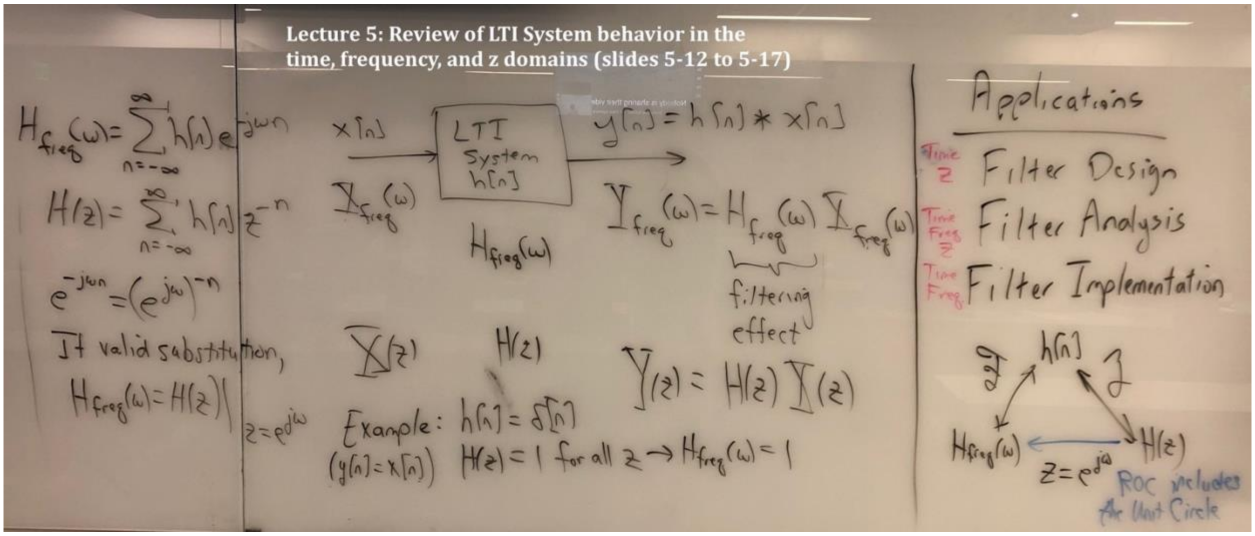
[10:30] Takeaways from last lecture

- Relation between z-transform and frequency response
  - If ROC includes unit circle, then the substitution $z = e^{j\omega}$ is valid



- Image processing demo #1: [imageRampsCosines.m]
  - Images are two dimensional signals. x and y coordinates replace time variable t/n. The amplitude of the signal represents brightness.
  - Brightness (sometimes called luminance) represented as an unsigned 8-bit number at each pixel location.
  - For display, luminance value of
    - zero corresponds to black (background intensity)
    - 255 corresponds to white (foreground intensity)
    - 127.5 is mid-gray
  - For printing, luminance value of
    - zero corresponds to white (background intensity of the paper)
    - 255 corresponds to black (foreground intensity of ink on the paper)
    - 127.5 is mid-gray

Linear ramp: Brightness $= \underbrace{\alpha}_{\text{slope}} \cdot \underbrace{x}_{\text{x−coordinate}}$

One period of cosine $2\pi \frac{1}{1024}$ radians per sample

Four periods of a cosine $2\pi \frac{4}{1024}$ radians per sample

Sixteen periods of a cosine $2\pi \frac{16}{1024}$ radians per sample

- Highest possible frequency for discrete-time cosine would alternate between columns of black and white pixels (π radians per sample).

[11:10] FIR filter examples

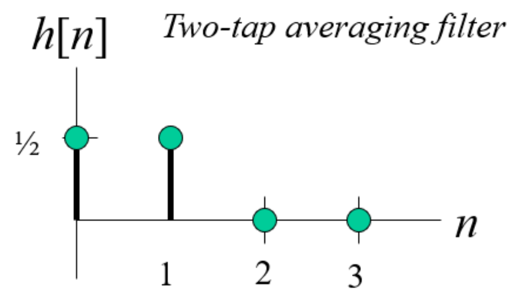
Two-sample averaging filter

$$y[n] = \frac{1}{2}x[n] + \frac{1}{2}x[n-1]$$

$$h[n] = \frac{1}{2}\delta[n] + \frac{1}{2}\delta[n-1]$$

$$H(\omega) = \frac{1}{2} + \frac{1}{2}e^{-j\omega}$$

$$= \cos(\omega/2) \quad \overbrace{e^{-j\omega/2}}^{\text{half sample delay}}$$



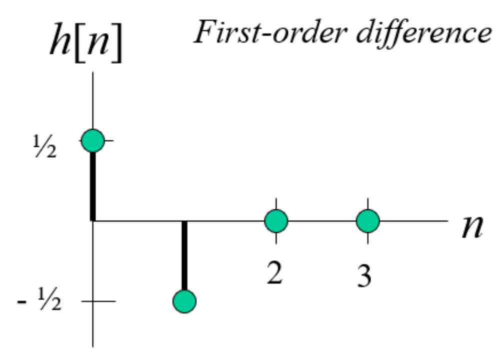
First order difference

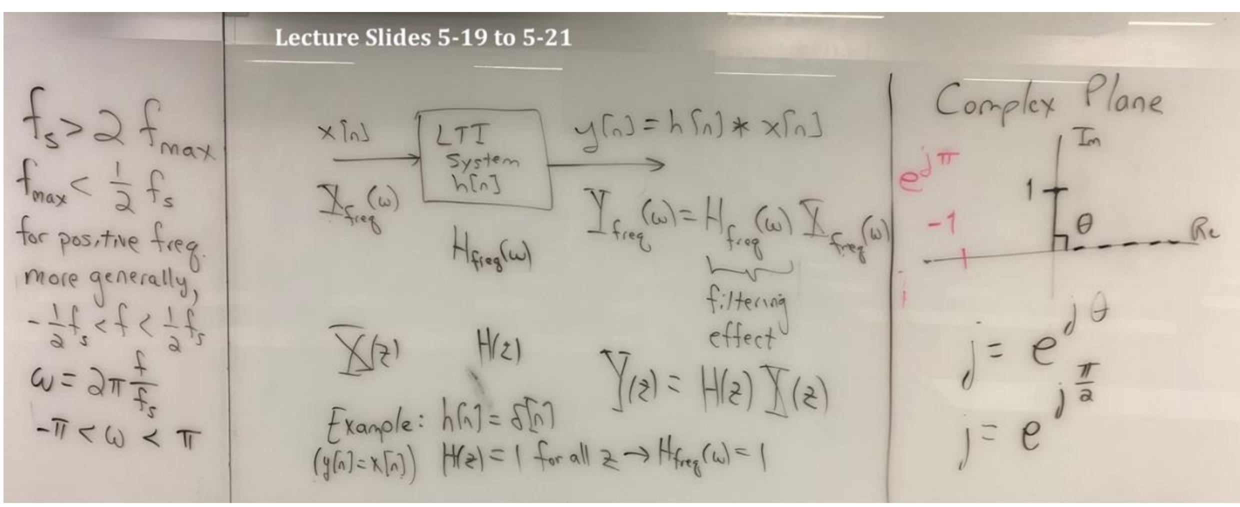
$$y[n] = \frac{1}{2}x[n] - \frac{1}{2}x[n-1]$$

$$h[n] = \frac{1}{2}\delta[n] - \frac{1}{2}\delta[n-1]$$

$$H(\omega) = \frac{1}{2} - \frac{1}{2}e^{-j\omega}$$

$$= \sin(\omega/2)\, e^{j(\pi-\omega)/2}$$





- Image processing demo #2: Filtering. What happens when five-tap averaging and first-order difference filters filter an image (either separately or in cascade)?
  - Averaging filter: blurring effect
  - Differencing filter: extracts fine detail, edges, textures.
  - Cascade of averaging and difference filters
    - extracts textures and edges of the blurred image
  - Any loss in precision in the calculations in the cascade? No.
    - The initial mandrill image has 256 rows and 256 columns and each pixel is an unsigned eight-bit number to represent gray levels [0, 255]
    - Averaging filter applied across each row produces a 256x260 image due to convolving a five-coefficient impulse response and the row of 256 elements – adding 5 numbers requires 3 extra bits worst case
    - Averaging filter applied down each column of the 256x260 image to produces a 260x260 image – 3 extra bits worst case
    - Difference filter applied across each row of 260x260 image produces a 260x261 image – adding 2 numbers requires 1 carry bit worst case
    - First-order difference filter is down each column of the 260x261 image to produce a 261x261 image – 1 carry bit worst case
    - The image produced by the cascade requires 16 bits in the worst case
    - All intermediate calculations were carried out in 64-bt floating point arithmetic in MATLAB with 53 bits of mantissa and 11 bits exponent