Homework #5: Chapter 6 (due Fri. Oct. 19, 2012)

Preliminary

- All problems are worth +10 points unless otherwise noted.
- Download the CompanionFiles5.zip file

  http://www.ece.nmsu.edu/~pdeleon/Teaching/EE395/Homework/CompanionFiles5.zip
- Code the following tool from *DSP Software Toolkit* Chapter 6: **freqresp.m**, **fr_plot.m**, and **polezero.m**. A code skeleton of **fr_plot.m** can be found at the end of this assignment.
- Please attach at the *end* of your assignment printouts of **main1.m** (code to solve software problem 1), **main2.m** (code to solve software problem 2), etc... as well as any new tools developed in this assignment.
- A simple way to determine the coefficients of the polynomial $(1 + \alpha z^{-1})(1 + \beta z^{-1} + \gamma z^{-2})$ is via

```
poly1 = [1;alpha];
poly2 = [1;beta;gamma];
poly = convltn(poly1,poly2)
```

- In order to reproduce some magnitude response plots, especially those with narrow and deep notches, you may need to increase the computational resolution of freqresp.m from 1024 samples to a value much higher.
- In some cases, numerical precision errors may cause filter calculations to have very small, i.e. 10^{-10} or less, imaginary components. In these cases, you may discard the small imaginary values.

Software Problems

Use your software tools to solve the following problems.

1. Consider the transfer function

$$H(z) = \frac{5 + 2z^{-1}}{1 - 0.8z^{-1}}. \tag{1}$$

   Plot the pole/zero pattern and magnitude response [x-axis is ω (rads/sample) and y-axis is $|H(\omega)|$ (normal units)] of the system. Compare your results to Fig. 6.2.2 in the text.

2. Recreate the pole/zero pattern and magnitude response plot of Example 6.2.1.
3. Recreate the pole/zero patterns and magnitude response plots of Example 6.2.3.
4. Consider Example 6.4.2 for $r = 1$. Run the following code which computes filter coefficients from (6.4.6), plots the magnitude response, computes and plots the impulse response, and plots the pole/zero pattern. Please submit resulting plots.

```
% notch filter params
R = 0.98;
r = 1;
omega0 = 0.4*pi;

% build out filter coefs, verify coefs in Example 6.4.2
b = convltn([1;-r*exp(j*omega0)],[1;-r*exp(-j*omega0)])  % put notches at +/- omega0
```

```
a = convltn([1;-R*exp(j*omega0)],[1;-R*exp(-j*omega0)])

% code to plot frequency response
fr_plot(b,a,2*pi,1);

% code to compute and plot impulse response
delta = [1;zeros(100,1)]; % create impulse
h = fltr(b,a,delta); % filter it to get response to impulse
plotdsig(h,1); % students can supply proper labels and embellishments

% code for pole/zero pattern
polezero(b,a);
```

5. Recreate the pole/zero pattern and magnitude response plot of Example 6.4.3. For the magnitude response, please plot in units of Hz assuming $f_s = 600$ Hz and note the notches at the harmonic frequencies.

Textbook Problems

6.1 c, e [you are free to use your tools for the magnitude response (also provide a phase response) and pole/zero pattern; direct block diagram realization only]

6.3 a

6.5 d, e [you are free to use your tools for the magnitude response (also provide a phase response) and pole/zero pattern; direct block diagram realization only]

6.12

Code skeleton for fr_plot.m

```
if ((nargin ~= 2) & (nargin ~= 4))
   error('Error (fr_plot): must have 2 or 4 input arguments.');
end;
if (nargin == 2) % assume fs, option defaults
   fs = 2*pi;
   option = 0;
end;

N = 1024;
[H_mag,H_phase] = freqresp(b,a,N);

if (fs == 2*pi) % Create frequency axis depending on fs
   f = [0:2*pi/N:2*pi-2*pi/N]'; % rads/sample
elseif (fs == 1)
   f = [0:1/N:1-1/N]'; % normalized f/fs
else
   f = [0:fs/N:fs-fs/N]'; % Hz
end;

figure(1); % Magnitude Response
if (any(imag(b))|any(imag(a))) % complex case
   if (option == 0)
      plot(f,20*log10(H_mag),'k');
   elseif (option == 1)
      plot(f,H_mag,'k');
   else % (option == 2)
      plot(f,H_mag.*H_mag,'k');
  end;
else  % b,a are real
   if (option == 0)
      plot(f(1:N/2),20*log10(H_mag(1:N/2)),'k');
   elseif (option == 1)
      plot(f(1:N/2),H_mag(1:N/2),'k');
   else % (option == 2)
      plot(f(1:N/2),H_mag(1:N/2).*H_mag(1:N/2),'k');
   end;
end;
% --> Code for appropriate x- and y-labels depending on fs and option <--
grid;

figure(2); % Phase Response
if (any(imag(b))|any(imag(a))) % complex case
   plot(f,H_phase,'k');
   axis([0 fs -pi pi]);
else  % b,a are real
   plot(f(1:N/2),H_phase(1:N/2),'k');
   axis([0 fs/2 -pi pi]);
end;
% --> Code for appropriate x- and y-labels depending on fs <--
grid;
```