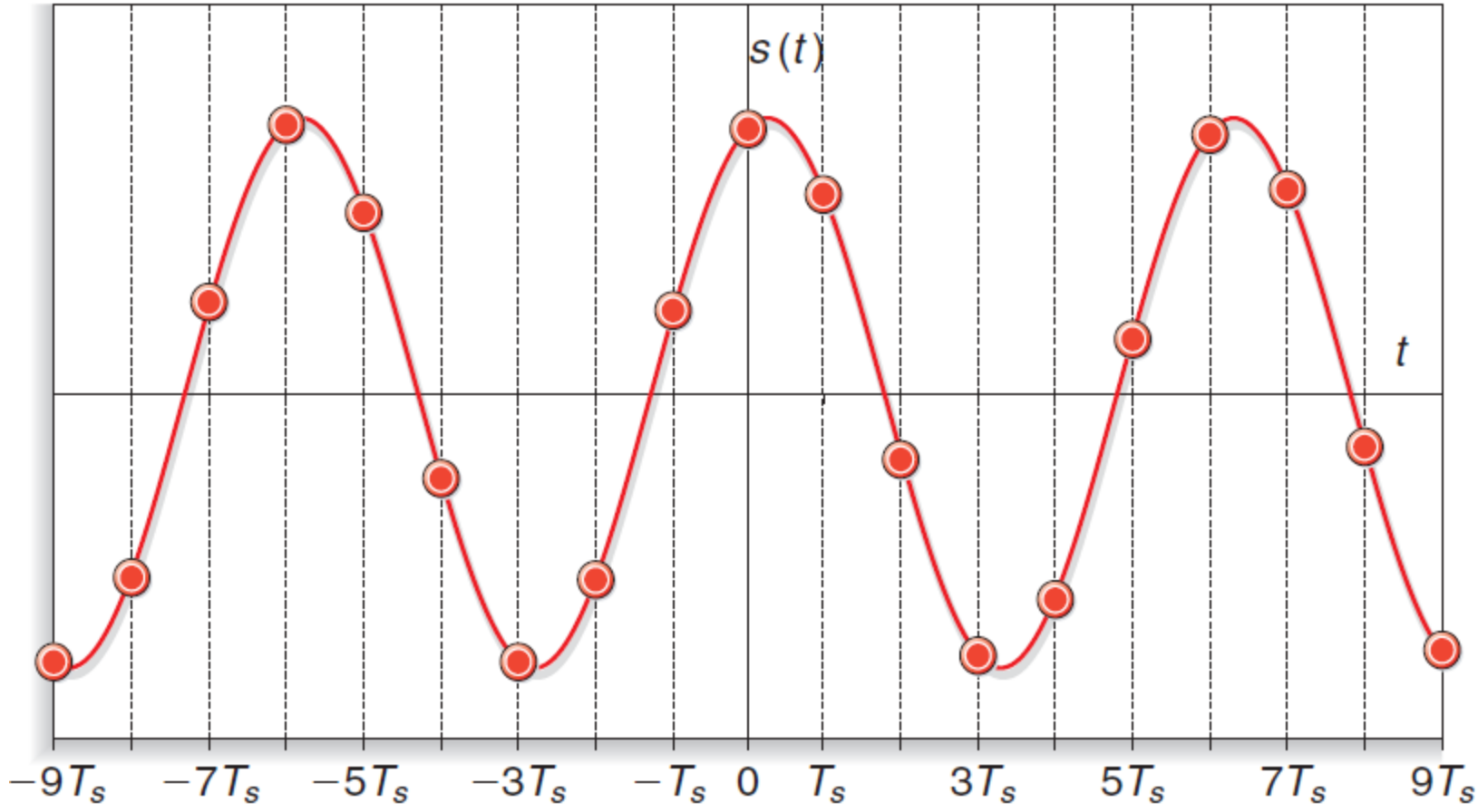


Real-World Signal

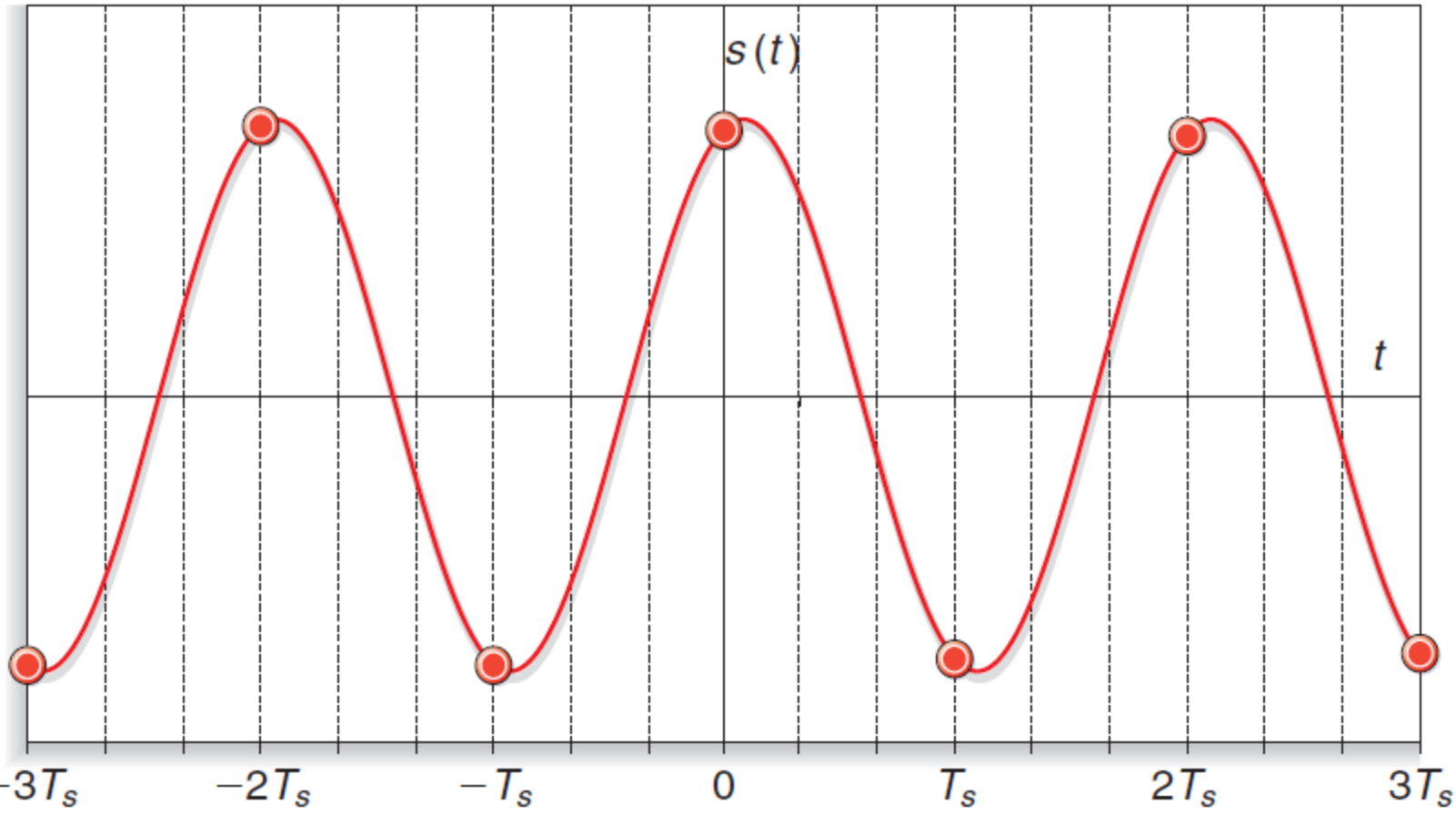
Three Component Signals

Any real-world signal can be broken down into a sum of sine waves and this combination of sine waves is unique.

- Below we see a sinusoid sampled at a moderate rate of several samples per period. T_s is the spacing between samples; $f_s = 1/T_s$ is the sampling frequency.



- Below we see the sampled sinusoid at a lower rate (two samples per period), which means that the sampling period is larger. Here $T_s = T/2$ and $f_s = 2f$. T = sinusoid period and f = sinusoid frequency.



- Some Observations
  - If we were given the samples s[n] for the figure on slide 15 and told that they came from some sinusoidal signal having a frequency no higher than that shown, we would have little trouble recreating the original analog signal.
  - However, for the situation shown on slide 16, notice that if the sinusoid were shifted to the right slightly so that the sinusoidal signal were equal to zero at time t = 0, then all the sample times would occur at times when the signal had a value of zero. Therefore, all of the samples would be zero. We would not be able to tell the difference between the original sinusoid and a signal that was always zero.
  - It seems that the sampling frequency $f_s = 2f$ is on the borderline of the minimum number of samples per second that is necessary in order to be able to recreate the simple sine wave from a list of numbers.

- What we have just observed leads us to *one of the most powerful mathematical results of the digital era*.
- Nyquist Sampling Theorem
  - A sampled signal can be converted back to its original analog signal without any error if the sampling rate is more than twice as large as the highest frequency of the signal. The Nyquist Frequency = ½ f_s and it is a discrete-time system property.
  - Restated in mathematical form, if T_s is the spacing between samples and $f_s = 1/T_s$ is the sampling frequency, then theoretically we can convert the samples of an analog signal back into the original signal if $f_s > 2f_{highest}$, where $f_{highest}$ is the highest frequency contained in the analog signal.
  - The value $2f_{highest}$ is called the Nyquist rate. $f_{highest}$ is the highest frequency contained in a signal. The Nyquist rate is the lower bound of the sampling frequency that satisfies the Nyquist sampling criterion. It is a continuous-time signal property.

- Shown below are the Minimum Sampling Rates for Various Signals. Here the signal bandwidth refers to the difference between the highest frequency in the signal and the lowest frequency in the signal. In general, the Nyquist rate can be taken to be two times the bandwidth of the analog signal. If the lowest frequency is zero, then the signal's bandwidth is equal to the highest frequency.

Signal Type	Signal Bandwidth	Minimum Sampling Rate	Rate Actually Used
Telephone-quality speech	3.5 kHz	7000 samples/s	8000 samples/s
Music	20 kHz	40,000 samples/s	44,100 samples/s
FM radio	200 kHz	400,000 samples/s	500,000 samples/s
Standard-definition television	6 MHz	12,000,000 samples/s	14,400,000 samples/s

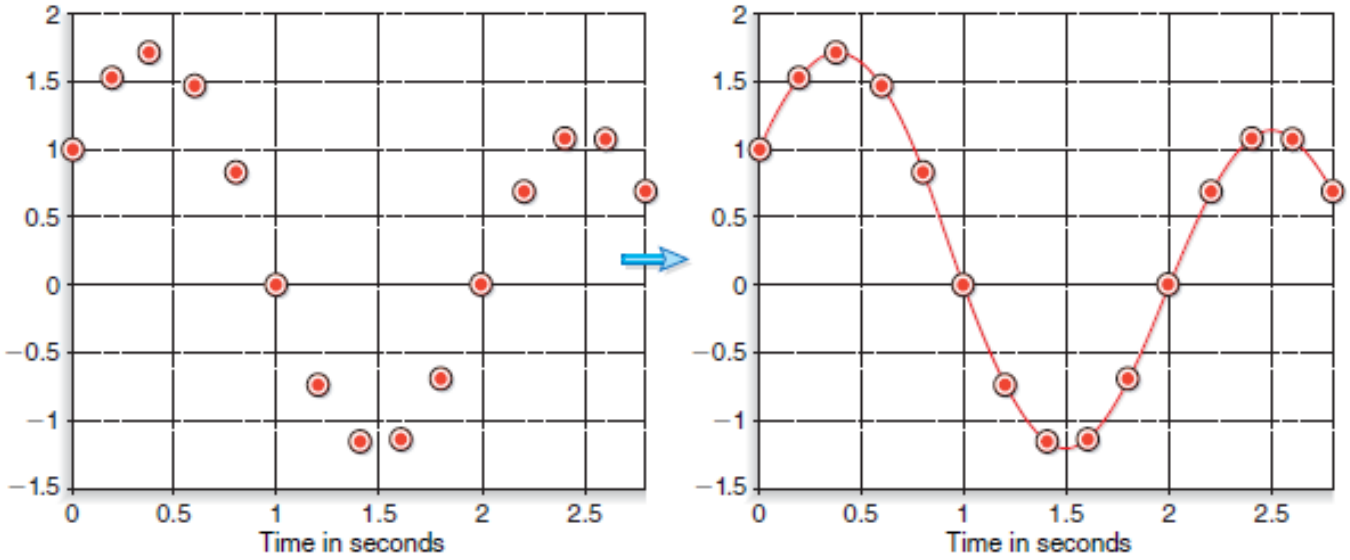
Nyquist Rate

- Mathematicians and engineers added to Nyquist's original sampling result by discovering precisely how to recreate the original signal from only its samples. They showed that if an analog signal is sampled at a rate greater than two times the bandwidth, then it can be exactly reconstructed from its samples. In fact, there is a mathematical formula for reconstructing the signal and it can be implemented in a very practical way. The reconstruction of the original signal from samples of the signal is done with a digital-to-analog (D/A) converter.
- The implications of the Nyquist sampling theorem are nothing short of remarkable!
- The process of sampling a signal, manipulating a sequence of numbers that results from sampling to remove noise or emphasize certain features, for example, and producing an analog output signal after the manipulations is called Digital Signal Processing (DSP).

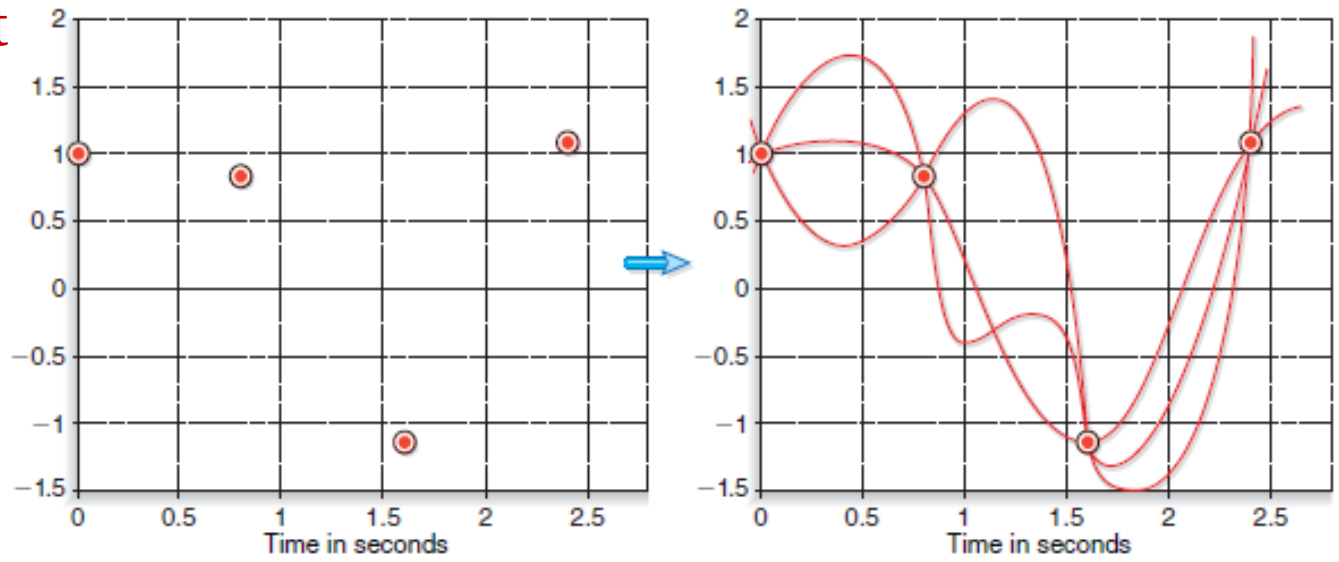
Bad Effects of Sampling Too Slowly

- Band-limited signals (signals whose highest frequency falls below some finite value) can be reconstructed perfectly from their samples, as long as the sampling rate is greater than twice the bandwidth of the signal that was sampled.
- What happens if we unfortunately sample a signal too slowly and fail to meet the requirement of the Nyquist sampling theorem?
- If we have sampled the original signal at a rate higher than the Nyquist rate, then there is only one analog signal that satisfies the measurements s[n] and has the bandwidth specified by $f_{highest} - f_{lowest}$. And there is a formula for finding that original signal!
- If we don't sample fast enough, it turns out that there are many signals that pass through the samples s[n].

Signal sampled above the Nyquist rate and reconstructed.



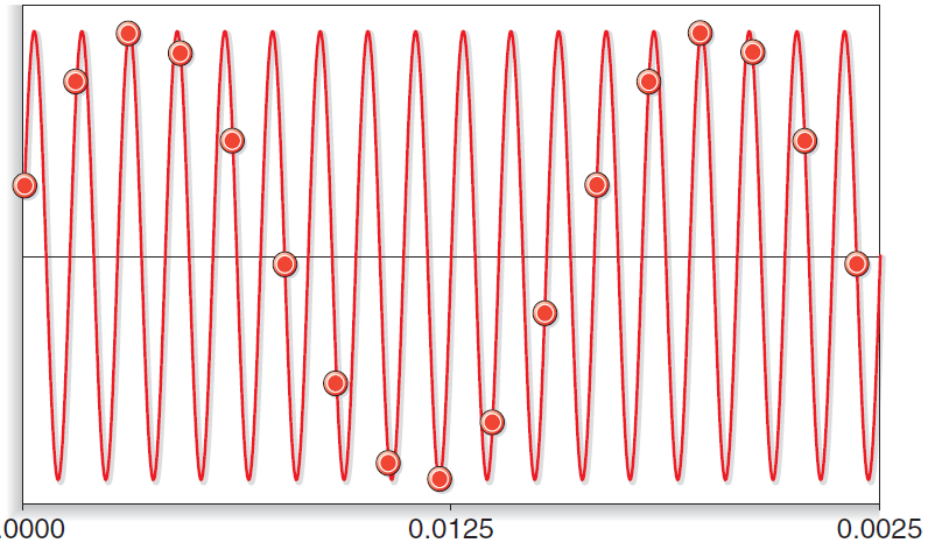
Signal sampled below the Nyquist rate. Many band-limited signals with bandwidth > $f_s/2$ that pass through the samples can be reconstructed.



Which signal is correct?

- Aliasing
  - Aliasing refers to one signal "pretending to be" another signal when their samples are the same. It is an undesired effect due to undersampling whereby one signal can masquerade as another.

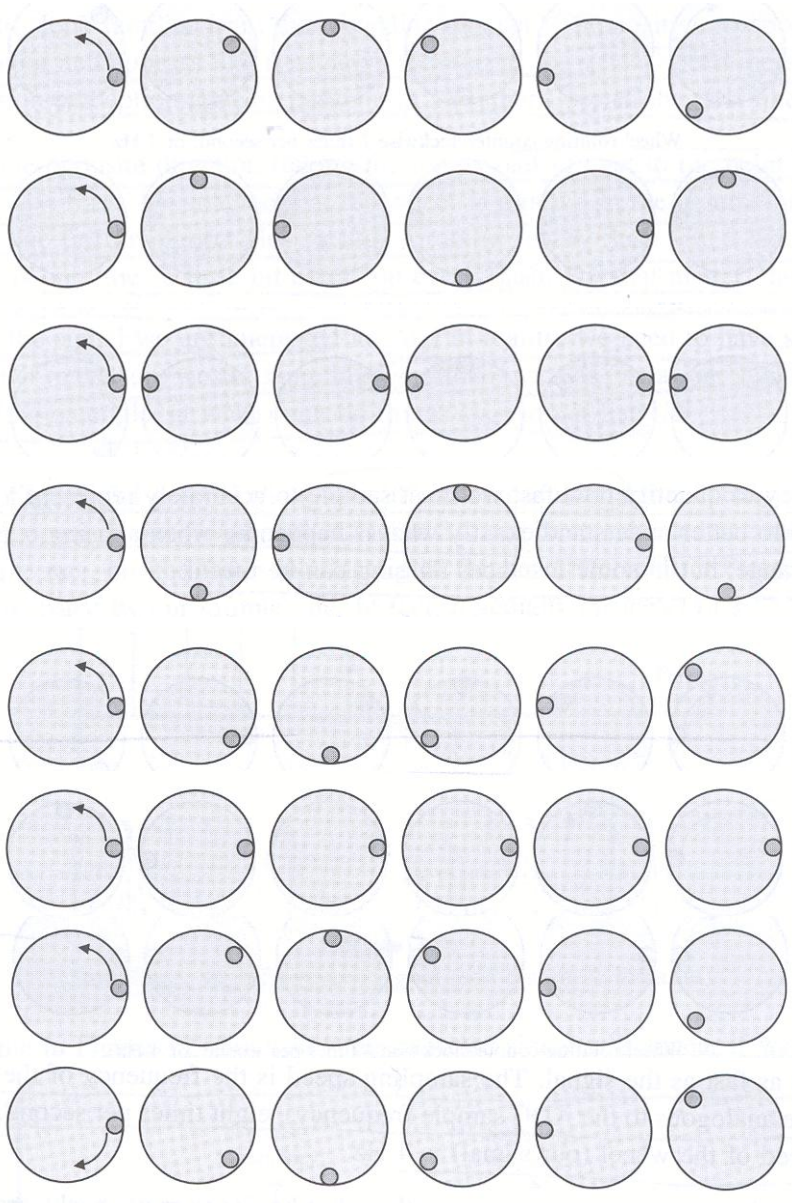
This 720-Hz sine signal has a Nyquist rate of 1440 Hz. What happens if we sample this signal at 660 samples per second – far below the Nyquist rate? A 60-Hz sine signal appears, not the 720-Hz sine signal.



Sampling a 720-Hz sinusoid at the rate of 660 Hz. Note that the samples appear to be coming from a sinusoid at a much lower frequency that does not exist. This effect is called aliasing.

How fast must we sample?

Imagine a spinning wheel rotating CCW at f Hz with a single dot near its edge. View the wheel with a strobe light set to flash 8 times per second (8 Hz).



← Which direction is wheel rotating?

← Dot appears to be moving backward.

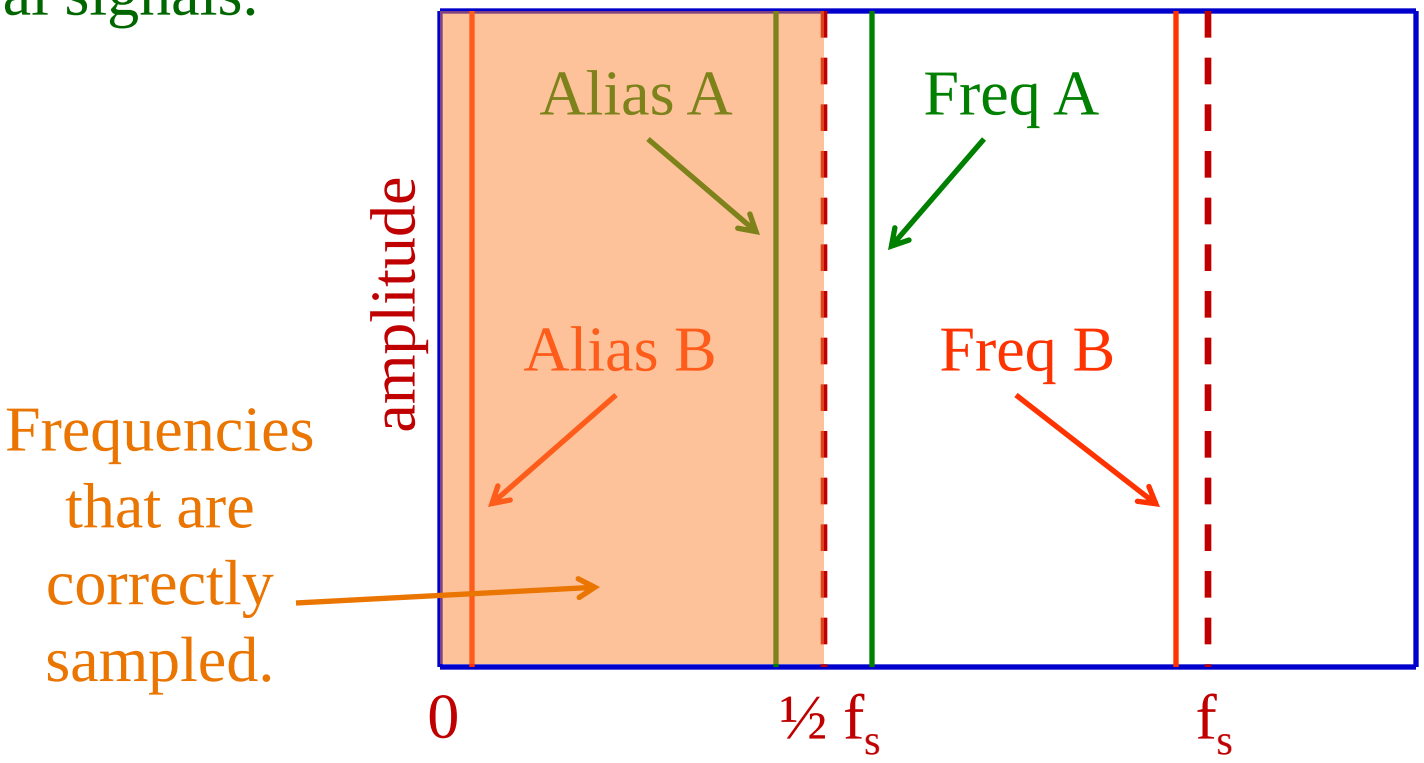
← Dot definitely appears to be moving backward.

← Is the wheel spinning forward or backward, or has it stopped?

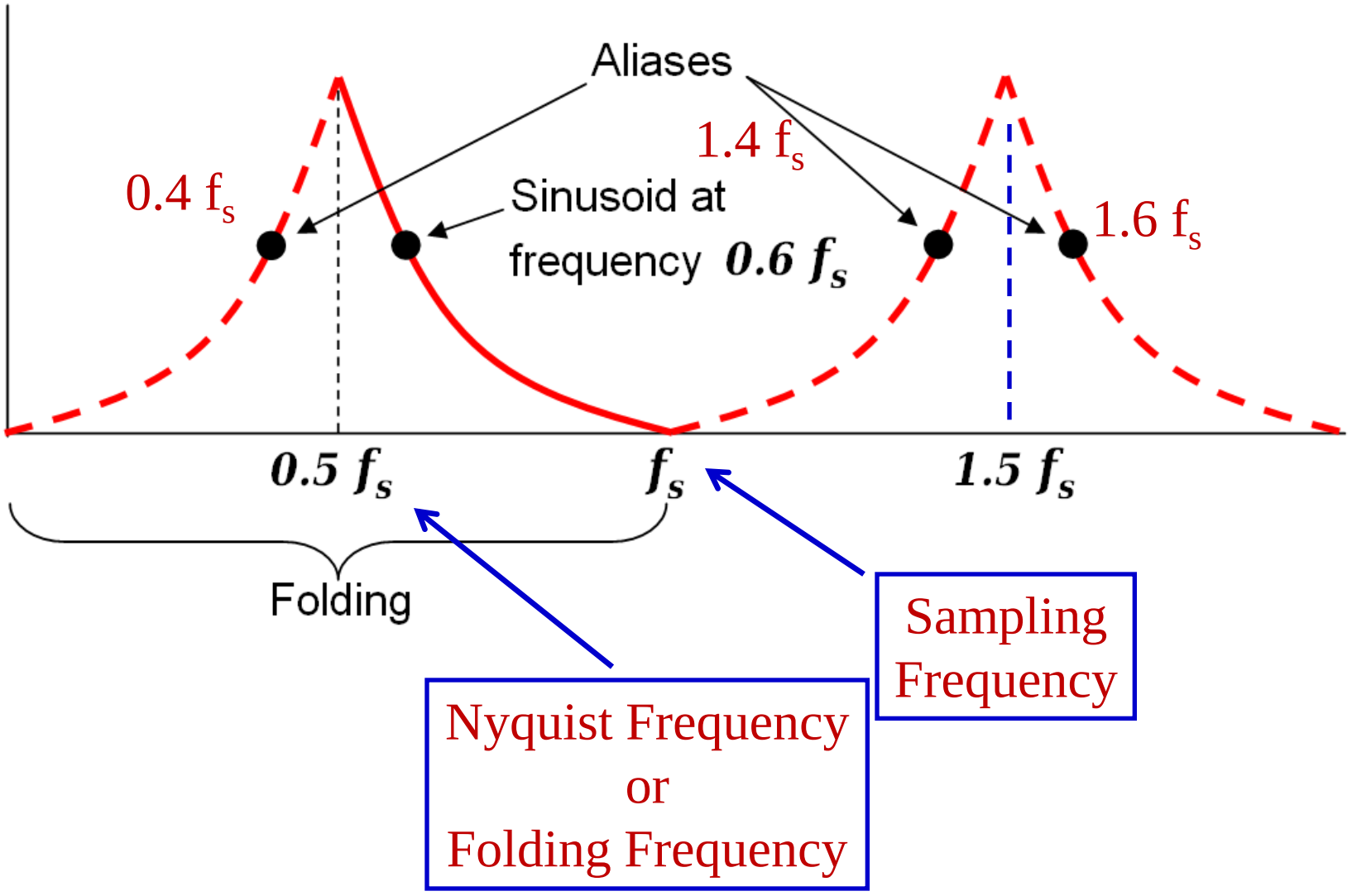
← Image looks the same as when the wheel was rotating at 1 Hz.

← Image looks the same as when the wheel was rotating at 7 Hz.

- Aliasing is an inevitable, irreversible process which shifts frequencies. It cannot be completely eliminated, only reduced.
- f_s is the sampling frequency and ½ f_s is the Nyquist frequency, also called the folding frequency. Frequencies A and B are above the Nyquist frequency and are aliased, folded back into the useful frequency range. They are fictitious signals indistinguishable from the real signals.



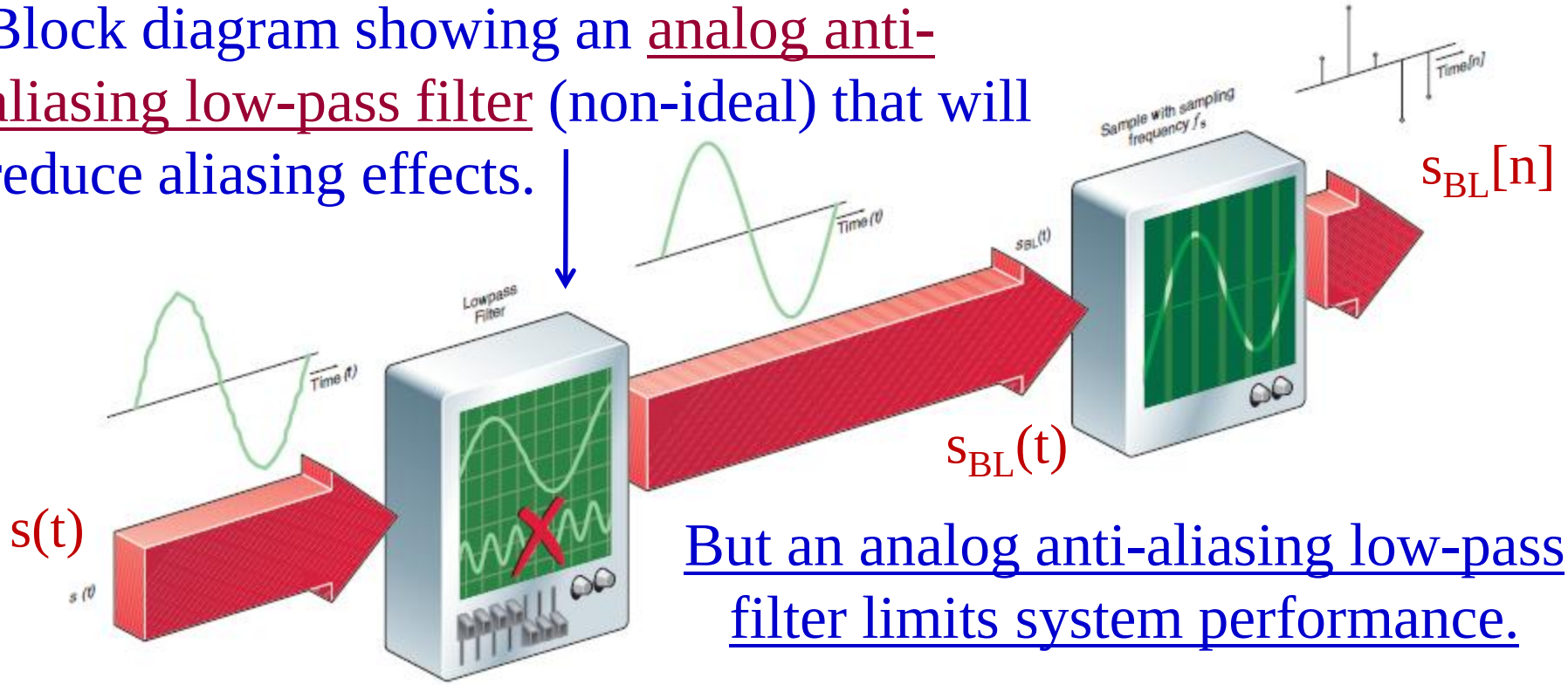
Aliasing


Aliases
1.4 f_s
0.4 f_s
Sinusoid at
frequency $0.6\ f_s$
1.6 f_s
$0.5\ f_s$
f_s
$1.5\ f_s$
Folding
Sampling
Frequency
Nyquist Frequency
or
Folding Frequency

- Limiting the Effects of Aliasing
  - Aliasing is a serious effect that is inevitable and irreversible and must be avoided if we are to properly digitize information.
  - Aliasing causes frequencies to change, whether the signals are sounds, images, or video.
  - For typical signals, images, and time-varying scenes that are composed of many frequencies, the effects are very complex, so we have to take measures to prevent aliasing.
  - In most cases where we can measure the bandwidth of s(t) and where we can choose the sampling rate, we can avoid aliasing by simply sampling faster than the Nyquist rate.
  - However, what if s(t) is not band limited (i.e., has an infinitely broad range of frequencies in it) or if we are not able to change the sampling rate?

- We must first pass the analog signal s(t) through a device called an analog low-pass filter (device that attenuates all frequencies in a signal above some selected frequency and passes all others). This filter will reduce the effect of frequency components above ½ the sampling frequency. Such a filter is called an anti-aliasing filter.
- This filtering operation will cause the signal to be different, because it will attenuate the high frequencies in s(t). However, we would not be able to use these high frequencies anyway because they are above the Nyquist limit. Attenuating them prevents them from corrupting the lower frequencies that we can use. The filter does add time delay and thus limits system performance.
- The overall scheme is shown on the next slide. Here, s(t) is the original analog signal, which is not band-limited. The analog anti-aliasing filter attenuates the high frequencies in s(t), creating the new, band-limited, analog signal $s_{BL}(t)$. This signal is then sampled faster than the Nyquist rate, producing the set of samples $s_{BL}[n]$.

Block diagram showing an analog anti-aliasing low-pass filter (non-ideal) that will reduce aliasing effects.



But an analog anti-aliasing low-pass filter limits system performance.

A common example is the telephone system. Our voices have a maximum frequency of about 3600 Hz. At the microphone, our voice is filtered by an analog filter to substantially reduce frequencies above 3600 Hz. Then the microphone signal is sampled at 8 kHz. If you hear music in the background while on a telephone, the music will sound flat or distorted. Our ears can detect up to about 15 kHz frequencies, and music generally has frequency content exceeding 3600 Hz. Little of this frequency content will be passed through the telephone system.

Concept Questions

- What is the Nyquist Sampling Theorem?
- What is the Nyquist rate? What is the Nyquist frequency?
- What is the folding frequency? What does that mean?
- What is the definition of band limited?
- What is DSP?
- What happens when a sinusoid is sampled above the Nyquist rate? below the Nyquist rate?
- What is an anti-aliasing filter? Why is it not ideal? Why does it limit system performance?
- Suppose a wheel with 12 spokes is rotating at 1.5 revolutions per second. Will a camera frame rate of 24 HZ cause aliasing?

Binary Numbers – The Digital Choice

- Computers and digital devices represent information in a binary number code. The smallest unit is called a bit (binary digit) which can represent only one of two states – 0 or 1. To represent more than just two possible states, we group many bits together.
- Why bits? Why binary number representation and arithmetic?
  - The transistor is the basic building block of nearly all digital technologies and this can be designed to act like a switch having two distinct states that can be controlled by currents and voltages.
  - In addition, semiconductor memory (RAM), magnetic disks, and optical discs store only one of two possible states – 0 or 1 – at each physical location on the device.

- All samples, or measurements, of signals, such as music or pictures, can be stored as numbers in a computer. However, computers and digital memories store only bits.
- Computers and digital storage devices use the binary number system rather than the decimal number system. The binary number system is an arithmetic system for representing numbers as a series of bits. It has only two symbols – 0 and 1. We write larger numbers with groups of 0's and 1's.
- Two bits together can represent (0, 0), (0, 1), (1, 0), and (1, 1) or four different states or values. Three bits can represent (0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), and (1, 1, 1) or eight different states or values.
- In general, B bits, where B is a positive integer, can represent 2^B states or values, e.g., four bits can represent 16 states; five bits can represent 32 states. The largest integer that can be represented by a B-bit binary number has all bits set to 1 and a value of $2^B - 1$.

- In the decimal number system, the number 10 is the base or radix.

100 10 1

10^2 10^1 10^0

Decimal representation of the number 238.

$$238 = 2(100) + 3(10) + 8(1)$$

$$238_{10} = (2\times10^2)+(3\times10^1)+(8\times10^0)$$

- In the binary number system, the number 2 is the base.

8 4 2 1

2^3 2^2 2^1 2^0

Binary representation of the decimal number 13.

$$13 = 1(8) + 1(4) + 0(2) + 1(1)$$

$$1101_2 = (1\times2^3)+(1\times2^2)+(0\times2^1)+(1\times2^0)$$

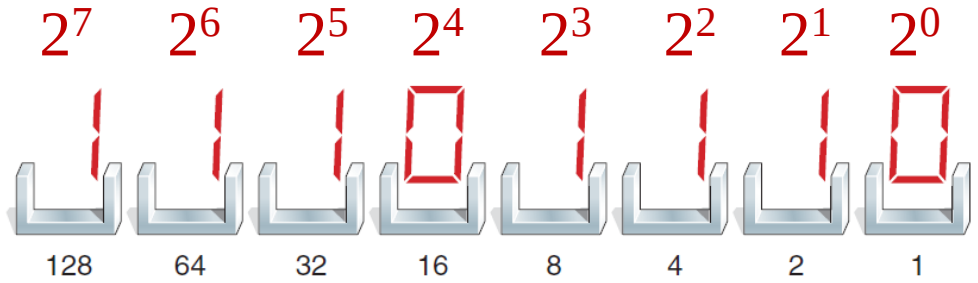
$$= 13_{10}$$

- Converting from binary to decimal is easy. For example:

$10101_2 = 21_{10}$

$101100_2 = 44_{10}$

$11101110_2 = 238_{10}$ ⟶

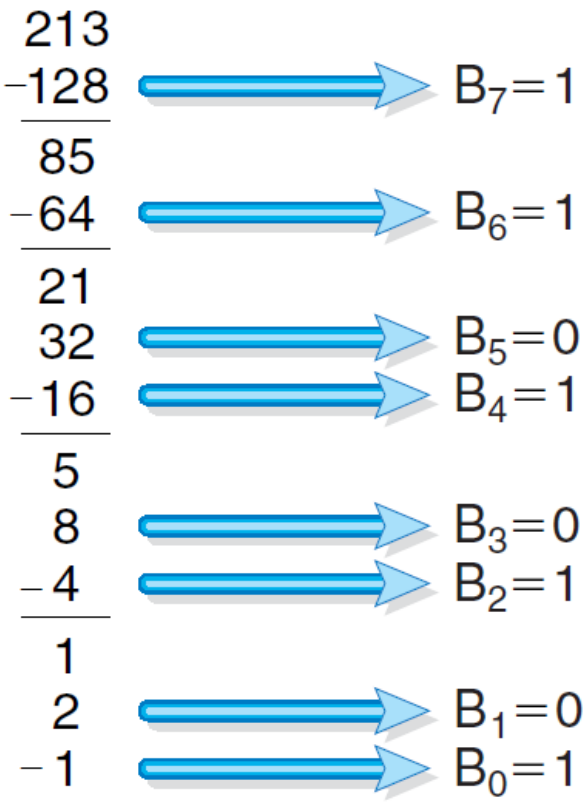


- Conversion from decimal to binary is also straightforward. As an example, convert the decimal number 213 to binary.

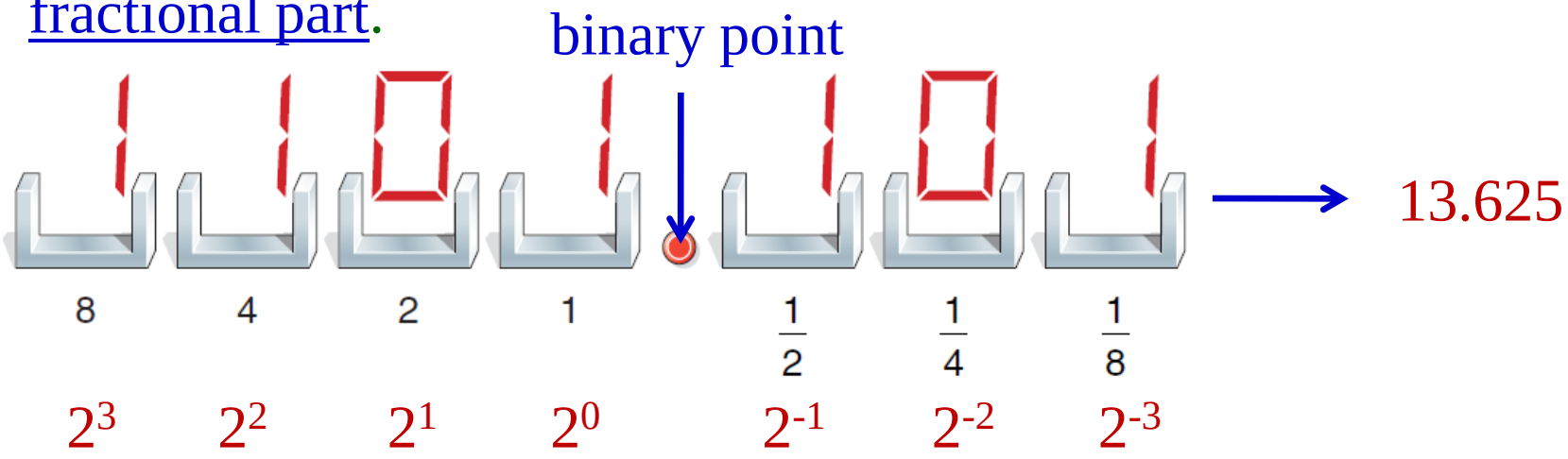
$$B_7(2^7)+B_6(2^6)+B_5(2^5)+B_4(2^4)+B_3(2^3)+$$
$$B_2(2^2)+B_1(2^1)+B_0(2^0)=213$$

$$B_7(128)+B_6(64)+B_5(32)+B_4(16)+B_3(8)+$$
$$B_2(4)+B_1(2)+B_0(1)=213$$

$11010101_2 = 213_{10}$



- In a binary number, the leftmost 0 or 1 is called the most significant bit because it has the most weight in the binary representation. The rightmost 0 or 1 is called the least significant bit because it has the least weight in the binary representation.
- In most cases, the numbers we want to store in a computer or digital device are not integers. Usually, there is a fractional part.



$$1101.101_2 = (1\times 2^3)+(1\times 2^2)+(0\times 2^1)+(1\times 2^0)+(1\times 2^{-1})+(0\times 2^{-2})+(1\times 2^{-3})$$

$$13.625_{10} = (1\times 10^1)+(3\times 10^0)+(6\times 10^{-1})+(2\times 10^{-2})+(5\times 10^{-3})$$

- Signals often can be negative numbers. Sinusoidal signals are negative half of the time. There are numerous ways to represent negative numbers in the binary number system.
- The method most similar to the way we handle decimal numbers is to a add a single leftmost bit to the representation, where a 0 value for that bit indicates a positive number and a 1 indicates a negative number.
  - For example, if 101_2 is two bits plus a sign (leftmost) bit, it represents the decimal number -1. Similarly, 01010_2 represents +10.
  - The combined representation of negative and positive numbers is called sign-magnitude form. It is one of several ways to represent both positive and negative binary numbers.
  - Note that there are two ways to represent zero in binary form, just as in decimal form, where +0 = -0.

Concept Questions

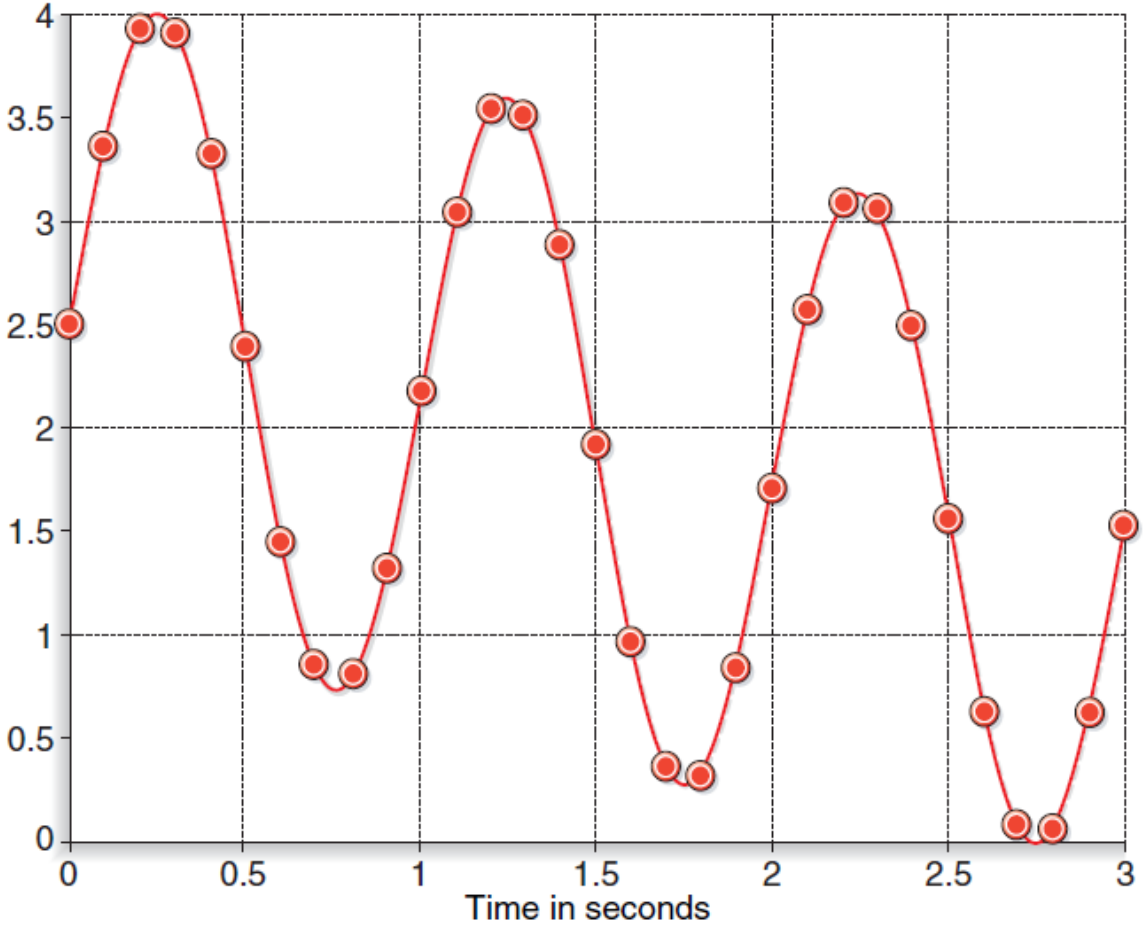
- What is a bit? What is a byte?
- Explain why computer storage devices store information as bits as opposed to some other way.
- What are the three types of storage methods commonly used in digital devices today?
- What is meant by most significant bit and least significant bit?
- What is the ratio of the most significant bit to the least significant bit?
- What is the base, or radix, of a numbering system?

Using Bits to Store Samples: Quantization

- In sampling, we assumed implicitly that computers store the numerical values of samples of a signal exactly, without any loss of accuracy or precision. This is not the case!
- Computers store numbers with fewer decimal places than the actual real number typically would have. This is really no different than when you take measurements in an experiment and choose to write down values using only a few decimal places.
- Real-world digital memory devices use only a fixed number of bits for the storage of each number, which implies limited or finite accuracy. The accuracy achieved in storing a sequence s[n] of numbers depends on how many bits are used to store each sample.

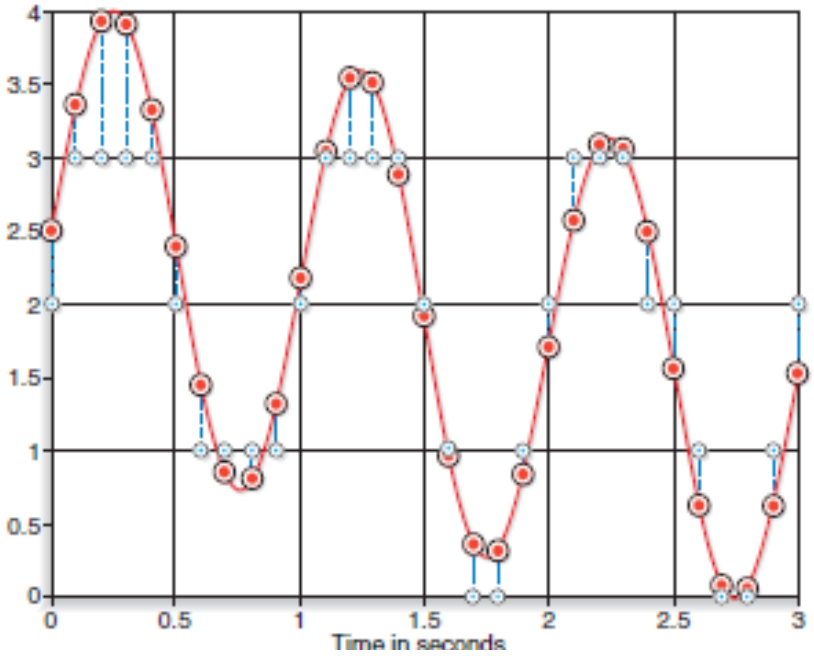
Shown is a sampled analog signal. Using binary numbers with a fixed number of 0's and 1's, we can store only a finite number of signal values.

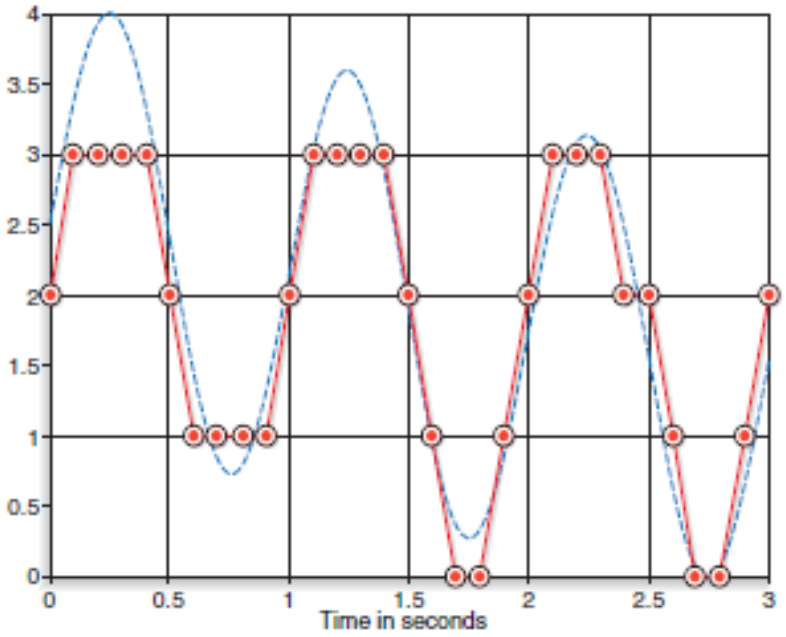
We should use the smallest number of bits that will provide the accuracy we need for the given engineering application.



sampled analog signal

True value	Quantized value	Binary number
2.5000	2	10
3.3783	3	11
3.9132	3	11
3.8966	3	11
3.3288	3	11
2.4183	2	10
1.4997	1	01
0.9179	1	01
0.8740	1	01
1.3712	1	01
2.2020	2	10
3.0304	3	11
3.5203	3	11
3.4644	3	11
2.8633	3	11
1.9261	2	10



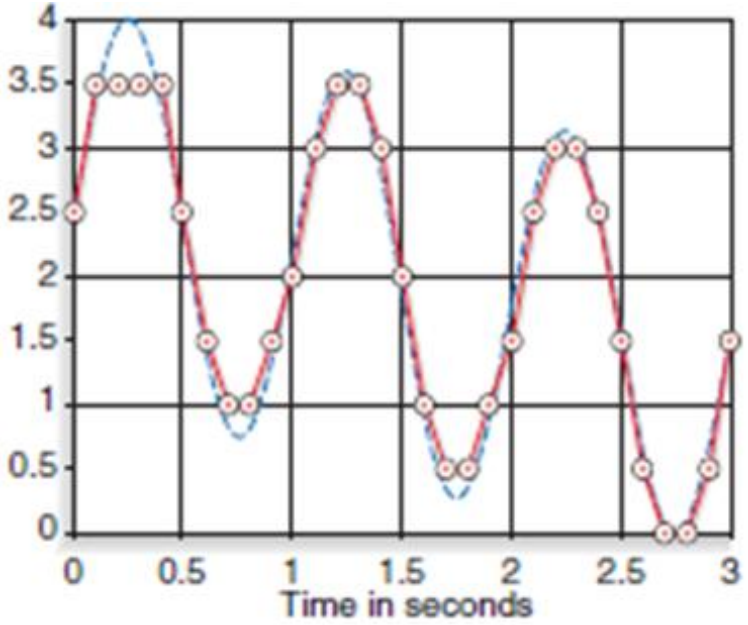


True value	Quantized value	Binary number
0.9901	1	01
0.3928	0	00
0.3430	0	00
0.8413	1	01
1.6800	2	10
2.5228	3	11
3.0333	3	11
3.0035	3	11
2.4338	2	10
1.5323	2	10
0.6359	1	01
0.0811	0	00
0.0761	0	00
0.6205	1	01
1.5060	2	10

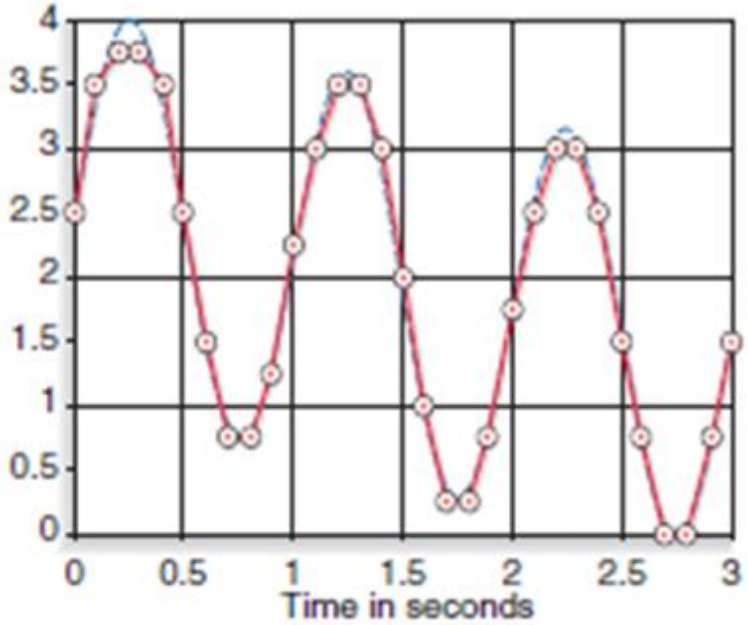
Here we are storing each sample with only two bits (4 possible stored states: 0, 1, 2, 3).

For each signal value s[n], we then round the true value to the nearest of the four levels. This process of signal rounding is a form of quantization.

- Quantization is the process of changing sample values to discrete levels. Rounding, whereby the sample values are changed to the nearest levels, is the most common form of quantization.
- The approximation of samples using only four quantization levels (2 bits) to store each sample is a very crude one. We could greatly improve the accuracy of our digitized information by using more bits per sample, which most engineering applications due.
- Using B bits to store each sample gives us 2^B quantization levels.
  - 2 bits give 4 levels, 3 bits give 8 levels, 4 bits give 16 levels, 8 bits give 256 levels, 10 bits give 1024 levels, 12 bits give 4096 levels, and 16 bits give 65,536 levels.
  - As the number of levels increases, the error associated with the quantization operation goes down.

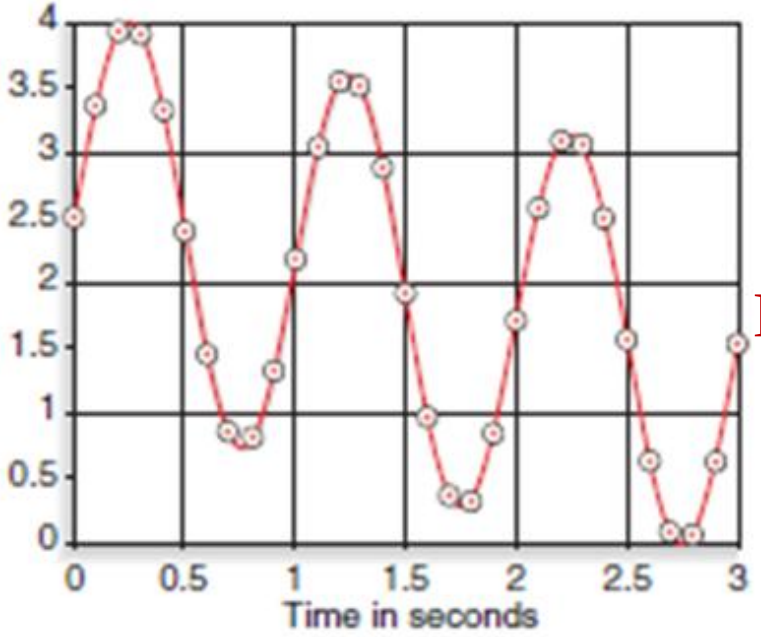


3 Bits



4 Bits

Shown is an analog signal quantized using 3 bits, 4 bits, and 16 bits. Notice how accurate the samples are when stored using 16 bits.

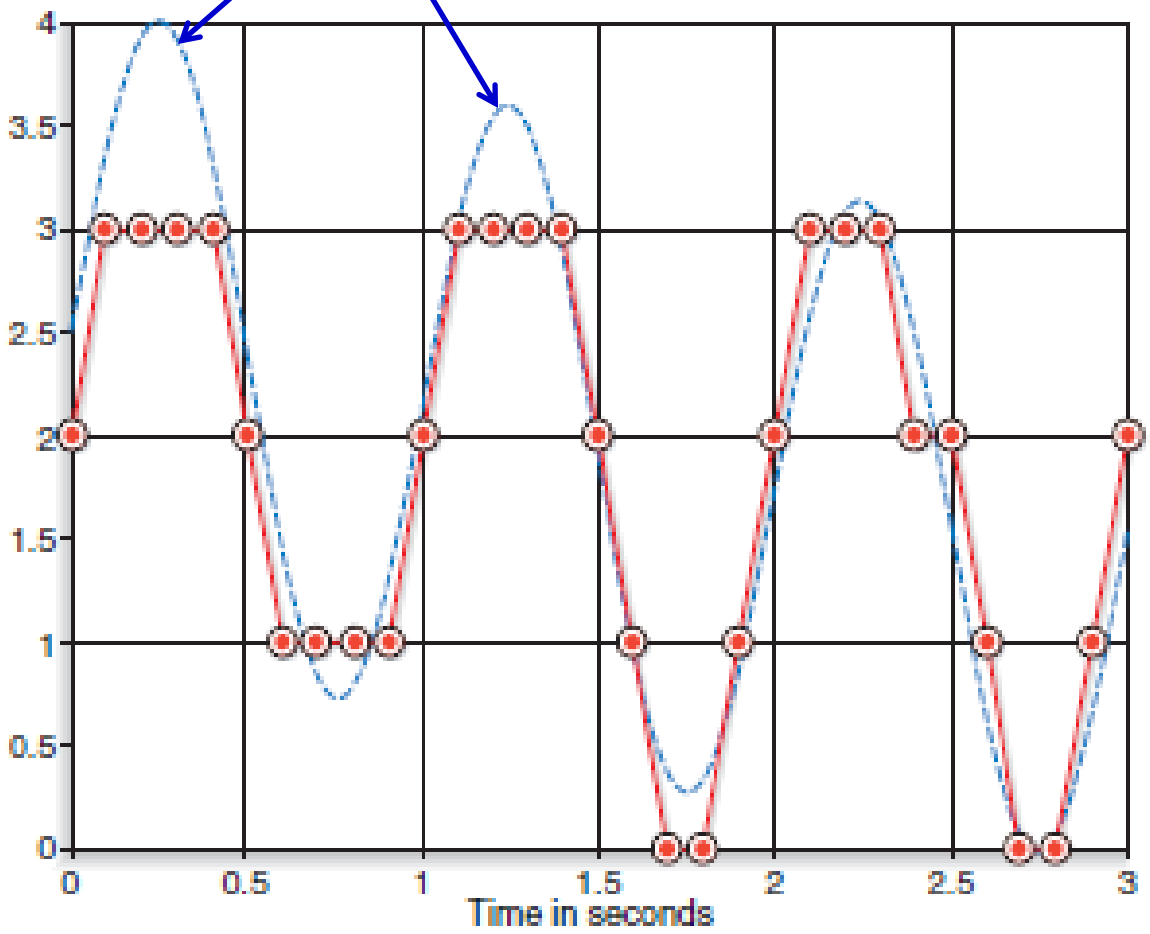


16 Bits

- Samples can have fractional values. Fractions are not a problem, because they can be represented in binary form. Usually, the quantization interval is not equal to 1, and the signal can be both positive and negative.
- In general, a signal may have a dynamic range (the difference between the largest and smallest amplitudes or intensities that a signal takes on) extending from +A to –A, in which case the total interval size of 2A would be divided up into $L = 2^B$ amplitude levels for B-bit binary representation.
- The levels should be set to cover the full dynamic range of the signal; otherwise, there will be amplitude clipping.
- Amplitude Clipping is an error caused in the quantization operation if the quantizer levels do not cover the full dynamic range and results in the tops and bottoms of the signal being "clipped off."

Severe Clipping

We could have greatly reduced the clipping by making a different choice for the quantization levels. It would be better to set the levels higher at 0.5, 1.5, 2.5, and 3.5, rather than at 0, 1, 2, and 3, as shown.



Signal Quantized using 2 Bits

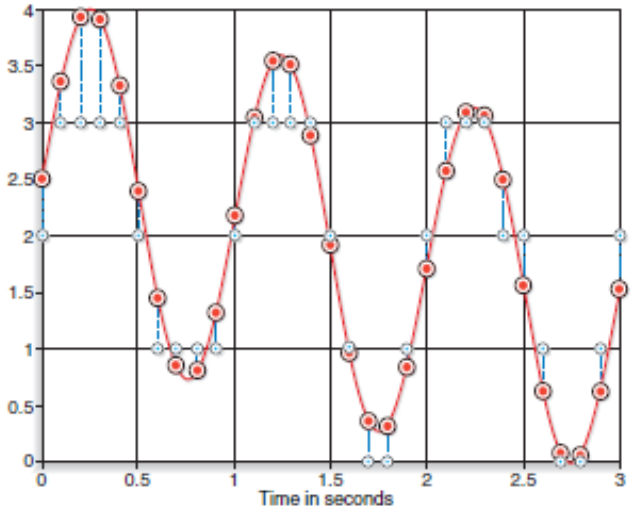
- It is equally important that the quantization levels not be set too high. The signal being sampled may not rise above the first couple of levels, in which case the extra bits used to represent the higher levels would always be 0 and would be wasted.
- Keep in mind that a binary representation with B bits can represent 2^B levels of signal height or image intensity. More bits provide higher fidelity – better music and better pictures, for example. However, more bits also require more storage, more processing, and longer transmission times.
  - Interesting Fact: Music on CDs is stored using 16 bits per sample. With a sampling rate of 44,100 samples per second, 254,016,000 bits are used to store a 3-minute song in stereo.

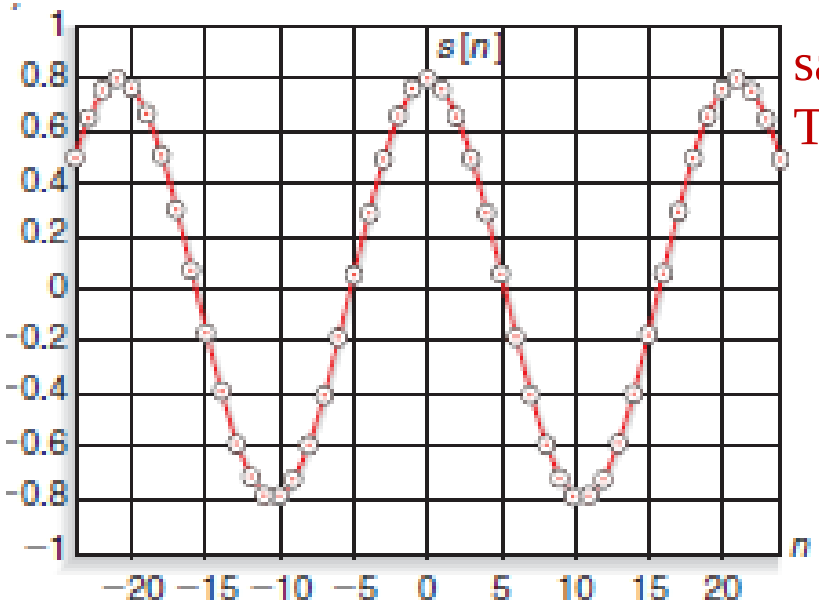
- No matter how quantization is performed, the number of bits per sample places the ultimate limit on the accuracy of the digitized values of the information.
- The combined processes of sampling of an analog signal and quantization of the samples to binary numbers are called digitization or analog-to-digital (A/D) conversion.
- The electric circuit that performs the conversion is called an analog-to-digital (A/D) converter. Most A/D converters are fabricated on integrated circuits and can be found any place where analog signals are to be stored, processed, or transmitted digitally, e.g., cell phone, computer, and automobile.
- The quantization operation causes errors in representing digitized information. These errors are called quantization noise, because the effect of quantization errors sounds like noise in a digitized music signal and looks like noise in a digital image.

- The more accurately we store the signal samples or measurements, the smaller the errors or quantization noise will be. We can reduce the errors by using more bits to represent each numerical value on a computer or digital system. The amount of noise decreases as the number of bits increases and the corresponding number of quantization levels increases.
- The amount of noise also depends on the dynamic range of the signal, since a wider dynamic range implies that the quantization intervals must be spaced more widely.
- To determine the impact of quantization noise in various applications, we must construct a way of measuring the amount of noise. The quantization error is simply the difference between the original signal samples and the rounded or quantized values.

- Since the size of an error in any given measurement or signal is always relative to the measurement itself, when constructing our definition of quantization noise, we must take into account the size of the signal we are measuring.
- Engineers have come up with a very important measure of the relative size of any type of noise, even quantization noise. This measure is called signal-to-noise ratio (SNR).
- The SNR is the ratio of the maximum signal level magnitude to the maximum noise level magnitude. For example:

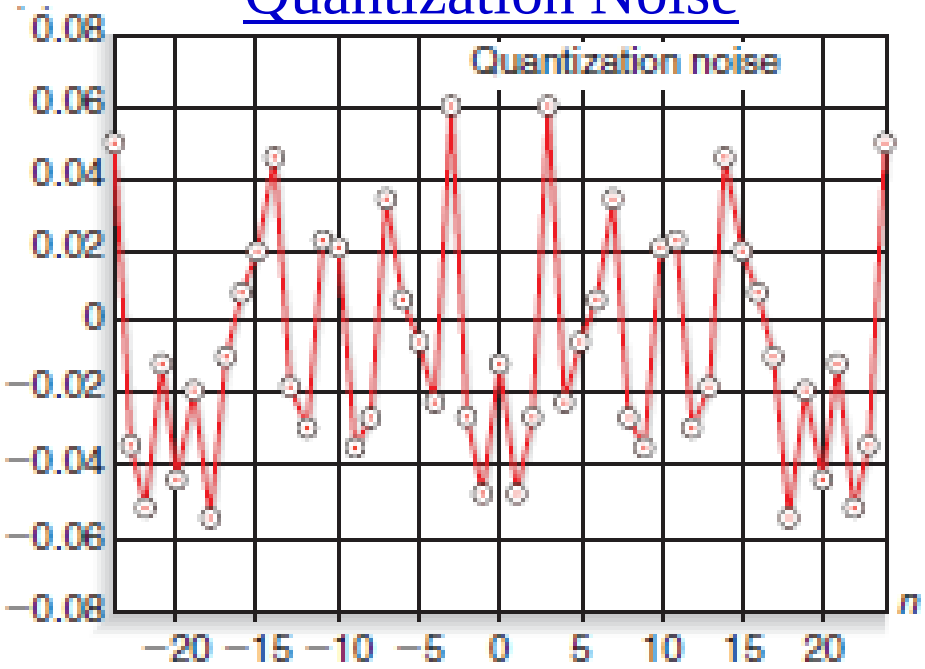
$$\text{SNR} = \frac{4}{1} = 4$$



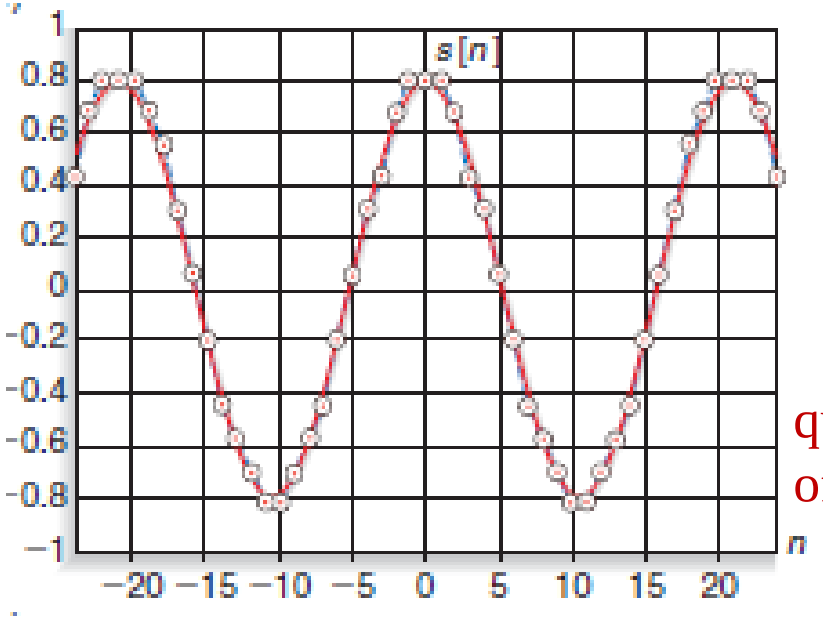


sampled sinusoid
T = 0.01 sec

Quantization Noise



Difference between original signal and quantized version



quantized (4 bits) version of sampled sinusoid

- What is the SNR? If we plot the quantization noise for a broader range of n, we would see that the maximum noise is 0.0625 or 1/16, which is half the distance between the levels in the quantized version because the quantization operation uses rounding.

$$\text{SNR} = \frac{0.8}{0.0625} = 12.8$$

- There is a simple formula for the SNR that will work for all signals.
  - Suppose that the quantizer rounds the signal sample values to numbers stored using only B bits. In general, a signal can be negative as well as positive, and so let's use sign-magnitude binary representation. By reserving the leftmost bit as the sign bit, we have B-1 bits remaining to store the magnitude of the signal. These B-1 bits can represent 2^{B-1} possible signal values.

- Suppose we set the highest quantized signal level, 2^{B-1}, to represent the maximum signal amplitude, so that the quantization levels cover the full dynamic range of the signal.
- For rounding, the quantization error can be no larger than half a quantization interval.
- Therefore

$$\text{SNR} = \frac{2^{B-1}}{1/2} = 2^B$$

- The SNR associated with quantization increases quickly when we increase the number of bits used to store the samples.
- In typical digital storage applications, B ranges from about 6 bits to 24 bits, giving us SNR values from 64 to 16,777,216.

The Decibel Scale

- When engineers deal with numerical quantities that take on a wide range of values, they sometimes express the quantity on a logarithmic scale called decibels (dB), to make numbers easier to handle.

x	$\log_{10}(x)$	dB
0.001	-3	-60
0.01	-2	-40
0.1	-1	-20
1	0	0
10	1	20
100	2	40
1000	3	60

$$x \text{ dB} = 20 \log_{10} x$$

Very large and very small numbers turn out to be reasonable numbers when expressed as decibels.

$$\begin{aligned} \text{SNR} &= 20 \log_{10}(2^B) \\ &= 20 B \log_{10}(2) \\ &= 6.02 B \text{ (dB)} \end{aligned}$$

The SNR grows by about 6 dB for every bit we use to represent the samples.

Concept Questions

- Define the operation of quantization.
- What is meant by clipping?
- What is a digital signal?
- What is analog-to-digital conversion?
- What is quantization noise?
- What is the definition of SNR?

Advanced Concepts

- Non-Uniform Quantization
  - The telephone system uses non-uniform quantization to achieve an improved SNR for the number of bits used.
  - One of the most popular types of non-uniform quantizers is the logarithmic quantizer. It uses a logarithmic function to map the input signal samples to new values such that small-valued input signal samples have smaller steps between them than larger values.
  - This effectively allows a greater SNR for low-amplitude signals (i.e., signals for which the amplitude does not span the full dynamic range of the quantizer) since more bits are allocated to the small changes in the signal.

- Signed Numbers (Twos Complement Numbers)
  - In order to avoid the fact that there are two zeros, one +0 and one -0, with signed magnitude representations, a method called twos complement is used to represent both positive and negative binary numbers.
  - The MSB is used to denote the sign of a number.
  - All positive numbers have 0 in the MSB. All negative numbers have 1 in the MSB. This bit is called the sign bit.
  - Twos complement positive numbers range from 0 to $(2^{n-1} – 1)$. Twos complement negative numbers range from -1 to $-(2^{n-1})$. The range of values is not symmetrical since there is one more negative number than there is positive number.

- To find the negative or twos complement of a number, follow these steps:
  - Write the binary form of the positive number.
  - Complement all the bits of the number (i.e., change 1's to 0's and 0's to 1's).
  - Add 1 to the result.
- Example: Convert -72 to an 8-bit two's complement number.

$$72_{10} = 01001000_2$$

$$-72_{10} = 10110111 + 00000001$$

$$= 10111000_2$$

Check:

$$01001000_2 + 10111000_2 = 00000000_2$$

Note: When adding, any carries out of the MSB are meaningless and should be ignored. The sum is correct only if it is within the allowable range.

- Fixed Point vs. Floating Point Representations
  - There are two basic formats used in DSP: floating point and fixed point.
    - Fixed-point numbers have a fixed decimal point as part of the number, e.g., 12.34.
    - A floating-point number has a mantissa M and an exponent E where the number is $M \times 10^E$. M and E are stored using bits.
  - In DSP, fixed-point signed numbers are used most often.
  - Floating-point format is basically like scientific notation. The advantage is that a floating-point number can have a much greater dynamic range (ability to represent very small numbers to very large numbers) than a fixed-point number with an equivalent number of bits.