Administrative

- Quiz 1 back after class - mean 29 (of 35), std dev ~5 (but curve was skewed)
- Exam 1 is next Monday
  - I'll make sure all HW up through today's is posted
  - Today's lecture is NOT on exam 1
  - Sunday night review session of interest?
    - I also have 3-5 pm office hours tomorrow
- MATLAB3 is due next Wednesday; no other HW due next week
  - If you haven't yet, sign up for Piazza

EE-125:
Digital Signal Processing

Finish All-pass Systems

Discrete Fourier Transform

Professor Tracey

Tufts

Outline

- Finishing all-pass
  - Usually, I cover minimum phase too, but am skipping this year to make room for wavelets
- DFT/FFT: where we are in the overall class
- Discrete Fourier Transform (P&M 7.1)
  - Sampling $X(\omega)$ evenly in frequency – slightly different derivation than in the book
  - Time domain aliasing and how to avoid it
  - Math-y, so clean copy of lecture notes will by on Trunk
- DFT properties (P&M 7.1)
  - Shifting, even/odd, etc.

All-pass filters (P&M 5.4.6)

- Definition: $|H_{AP}(\omega)|$ = constant
- Simplest (but most useful?) case: time delay
- More interesting example: 1 zero, 1 pole

$$H(z) = \frac{z^{-1} - a}{1 - az^{-1}}$$

- Magnitude of this is:

$$\begin{aligned} | H_{ap}(\omega) |^2 &= H(z)H(z^{-1}) \,|_{z=e^{j\omega}} \\ &= \left(\frac{z^{-1} - a}{1 - az^{-1}}\right)\left(\frac{z - a}{1 - az}\right) |_{z=e^{j\omega}} \\ &= 1 \end{aligned}$$

All-pass filters, con’t

- More general case: poles/zeros can be real or complex, and there can be many of them:

$$H_{ap}(z) = \prod_{k=1}^{N_R} \frac{z^{-1} - \alpha_k}{1 - \alpha_k z^{-1}} \prod_{k=1}^{N_C} \frac{(z^{-1} - \beta_k)(z^{-1} - \beta_k^*)}{(1 - \beta_k z^{-1})(1 - \beta_k^* z^{-1})}$$

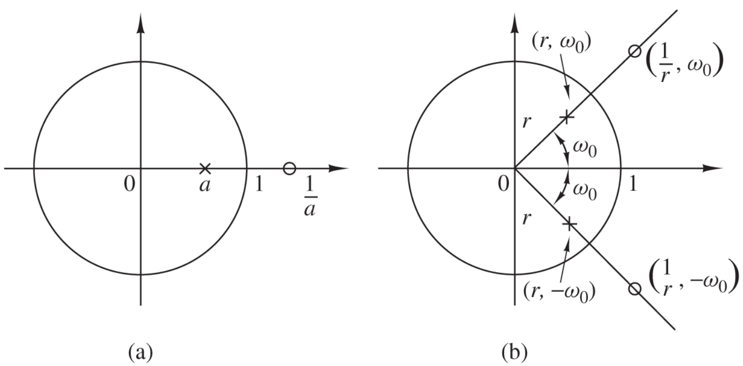

Figure 5.4.16 Pole–zero patterns of (a) a first-order and (b) a second-order all-pass filter.

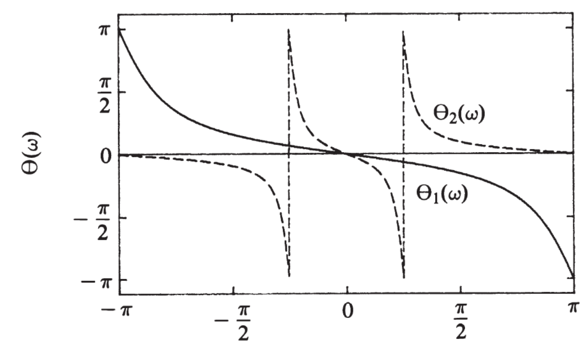

Figure 5.4.17 Frequency response characteristics of an all-pass filter with system functions (1) $H(z) = (0.6 + z^{-1})/(1 + 0.6z^{-1})$, (2) $H(z) = (r^2 - 2r\cos\omega_0 z^{-1} + z^{-2})/\ (1 - 2r\cos\omega_0 z^{-1} + r^2 z^{-2})$, $r = 0.9$, $\omega_0 = \pi/4$.

Outline

- Finishing all-pass
  - Usually, I cover minimum phase too, but am skipping this year to make room for wavelets
- DFT/FFT: where we are in the overall class
- Discrete Fourier Transform (P&M 7.1)
  - Sampling X(ω) evenly in frequency – slightly different derivation than in the book
  - Time domain aliasing and how to avoid it
  - Math-y, so clean copy of lecture notes will by on Trunk
- DFT properties (P&M 7.1)
  - Shifting, even/odd, etc.

Organizing the Fourier transforms (From Lecture 3)

Chapter 4:

	Periodic	**Aperiodic**
CT	CT Fourier series $x(t)$, c_k	CTFT $x(t)$, $X(f)$
DT	DT Fourier series $x(n)$, c_k	DTFT $x(n)$, $X(\omega)$

← Our focus until today

Looking ahead:

- For computer implementation, *discrete* quantities are most natural
- Later (in Chap 7) we'll discretize frequency as well as time:
  - Discrete Fourier Transform (DFT) ; $x(n) \leftrightarrow X(k)$
  - We'll do this by repeating our data to make signals periodic; close link between DFT and DT Fourier series
  - The FFT is just a fast algorithm for computing the DFT

Organizing the Fourier transforms (today)

Chapter 4:

	Periodic	**Aperiodic**
CT	CT Fourier series $x(t)$, c_k	CTFT $x(t)$, $X(f)$
DT	DT Fourier series $x(n)$, c_k	DTFT $x(n)$, $X(\omega)$

Our
focus today

Looking ahead:

- For computer implementation, *discrete* quantities are most natural
- Later (in Chap 7) we'll discretize frequency as well as time:
  - Discrete Fourier Transform (DFT) ; $x(n)$ <-> $X(k)$
  - We'll do this by repeating our data to make signals periodic; close link between DFT and DT Fourier series
  - The FFT is just a fast algorithm for computing the DFT

Idea: Sampling in freq vs. time

- Before we studied sampling in time:
  - –Multiply x(t) by an impulse train D(t) in time (impulse spaced at T)
  - –This gives convolution in frequency, so X(F) repeats every Fs=1/T or $X(\omega)$ repeats every 2π
  - – By sampling more finely (smaller T) we avoid aliasing
  - – We can recover the signal by selecting out just the part of $X(\omega)$ we want (via a lowpass filter)

Idea: Sampling in freq vs. time

- Before we studied sampling in time:
  - Multiply x(t) by an impulse train D(t) in time (impulse spaced at T)
  - This gives convolution in frequency, so X(F) repeats every Fs=1/T or X(ω) repeats every 2 π
  - By sampling more finely (smaller T) we avoid aliasing
  - We can recover the signal by selecting out just the part of X(ω) we want (via a lowpass filter)
- Sampling in frequency has the same structure:
  - Multiply X(ω) (or X(f)) by an impulse train in frequency, spacing $\delta\omega$, giving N points over one period.
  - This gives convolution in time, so it corresponds to analyzing a periodic signal xp(n)
  - By sampling more finely (bigger N) we avoid aliasing
  - We can recover x(n) from xp(n) by just taking the first N points

Tufts

Math for finding xp(n)

From the CTFT, we have

$$x_p(n) = \int_0^1 \left[\frac{1}{N} \sum_{k=0}^{N-1} X(f)\delta(f - \frac{k}{N}) \right] e^{+j2\pi f n} df \qquad (1.1)$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} \int_0^1 X(f) e^{+j2\pi f n} \, \delta(f - \frac{k}{N}) df \qquad (1.2)$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} X(\frac{k}{N}) e^{+j2\pi \frac{k}{N} n} \qquad (1.3)$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{+j2\pi kn/N} \qquad (1.4)$$

Aliasing in time

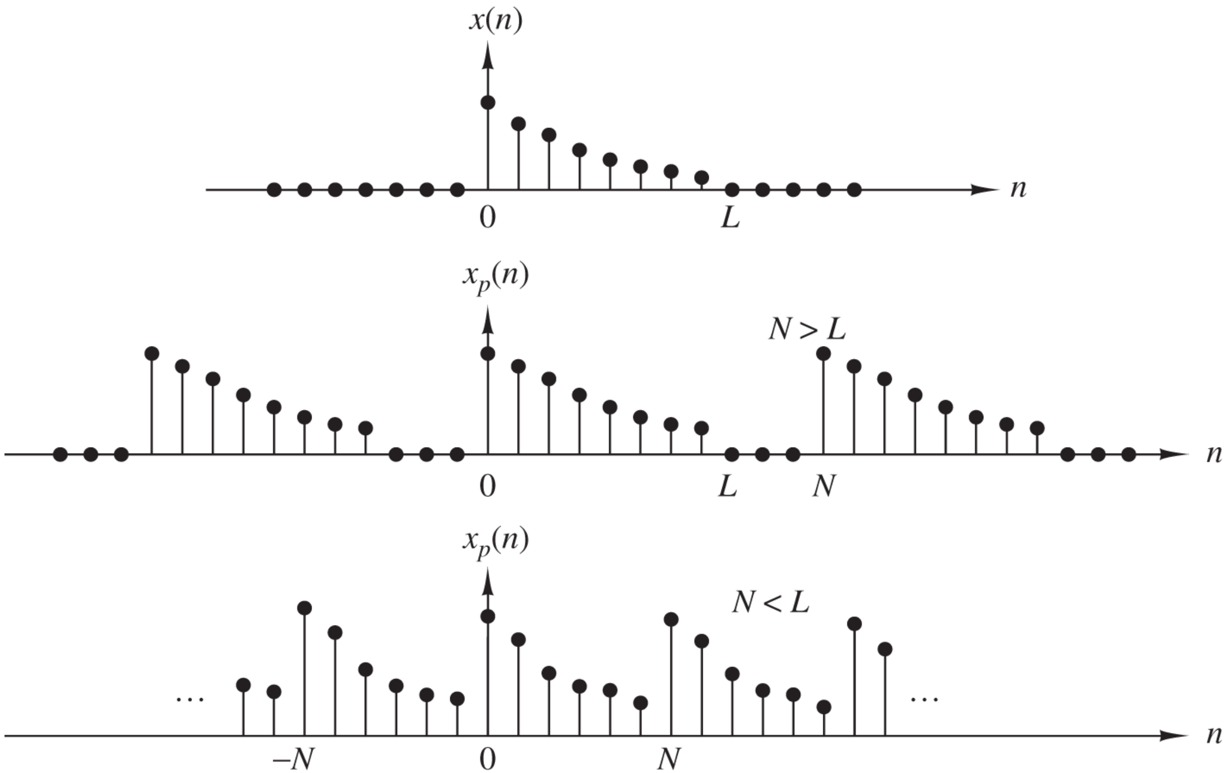

Figure 7.1.2 Aperiodic sequence $x(n)$ of length L and its periodic extension for $N \geq L$ (no aliasing) and $N < L$ (aliasing).

Circular shift of a sequence

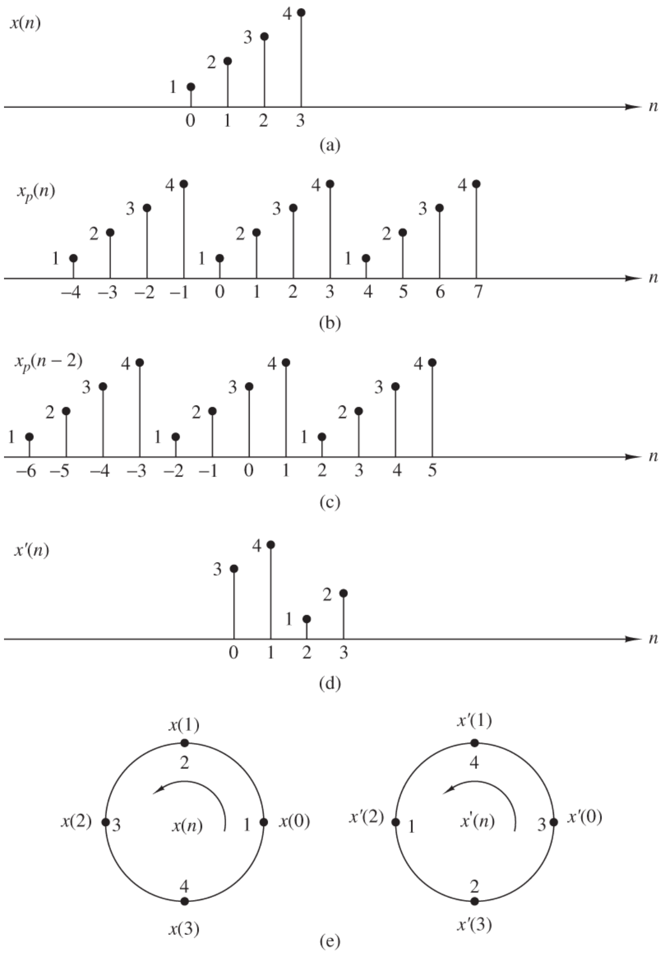

Figure 7.2.1 Circular shift of a sequence.

Tufts

Time reversal of a sequence

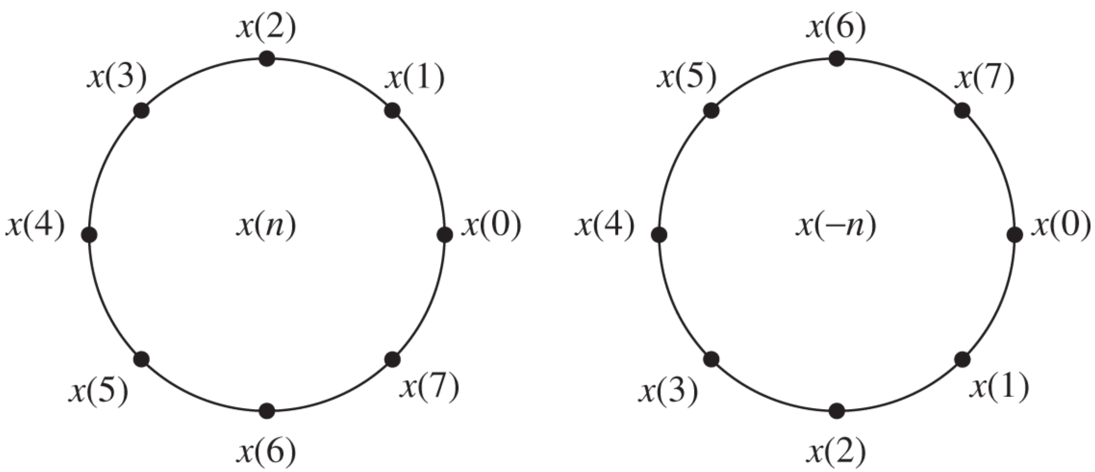

Figure 7.2.3 Time reversal of a sequence.