Music 171: Fundamentals of Digital Audio

Tamara Smyth, trsmyth@ucsd.edu
Department of Music,
University of California, San Diego (UCSD)

October 15, 2019

Analog Signals

- A **signal**, of which a sinusoid is only one example, is a set, or sequence of numbers.
- The term "analog" refers to the fact that it is "analogous" of the signal it represents.
- A "real-world" signal is captured using a microphone which has a diaphragm that is pushed back and forth according to the compression and rarefaction of the sounding pressure waveform.

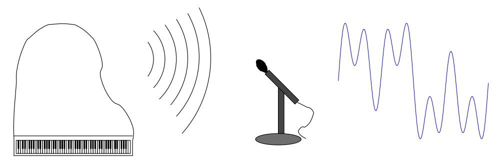


Figure 1: The electrical signal used to represent the pressure variations of a sound wave is an analog signal.

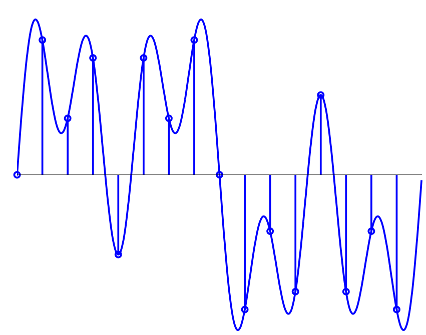
- The microphone transforms this displacement into a time-varying voltage—an analog electrical signal.

Continuous-Time Signals

- Analog signals are *continuous* in time.
- A **continuous-time** signal is an *infinite* and *uncountable* sequence of numbers, as are the possible values each number can have.
  - between a start and end time, there are infinite possible values for time t and the waveform's instantaneous amplitude $x(t)$.
- A continuous signal cannot be stored, or processed, in a computer since it would require infinite data.
- Analog signals must be *discretized* (*digitized*) to produce a finite set of numbers for computer use.

Discrete-Time, Digital Signals

- When analog signals are brought into a computer, they must be made *discrete* (finite and countable).
- A **discrete-time signal** is a *finite* sequence of numbers, with finite possible values for each number.
  - number values are limited by how many bits are used to represent them (*bit depth*);
  - can be stored on a digital storage medium.



- **Sampling**: the process of taking individual values of a continuous-time signal (at regular time intervals).

Analog to Digital Conversion

- The process by which an analog signal is digitized is called *analog-to-digital* or "a-to-d" conversion
- **analog-to-digital converter (ADC)**: the device that discretizes (digitizes) an analog signal.
- The ADC must accomplish two (2) tasks:
  1. **sampling**:
     - taking values (samples) at regular time intervals;
  2. **quantization**:
     - assign a number to the value (using limited computer bits).

Sampling

- **Sampling**: process of taking values (samples) of the analog waveform at regularly spaced time intervals.

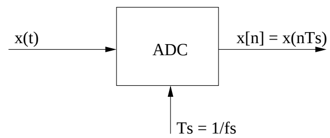


Figure 2: The ideal analog-to-digital converter.

- **Sampling Rate**: number of samples taken per second (Hz);
  - $f_s = 48$ kHz (professional studio)
  - $f_s = 44.1$ kHz (CD)
  - $f_s = 32$ kHz (broadcasting)
- **Sampling Period**: time interval (in seconds) between samples:

$$T_s = 1/f_s \text{ seconds.}$$

Sampled Sinusoids

- Sampling corresponds to transforming the continuous time variable t into a set of discrete times that are integer n multiples of the sampling period T_s:

$$t \longrightarrow nT_s.$$

- Integer n corresponds to the *index* in the sequence.
- **Continuous sinusoid**:

$$x(t) = A\sin(\omega t + \phi).$$

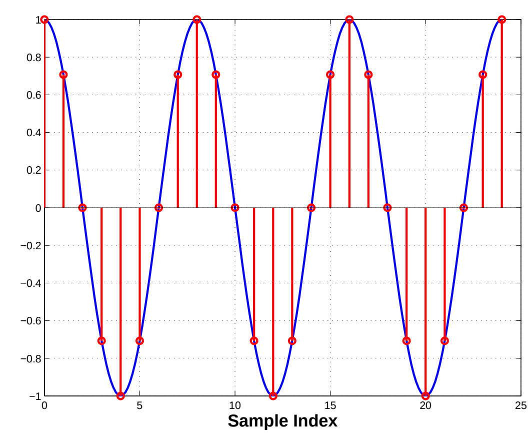
- **Discrete sinusoid**

$$x(n) = A\sin(\omega n T_s + \phi),$$

  a sequence of numbers that may be indexed by n.

Question 1

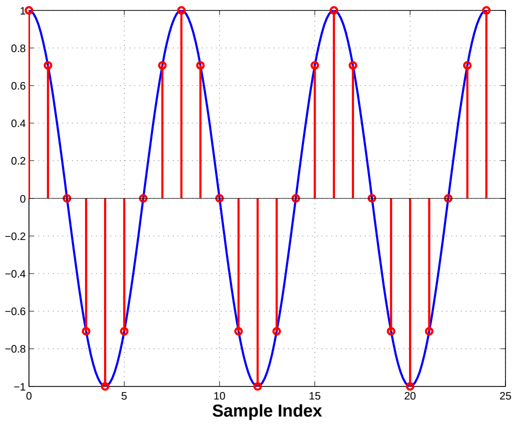
- If the following sinusoid was sampled at $f_s = 16$ Hz,
  - what is the **duration** of the signal shown?



- What would the duration be if $f_s = 32$?

Answer 1

- If $f_s = 16$, what is the **duration** shown?



- The **sampling period** (time between samples) is

$$T_s = 1/16 \text{ s.}$$

Since 24 samples are shown, **the duration is**

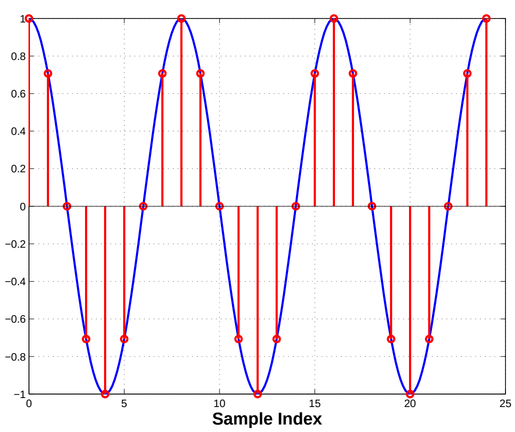
$$24 \times T_s = \frac{24}{16} = \frac{3}{2} = 1.5 \text{ s.}$$

- If $f_s = 32$, **the duration is shorter**:

$$24 \times T_s = \frac{24}{32} = \frac{3}{4} = .75 \text{ s.}$$

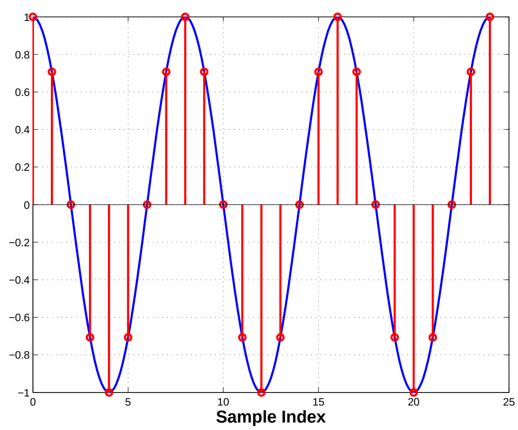
Question 2

- If the following sinusoid was sampled at $f_s = 16$ Hz,
  - what is the **frequency** of the sinusoid?



Answer 2 (Method 1)

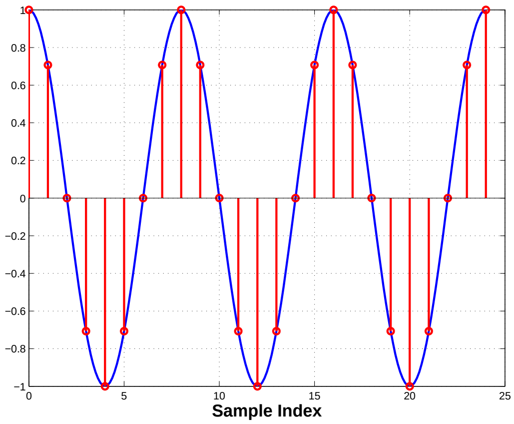
- If $f_s = 16$, what is the **frequency** of the sinusoid?



- **Method 1**:
  - 16 samples corresponds to 1 second,
  - there are 2 cycles in 1 second (after 16 samples),
  - the **frequency** is 2 Hz.

Answer 2 (Method 2)

- If $f_s = 16$, what is the **frequency** of the sinusoid?



- **Method 2**: the **period** of the sinusoid is

$$T = 8 \times T_s = \frac{8}{16} = \frac{1}{2},$$

and the **frequency** is

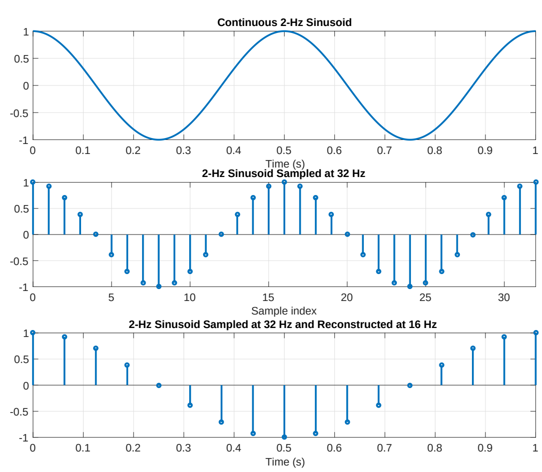
$$f = \frac{1}{T} = \frac{1}{2} = 2 \text{ Hz.}$$

- If $f_s = 32$, then the **frequency is higher**:

$$f = \frac{1}{T} = \frac{32}{8} = 4 \text{ Hz.}$$

Sampling and Reconstruction

- Once $x(t)$ is sampled to produce $x(n)$, *time scale information is lost*.
- $x(n)$ may represent a number of possible waveforms.



- Reconstructing at half the sampling rate $(F_s/2)$ will double the time between samples $(2/F_s)$, making the sinusoid **twice as long** and **halving the frequency**.

Importance of Knowing Sampling Rate

- If the signal is digitized and reconstructed *using the same sampling rate*, the frequency and duration will be preserved.
- If reconstruction is done *using a different sampling rate*, the **time interval between samples changes** resulting in a change in
  - overall signal **duration**,
  - time to complete one cycle (**period**) and thus sounding **frequency**.

Question 3

- A 220-Hz sinusoid is sampled at $f_{s1} = 44100$. It is then played on an audio system having a different sampling rate of $f_{s2} = 22050$ Hz.
  - At what **frequency** will the sinusoid sound?

Answer 3

- If a 220-Hz sinusoid sampled at $f_{s1} = 44100$ Hz is played back at $f_{s2} = 22050$ Hz ($f_{s1}/2$), then the
  - **period** between the samples will be twice as long:

$$T_{s2} = \frac{1}{f_{s2}} = \frac{1}{22050} = \frac{2}{44100} = \frac{2}{f_{s1}},$$

  - and the **period** of oscillation will double:

$$T_2 = 2T_1 = \frac{2}{f_1} = \frac{2}{220} \text{ s},$$

  - and the corresponding **frequency** will be halved (sound an octave lower):

$$f_2 = \frac{1}{T_2} = \frac{220}{2} = 110 \text{ Hz}.$$

Sampling in Practice II

- In the Beatles track "In My Life" (1:28) there is a Baroque-style piano solo composed and played by George Martin:
  - piano solo

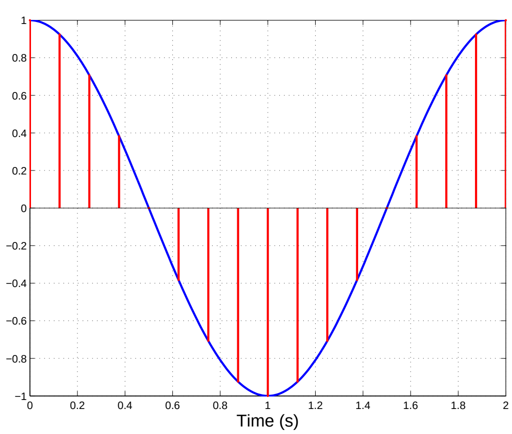
  Problem: George Martin's couldn't play fast enough!

  Solution:
  - play an octave lower and at half tempo
  - record at 1/2 sampling rate (or "speed" if analog),
  - play back at twice the rate at which it was recorded.

  Effect: The distortion creates a sort of harpsichord sound at the regular tempo and pitch. Why?
- Listen to how the original sound regains a piano quality when played at 1/2 the sampling rate:
  - piano solo at 1/2 sampling rate
- Note: Beatles recordings were actually analog, but the same principles apply!

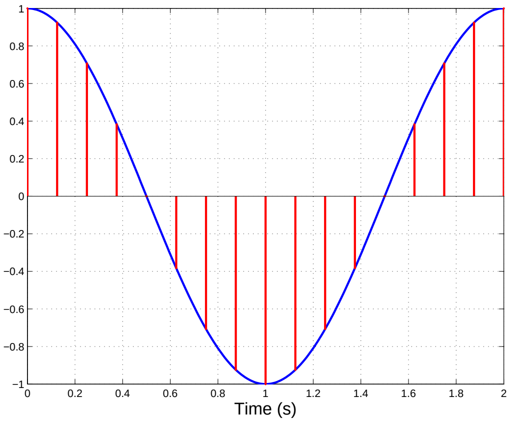
Question 4

- What is the **sampling rate** and **frequency** of the following sinusoid?



Answer 4

- Give the sinusoid's **sampling rate** and **frequency**.



- The period is 2 seconds (or, there is 1/2 cycle after 1 second) and the **frequency** is 1/2 Hz.
- The sinusoid has 8 samples in 1 second and thus **the sampling rate** is $f_s = 8$ Hz.
- The period has 16 samples and is thus $16 \times 1/8 = 2$ seconds long. The frequency is thus .5 Hz.

Implications of Sampling

- Is a sampled sequence only an approximation of the original?
- Is it possible to *perfectly* reconstruct a sampled signal?
- Will anything less than an infinite sampling rate introduce error?
- How frequently must we sample in order to "faithfully" reproduce an analog waveform?

Nyquist Sampling Theorem

- The **Nyquist Sampling Theorem** states that:

  A bandlimited continuous-time signal can be sampled and perfectly reconstructed from its samples if the waveform is sampled over twice as fast as it's highest frequency component.

- **Nyquist limit**: the highest frequency component that can be accurately represented:

$$f_{\text{max}} < f_s/2.$$

- **Nyquist frequency**: sampling rate required to accurately represent up to f_{max}:

$$f_s > 2f_{\text{max}}.$$

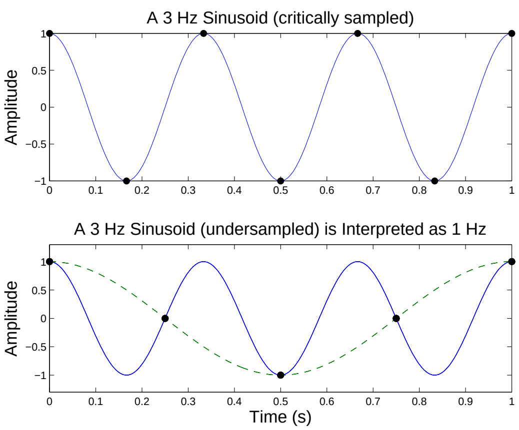
- **No information is lost** if sampling above $2f_{\text{max}}$.
- **No information is gained** by sampling much faster than $2f_{\text{max}}$.
- Is $f_s = 44,100$ Hz (CD-quality) enough?

Digital Audio System

- **Low-pass filter (left)**:
  - prevents components with frequency $> f_s/2$ from entering the ADC.
- **COMPUTER**:
  - sound storage/processing;
  - processing may introduce components with frequencies $> f_s/2$.
- **Low-pass filter (right)**:
  - defines audible bandwidth up to $f_s/2$;
  - **does not** prevent introduction of components with frequency $> f_s/2$ ("damage" already done)!
- What happens when frequencies exceed $f_s/2$?

Undersampling

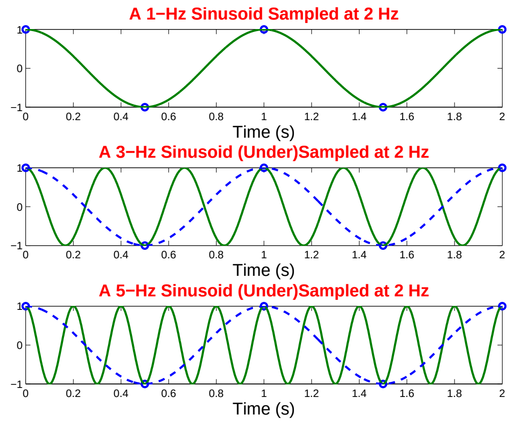
- If a signal is undersampled, it will be interpreted as the **alias** lying in the permitted range ($f < f_s/2$);



- Undersampling a 3-Hz sinusoid at $f_s = 4$ causes its frequency to be interpreted as 1Hz.

What is an Alias?

- Discrete sinusoids have infinite aliases but it is the one lowest in frequency ($< f_s/2$) that will sound.

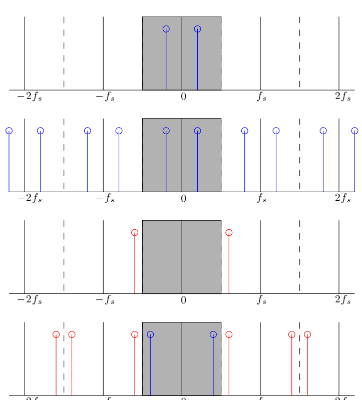


- Infinite sinusoids (with frequencies $f_0 \pm l f_s$) will produce the same sample values as the sinusoid at f_0.
- **Challenge**: show by proving the following equality:

$$A\cos(2\pi f_0 nT_s + \phi) = A\cos(2\pi(f_0 \pm l f_s)nT_s + \phi),$$

Aliasing / Folding over

- **Top 2 plots (blue)**: sinusoid has (infinite) aliases, but its frequency does not exceed $f_s/2$—GREAT!



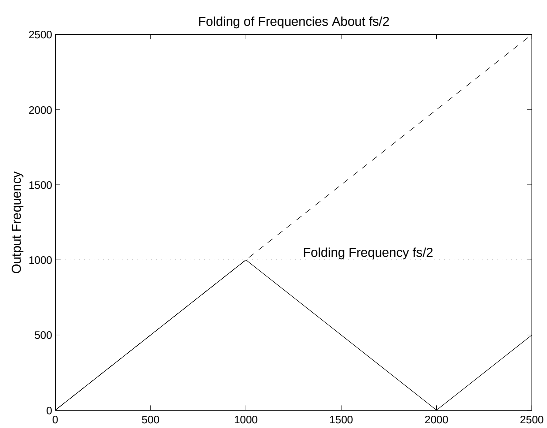
- **Bottom 2 plots (red)**: sinusoid with frequency $> f_s/2$ will have a negative frequency component with an alias in the sounding bandwidth (shaded).

Folding Frequency (Nyquist Limit)

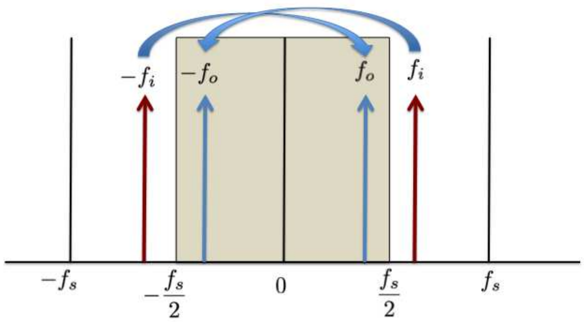
- Let f_{in} be the input signal and f_{out} be the signal at the output (after the lowpass filter).
- If f_{in} is less than the Nyquist limit ($f_s/2$)

$$f_{\text{out}} = f_{\text{in}},$$

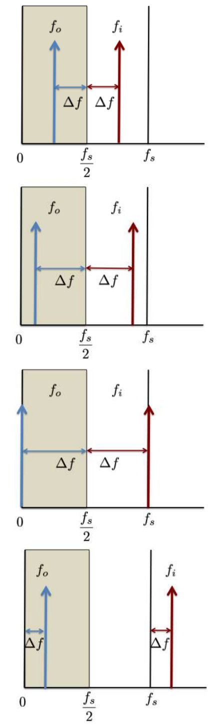
otherwise there is a folding over $f_s/2$.



- The *folding* occurs because of aliases of the negative frequency components.



Foldover

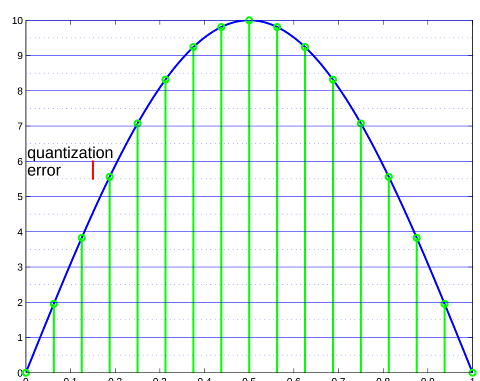


Quantization

- Where sampling is the process of taking a sample at regular time intervals...
- **Quantization** is the process of assigning a value (from a finite number of possibilities) to the amplitude of the signal at a time sample.
- Computers use bits to store such data.
- The higher the number of bits used to represent a value, the greater the number of possible values and the more precise the sampled amplitude will be.
- With n bits, 2^n possible values that can be represented.
- For CD quality audio, the number of bits is $n = 16$:
  - each sample can have $2^{16} = 65,536$ possible values;
  - the highest possible amplitude is $2^{15} = 32,768$, (since audio signals are positive and negative).
- Does quantization introduce error?

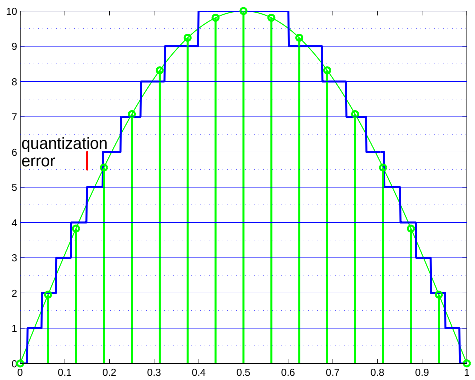
Quantization (linear)

- The signal below has 11 possible values to which the instantaneous amplitude may be quantized.



Quantization Error

- In a linear converter, when the actual sample value falls between these values (solid blue lines), it is quantized (or rounded) to the nearest value.



- This introduces a quantization error that will be uniformly distributed between 0 and 1/2 (error will never be greater than a factor of 1/2 the increment).

Bit Depth

- Computers use bits to store sample values—the number of bits used is called the **bit depth**: the greater the bit depth,
  - the greater the number of possible values,
  - the more precise the sampled amplitude will be.
- With n bits, 2^n possible values that can be represented.
  - for CD quality audio, $n = 16$,
  - each sample can have $2^{16} = 65,536$ possible values,
  - the highest possible amplitude is $2^{15} = 32,768$, (since audio signals are positive and negative).

Signal to Quantization Error (SQNR)

- When noise is a result of quantization error, we determine its audibility using the signal-to-quantization-noise-ratio (SQNR), determined by the ratio of
  - the maximum amplitude (2^{n-1}) to
  - the maximum quantization noise (1/2 for a linear converter).
- To determine audibility, the SQNR is provided in decibels (dB):

$$\begin{aligned} 20\log_{10}\left(\frac{2^{n-1}}{1/2}\right) \text{ dB} &= 20\log_{10}(2^n) \text{ dB} \\ &= n \times 20\log_{10}(2) \text{ dB} \\ &\approx n \times 6 \text{ dB} \\ &\approx 96 \text{ dB (for 16 bits).} \end{aligned}$$