Lecture 5 Outline:
Zero-Order Hold Sampling, Upsampling and Downsampling

Announcements:
- Updated OHs:
  - Me: MWF after class & by appt. (made by email or before/after class, late MWF ok)
  - Wed 4PM-5PM (Malavika), 6PM-7PM (John); Thu 4PM-5PM (Georgia), Pack 3rd fl.
- HW 1 due Friday 4pm. HW 2 will be posted Friday
- Pauly’s FT handouts will be posted (different notation than Kahn’s)

- Review of Lecture 3 (Lecture 4 was Matlab tutorial)
- Sampling and Reconstruction with Zero-Order Hold
- Discrete-Time Upsampling and Reconstruction
- Digital Downsampling
Review of Lecture 3

- **Digital to Analog Conversion**

  \[
  x_d[n] \rightarrow \text{Discrete-To Continuous} \rightarrow x_d(nT_s) \rightarrow \text{LPF} \rightarrow x_r(t)
  \]

- **Quantization**

  \[
  x(t) \rightarrow x^Q(nT_s) \rightarrow \text{ADC} \rightarrow 101101... \\
  101101... \rightarrow \text{DAC} \rightarrow x^Q_d[n] = \sum_{i=-\infty}^{\infty} (-A + k[i]\Delta)\delta(i-n)
  \]
**Review Continued:**

**Analog to Bits**

Continuous-Time Unquantized \( x(t) \) → Anti-Aliasing Lowpass Filter → Sampler \( t = nT \) → Discrete-Time Unquantized \( x_d[n] = x(t) \big|_{t = nT} \) → Quantizer → Discrete-Time Quantized Representation of \( x_d[n] \)

**Bits to Analog**

Discrete-Time Quantized \( y[n] \) → Reconstruction System → Continuous-Time \( y(t) \)

**ADC**

Digital Storage, Transmission, Signal Processing, …

**DAC**
• Ideal sampling not possible in practice
  • delta function train cannot be realized

• In practice, ADC uses zero-order hold to produce $x_d[n]$ from $x_0(t)$
  • Design uses switch, capacitor and resistors

• Reconstruction of $x(t)$ from $x_0(t)$ removes $h_0(t)$ distortion
  • Multiplication in frequency domain by $1/H_0(j\omega)$

• Also use zero-order hold for reconstruction in practice
Discrete-Time Upsampling

- Inserts L-1 zeros between each $x_d[n]$ value to get upsampled signal $x_e[n]$
- Compresses $X_d(e^{j\Omega})$ by L in $\Omega$ domain and repeats it every $2\pi/L$; So $X_e(e^{j\Omega})$ is periodic every $2\pi/L$
- Leads to less stringent reconstruction filter design than ideal LPF: zero-order hold often used
Reconstruction of Upsampled Signal

- Pass through an ideal LPF $H_i(e^{j\Omega})$ to get $x_i[n]$
- $x_i[n]=\hat{x}_e[n]=x(nT_s/L)$ if $x(t)$ originally sampled at Nyquist rate ($T_s < \pi/W$)
- Relaxes DAC filter requirements (approximate LPF/zero-order hold reconstructs $x(t)$ from $x_i[n]$; better filter to reconstruct from $x_d[n]$ needed)

$X_i(e^{j\Omega}) \iff x_i[n]$

Reconstruct $x(t)$ from $x_i[n]$ by passing it through a DAC
Digital Downsampling: Fourier Transform and Reconstruction

- **Digital Downsampling**
  - Removes samples of \( x(nT_s) \) for \( n \neq MT_s \)
  - Used under storage/comm. constraints

- **Repeats** \( X_d(e^{j\Omega}) \) every \( 2\pi/M \) and scales \( \Omega \) axis by \( M \)
  - This results in a periodic signal \( X_c(e^{j\Omega}) \) every \( 2\pi \)
  - Introduces aliasing if \( X_d(e^{j\Omega}) \) bandwidth exceeds \( \pi/M \)
  - Can prefilter \( X_d(e^{j\Omega}) \) by LPF with bandwidth \( \pi/M \) prior to downsampling to avoid downsample aliasing
Main Points

- Sampling in practice uses a zero-order hold, whose distortion can be completely removed
  - Zero order hold easily implemented with a switch, capacitor, and resistors
  - Higher-order holds provide better performance in practice

- Can also use a zero-order hold in DAC for reconstruction
  - Allows for perfect recovery of original sampled signal with ideal filters
  - In practice, use filters that approximate ideal filter but are easier to implement

- Upsampling of a discrete-time signal by $L$ compresses its Fourier Transform by $1/L$
  - Equivalent to sampling $L$ times faster if original sampling rate above Nyquist; much easier to add zeros!!!!
  - Allows less stringent requirements on reconstruction filter than an ideal LPF
  - A zero-order hold is often used for the reconstruction

- Digital downsampling used when system constraints preclude storing/communicating samples every $T_s$ seconds.