6.003: Signal Processing

Sounds as Signals

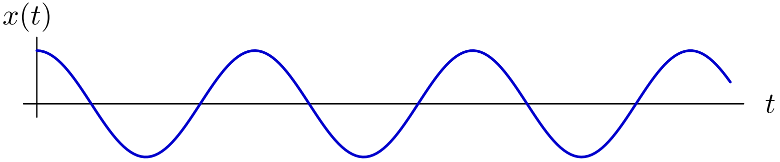
good morning!
we will start at 8:05am Eastern

adam hartz
hz@mit.edu

18 February 2021

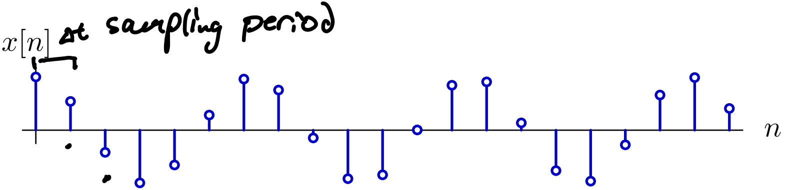
Tones and Sinusoids

A "tone" is a pressure that changes sinusoidally with time.



In 6.003, we will think of this as a "continuous-time" (CT) signal.

In contrast, a "discrete-time" (DT) signal is a sequence of numbers.



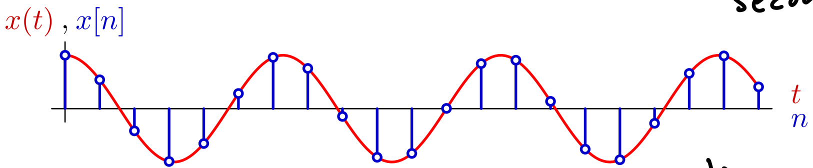
Mathematically:

$$x(t) = A\cos(\omega t)$$

$$x[n] = A\cos(\Omega n)$$

CT and DT Representations

Assume that $x[n]$ represents "samples" of $x(t)$: f_s $\frac{\text{samples}}{\text{second}}$



radians/second 2π radians/cycle

$$x(t) = A\cos(\omega t)$$

$$x[n] = A\cos(\Omega n)$$

- What are the units of ω, t, Ω, and n?

ω: radians/second; t: seconds; Ω: radians/sample; n: samples; f: cycles/seconds

Let f represent the "frequency" of the tone in cycles/second.

- Determine ω in terms of f.
- Determine Ω in terms of ω. [$\to f_s$]
- Determine Ω in terms of f.

$\omega \frac{\text{radians}}{\text{second}} = f \frac{\text{cycles}}{\text{second}} \cdot 2\pi \frac{\text{rad}}{\text{cycle}}$

$\Omega \frac{\text{rad}}{\text{sample}} = \omega \frac{\text{rad}}{\text{sec}} \cdot \frac{1}{f_s} \frac{\text{sec}}{\text{samp}}$

$\Omega = \omega / f_s = \frac{2\pi f}{f_s}$

Generating Sounds

Write a program to generate a tone.

We have provided some Python utilities to manipulate digital audio (in the `lib6003.audio` module)

The function `wav_write` creates a .wav file from 3 input arguments:

- `samples`: list of discrete samples
- `sample_frequency`: in samples/second
- `filename`: of resulting .wav file

Plotting

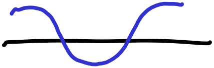
Make a plot of the numbers in list `x`.

Use `matplotlib`.

```
import matplotlib.pyplot as plt
```

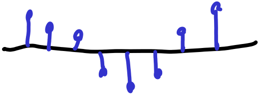
Line Plot

```
plt.plot(x)
plt.show()
```



Stem Plot

```
plt.stem(x)
plt.show()
```



Examples

See `rec01b.py`, parts 1 and 2.

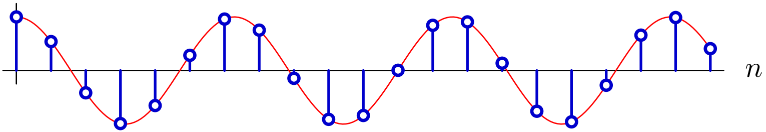
An Interesting Phenomenon

See `rec01b.py`, part 3.

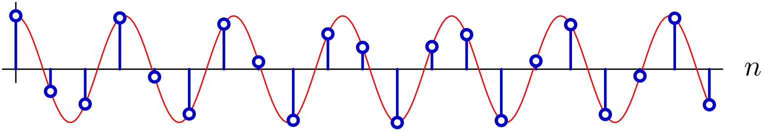
Aliasing

As the frequency Ω increases, the shapes of the sampled signals deviate from those of the underlying CT signals.

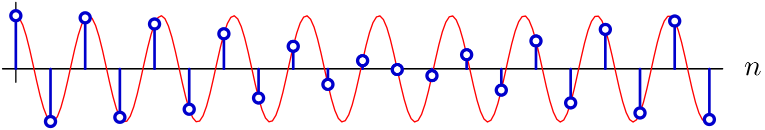
$\Omega = 1 : x[n] = \cos(n)$



$\Omega = 2 : x[n] = \cos(2n)$



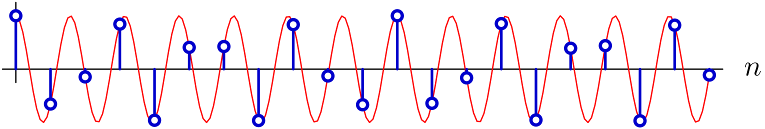
$\Omega = 3 : x[n] = \cos(3n)$



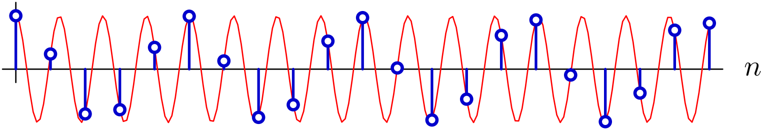
Aliasing

Worse and worse representation.

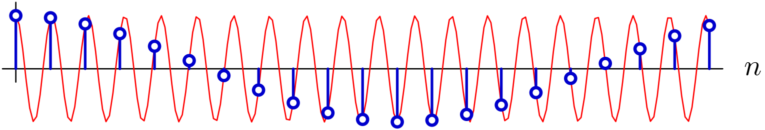
$$\Omega = 4 : x[n] = \cos(4n) = \cos((2\pi - 4)n) \approx \cos(2.283n)$$



$$\Omega = 5 : x[n] = \cos(5n) = \cos((2\pi - 5)n) \approx \cos(1.283n)$$



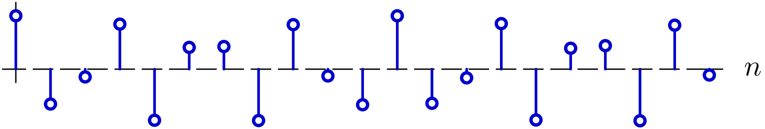
$$\Omega = 6 : x[n] = \cos(6n) = \cos((2\pi - 6)n) \approx \cos(0.283n)$$



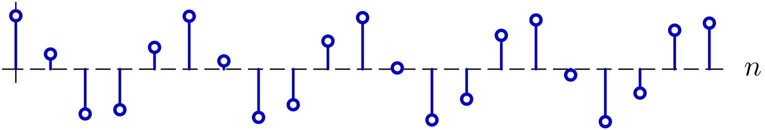
Aliasing

Worse and worse representation.

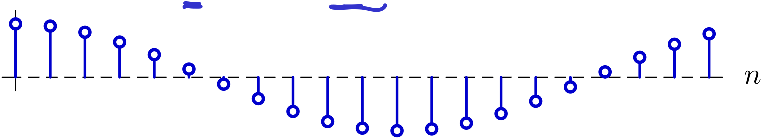
$$\Omega = 4 : x[n] = \cos(4n) = \cos((2\pi - 4)n) \approx \cos(2.283n)$$



$$\Omega = 5 : x[n] = \cos(5n) = \cos((2\pi - 5)n) \approx \cos(1.283n)$$



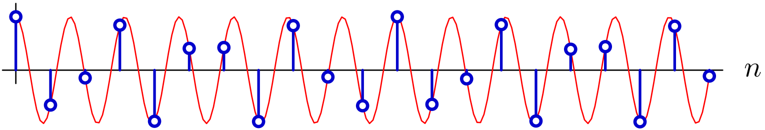
$$\Omega = 6 : x[n] = \cos(6n) = \cos((2\pi - 6)n) \approx \cos(0.283n)$$



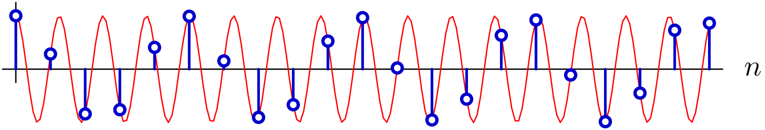
Aliasing

Worse and worse representation.

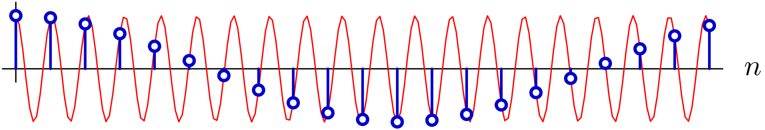
$$\Omega = 4 : x[n] = \cos(4n) = \cos((2\pi - 4)n) \approx \cos(2.283n)$$



$$\Omega = 5 : x[n] = \cos(5n) = \cos((2\pi - 5)n) \approx \cos(1.283n)$$



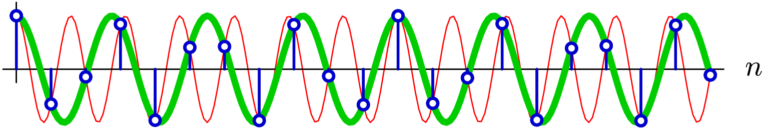
$$\Omega = 6 : x[n] = \cos(6n) = \cos((2\pi - 6)n) \approx \cos(0.283n)$$



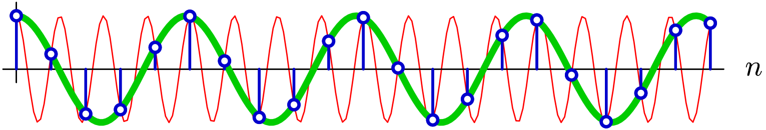
Aliasing

For $\Omega > \pi$, a lower frequency Ω_L has the same sample values as Ω.

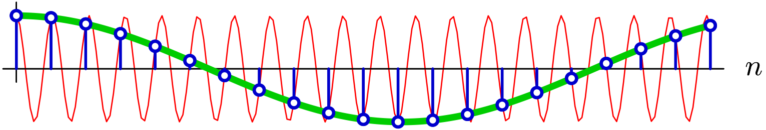
$$\Omega = 4 : x[n] = \cos(4n) = \cos((2\pi - 4)n) \approx \cos(2.283n)$$



$$\Omega = 5 : x[n] = \cos(5n) = \cos((2\pi - 5)n) \approx \cos(1.283n)$$



$$\Omega = 6 : x[n] = \cos(6n) = \cos((2\pi - 6)n) \approx \cos(0.283n)$$



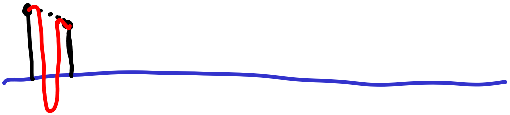
The same DT sequence represents many different values of Ω.

Aliasing

The same DT sequence can result from uniform sampling of CT sinusoids with different frequencies.

For example, let's consider and compare the following signals using Python:

$$x_1[n] = \cos(0.2\pi n)$$
$$x_2[n] = \cos(2.2\pi n)$$
$$x_3[n] = \cos(1.8\pi n)$$



Aliasing

The same DT sequence can result from uniform sampling of CT sinusoids with different frequencies.

For example, let's consider and compare the following signals using Python:

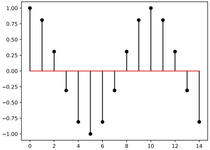
$$x_1[n] = \cos(0.2\pi n)$$
$$x_2[n] = \cos(2.2\pi n)$$
$$x_3[n] = \cos(1.8\pi n)$$

These all represent the exact same signal! They are all *aliases* for that signal.

Aliasing

The same DT sequence can result from uniform sampling of CT sinusoids with different frequencies.

$f = \frac{\Omega f_s}{2\pi}$



$f_{max} = \frac{\pi f_s}{2\pi} = \frac{f_s}{2}$

Nyquist rate

Although there are multiple frequencies Ω that we could use to refer to this signal, it is difficult to "see" anything but $\Omega = 0.2\pi$ by looking at this graph.

We can remove the ambiguity of which frequency is represented by a set of samples by choosing the one in the range $0 \leq \Omega \leq \pi$.

We call that range of frequencies the **base band** of frequencies, and and the value of Ω that falls in that range is often referred to as the *principal alias*.

Aliasing

The same DT sequence can result from uniform sampling of CT sinusoids with different frequencies.

For example, starting from a signal $\cos(\Omega n)$, where $0 \le \Omega \le \pi$, what can we say about each of the following?

$\cos((2\pi + \Omega)n)$ $\qquad \cos((2\pi - \Omega)n)$ $\qquad \sin((2\pi + \Omega)n)$ $\qquad \sin((2\pi - \Omega)n)$

$= \cos(2\pi n + \Omega n)$

$= \cos(\Omega n)$

$\cos(2\pi n - \Omega n)$

$= \cos(-\Omega n)$

$= \cos(\Omega n)$

Aliasing

The same DT sequence can result from uniform sampling of CT sinusoids with different frequencies.