Music 320
Autumn 2010–2011
Homework #8
Z-transform, FIR & IIR Filters
130 points
Due in one week (11/19/2010)

Theory Problems

1. (20 points) Consider the filter

$$y(n) = x(n) - x(n-1)$$

which is identical to the simplest low-pass filter except that adjacent input samples are subtracted rather than added. Derive the amplitude response and the phase response. How has the response changed? Would you call this a low-pass filter, high-pass filter, or something else? In the time domain, we may call it a *first-order difference*.
Solution: Let's choose the input signal $x(n) = e^{j\omega nT}$, then

$$\begin{aligned} y(n) &= x(n) - x(n-1) \\ &= e^{j\omega nT} - e^{j\omega(n-1)T} \\ &= e^{j\omega nT}(1 - e^{-j\omega T}) \\ &= (1 - e^{-j\omega T})x(n) \\ &\triangleq H(e^{j\omega T})x(n) \end{aligned}$$

Expressing $H(e^{j\omega T})$ in polar form, we have

$$H(e^{j\omega T}) \triangleq G(\omega)e^{j\Theta(\omega)} \triangleq |H(e^{j\omega T})|\angle H(e^{j\omega T})$$

where $G(\omega)$ is the amplitude response, and $\Theta(\omega)$ is the phase response.

$$\begin{aligned} H(e^{j\omega T}) &= 1 - e^{-j\omega T} \\ &= (e^{j\omega T/2} - e^{-j\omega T/2})e^{-j\omega T/2} \\ &= 2j\sin(\omega T/2)e^{-j\omega T/2} \\ &= 2\sin(\omega T/2)e^{j(\pi/2-\omega T/2)} \end{aligned}$$

Therefore,

$$\begin{aligned} G(\omega) &= |2\sin(\omega T/2)e^{j(\pi/2-\omega T/2)}| \\ &= |2\sin(\omega T/2)| \\ &= 2\sin(\omega T/2), \quad |\omega T| \le \pi \end{aligned}$$

and

$$\Theta(\omega) \;=\; \pi/2 - \omega T/2, \quad |\omega T| \leq \pi$$

Clearly, we have a gain of 0 at DC ($\omega T = 0$), and a gain of 2 at half the sampling rate ($\omega T = \pi$), which is the opposite of the simplest low-pass filter. Hence this filter can be regarded as a high-pass filter. The phase response is the same but with an offset of $\pi/2$.

2. (10 points) For the two input sequences

$$x_1(n) = [1, 1, 1, 1, 1, 1, 1, 1]$$

and

$$x_2(n) = [1, -1, 1, -1, 1, -1, 1, -1]$$

find the output $y(n)$ using the first-order difference filter given in the previous problem. How would you relate your answers to the results you got in the previous problem?
Solution: Simply plugging the input sequences into the system yields the outputs

$$y_1(n) = [1, 0, 0, 0, 0, 0, 0, 0]$$

and

$$y_2(n) = [1, -2, 2, -2, 2, -2, 2, -2]$$

We can relate these results to the amplitude response obtained in the previous problem as follows. $x_1(n)$ can be viewed as a DC signal since there is no change in the signal. Thus this DC signal is completely filtered out by a high-pass filter except for the first transient response. On the other hand, $x_2(n)$, whose frequency is half the sampling rate, is boosted up by a factor of 2, which is the maximum gain of the filter. Again, there is a transient at the beginning.

3. For the following filter:

$$H(z) = \frac{6z^2 - 6z}{1 - 5z + 6z^2}$$

(a) (5 pts) Draw the direct-form-II realization.
Solution: Get into canonical form as

$$H(z) = \frac{1 - z^{-1}}{1 - \frac{5}{6}z^{-1} + \frac{1}{6}z^{-2}}$$

and draw by inspection.

(b) (5 pts) Draw the transposed direct-form-II realization.

(c) (5 pts) Find the partial fraction expansion.
Solution:

$$H(z) = \frac{-3}{1 - \frac{1}{2}z^{-1}} + \frac{4}{1 - \frac{1}{3}z^{-1}}$$

(d) (5 pts) Draw a realization as parallel one-pole sections.

4. (30 points) [Partial Fraction Expansion] Express the following transfer functions as a sum of one pole filters using *partial fraction expansion* (PFE):

(a)

$$H_1(z) = \frac{-2}{1 - \frac{4}{3}z^{-1} + \frac{1}{3}z^{-2}}$$

Solution:

$$H_1(z) = \frac{-2}{1 - \frac{4}{3}z^{-1} + \frac{1}{3}z^{-2}} = \frac{-2}{(1 - \frac{1}{3}z^{-1})(1 - z^{-1})} = \frac{1}{1 - \frac{1}{3}z^{-1}} - \frac{3}{1 - z^{-1}}$$

(b)

$$H_2(z) = \frac{4 - \frac{7}{2}z^{-1}}{1 - \frac{3}{2}z^{-1} + \frac{1}{2}z^{-2}}$$

Solution:

$$H_2(z) = \frac{4 - \frac{7}{2}z^{-1}}{1 - \frac{3}{2}z^{-1} + \frac{1}{2}z^{-2}} = \frac{4 - \frac{7}{2}z^{-1}}{(1 - \frac{1}{2}z^{-1})(1 - z^{-1})} = \frac{3}{1 - \frac{1}{2}z^{-1}} + \frac{1}{1 - z^{-1}}$$

(c)

$$H_3(z) = \frac{1 - j2 + (\frac{7}{4} - j\frac{9}{4})z^{-1}}{1 + (\frac{3}{4} - j)z^{-1} - j\frac{3}{4}z^{-2}}$$

Solution:

$$H_3(z) = \frac{1 - j2 + (\frac{7}{4} - j\frac{9}{4})z^{-1}}{1 + (\frac{3}{4} - j)z^{-1} - j\frac{3}{4}z^{-2}} = \frac{1 - j2 + (\frac{7}{4} - j\frac{9}{4})z^{-1}}{(1 + \frac{3}{4}z^{-1})(1 - jz^{-1})} = \frac{j}{1 + \frac{3}{4}z^{-1}} + \frac{1 - j3}{1 - jz^{-1}}$$

5. (15 points) [Inverse z Transform] Give the impulse response of each of the filters in the previous problem by inverting the z transform. (*Hint*: The z transform is linear, so you do this by inverting the one pole filters you found with PFE)

Solution:

Using linearity of the z transform and that $\frac{r_i}{1 - p_i z^{-1}} \leftrightarrow r_i(p_i)^n u(n)$:

(a)

$$H_1(z) = \frac{1}{1 - \frac{1}{3}z^{-1}} - \frac{3}{1 - z^{-1}} \leftrightarrow h_1(n) = 1(\frac{1}{3})^n u(n) - 3(1)^n u(n)$$

(b)

$$H_2(z) = \frac{3}{1-\frac{1}{2}z^{-1}} + \frac{1}{1-z^{-1}} \leftrightarrow h_2(n) = 3(\frac{1}{2})^n u(n) + 1(1)^n u(n)$$

(c)

$$H_3(z) = \frac{j}{1+\frac{3}{4}z^{-1}} + \frac{1-j3}{1-jz^{-1}} \leftrightarrow h_3(n) = j(-\frac{3}{4})^n u(n) + (1-j3)(j)^n u(n)$$

Lab Assignments

Follow the same file naming convention of the previous lab.

1. (20 points) Simple FIR Digital Filter Design
   (a) (5 points) Use the Matlab function `fir1` to design a 10th order FIR lowpass filter that cuts off at one-fourth the sampling rate. Plot the impulse response.
   (b) (5 points) Use `freqz` to display the amplitude and phase response of this filter.
   (c) (10 points) Generate 4096 samples of a white noise signal using `randn` and apply the FIR filter to it. With your sound volume TURNED WAY DOWN (at first), listen to the input and output signals. Plot the magnitude of a length 8192 FFT of the input and output signals.

   Turn in your Matlab code.

   Solution:

```
clear all
close all

b = fir1(10,.5);
a=1;
d=[1 zeros(1,99)];
h = filter(b,a,d);

plot([0:99],h)
grid
title('Impluse Respone 10th order FIR lowpass (cutoff=fs/4)')
xlabel('samples (n)')
ylabel('amplitude')
```

```
figure
freqz(b,a);
title('10th order FIR lowpass (cutoff=fs/4)')

w_noise=randn(1,1000);
w_noiseF=filter(b,a,w_noise);
sound(w_noise,44100)
pause
sound(w_noiseF,44100)

plotspec (w_noise)
title('White Noise Spectrum (INPUT)')
plotspec (w_noiseF)
title('Filtered White Noise Spectrum (OUTPUT)')
```

2. (20 points) [Convolution measurement] A second-order IIR filter is given like below

$$y(n) = 0.3024\,x(n) - 0.3024\,x(n-2) + 1.749\,y(n-1) - 0.9244\,y(n-2)$$

Using the `jobs.wav` sound file (downloaded with the pdf), write a script which applies the given filter to the `jobs.wav` in the following ways:

(a) directly using the difference equation above

(b) using `conv`

(c) using `filter`

(d) using `fftfilt`

Measure the run time of each operation using `tic` and `toc` function. The usage of `tic` and `toc` function is as follows.

`tic`

...perform your calculation...

`toc` // returns an elapsed time.

Compare the performances of these operations when $N = 128$, 1024, and 8092. Turn in your code, and the run time results for each N.

Note that you need a finite impulse response `h` to calculate them in (a), (b) and (d), which is contrary to the given filter. But, you can get an approximated finite impulse response like this.

```
h = filter(b,a,[1; zeros(N-1,1)])
```

Use the impulse response `h` or filter coefficients `a` and `b` as arguments of Matlab functions above. Also, make sure that the filtered outputs are the same by listening to them.

Solution:

```
clear all
close all
% measuring the calculation that runs quickly requires some subtlety
% because there is no guarantee that the system clock unit is precise
% enough to measure the elapsed time. You can see that the elapsed time
% changes every time you measure it. So, the best way to get around
% this problem is to repeat the calculation many times and average it.
%
[x, fs] = wavread('jobs.wav');
M=length(x);
N=128; % N = 1024; N = 8092;

% This filter is meant to be a bandpass filter.
%[b,a] = butter(1,[500 600]*2/fs);
%b = 8*b;

b = [0.3024 0 -0.3024];
a = [1.0000 -1.7490 0.9244];
h = filter(b,a,[1;zeros(N-1,1)]);

%Impulse response of the filter

% 1. Convolution by hand
% ----------------------
xc=[zeros(N-1,1);x;zeros(N-1,1)];
hc=h(N:-1:1);
y0 = zeros(M+N-1,1);
tic
for ii=1:M+N-1
y0(ii)=hc'*xc(ii:ii+N-1);  % inner product = sum of point-wise multiplication
end
myConvTime = toc;

% 2 ."conv" function
% ---------------
tic
```

```
y1 = conv (h, x);
convTime = toc;

% 3 ."filter" function
% --------------------
tic
y2 = filter (b, a, x);
filterTime = toc;

% 4 ."fftfilt" function
% --------------------
tic
y3 = fftfilt (h, x);
fftfiltTime = toc;

sprintf('N: %d', N)
sprintf('My convolution: %f [s], %d samples', myConvTime, length(y0))
sprintf('Conv: %f [s], %d samples', convTime, length(y1))
sprintf('Filter: %f [s], %d samples', filterTime, length(y2))
sprintf('FftFilt: %f [s], %d samples', fftfiltTime, length(y2))
wavwrite(y0, fs, 'jobsRA_myConv.wav');
wavwrite(y1, fs, 'jobsRA_conv.wav');
wavwrite(y2, fs, 'jobsRA_filter.wav');
wavwrite(y3,fs,'jobsRA_fftfilt.wav');
```