Music 320
Autumn 2010–2011
Homework #9
Filters & Turkey Analysis
135 points
Due in two weeks (12/03/2010)

Theory Problems

1. (15 pts) [Symmetric Filter Chain]

   Given a filter of the form $y(n) = x(n) + 2x(n-1) + 3x(n-2) + 2x(n-3) + x(n-4)$

   (a) (5 pts) Find an expression for the group delay

   Solution:

   Let $H(e^{j\omega})$ be the transfer function of the given filter. Note that the filter is real and symmetric around $n = 2$. Using the fact that real and even systems in the time domain are also real and even in the frequency domain is easy to see that the filter $y_{even}(n) = x(n+2) + 2x(n+1) + 3x(n) + 2x(n-1) + x(n-2)$ has a real transfer function.

   Using the shift theorem we can write

   $$y(n) = y_{even}(n-2) \leftrightarrow H(e^{j\omega T}) = e^{-2j\omega T} H_{even}(e^{j\omega T})$$

   Then:

   $$\angle H(e^{j\omega T} = \angle e^{-2j\omega T} + \angle H_{even}(e^{j\omega T}) = -2\omega T$$

   Then, the group delay is

   $$\mathcal{D}_g = -\frac{d\angle H(e^{j\omega T}}{d\omega} = 2T$$

   (b) (5 pts) Find an expression for the phase delay

   Solution:

   $$\mathcal{D}_p = -\frac{\angle H(e^{j\omega T}}{\omega} = 2T$$

   (c) (5 pts) Find and expression for the phase and group delay for a chain of N of these filters

   Solution: The phase of a chain of N filters in cascade of just the sum of the phases of each filter. Then

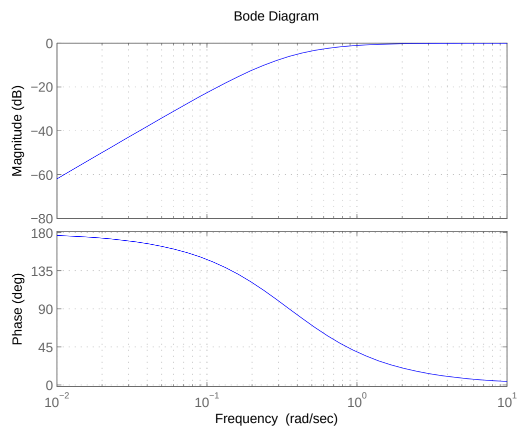
   $$\mathcal{D}_{pN} = \mathcal{D}_{gN} = 2NT$$

2. (25 pts) [Butterworth filter]

A continuous time second-order high-pass Butterworth filter with normalized cutoff frequency $\omega_c = 0.5$ has the following frequency response in the S-domain:

$$H(s) = \frac{s^2}{s^2 + \frac{1}{\sqrt{2}}s + \frac{1}{4}}$$

(a) (10 pts) Sketch the bode plots of the filter (magnitude and phase). Use a log scale for the frequency axis, dB for the magnitude axis and degree for the phase axis.



Solution:

(b) (10 pts) One common method for discretizing a continuous-time filter is by using the Bilinear Transform $s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}$

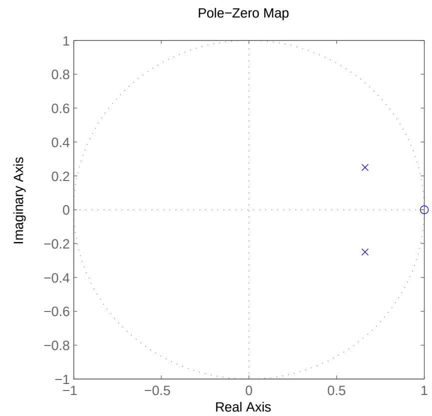
Find the discrete-time frequency response $H(z)$ of the filter $H(s)$ and its corresponding difference equation, assuming $T = 1$

Solution:

$$H(z) = \frac{16 - 32z^{-1} + 16z^{-2}}{(17 + 4\sqrt{2}) - 30z^{-1} + (17 - 4\sqrt{2})z^{-2}}$$

$$y(n) = \frac{16}{17 + 4\sqrt{2}}x(n) - \frac{32}{17 + 4\sqrt{2}}x(n-1) + \frac{16}{17 + 4\sqrt{2}}x(n-2) + \frac{30}{17 + 4\sqrt{2}}y(n-1) - \frac{17 - 4\sqrt{2}}{17 + 4\sqrt{2}}y(n-2)$$

(c) (5 pts) Plot the poles and zeros of the filter in the Z-plane



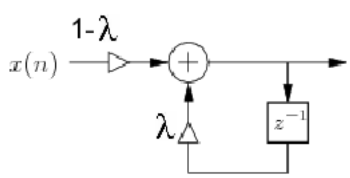
Solution:

3. (45 points) [Thanksgiving Turkey] You wake up on Thanksgiving, put your turkey into a 350 degree oven ($T_{oven} = 350$), go back to sleep, wake up again, and...oh no! You forgot to write down what time you put the Turkey in the oven! You must figure out when (t_{cooked}) the turkey will reach the desired temperature $T_{cooked} = 160$. To do so, you measure the temperature $T_1 = 100$ at $t_1 = 0$ and $T_2 = 110$ at $t_2 = 30$ where T is in degrees farenheit and t is in minutes.

   Using these measurements can now solve for t_{cooked} by approximating the system as a leaky integrator (one-pole) filter $y[n] = (1 - \lambda)x[n] + \lambda y[n - 1]$. The input $x[n]$ is the oven temperature over time (assumed constant), while the output $y[n]$ is the turkey temperature over time.

   Using this information, solve for the following:

   (a) (5 points) Draw the signal flow diagram of the turkey/oven system.

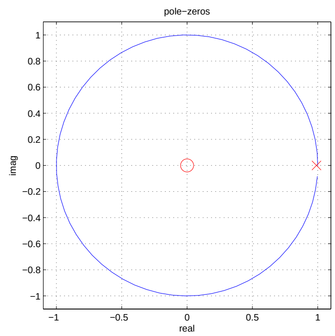
   Solution:

   (b) (5 points) Write the transfer function of the turkey/oven system.

   Solution: $H(z) = \frac{1-\lambda}{1-\lambda z^{-1}}$

   (c) (5 points) Plot the poles and zeros of the turkey/oven system.

   Solution: Zeros: 0 Poles: λ The plot should look something like:



(d) (15 points) Write the turkey temperature trajectory (step response with the input at 350 instead of 1).

Solution: $T(t) = T_{oven} - \lambda^t(T_{oven} - T(t_1)) = 350 - 250\lambda^t$

(e) (5 points) Solve for the forgetting factor λ

Solution: Using the temperature trajectory and the given data points: $110 = (100 - 350)\lambda^{30} + 350$ So, $\lambda = (\frac{110-350}{100-350})^{1/30} = 0.9986$

(f) (5 points) Solve for the turkey time constant τ.

Solution: The time constant of one pole (R) can be approximated as: $\tau \approx \frac{T}{1-R}$ so, $\tau \approx 735$ minutes.

(g) (5 points) Solve for the time t_{cooked} when the turkey will reach T_{cooked}.

Solution: $T_{cooked} = 350 - 250(.9986^{t_{cooked}}) = 160$
So, $t_{cooked} = \log((160 - 350)/-250)/\log\lambda \approx 195$ or 201 minutes depending on rounding

Lab Assignments

Follow the same file naming convention of the previous lab.

1. (30 pts) The purpose of this problem is to introduce you to a couple of useful Matlab commands for filter design.

   (a) (10 pts) (5 points) Use `butter` to design a digital second order low-pass filter that cuts off at fs/3. Design a filter with the same cutoff frequency using `cheby1`. Turn in the coefficients for both filters.

   Solution:

```
clear all
close all
```

```
N = 2048;
ripple = 3;

% LPF
[bb\_lpf, ba\_lpf] = butter(2, 2/3, 'low')
[cb\_lpf, ca\_lpf] = cheby1(2, ripple, 2/3, 'low')
```

(b) (10 pts) Use `freqz` to display the amplitude and phase response of both filters designed in part 1a (plot at least 2048 points). Plot both filters in the same figure using `hold` and comment briefly on the spectral differences between the filters.

Solution:

```
[H_blpf, w] = freqz(bb_lpf, ba_lpf, N);
[H_clpf, w] = freqz(cb_lpf, ca_lpf, N);
figure(1);
plot(w, 20*log10(abs([H_blpf H_clpf])));
legend('Butterworth','Chebyshev');

% Both filters have the same cutoff frequency (-3 dB at 2*pi/3), but the
% Butterworth filter has a flatter response in the passband, while the
% Chebyshev filter has a better rejection in the stop-band
```

(c) (5 pts) Repeat parts 1a and 1b for a high-pass filter with cutoff frequency of fs/6.

Solution:

```
% HPF
[bb_hpf, ba_hpf] = butter(2, 1/3, 'high');
[cb_hpf, ca_hpf] = cheby1(2, ripple, 1/3, 'high');

[H_bhpf, w] = freqz(bb_hpf, ba_hpf, N);
[H_chpf, w] = freqz(cb_hpf, ca_hpf, N);
figure(2);
plot(w, 20*log10(abs([H_bhpf H_chpf])));
legend('Butterworth','Chebyshev');
```

(d) (5 pts) Repeat parts 1a and 1b for a band-pass filter with a low cutoff frequency of fs/6 and high cutoff frequency of fs/3.

Solution:

```
% BPF
[bb_hpf, ba_hpf] = butter(2, [1/3 2/3]);
[cb_hpf, ca_hpf] = cheby1(2, ripple, [1/3 2/3]);

[H_bhpf, w] = freqz(bb_hpf, ba_hpf, N);
[H_chpf, w] = freqz(cb_hpf, ca_hpf, N);
```

```
figure(3);
plot(w, 20*log10(abs([H_bhpf H_chpf])));
legend('Butterworth','Chebyshev');
```

Turn in your Matlab code

2. (20 pts) [Allpass Filter Group Delay]

   Form the allpass filters `b = [rho 1]; a = [1 rho];` for `rho = [-0.5, 0, 0.5];`. For each filter, compute the group delay by using `diff()` on `angle()` of the transfer function formed by `freqz(b, a, 0:1/(nbins-1):pi)` (you probably don't have to do `unwrap()`, but verify if you need to). Compare that group delay to the output of `grpdelay()`. Plot the poles and zeros of the system using `zplane()` (note the different way `zplane()` treats row and column vectors).

   Turn in your Matlab code.

```
clear all; close all;

rho = -0.5;
%rho = 0.0;
%rho = 0.5;

b = [rho 1]; a = [1 rho];
nbins = 512;
[H,w] = freqz(b, a, 0:1/(nbins-1):pi);

% verify the phase, to check if unwrapping is necessary
phase = angle(H);
plot(w, phase)

% using diff
gd1 = -diff(phase)*nbins;

% using grpdelay
gd2 = grpdelay(b,a,w);

plot(w(1:end-1),gd1,'b');
hold on;
plot(w,gd2,'r');
xlabel('Frequency [rad]');
ylabel('Delay in cycles');
legend('Using diff','Using grpdelay');
```

```
title('Group delay comparisson')
hold off;

% plot zero and pole
figure;
zplane(b,a);
title('Pole and zero of the filter')
```