Lecture 6 Outline:
Upsampling and Downsampling

- Announcements:
  - HW 1 due today 4pm. HW 2 will be posted later today

- Review of Last Lecture

- Discrete-Time Upsampling

- Reconstruction of Upsampled Signal

- Discrete-Time Downsampling

- Reconstruction of Downsampled Signal
Review of Last Lecture

Sampling with Zero-Order Hold

- Ideal sampling not possible in practice
- In practice, ADC uses zero-order hold to produce \( x_d[n] \) from \( x_0(t) \)
- Reconstruction of \( x(t) \) from \( x_0(t) \) removes \( h_0(t) \) distortion
  - Multiplication in frequency domain by \( T \cdot \Pi(\frac{\omega}{\omega_s})/H_0(j\omega) \)
- Also use zero-order hold for reconstruction in practice

\[
\sum_n \delta(t-nT_s) \quad \rightarrow \quad h_0(t) \quad \rightarrow \quad x_0(t) \quad \rightarrow \quad x(t)
\]
\[
\sum_n \delta(t-nT_s) \rightarrow x(t) \rightarrow h_0(t) \rightarrow x_0(t) \rightarrow H_r(j\omega) \rightarrow x_r(t) = x(t)
\]

\[
H_0(j\omega) = Te^{-j\omega T/2} \cdot \frac{\operatorname{sinc}(\omega T/2\pi)}{2\pi} \quad H_r(j\omega) = \frac{e^{-j\omega T/\omega_s}}{\omega_2} \cdot \prod \left( \frac{\omega}{\omega_2} \right)
\]

Zero-order hold model

\[
H_0(j\omega)H_r(j\omega) = T \cdot \prod \left( \frac{\omega}{\omega_s} \right)
\]

\[
x(t) \cdot x_0(t) = x_r(t)
\]

\[
|X(j\omega)|
\]

\[
X_s(j\omega)
\]

\[
\left| H_0(j\omega) \right|
\]

\[
\left| H_r(j\omega) \right|
\]

\[
X_0(j\omega)
\]

\[
X(j\omega)
\]

\[
H_r(j\omega) = \frac{e^{-j\omega T/\omega_s}}{\omega_2} \cdot \prod \left( \frac{\omega}{\omega_2} \right)
\]

\[
T \sum_{k=-\infty}^{\infty} X(j(\omega - k\omega_s))
\]

\[
\text{reconstruction filter} \quad \text{zero-order hold} \quad \text{copies of } X(j\omega) \text{ at } k\omega_s
\]
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 $x_i[n]$
- $x_i[n] = \dot{x}_e[n] = x(nT_s/L)$ 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

$x_d[n] = x(nT_s) \leftrightarrow X_d(e^{j\Omega})$

$x_i[n] = \dot{x}_e[n] = x(nT_s/L)$ if $T_s < \pi/W$

Reconstruct $x(t)$ from $x_i[n]$ by passing it through a DAC
**Discrete-Time 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 bandwith \(\pi/M\) prior to downsampling to avoid downsample aliasing
Main Points

- **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

- **Reconstruct upsampled signal by passing it through an ideal/approximate LPF**
  - Upsampling eases the requirements on the reconstruction filter; a zero-order hold often used instead of a LPF
  - Reconstructed signal is sinc interpolation of signal sampled every $T_s$ and upsampled at rate L or signal sampled every $T_s/L$

- **Digital downsampling used when system constraints preclude storing/communicating samples every $T_s$ seconds.**
  - Removes samples of $x(nT_s)$ for $n\neq MT_s$; Repeats $X_d(e^{j\Omega})$ every $2\pi/M$
  - Introduces aliasing if $X_d(e^{j\Omega})$ bandwidth exceeds $\pi/M$