ELEC 221 Lecture 17
The sampling theorem

Thursday 3 November 2022

- Midterms available for pickup at my office
- Assignment 5 available; due 11:59 Friday Nov. 11 (**no extensions**; solutions to be posted immediately after for studying)

Important: on Zoom for the next week.

- Nov. 8 class
- Office hours this Friday and next Friday
- Still available by appointment

Links will be distributed on Canvas.

We introduced the **step response** of filters.

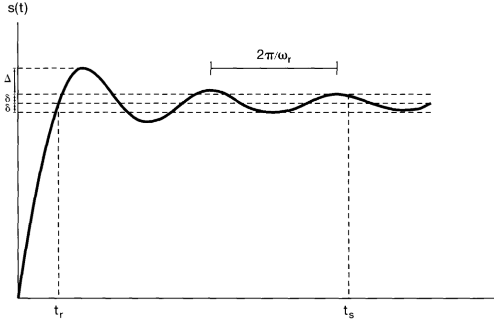


Figure 6.17 Step response of a continuous-time lowpass filter, indicating the rise time t_r, overshoot Δ, ringing frequency ω_r, and settling time t_s—i.e., the time at which the step response settles to within $\pm\delta$ of its final value.

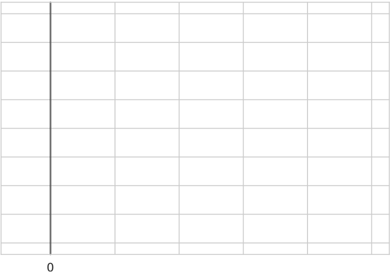
Image credit: Oppenheim 6.4

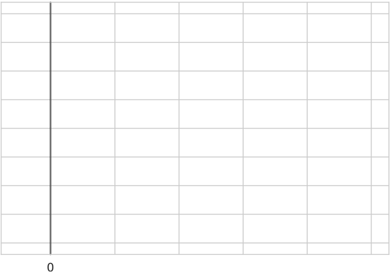
Last time

$$\tau \frac{dy(t)}{dt} + y(t) = x(t), \qquad H(j\omega) = \frac{1}{1 + j\omega\tau}$$

The impulse and step response of the system are

τ is the **time constant** of the system.





Last time

We drew some simple Bode plots.

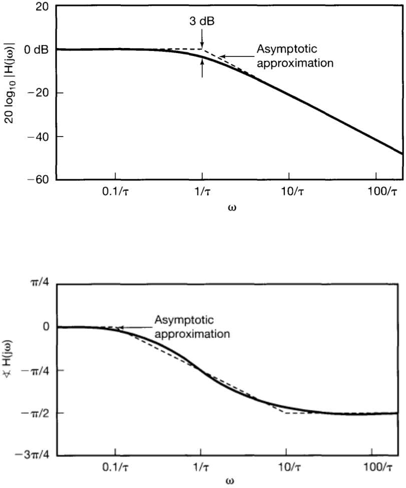
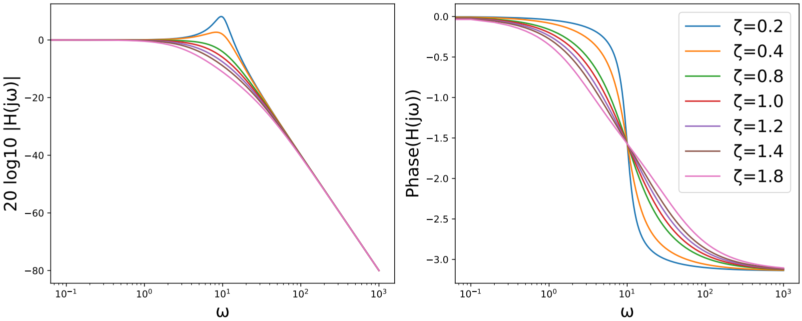


Image credit: Oppenheim 6.5

Last time

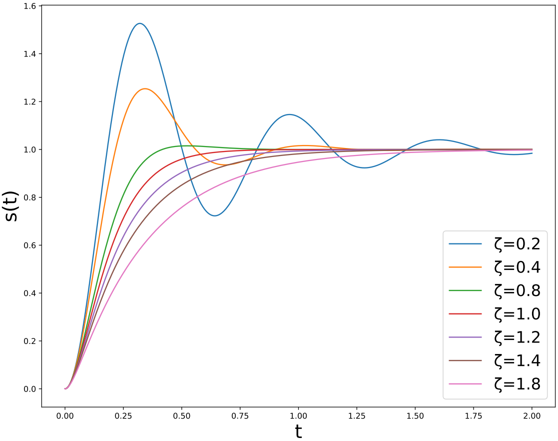
We looked at systems described by second-order ODEs.

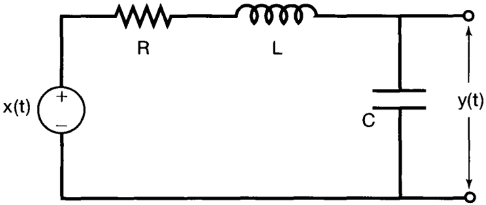
$$\frac{d^2y(t)}{dt^2} + 2\zeta\omega_n \frac{dy(t)}{dt} + \omega_n^2 y(t) = \omega_n^2 x(t)$$



Last time

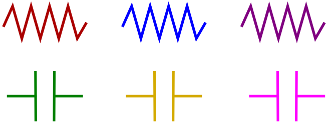
ζ is the damping ratio (can be under, over, or critically damped).





Suppose $L = 6\text{H}$. We have a box of capacitors and resistors:

300 Ω   500 Ω   1000 Ω



100 μF   200 μF   400 μF

What is the best choice to ensure step response doesn't oscillate?

Image credit: Oppenheim P6.19.

Last time

$$x(t) = LC\frac{d^2y(t)}{dt^2} + RC\frac{dy(t)}{dt} + y(t)$$

Solution: compute the frequency response

Find that

Last time

If $\zeta = (R/2)\sqrt{C/L}$, and $L = 6$H, we want

Best choice is

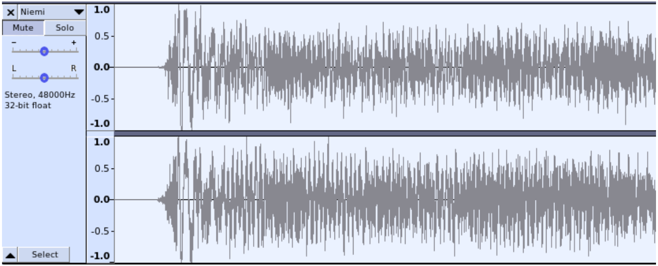
- state the sampling theorem
- define the Nyquist sampling rate and determine if a sampling rate is sufficient to reconstruct a signal
- construct systems of filters to interpolate a signal from its samples
- describe the phenomenon of aliasing

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
```



Sampling

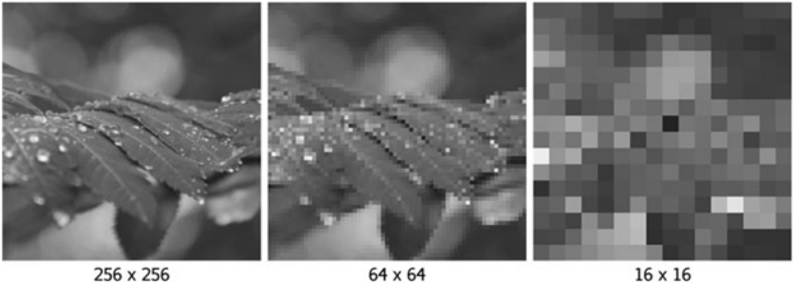


Image credit: `https://what-when-how.com/introduction-to-video-and-image-processing/image-acquisition-introduction-to-video-and-image-processing-part-2/`

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

Sampling

We saw that the discrete Fourier transform was a set of equally-spaced samples of the discrete-time Fourier transform.

The discrete Fourier transform

What if we sample this signal at particular values of $k\omega = k2\pi/N$?

$$\begin{aligned} X(e^{jk2\pi/N}) &= \sum_{n=0}^{N-1} x[n]e^{-jk2\pi n/N} = N\tilde{X}[k] \\ \frac{1}{N}X(e^{jk2\pi/N}) &= \tilde{X}[k] \end{aligned}$$

The discrete Fourier transform is a set of equally spaced samples of the full discrete-time Fourier transform.

Key point 1: Any signal $x[n]$ can be uniquely specified by a finite set of samples from its DTFT (i.e., its DFT).

22 / 41

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.**

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

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 is

This is all time domain; what happens in the frequency domain?

By the multiplication property,

But what is $P(j\omega)$? We haven't evaluated this yet...

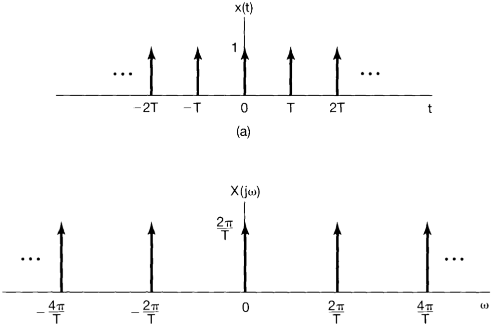
Impulse train sampling

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

Impulse train sampling

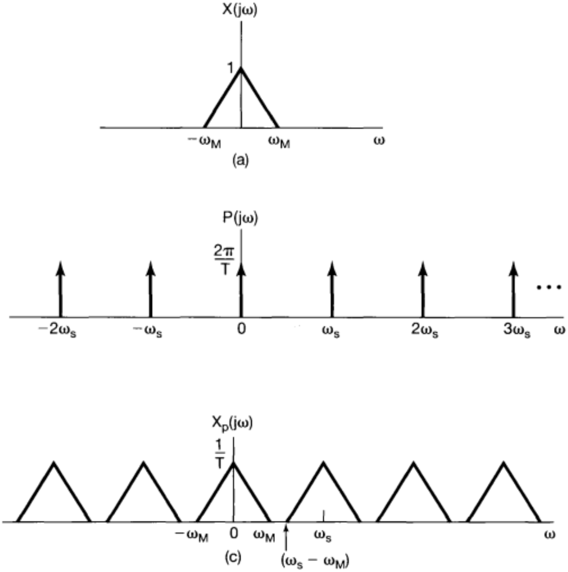
We need to find the Fourier series coefficients of the periodic impulse train.

Impulse train sampling



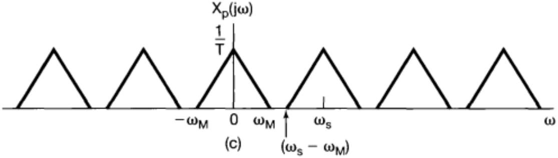
Impulse train sampling

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

"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 $x(t)$."

The sampling theorem

Let's show this graphically:

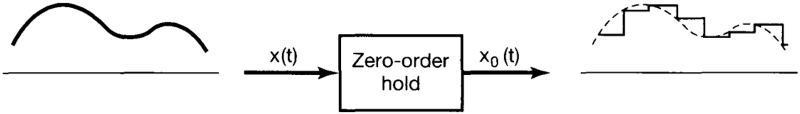
The sampling frequency is key:

- $\omega_s = 2\omega_M$ 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?

Sampling in practice: zero-order hold

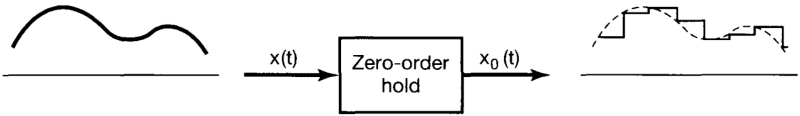
In reality we cannot generate perfect narrow, large-amplitude impulses. Instead:



Can we still reconstruct our signal?

Image credit: Oppenheim 7.1

Sampling in practice: zero-order hold



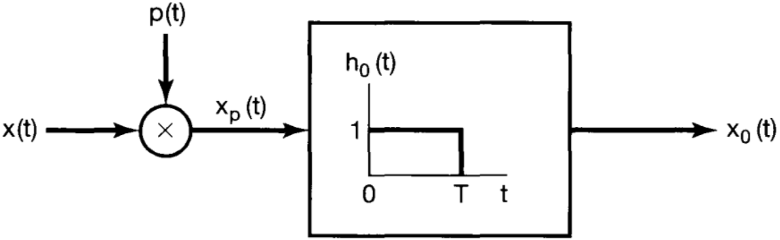
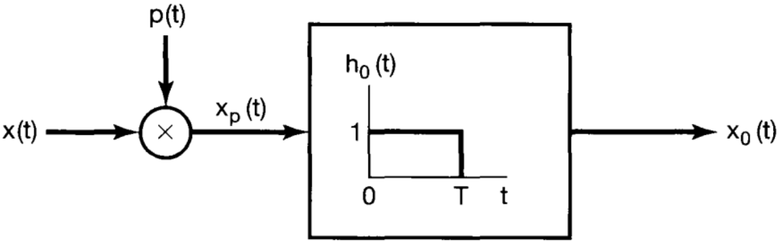
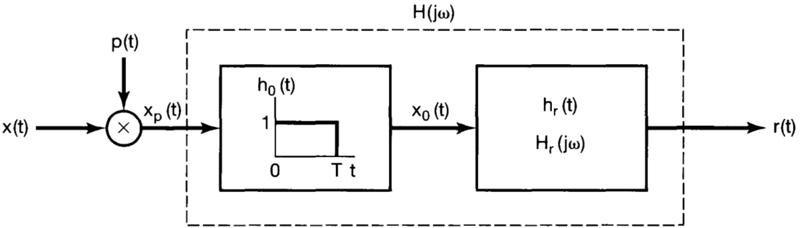


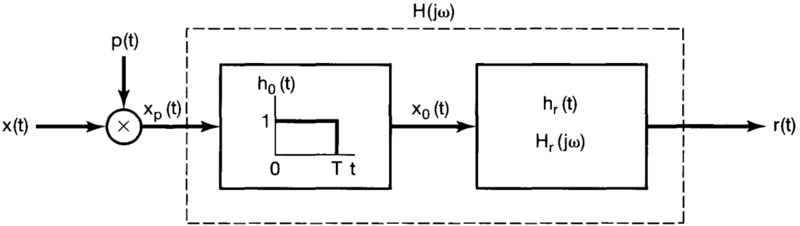
Image credit: Oppenheim 7.1

Sampling in practice: zero-order hold





Sampling in practice: zero-order hold

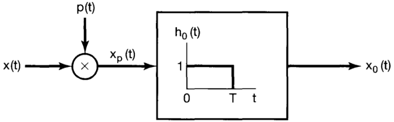


To obtain $r(t) = x(t)$, need $H_r(j\omega)H_0(j\omega) = H(j\omega)$ for ideal lowpass filter.

But what is $H_0(j\omega)$?

Image credit: Oppenheim 7.1

Sampling in practice: zero-order hold

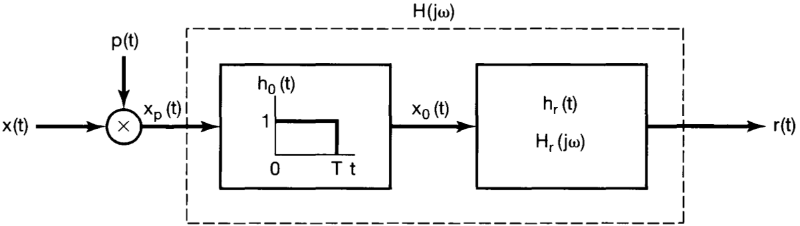


Square pulse between $-T_1$ and T_1:

Use properties of the Fourier transform to obtain

Image credit: Oppenheim 7.1

Sampling in practice: zero-order hold



If $H_r(j\omega)H_0(j\omega) = H(j\omega)$ (ideal lowpass filter) and

then we need

Sampling in practice: zero-order hold

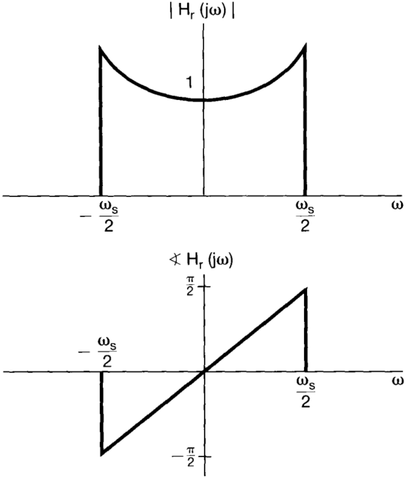
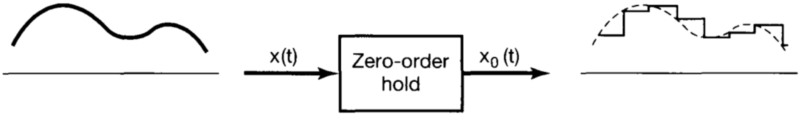


Image credit: Oppenheim 7.1

In some cases, the ZOH actually provides a good enough interpolation:



But we can do a lot better using, e.g., linear (first-order hold) or higher-order polynomial reconstruction methods.

Image credit: Oppenheim 7.1

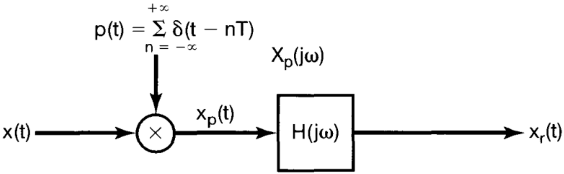
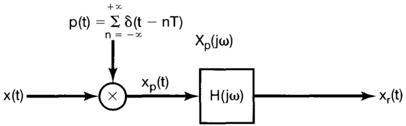


Image credit: Oppenheim 7.2



For lowpass filter with cutoff ω_c and gain T,

Then

Image credit: Oppenheim 7.2

Sometimes zero- or first-order are good enough; increasing the order will improve interpolation at the cost of complexity.

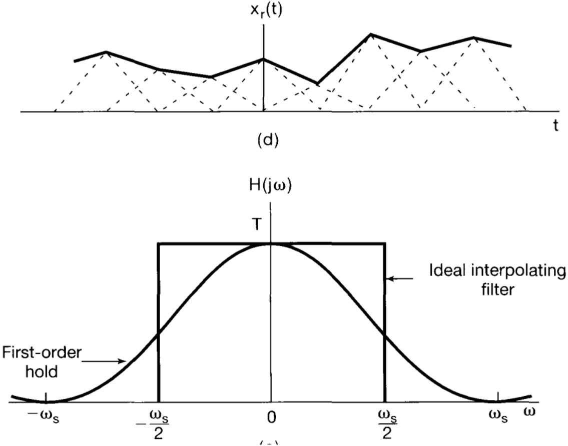


Image credit: Oppenheim 7.2

Aliasing

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

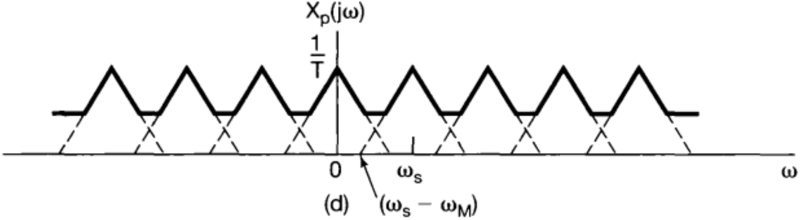


Image credit: Oppenheim 7.1

Aliasing

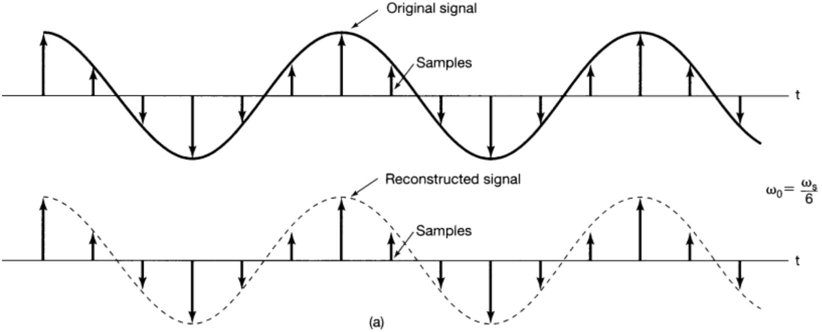


Image credit: Oppenheim 7.3

Aliasing

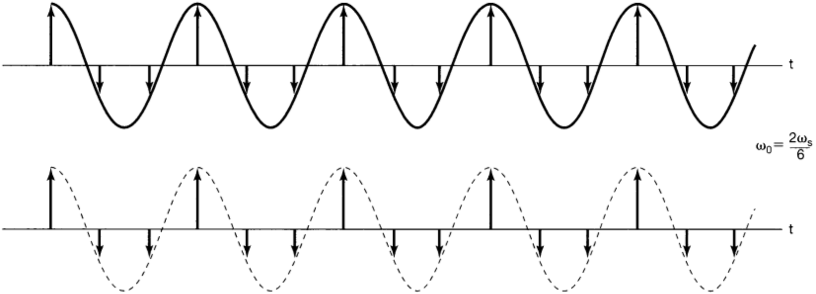


Image credit: Oppenheim 7.3

Aliasing

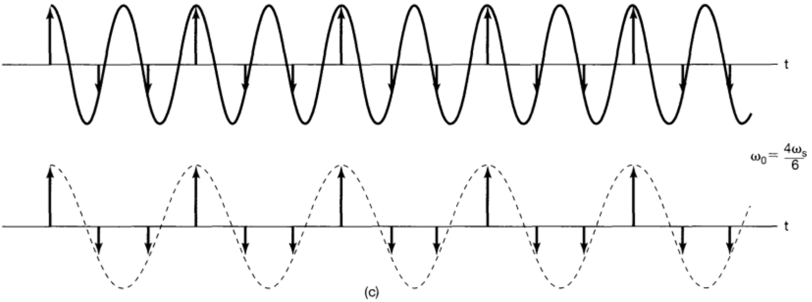


Image credit: Oppenheim 7.3

Aliasing

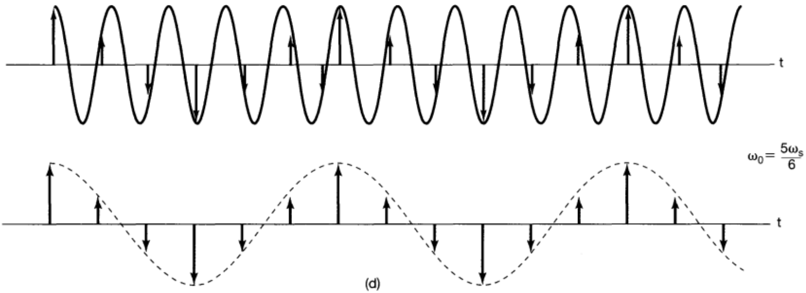
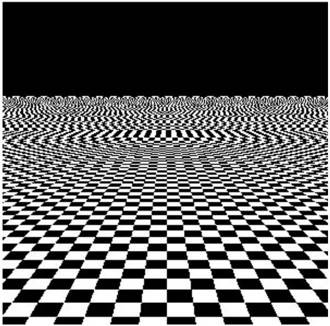


Image credit: Oppenheim 7.3

Aliasing

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/`

Today

Learning outcomes:

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

Oppenheim practice problems: 7.1-7.7, 7.21, 7.25

For next time

Content:

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

Action items:

1. Assignment 5 due 11:59pm Friday 11 Nov
2. Midterm 2 Monday 14 Nov during tutorial

Recommended reading:

- From this class: Oppenheim 7.1-7.3
- For next class: Oppenheim 7.4-7.6