6.003: Signal Processing

Sampling and Aliasing

Announcements:

- HW3 is posted
  - LAB3 check-in due Friday, September 24, at 3pm.
  - the rest of HW3 due Tuesday, September 28, at noon.
- Quiz 1: October 5, 2-4pm, 50-340 (Walker)
  - Coverage up to and including all of week 3, including HW3.
  - Closed book except for one page of notes (8.5"x11" both sides).
  - No electronic devices. (No headphones, cellphones, calculators, ...)
- No HW4 – a practice quiz will be posted on September 28.

September 21, 2021

Last Time

Fourier Series – Complex Exponential Form

- complex numbers
- complex exponentials and their relation to sinusoids
- complex exponential form of Fourier series
- **delay property** of Fourier series

Is the Complex Exponential Form Actually Easier?

We previously determined the effect of a half-period shift on the Fourier coefficients of the trig form. The result was a bit complicated.

Assume that $f(t)$ is periodic in time with period T:

$$f(t) = f(t+T)\,.$$

Let $g(t)$ represent a version of $f(t)$ shifted by half a period:

$$g(t) = f(t-T/2)\,.$$

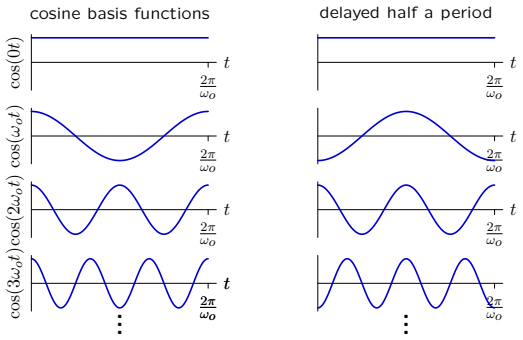
How many of the following statements correctly describe the effect of this shift on the Fourier series coefficients.

- cosine coefficients c_k are negated ✗
- sine coefficients d_k are negated ✗
- odd-numbered coefficients $c_1, d_1, c_3, d_3, \ldots$ are negated ✓
- sine and cosine coefficients are swapped: $c_k \to d_k$ and $d_k \to c_k$ ✗

Half-Period Shift

Shifting $f(t)$ shifts the underlying basis functions of it Fourier expansion.

$$f(t-T/2) = \sum_{k=0}^{\infty} c_k \cos\left(k\omega_o(t-T/2)\right) + \sum_{k=1}^{\infty} d_k \sin\left(k\omega_o(t-T/2)\right)$$

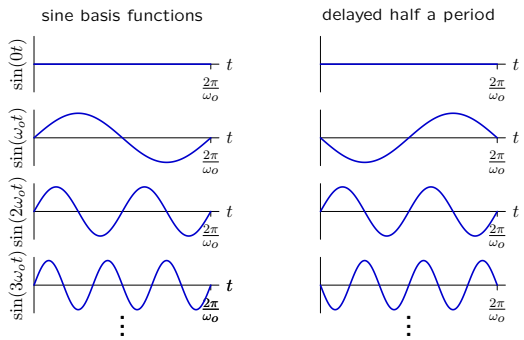


Half-period shift inverts c_k terms if k is odd. It has no effect if k is even.

Half-Period Shift

Shifting $f(t)$ shifts the underlying basis functions of it Fourier expansion.

$$f(t-T/2) = \sum_{k=0}^{\infty} c_k \cos\left(k\omega_o(t-T/2)\right) + \sum_{k=1}^{\infty} d_k \sin\left(k\omega_o(t-T/2)\right)$$

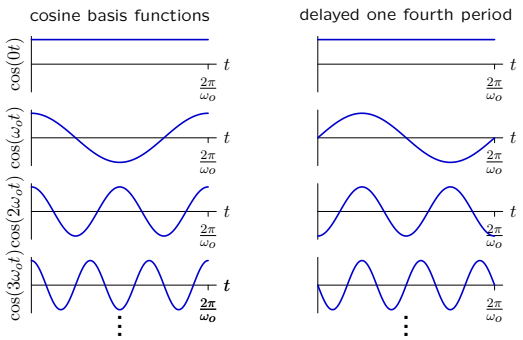


Half-period shift inverts d_k terms if and only if k is odd.

Quarter-Period Shift

Shifting by $T/4$ is **even more complicated**.

$$f(t-T/4) = \sum_{k=0}^{\infty} c_k \cos\left(k\omega_o(t-T/4)\right) + \sum_{k=1}^{\infty} d_k \sin\left(k\omega_o(t-T/4)\right)$$

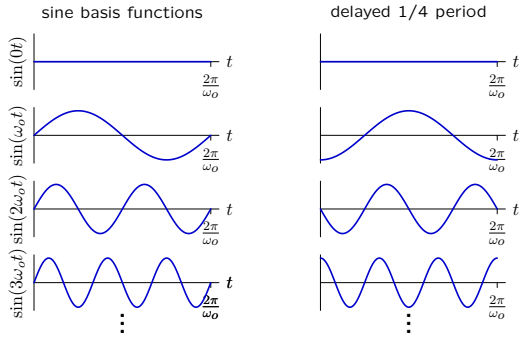


$\cos(\omega_o t) \to \sin(\omega_o t)$; $\cos(2\omega_o t) \to -\cos(2\omega_o t)$; $\cos(3\omega_o t) \to -\sin(3\omega_o t)$

Check Yourself: Alternative (more intuitive) Approach

Shifting $f(t)$ shifts the underlying basis functions of it Fourier expansion.

$$f(t-T/4) = \sum_{k=0}^{\infty} c_k \cos\left(k\omega_o(t-T/4)\right) + \sum_{k=1}^{\infty} d_k \sin\left(k\omega_o(t-T/4)\right)$$



$\sin(\omega_o t) \rightarrow -\cos(\omega_o t)$; $\sin(2\omega_o t) \rightarrow -\sin(2\omega_o t)$; $\sin(3\omega_o t) \rightarrow \cos(3\omega_o t)$

Summary of Shift Results

Let c_k and d_k represent the Fourier series coefficients for $f(t)$

$$f(t) = f(t+T) = c_0 + \sum_{k=1}^{\infty} c_k \cos(k\omega_o t) + \sum_{k=1}^{\infty} d_k \sin(k\omega_o t)$$

and c'_k and d'_k represent those for a half-period delay.

$$g(t) = f(t-T/2) = c_0 + \sum_{k=1}^{\infty} c'_k \cos(k\omega_o t) + \sum_{k=1}^{\infty} d'_k \sin(k\omega_o t)$$

Then $c'_k = (-1)^k c_k$ and $d'_k = (-1)^k d_k$.

Let c''_k and d''_k represent those for a quarter-period delay.

$$g(t) = f(t-T/2) = c_0 + \sum_{k=1}^{\infty} c'_k \cos(k\omega_o t) + \sum_{k=1}^{\infty} d'_k \sin(k\omega_o t)$$

Then

$$c''_k = \begin{cases} c_k & \text{if } k = 0,4,8,12,\ldots \\ d_k & \text{if } k = 1,5,9,13,\ldots \\ -c_k & \text{if } k = 2,6,10,14,\ldots \\ -d_k & \text{if } k = 3,7,11,15,\ldots \end{cases} \qquad d''_k = \begin{cases} d_k & \text{if } k = 0,4,8,12,\ldots \\ -c_k & \text{if } k = 1,5,9,13,\ldots \\ -d_k & \text{if } k = 2,6,10,14,\ldots \\ c_k & \text{if } k = 3,7,11,15,\ldots \end{cases}$$

Other Shifts Yield Even More Complicated Results

Let c_k and d_k represent the Fourier series coefficients for $f(t)$

$$f(t) = f(t+T) = c_0 + \sum_{k=1}^{\infty} c_k \cos(k\omega_o t) + \sum_{k=1}^{\infty} d_k \sin(k\omega_o t)$$

and c'''_k and d'''_k represent those for n eighth-period delay.

$$g(t) = f(t-T/8) = c_0 + \sum_{k=1}^{\infty} c'_k \cos(k\omega_o t) + \sum_{k=1}^{\infty} d'_k \sin(k\omega_o t)$$

$$c'''_k = \begin{cases} c_k & \text{if } k = 0,8,16,24,\ldots \\ \frac{\sqrt{2}}{2}(c_k + d_k) & \text{if } k = 1,9,17,25,\ldots \\ d_k & \text{if } k = 2,10,18,26,\ldots \\ \frac{\sqrt{2}}{2}(-c_k + d_k) & \text{if } k = 3,11,19,27,\ldots \\ -c_k & \text{if } k = 4,12,20,28,\ldots \\ \frac{\sqrt{2}}{2}(-c_k - d_k) & \text{if } k = 5,13,21,29,\ldots \\ -d_k & \text{if } k = 6,14,22,30,\ldots \\ \frac{\sqrt{2}}{2}(c_k - d_k) & \text{if } k = 7,15,23,31,\ldots \end{cases} \qquad d'''_k = \ldots$$

Effects of Time Shifts on Complex Exponential Series

Delaying time by τ multiplies the complex exponential coefficients of a Fourier series by a constant $e^{-jk\omega_o\tau}$.

Let a_k represent the complex exponential series coefficients of $f(t)$ and a'_k represent the complex exponential series coefficents of $g(t) = f(t-\tau)$.

$$\begin{aligned} a'_k &= \frac{1}{T}\int_T g(t)e^{-jk\omega_o t}dt \\ &= \frac{1}{T}\int_T f(t-\tau)e^{-jk\omega_o t}dt \\ &= \frac{1}{T}\int_T f(s)e^{-jk\omega_o(s+\tau)}ds \\ &= e^{-jk\omega_o\tau}\frac{1}{T}\int_T f(s)e^{-jk\omega_o s}ds \\ &= e^{-jk\omega_o\tau}a_k \end{aligned}$$

Each coefficient a'_k in the series for $g(t)$ is a constant $e^{-jk\omega_o t}$ times the corresponding coefficient a_k in the series for $f(t)$.

Summary

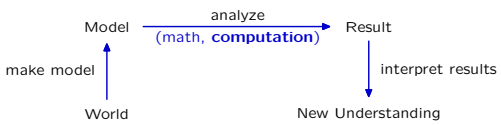
We introduced the complex exponential form of Fourier series.

- complex numbers
- complex exponentials and their relation to sinusoids
- analysis and synthesis with complex exponentials
- delay property: much simpler with complex exponentials

Today

Importance of Discrete Representations

Our goal is to develop **signal processing** tools to model interesting aspects of the world, to analyze the model, and to interpret the results.



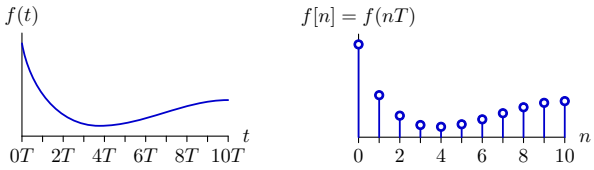
The **increasing power** and **decreasing cost** of computation makes the use of computation increasingly attractive.

However, many important signals are naturally described with continuous functions, that must be **sampled** in order to be analyzed computationally.

Today: understand relations between **continuous** and **sampled** signals.

Sampling

How does sampling affect the information contained in a signal?



T = sampling interval

Notation:
We will use parentheses to denote functions of continuous domain ($f(t)$) and square brackets to denote functions of discrete domain ($f[n]$).

Effects of Sampling Are Easily Heard

Sampling Music

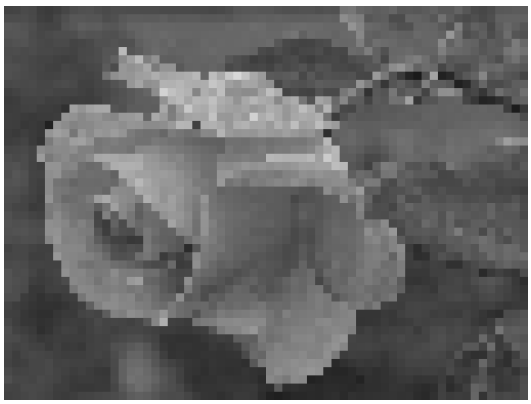
$$f_s = \frac{1}{T}$$

- $f_s = 44.1$ kHz
- $f_s = 22$ kHz
- $f_s = 11$ kHz
- $f_s = 5.5$ kHz
- $f_s = 2.8$ kHz

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto
Nathan Milstein, violin

Effects of Sampling are Easily Seen

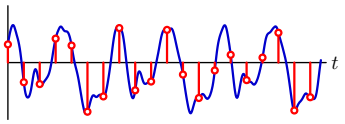
Sampling Images



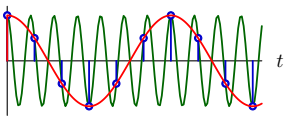
downsampled: 64×48

Characterizing Sampling

We would like to sample in a way that preserves **information**.
However, information is generally **lost** in the sampling process.
Example: samples provide no information about the intervening values.



Furthermore, information that is retained by sampling can be misleading.
Example: samples can suggest patterns not contained in the original.



Samples (blue) of the original high-frequency signal (green) could just as easily have come from a much lower frequency signal (red).

Characterizing Sampling

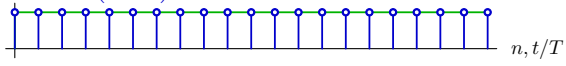
Our goal is to understand sampling so that we can mitigate its effects on the information contained in the signals we process.

Characterizing Sampling

Begin by sampling sinusoids with different frequencies ω.
Sample $x(t) = \cos(\omega t)$ every T seconds to obtain $x[n]$:

$$x[n] = x(nT) = \cos(\omega nT) = \cos\Big((\omega T)n\Big)$$

$\omega T = 0.0 \times 2\pi$
$x(t) = \cos(\omega t)$
$x[n] = \cos\Big((\omega T)n\Big)$



Characterizing Sampling

Begin by sampling sinusoids with different frequencies ω.

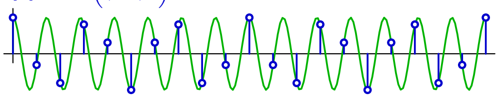
Sample $x(t) = \cos(\omega t)$ every T seconds to obtain $x[n]$:

$$x[n] = x(nT) = \cos(\omega nT) = \cos\Big((\omega T)n\Big)$$

$\omega T = 0.7 \times 2\pi$

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

$x[n] = \cos\Big((\omega T)n\Big)$



Characterizing Sampling

Begin by sampling sinusoids with different frequencies ω.

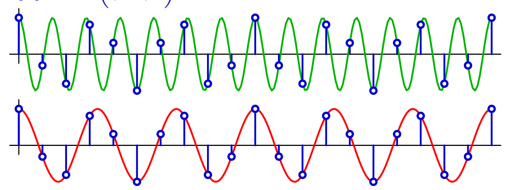
Sample $x(t) = \cos(\omega t)$ every T seconds to obtain $x[n]$:

$$x[n] = x(nT) = \cos(\omega nT) = \cos\Big((\omega T)n\Big)$$

$\omega T = 0.7 \times 2\pi$

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

$x[n] = \cos\Big((\omega T)n\Big)$



$\omega T \to 2\pi - \omega T = 0.3 \times 2\pi$

$x[n] = \cos\Big((2\pi - \omega T)n\Big) = \cos\Big((\omega T)n\Big)$

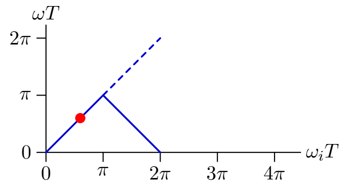
$x(t) = \cos\Big((\frac{2\pi}{T} - \omega)t\Big) \qquad \omega \to 2\pi/T - \omega$

Aliasing

Two different sinusoids

$$\cos\Big((\omega T)n\Big) \text{ and } \cos\Big((2\pi - \omega T)n\Big)$$

will generate the same sequence of samples.

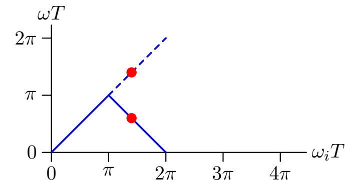


Aliasing

Two different sinusoids

$$\cos\Big((\omega T)n\Big) \text{ and } \cos\Big((2\pi - \omega T)n\Big)$$

will generate the same sequence of samples.

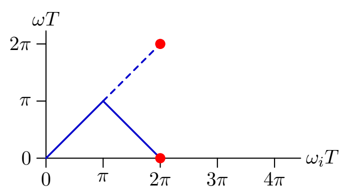


Aliasing

Two different sinusoids

$$\cos\Big((\omega T)n\Big) \text{ and } \cos\Big((2\pi - \omega T)n\Big)$$

will generate the same sequence of samples.



Mathematically:

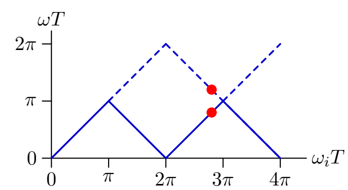
$$\cos\Big((2\pi - \omega T)n\Big) = \cos(2\pi n)\cos\Big((\omega T)n\Big) + \sin(2\pi n)\sin\Big((\omega T)n\Big)$$
$$= \cos\Big((\omega T)n\Big)$$

Aliasing

Two different sinusoids

$$\cos\Big((\omega T)n\Big) \text{ and } \cos\Big((2\pi - \omega T)n\Big)$$

will generate the same sequence of samples.

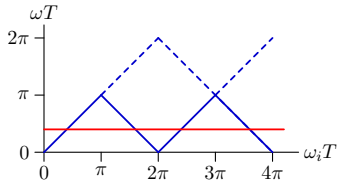


And the pattern continues (since the cosine function is periodic):

$$\cos\Big((2\pi + \omega T)n\Big) = \cos\Big((\omega T)n\Big)$$

Aliasing

Many input frequencies will generate the same sequence of samples.



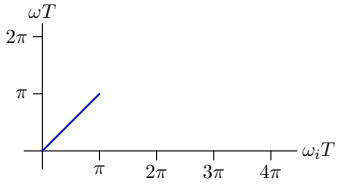
The same samples would result if $\omega_i T$ were 0.4π or 1.6π or 2.4π or ...

It's impossible to determine what frequency produced a particular output sequence of samples.

Since multiple frequencies ω_i generate the same discrete samples, we say that these frequencies are **aliases** of each other, and the process is called **aliasing**.

Anti-Aliasing

We can prevent aliasing by removing **input** frequencies with $\omega_i T > \pi$.



We call this low-frequency range of acceptable frequencies the **baseband**.

The maximum frequency that can be represented using this scheme is called the **Nyquist frequency**:

$$\omega_m = \pi/T$$

The Nyquist frequency is equal to half the sampling rate f_s

$$f_m = \frac{\omega_m}{2\pi} = \frac{\pi}{2\pi T} = \frac{1}{2T} = \frac{f_s}{2}$$

where f_s has dimensions of samples per second and f_m has dimensions of cycles per second.

Anti-Aliasing Demonstration

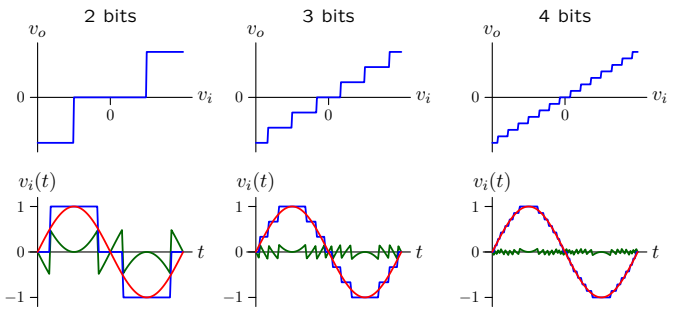
Sampling Music.

- $f_s = 11$ kHz without anti-aliasing
- $f_s = 11$ kHz with anti-aliasing
- $f_s = 5.5$ kHz without anti-aliasing
- $f_s = 5.5$ kHz with anti-aliasing
- $f_s = 2.8$ kHz without anti-aliasing
- $f_s = 2.8$ kHz with anti-aliasing

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto
Nathan Milstein, violin

Quantization

The information content of a signal depends not only with sample rate but also with the number of bits used to represent each sample.



Bit rate = (# bits/sample)×(# samples/sec)

Check Yourself

We hear sounds that range in amplitude from 1,000,000 to 1.

How many bits are needed to represent this range?

1. 5 bits
2. 10 bits
3. 20 bits
4. 30 bits
5. 40 bits

Quantization Demonstration

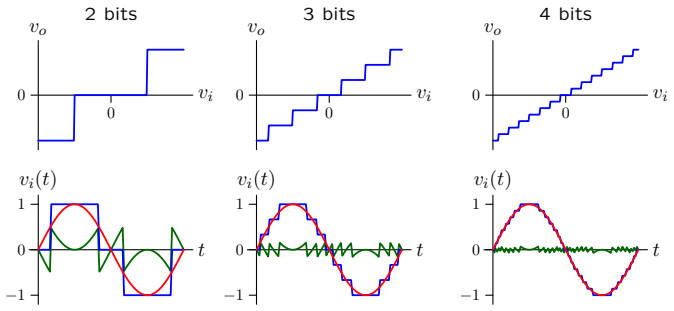
Quantizing Music

- 16 bits/sample
- 8 bits/sample
- 6 bits/sample
- 4 bits/sample
- 3 bits/sample
- 2 bit/sample

J.S. Bach, Sonata No. 1 in G minor Mvmt. IV. Presto
Nathan Milstein, violin

Quantization

We measure discrete amplitudes in bits.



Example: high-quality audio

$$2\,\text{channels} \times 16\,\frac{\text{bits}}{\text{sample}} \times 44,100\,\frac{\text{samples}}{\text{sec}} \times 60\,\frac{\text{sec}}{\text{min}} \times 74\,\text{min} \approx 6.3\,\text{G bits}$$

$$\approx 0.78\,\text{G bytes}$$

Quantizing Images

Converting an image from a continuous representation to a discrete representation involves the same sort of issues.

This image has 280×280 pixels, with brightness quantized to 8 bits.

Summary

Many important signals are naturally described with continuous functions and must be **sampled** in order to be analyzed computationally.

Information is generally lost in the sampling process.

Sampling can result in the same sequence of samples from very different signals – a process we call **aliasing**.

We can understand precisely how simple sinusoids alias, and that allows us to mitigate distortions caused by aliasing.

Quantization of a signal's amplitude can also introduce distortions.

We will study all of these distortion mechanisms as the semester progresses.