New Mexico State University
Klipsch School of Electrical Engineering

EE395 - Introduction to DSP

Fall 2012 - Final Exam

Name: ____________________

Solve Problems 1-3 and any three from Problems 4-9.
Circle below which three from Problems 4-9 you want graded.

Prob. 1	/ 16.6 points
Prob. 2	/ 16.6 points
Prob. 3	/ 16.7 points
Prob. 4	/ 16.7 points
Prob. 5	/ 16.7 points
Prob. 6	/ 16.7 points
Prob. 7	/ 16.7 points
Prob. 8	/ 16.7 points
Prob. 9	/ 16.7 points
Total	/ 100 points

Prob. 1

Flanging (p. 355, Section 8.2.2) is a popular audio effect similar to an echo but with a *variable* delay. The difference equation is given by

$$y(n) = x(n) + ax(n - d(n)) \tag{1.1}$$

where the variable delay is given by

$$d(n) = \frac{D}{2}\left[1 - \cos(2\pi F_d n)\right], \tag{1.2}$$

D is the maximum delay, and F_d is a low frequency, in units of [cycles/sample].

Write a C code to implement the flanging effect within `process_signal.c`. Use the functions `initialize_program.c` and `user_data.h` given on the next page. You may assume the DSP C functions, `main.c`, and `util.h` have been properly implemented as used in class and `tapi.c` has been implemented exactly as in the text on p. 356. At your option, you can disregard the supplied comments (hints) in `process_signal.c`

```
#include "user_data.h"
#include <math.h>
void process_signal(float inputRight, float inputLeft, float *outputRight,
    float *outputLeft)
{
                                      // local variable declarations (if any)

                                      // insert newest sample into the buffer


                                      // compute d


                                      // increment n (keep 0 <= n <= FS)


                                      // tap the delay line @ d using tapi()


                                      // compute right output

                                      // update circular delay line

    *outputLeft = 0.0;                // mute left output
}
```

Prob. 1 (cont.)

---

user_data.h

```
#ifndef USER_DATA
#define USER_DATA

void initialize_program(void);
void process_signal(float inputRight, float inputLeft, float *outputRight,
    float *outputLeft);

#define FS 48000            // sample rate
#define A 0.5               // echo gain
#define DELAY_LIMIT 480     // tapped delay line order also D (10 ms @ FS)
#define MAX_DELAY 481       // tapped delay line length
#define FD 1.0 / 48000.0    // Fd = 1 Hz

extern float buffer[];          // buffer to store samples
extern float *oldestSamplePtr; // pointer to oldest sample in buffer
extern int n;                   // make n a global counter for simplicity

#include "util.h"

#endif
```

---

initialize_program.c

```
#include "user_data.h"

float buffer[MAX_DELAY]={0};   // buffer to store samples initialized to zero
float *oldestSamplePtr=buffer; // pointer to oldest sample in buffer
int n=0;                       // counter

void initialize_program()
{
}
```

Prob. 2

Figure 2.1 (next page), shows magnitude spectrum plots. Each plot was computed using spectral analysis parameters:

- Signal length, L is 16 or 64 and there is no zero-padding
- DFT length, N is 16, 32, or 64
- A rectangular or Hamming window is applied prior to computing the DFT
- Each signal has either one or two complex exponentials of the form $e^{j\omega_0 n}$
- Complex exponentials frequencies, ω_0 are either -0.6π, -0.4π, 0.4π, or 0.6π

For each plot, fill in the table below with parameter values used in computing the plot. You may wish to justify your answer(s) if you believe the plot is ambiguous.

Figure	L	N	Window	# Complex Exp's	Frequency(ies)
(a)					
(b)					
(c)					
(d)					
(e)					
(f)					

Justifcations (optional):

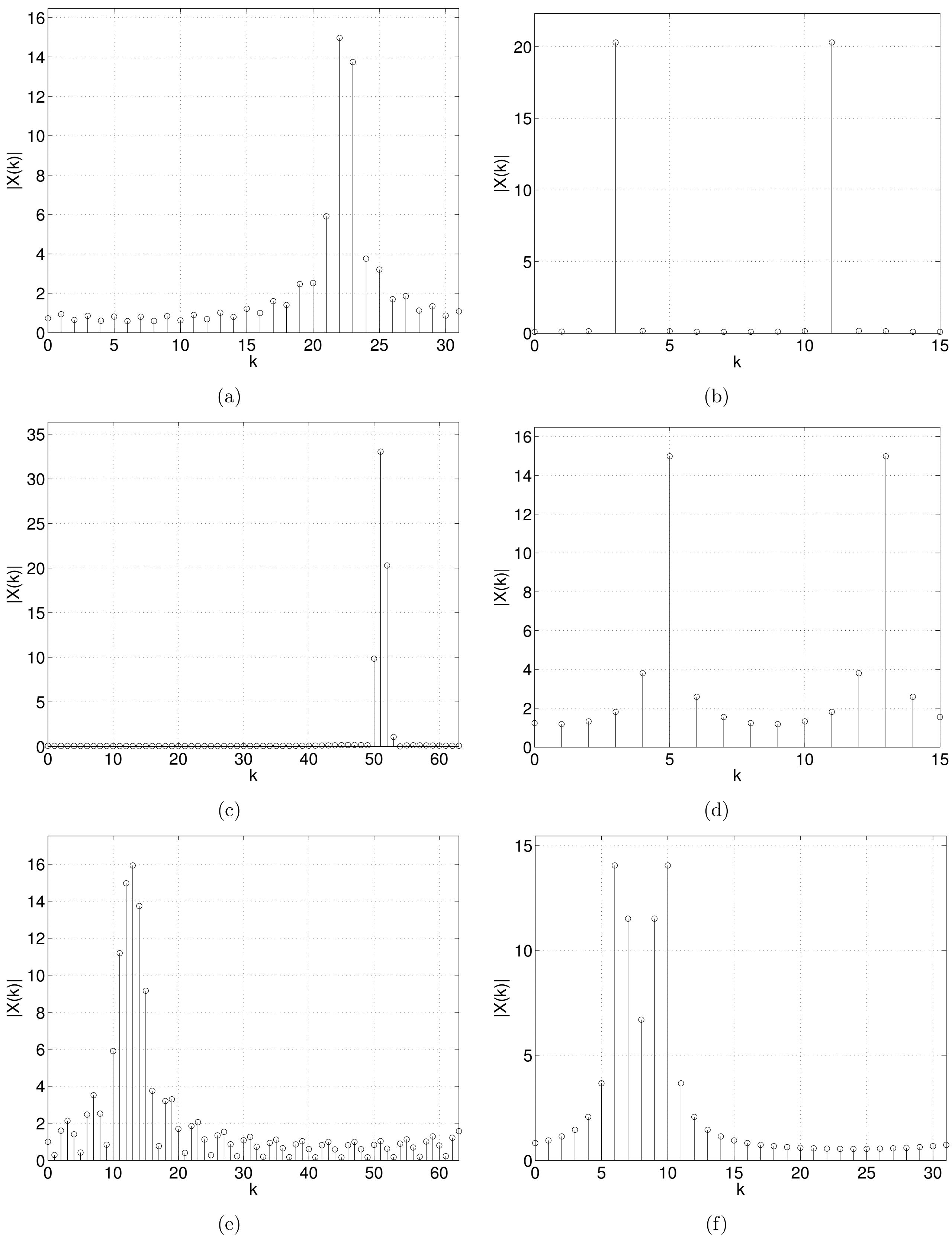


Figure 2.1: Magnitude spectrum plots for problem 2.

Prob. 3

Figure 3.1 (next two pages), shows magnitude response plots for digital filters. The filter designs are based on the following parameters:

- FIR filters are designed using the window method with either a rectangular or Hamming window; FIR filters have a *length* of either 8 or 32
- IIR filters are designed using the bilinear transform with either a Butterworth or Chebyshev prototype; IIR filters have an *order* of either 3 or 8
- The cutoff frequency, ω_c is the same for all filters.

For each plot, fill in the table with filter parameter values used in computing the plot. Some filters may occur twice. You may wish to justify your answer(s) if you believe the plot is ambiguous.

Figure	FIR or IIR	Window (FIR only)	Prototype (IIR only)	Length/Order
(a)				
(b)				
(c)				
(d)				
(e)				
(f)				
(g)				
(h)				

Justifcations (optional):

Prob. 3 (cont.)

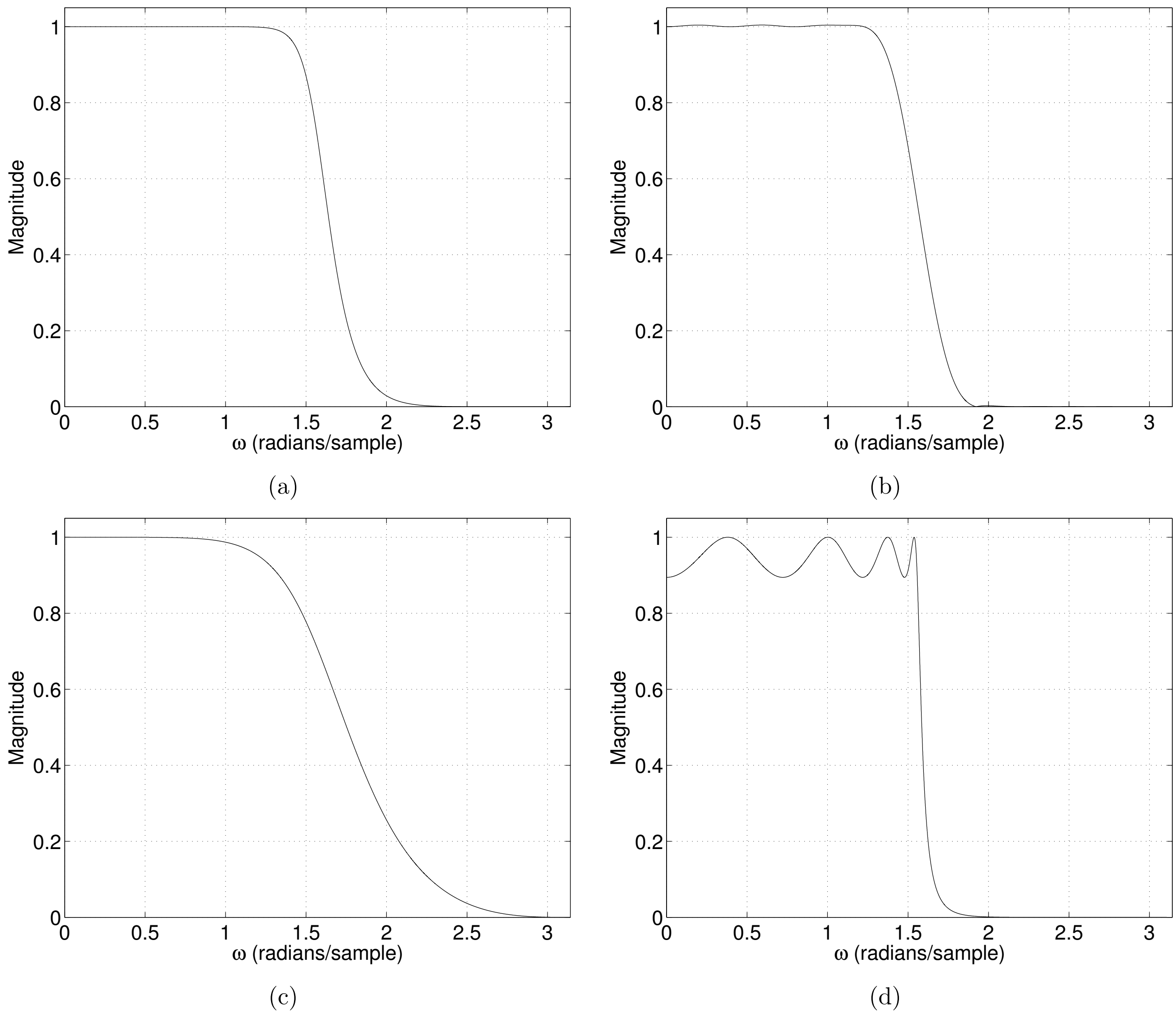


Figure 3.1: Magnitude response plots for problem 3.

Prob. 3 (cont.)

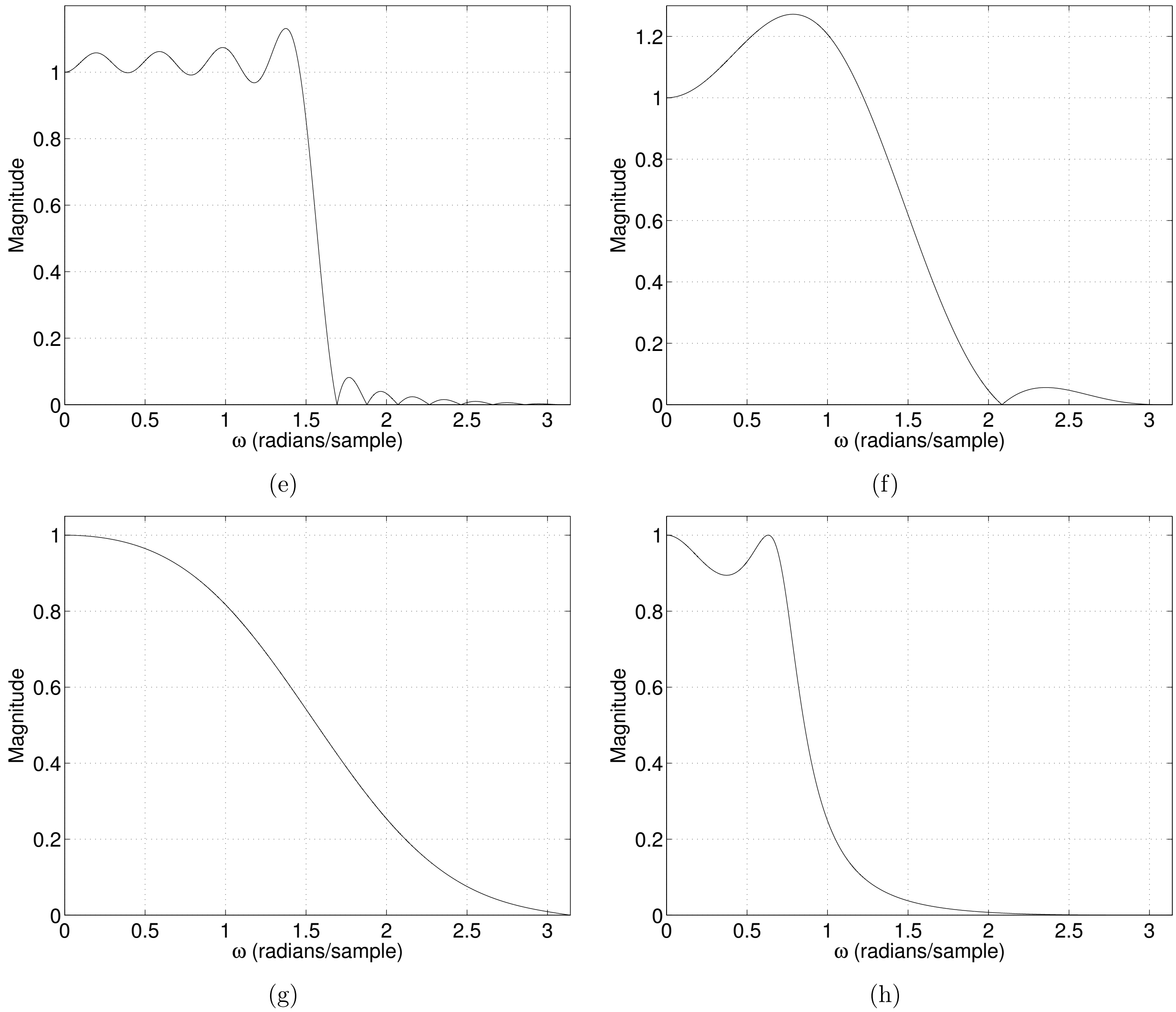


Figure 3.1: (cont.) Magnitude response plots for problem 3.

Prob. 4

If the Nyqiust rate for $x(t)$ is f_s, what is the Nyqiust rate (in terms of f_s) for each of the following signals, $y(t)$ that are derived from $x(t)$?

Note: The Fourier transform (FT) and inverse-Fourier transform (IFT) are given by

$$X(\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j\Omega t}dt \quad \leftrightarrow \quad x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\Omega)e^{j\Omega t}d\Omega$$

and FT properties useful for this problem are found below.

$r(t) = s(t)p(t) \leftrightarrow R(\Omega) = \frac{1}{2\pi}S(\Omega) * P(\Omega)$	$x(at) \leftrightarrow \frac{1}{\lvert a\rvert}X(\Omega/a)$
$x(t)\cos(\Omega_0 t) \leftrightarrow \frac{1}{2}[X(\Omega-\Omega_0)+X(\Omega+\Omega_0)]$	$\frac{dx(t)}{dt} \leftrightarrow j\Omega X(\Omega)$

(a) $y(t) = x^2(t)$

Nyquist rate for $y(t)$:

(b) $y(t) = x(t)\cos(2\pi f_0 t)$ (do not assume bandpass sampling)

Nyquist rate for $y(t)$:

(c) $y(t) = x(2t)$

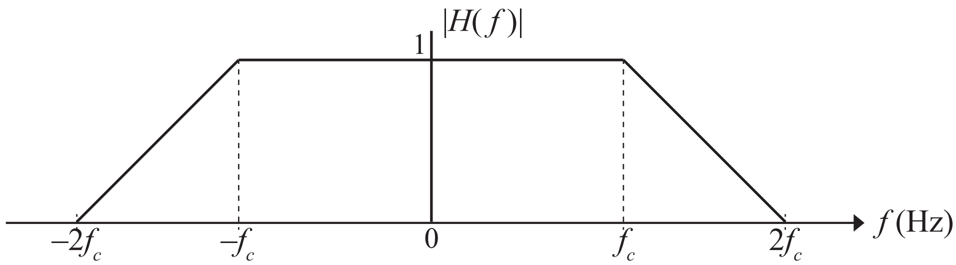
Nyquist rate for $y(t)$:

(d) $y(t) = \frac{dx(t)}{dt}$

Nyquist rate for $y(t)$:

Prob. 5

Consider the continuous-to-discrete (C/D) time converter on p. 7 Figure 1.3.5. Suppose the magnitude response of a practical, analog, anti-aliasing (lowpass) prefilter is shown below. Assume the passband edge frequency, $f_c = 400$ Hz and the magnitude response is zero for $|f| > 2f_c$.



(a) Determine the minimum value for f_s so that there is no aliasing in the discrete-time signal.

(b) Determine the minimum value for f_s so that the aliased spectral components within the Nyquist interval are suppressed by more than 50% relative to the signal components.

(c) Determine the minimum value for f_s so that the aliased spectral components within the Nyquist interval are suppressed by more than 1% relative to the signal components.

Prob. 6

For the given digital filter description, determine the equivalent descriptions. Assume the systems are at rest and causal. All impulse responses should be in closed-form (no recursions).

(a) $H(z) = 1 + 2z^{-1} + 3z^{-2}$

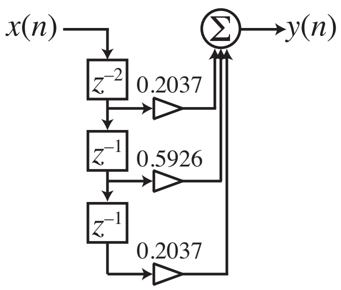
Determine the impulse response and difference equation and draw the canonical realization.

(b) $H(z) = \dfrac{-2/5}{1 - \dfrac{1}{2}z^{-1}} + \dfrac{7/5}{1 + \dfrac{3}{4}z^{-1}}$

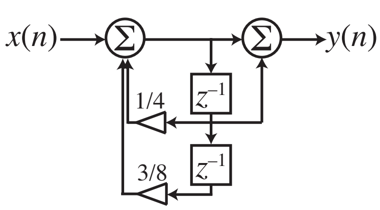
Determine the impulse response and difference equation and draw the canonical realization.

Prob. 6 (cont.)

(c) Determine the difference equation, transfer function, and impulse response for the following filter realization.



(d) Determine the difference equation, transfer function, and impulse response for the following filter realization.



Prob. 7

The N-point DFT, $X_N(k)$ of a length-L signal, $x(n)$ is given by

$$X_N(k) = \sum_{n=0}^{L-1} x(n) e^{-j2\pi nk/N}, \ 0 \leq k \leq N-1.$$

(a) Show that for N even, $X_{\frac{N}{2}}(k) = X_N(2k)$ for $0 \leq k \leq N/2 - 1$

(b) A length-8 signal, $\mathbf{x}$ has the following 8-point DFT, $\mathbf{X}_8$

$$\begin{aligned} \mathbf{x} &= [4, 2, 4, -6, 4, 2, 4, -6] \\ &\updownarrow \\ \mathbf{X}_8 &= [8, 0, -16j, 0, 24, 0, 16j, 0] \end{aligned}$$

Without performing any additional computations, determine the 4-point and 2-point DFTs of $\mathbf{x}$.

(c) Modulo-2 wrap $\mathbf{x}$ in (b) and directly compute the 2-point DFT using the formula. Show your result is that same as in (b).

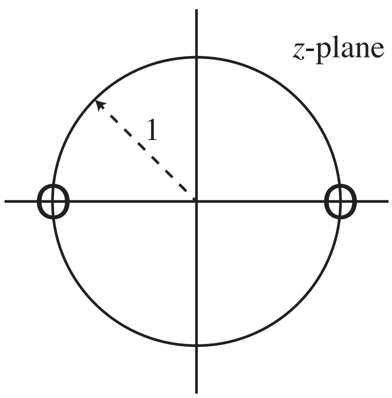
Prob. 8

For the following pole/zero patterns:

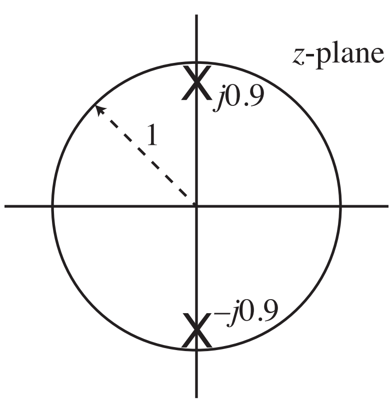
1. compute the transfer function, $H(z) = B(z)/A(z)$ (multiply out factors if necessary)
2. draw the magnitude response $|H(\omega)|$ vs. ω, $0 \leq \omega \leq \pi$ and indicate on the plot, the maximum and minimum values of $|H(\omega)|$

Pole values are indicated on the pattern; zeros are *on* the unit-circle.

(a)

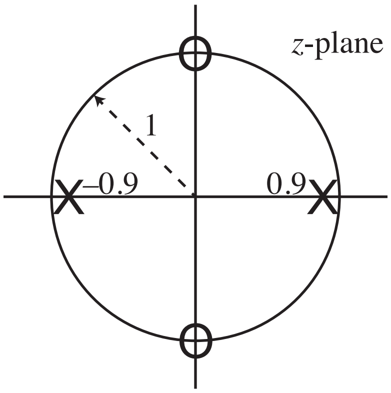


(b)

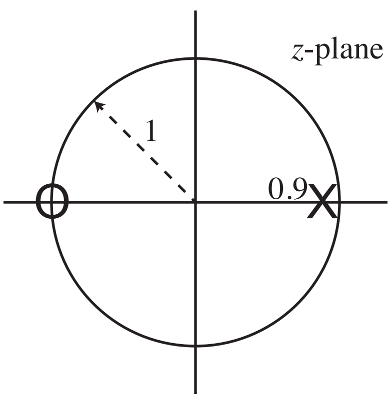


Prob. 8 (cont.)

(c)



(d)



Prob. 9

The transfer function of a second-order, analog Butterworth lowpass filter with cutoff-frequency, $\Omega_c = 2$ rad/s is given by

$$H_a(s) \quad = \quad \frac{1}{1 + \frac{\sqrt{2}}{2} s + \frac{1}{4} s^2}. \tag{9.1}$$

(a) Apply the bilinear transform, $s = \frac{1 - z^{-1}}{1 + z^{-1}}$ to $H_a(s)$ and determine $H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}}$. Please scale your coefficients so that $a_0 = 1$.

Prob. 9 (cont.)

(b) What is the cutoff frequency, ω_c of the digital filter, $H(z)$ in (a)?

(c) Sketch the $|H(\omega)|^2$ vs. ω. Please compute and note magnitude-squared response values for $\omega = 0$ and π rads/sample.