ECE 313 Lecture Notes

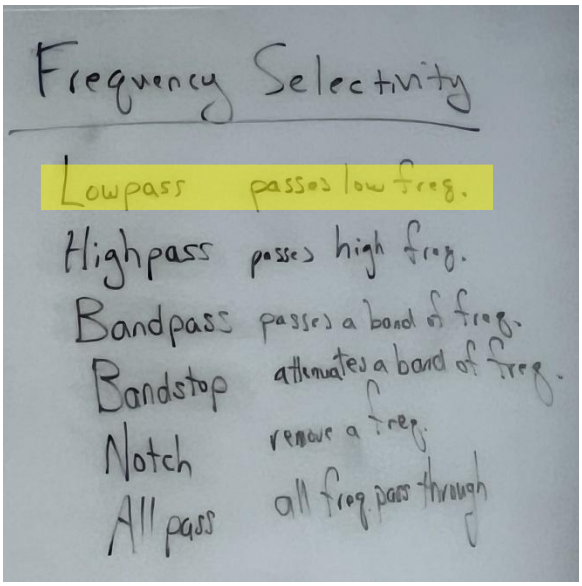
Lecture 9 on Frequency Responses Part 2

Thursday, October 22, 2026

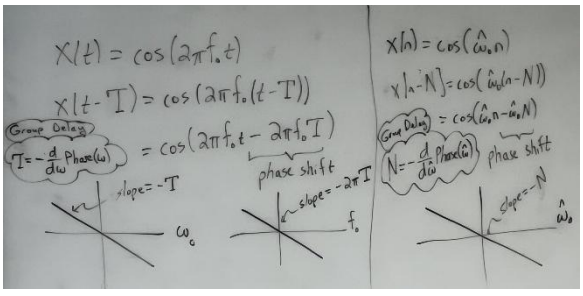
By TA Alice Liu

…

Sinusoidal Response of FIR Filters (Review)



- When the input to a Finite Impulse Response (FIR) filter is a sinusoidal signal at a fixed frequency, then the output in the same sinusoidal signal scaled by the FIR frequency response at the sinusoid's frequency.
- The FIR frequency response will scale the input sinusoid in magnitude and shift the sinusoid in phase.
- So far, in course, we've been examining Lowpass Filters (LPF) which pass low frequencies and attenuate high frequencies.
- If an FIR filter has even or odd symmetry in its impulse response, we are able to factor the filter's frequency response into closed-form expressions for the phase and magnitude responses:
  - $H(e^{j\omega}) = |H(e^{j\omega})|\, e^{j \measuredangle H(e^{j\omega})}$
  - $|H(e^{j\omega})|$ → frequency response
  - $\measuredangle H(e^{j\omega})$ → phase response
- When the *phase response/phase-shift*, $\measuredangle H(e^{j\omega})$, is LINEAR, all frequency components experience the same CONSTANT *delay* from input to output:
  - When a sinusoidal signal passes through a delay block, the phase shift is linear in the sinusoidal frequency and the delay, as shown on the right:
    - CT: $phase(\omega) = -T\,\omega$
    - DT: $phase(\hat{\omega}) = -N\,\hat{\omega}$



where the delay is T seconds for CT and N samples for DT.

- To compute the (group) delay from the phase, we negate the slope with respect to the frequency (more on next page)

....

The Matlab freqz command only plots frequency responses from 0 to π. Why?

- How come we only give you half of the *magnitude* response?
  - Because it's ***even*** symmetric
  - To get the other half, we flip the magnitude response across the y-axis
- How come we only give you half of the *phase* response?
  - Because it's ***odd*** symmetric
  - To get the other half, we flip it across **both** the x-axis & y-axis

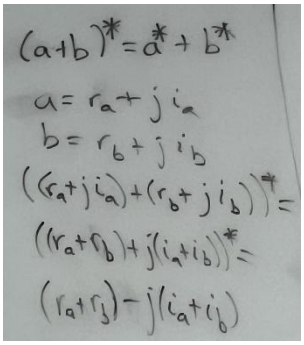
...

Back to Group Delay

- CT Group delay = $T = -\frac{\partial}{\partial \omega} Phase(\omega)$
- Change in phase (in rad) divided by change in frequency (in rad/s) to give seconds
- DT Group delay = $N = -\frac{\partial}{\partial \hat{\omega}} Phase(\hat{\omega})$
- Change in phase (in rad) divided by change in frequency (in rad/sample) to give samples

...

Complex Numbers



- Conjugate of Sum Review (slide 9-7)

$$H\left(e^{-j\hat{\omega}}\right) = \sum_{k=0}^{M} h[k]\, e^{j\hat{\omega}k} = H^*\left(e^{j\hat{\omega}}\right)$$

2-Point Averaging Filter

- Where is the midpoint of the impulse response?
  - For 2-point averaging filter, the midpoint is at sample index 0.5
  - Note: This value does not appear on the impulse response, but it is indeed the midpoint value to use for frequency response and other computations
- What is the group delay for this filter?
  - ½ sample

…

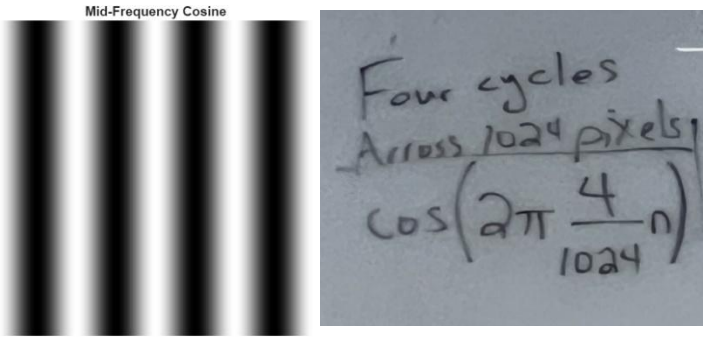
First-order Difference Filter

- Similar to the two-point (first-order) averaging filter, but instead of adding the current input value and the previous input value, it's a difference
- Shares same midpoint of the impulse response as 2-point averaging filter, n = 0.5
- What's the group delay?
    - Also just ½ sample

…

Image Processing Demo #1: imageRampsCosines.m

- Unsigned eight-bit grayscale image:
    - 0 is the background intensity (black for monitors and white for printing)
    - 255 is the foreground intensity (white for monitors and black for printing)
- Mid-gray is an intensity value of 127.5, which cannot be represented as an unsigned eight-bit value. The demo uses 127. (The other possible choice is 128.)
- Example: Cosine completes exactly four horizontal periods over 1024 samples.
    - Image has four vertical black bars and four vertical white bars
    - What's the discrete-time frequency?



- As we increase the horizontal frequency, we see narrower and narrower black and white lines alternating. Looks like texture.
- High-frequency content in images (i.e. where the image suddenly changes in intensity) includes…
    - Edges
    - Lines
    - Corners
    - High contrast areas
    - Textures

Image Processing Demo #2: [Image filtering demo]

- Lowpass filtering *reduces* high frequencies and *passes* low-frequencies
    - Results in blurring of the original image
- Thus, 1st order difference (highpass) filtering *keeps* high-frequency content and *reduces* low-frequency content
    - Results in extraction of edges, lines, corners, high contrast areas, and texture
- Image processing – how do we apply a 1-D filter to a 2-D image?
    - Apply the filter across rows and then down the columns of the result

