ELEC 221 Lecture 17
The sampling theorem

Thursday 7 November 2024

Announcements

- Assignment 4 to be released soon (focus on chapters 7/8)
- No class Tuesday (reading break)
- No prof office hours this Friday

Recap

Continuous time:

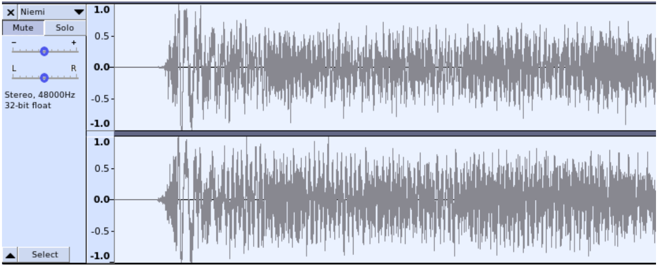
Discrete time:

Lecture 04 Demos

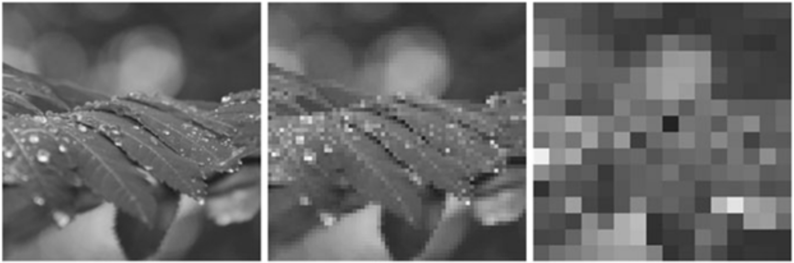
```
import numpy as np
import matplotlib.pyplot as plt
from IPython.display import Audio
```

Demo 1: fun with square waves

```
tone = 65            # A frequency in Hz
duration = 2          # The length of the audio signal (in seconds)
sample_rate = 48000  # The number of samples per second to take
t_range = np.linspace(0, duration, sample_rate * duration) # Range of time
```



Motivation



256 x 256 64 x 64 16 x 16

Image credit: https://what-when-how.com/introduction-to-video-and-image-processing/image-acquisition-introduction-to-video-and-image-processing-part-2/

Motivation

https://youtu.be/B8EMI3_0TO0?t=9

History of frame rate in film:
https://www.youtube.com/watch?v=mjYjFEp9Yx0

Image credit: https://www.mediacollege.com/video/frame-rate/img/frame-rates.jpg

Today

Core question: under what conditions can we recover a continuous time signal using only information from its samples?

Learning outcomes:

- state the sampling theorem
- define the Nyquist sampling rate and determine if a sampling rate is sufficient to reconstruct a signal from its samples
- describe the phenomenon of aliasing

Impulse train sampling

The unit impulse as a sampler

Multiplying the signal by a shifted impulse picks out the value of the signal at that point:

$$x[n] \cdot \delta[n-k] = x[k] \cdot \delta[n-k]$$

This allows us to write any signal as a **superposition of weighted impulses.**

$$x[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot \delta[n-k]$$

17 / 40

Impulse train sampling

In continuous time:

What if we have more than one?

where

Impulse train sampling

What does the following signal look like?

Impulse train sampling

The combined signal in the time domain is

What happens in the *frequency domain*?

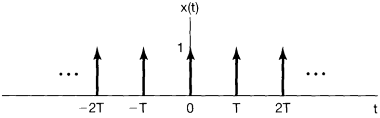
Impulse train sampling

We have a periodic impulse train. Recall what Fourier transforms of periodic signals looked like:

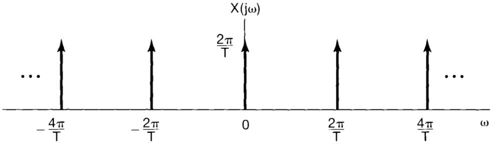
Impulse train sampling

We need to find the Fourier series coefficients of the periodic impulse train.

Impulse train sampling

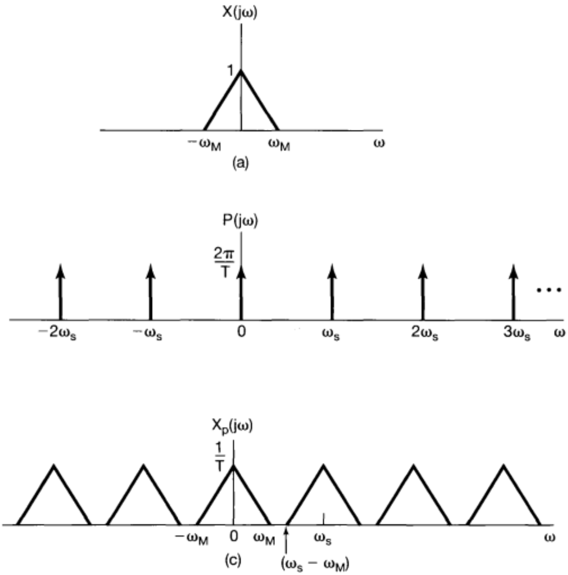


(a)



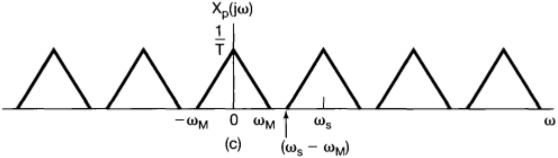
Impulse train sampling

Impulse train sampling



Impulse train sampling

Suppose we have sampled...



How do we recover our original signal from this spectrum?

Image credit: Oppenheim 7.1

The sampling theorem

"Let $x(t)$ be a **band-limited** signal with $X(j\omega) = 0$ for $|\omega| > \omega_M$. Then $x(t)$ is uniquely determined by its samples $x(nT)$, $n = 0, \pm 1, \pm 2, \ldots$, if

Given these samples, we can reconstruct $x(t)$ by generating a periodic impulse train in which successive impulses have amplitudes that are successive sample values. This impulse train is then processed through an ideal lowpass filter with gain T and cutoff frequency greater than ω_M and less than $\omega_s - \omega_M$. The resulting output signal will exactly equal $x(t)$."

Let's show this graphically:

The sampling theorem

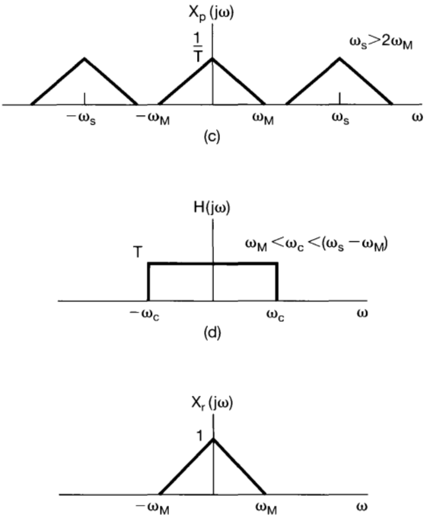


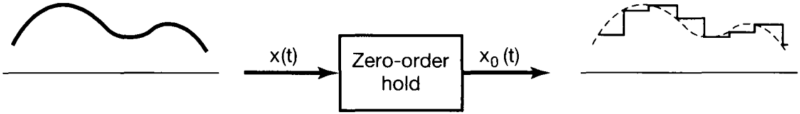
Image credit: Oppenheim 7.1

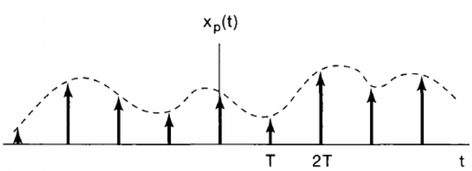
The sampling frequency is key:

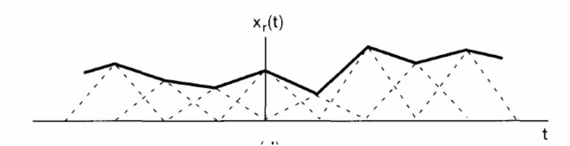
- $\omega_s = 2\omega_M$ is referred to as the **Nyquist rate**
- $\omega_M = \omega_s/2$ is referred to as the **Nyquist frequency**

Exercise: suppose we perform impulse-train sampling with period $T = 10^{-4}$. If a signal $x(t)$ has $X(j\omega) = 0$ for $|\omega| > 15000\pi$, can we reconstruct it exactly from the samples?

In reality we cannot generate a perfect, ideal impulse train. But, we can still interpolate (you will explore this in A4)







Aliasing

What happens when you don't sample at a high enough rate?

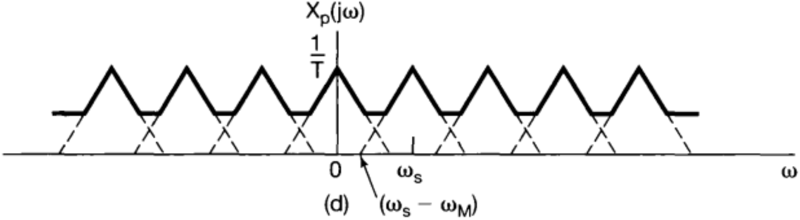


Image credit: Oppenheim 7.1

Aliasing

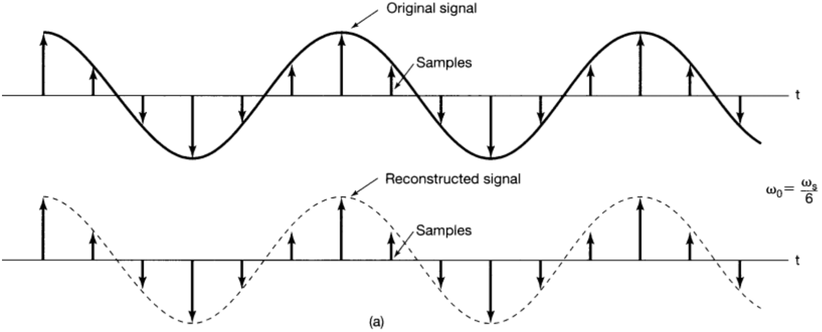


Image credit: Oppenheim 7.3

Aliasing

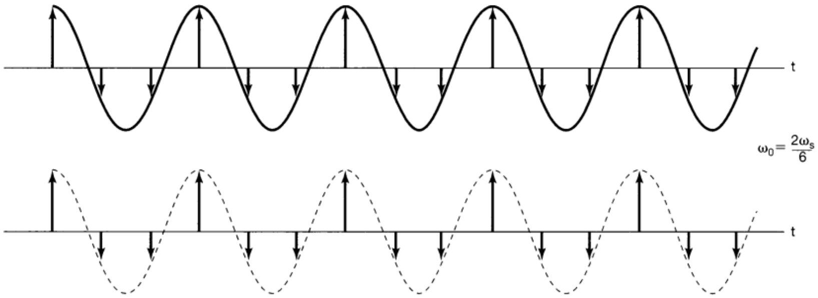
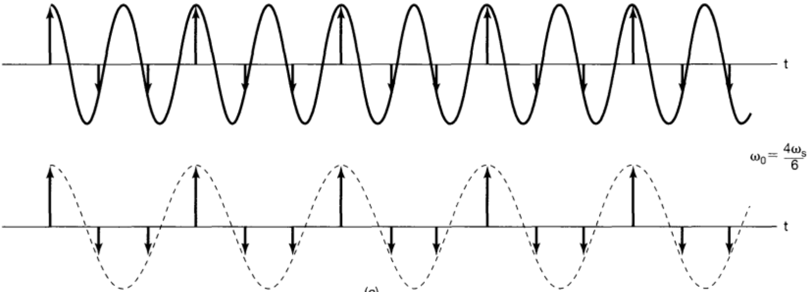


Image credit: Oppenheim 7.3

Aliasing



(c)

Image credit: Oppenheim 7.3

Aliasing

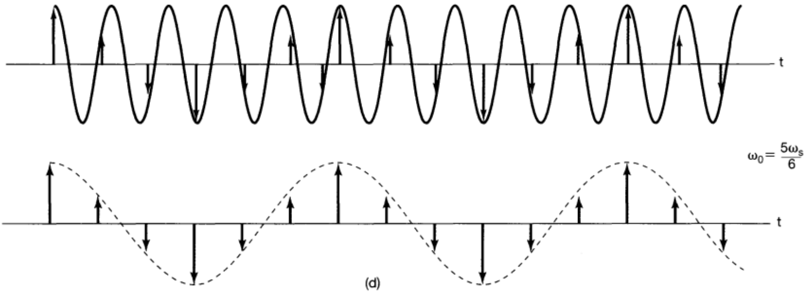


Image credit: Oppenheim 7.3

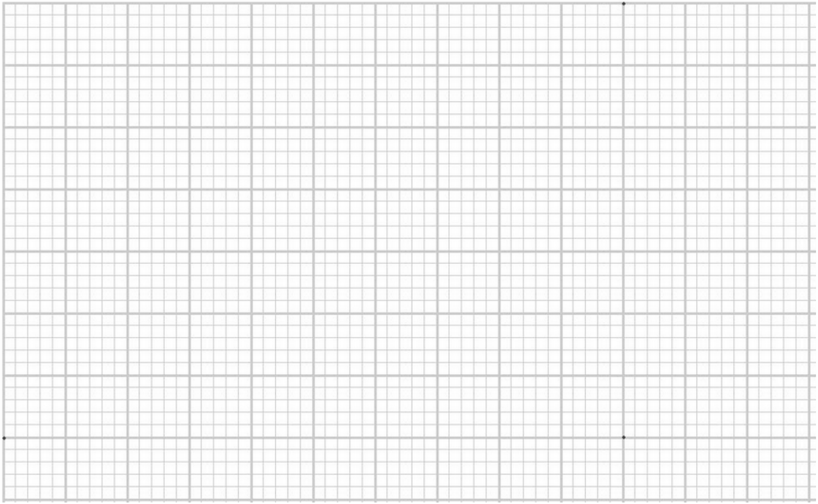
`https://visualize-it.github.io/stroboscopic_effect/simulation.html`

Two aspects to consider here:

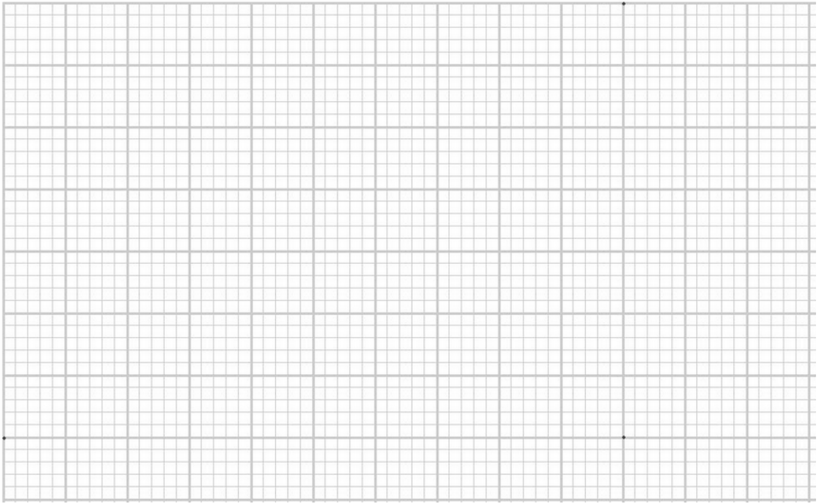
- Why does the interpreted frequency *decrease* as the true frequency increases?
- Why does it it look like it goes *backwards*?

We can understand both by looking at the spectra.

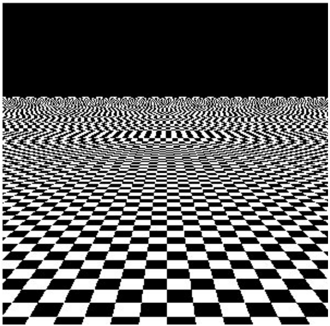
Frequency misattribution



Backwards-ness



Real-world examples



Fun on your own: read up about Moiré patterns, and various **anti-aliasing** techniques that are used in music/images/games!

Image credit: `https://textureingraphics.wordpress.com/what-is-texture-mapping/anti-aliasing-problem-and-mipmapping/`

For next time

Content:

- DT processing of CT signals
- Sampling in discrete time
- Decimation/interpolation

Action items:

1. Watch for A4

Recommended reading:

- From this class: Oppenheim 7.0-7.3
- Suggested problems: 7.1-7.6, 7.21, 7.25
- For next class: Oppenheim 7.4-7.6