[11:00-11:05] Logistics for midterm #1

The exam is open book, open note, open laptop/tablet, but no networking is allowed.

If you want the ability to use MATLAB on the exam, ensure that you have a working local version installed, since you will not be able to use the web version.

The exam will cover material up to and including the 9/30 lecture. The exam will cover topics relating to the lectures, homework assignments, and mini project.

[11:05-11:45] Sampling and aliasing

When sampling, any frequencies beyond $(-\frac{f_s}{2}, \frac{f_s}{2})$ will alias down to a frequency within this range.

A sinusoidal signal $x(t) = A\cos(2\pi f_0 t + \phi)$ becomes $x[n] = \cos\left(2\pi \frac{f_0}{f_s} n\right)$ when sampled. The samples $y[n]$ of $y(t) = A\cos(2\pi(f_0 + \ell f_s)t + \phi)$ are identical to the samples of $x[n]$ for any integer ℓ.

The scenario where $f_0 < f_s/2$ is called oversampling.

The scenario where $f_0 > f_2/2$ is called undersampling.

Example (undersampling):

$$x(t) = \cos(2\pi f_0 t), \qquad f_0 = 100 \text{ Hz}, \qquad f_s = 80 \text{ Hz}$$

$$\hat{\omega}_0 = 2\pi \frac{f_0}{f_s} = 2.5\pi$$

Since $x[n] = \cos(2.5\pi n) = \cos(0.5\pi n + 2\pi n) = \cos(0.5\pi n)$ the samples of $\cos(2\pi\ 100\ t)$ are the same as the samples of $\cos(2\pi\ 20\ t)$ when sampled at $f_s = 80$ Hz (aliasing).

However, applying a standard discrete-to continuous reconstruction procedure to $x[n]$ will result in a signal that resembles $\cos(2\pi\ 20\ t)$, since $-\frac{f_s}{2} < 20 \text{ Hz} + \ell f_s < \frac{f_s}{2}$ is only satisfied when $\ell = 0$.

To mitigate the effect of sampling, we can apply a low-pass analog filter (e.g. RC filter) to attenuate any frequencies above $f_s/2$.

Example (folding by undersampling)

$$x(t) = \cos(2\pi f_0 t), \qquad f_0 = 100 \text{ Hz}, \qquad f_s = 125 \text{ Hz}$$

$$x[n] = \cos\left(2\pi \frac{f_0}{f_s} n\right) = \cos(1.6\pi\, n) = \cos(1.6\pi n - 2\pi n) = \cos(-0.4\,\pi n)$$

[11:40-] Reconstruction (discrete-to-continuous conversion)

The general form of interpolation is a mixed (continuous and discrete) convolution:

$$\tilde{y}(t) = \sum_{-\infty}^{\infty} y[n]\, p(t - T_s n)$$

Input: discrete-time sequence $y[n] = y(nT_s)$

Output: continuous-time signal that is an approximation of $y(t)$

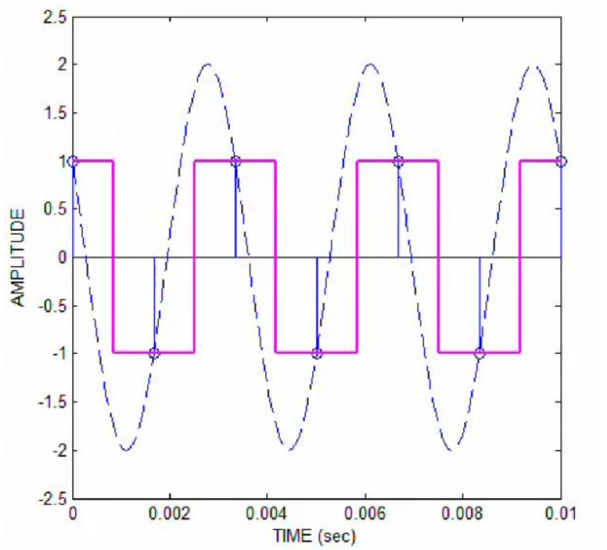
The pulse function $p(t)$ is chosen to have unit amplitude and/or area.

Rectangular pulse with height 1 and width T_s.

Linear interpolation: equivalent to $p(t) =$ **triangular pulse** with height 1 and width $2T_s$.

To avoid interfering with other samples, $p(t \pm \ell T_s) = 0$ for all non-zero integers $\ell \neq 0$.

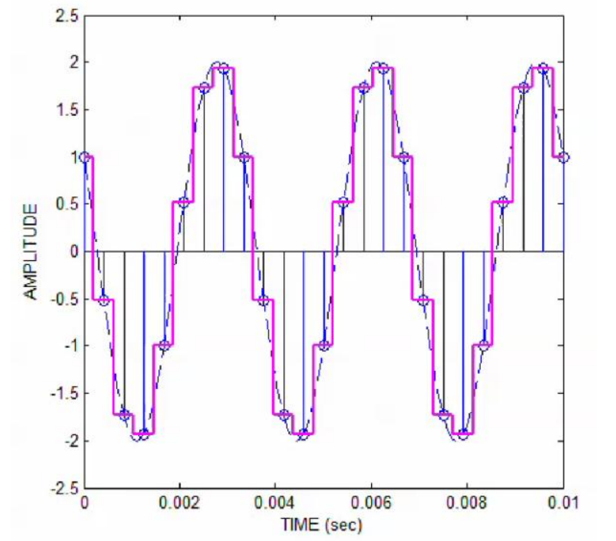
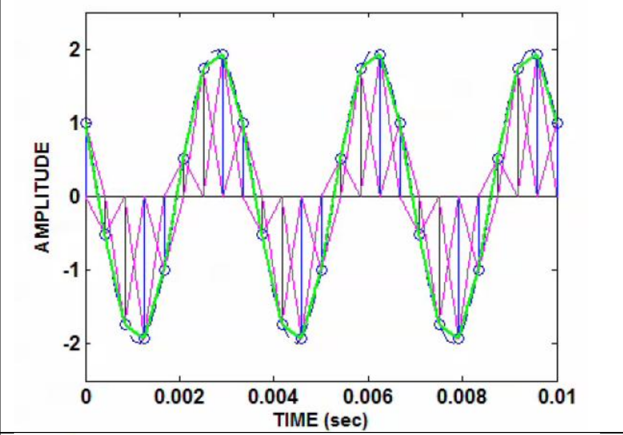
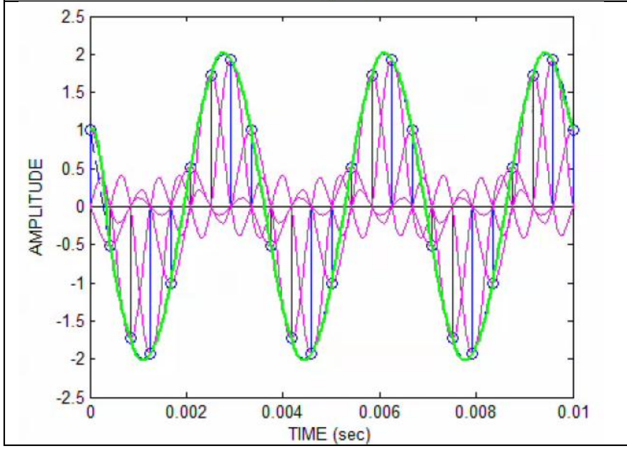
Sinc pulse: $p(t) = \text{sinc}(t/T_s) = \frac{\sin(\pi t/T_s)}{\pi t/T_s}$. The sinc pulse has infinite overlap with other sinc pulses, but the zero crossings occur at other sampling times to avoid interference.



Reconstruction with a square pulse

Sampling near the Nyquist rate

- Captures the correct number of zero crossings
- Amplitude is reduced
- Shape is not captured

	Reconstruction with a square pulse **Sampling at >8x the Nyquist rate** - Accurately tracks the amplitude - Shape is closer to a sinusoid
	Reconstruction with triangular pulse **Sampling at >8x the Nyquist rate** - Shape improves even more
	Reconstruction with truncated sinc pulse **Sampling at >8x the Nyquist rate** - Shape is nearly perfect - Amplitude is nearly perfect

[11:10] Power consumption

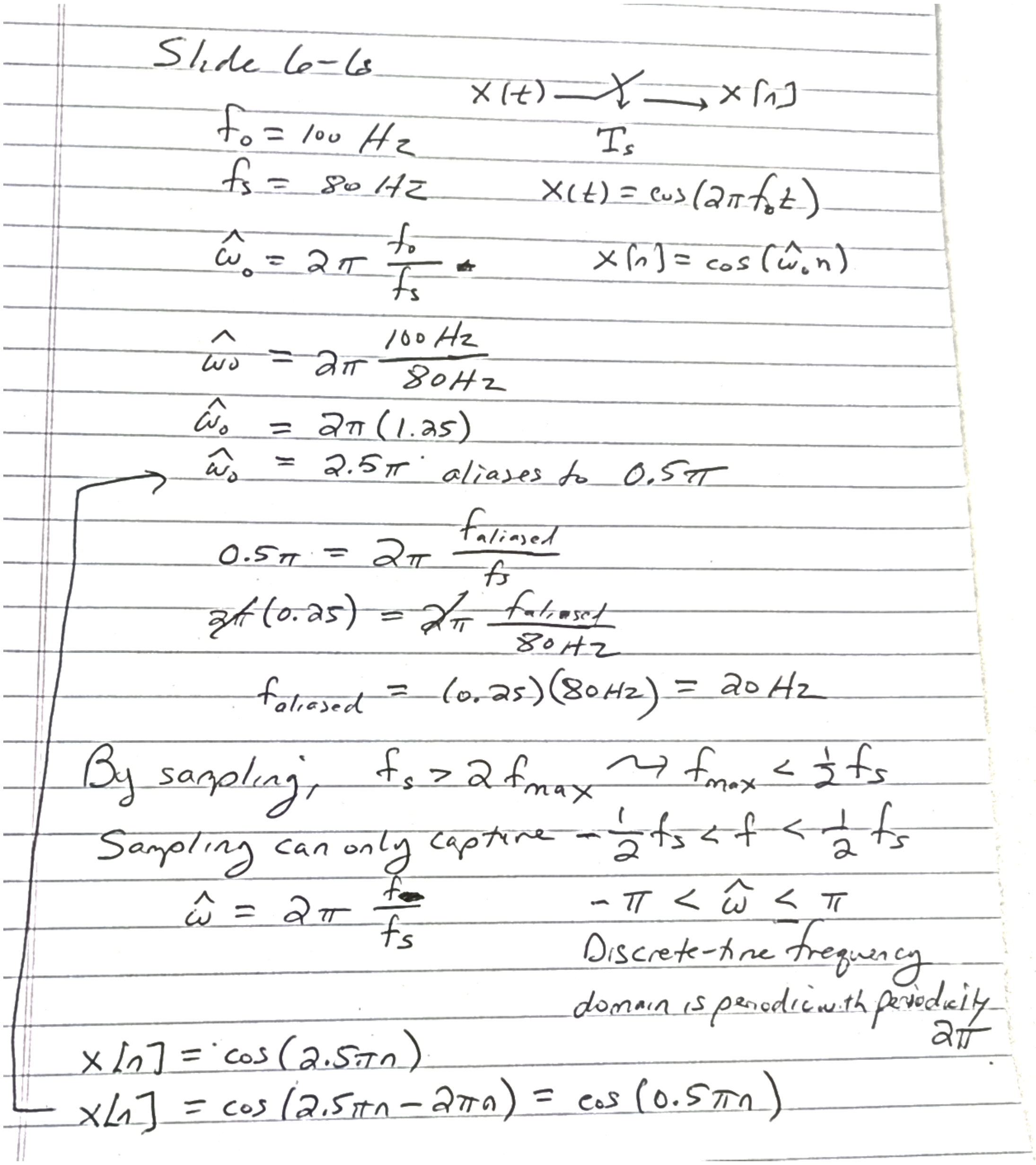
In a circuit, the dynamic power can be modeled by

$$P = ACV^2 f$$

Where f is the operating frequency of the circuit.

Some data converters have power $\propto f^2$. Additionally, many signal processing algorithms require complexity between n and n^2 in the number of samples n. Thus, oversampling can be very expensive from a power perspective.

Doc cam



Slide 6-6

$x(t) \longrightarrow \times \longrightarrow x[n]$

T_s

$f_0 = 100\ Hz$

$f_s = 80\ Hz$

$x(t) = \cos(2\pi f_0 t)$

$\hat{\omega}_0 = 2\pi \frac{f_0}{f_s}$

$x[n] = \cos(\hat{\omega}_0 n)$

$\hat{\omega}_0 = 2\pi \frac{100\ Hz}{80\ Hz}$

$\hat{\omega}_0 = 2\pi(1.25)$

$\hat{\omega}_0 = 2.5\pi$ aliases to 0.5π

$0.5\pi = 2\pi \frac{f_{aliased}}{f_s}$

$\cancel{2\pi}(0.25) = \cancel{2}\pi \frac{f_{aliased}}{80\ Hz}$

$f_{aliased} = (0.25)(80\ Hz) = 20\ Hz$

By sampling, $f_s > 2 f_{max} \rightsquigarrow f_{max} < \frac{1}{2} f_s$

Sampling can only capture $-\frac{1}{2} f_s < f < \frac{1}{2} f_s$

$\hat{\omega} = 2\pi \frac{f}{f_s}$

$-\pi < \hat{\omega} \leq \pi$

Discrete-time frequency domain is periodic with periodicity 2π

$x[n] = \cos(2.5\pi n)$

$x[n] = \cos(2.5\pi n - 2\pi n) = \cos(0.5\pi n)$

Power Consumption in a Circuit

Continuous Time Circuits $P \propto A\, C\, V_{dd}^2\, f$

$\uparrow$ activity factor

Discrete-Time Algorithms

Fast Fourier Transform (FFT)

for a block of N samples

$N \log_2 N$ for N is a power of two

N^2 N is prime

Some A/D converters: Power $\propto$ f_s^2

Overall, the power could increase by f_s to f_s^2 for increasing f_s