

## Homework 5

Due: November, 18, 2011

- Question 1.** Textbook 7.26
- Question 2.** Textbook 7.31
- Question 3.** Textbook 7.32
- Question 4.** Textbook 7.37
- Question 5.** Textbook 7.38
- Question 6.** Matlab Question 1
- Question 7.** Matlab Question 2

**7.26.** *Impulse invariance* and the *bilinear transformation* are two methods for designing discrete-time filters. Both methods transform a continuous-time system function  $H_c(s)$  into a discrete-time system function  $H(z)$ . Answer the following questions by indicating which method(s) will yield the desired result:

- (a) A minimum-phase continuous-time system has all its poles and zeros in the left-half  $s$ -plane. If a minimum-phase continuous-time system is transformed into a discrete-time system, which method(s) will result in a minimum-phase discrete-time system?
- (b) If the continuous-time system is an all-pass system, its poles will be at locations  $s_k$  in the left-half  $s$ -plane, and its zeros will be at corresponding locations  $-s_k$  in the right-half  $s$ -plane. Which design method(s) will result in an all-pass discrete-time system?
- (c) Which design method(s) will guarantee that

$$H(e^{j\omega})|_{\omega=0} = H_c(j\Omega)|_{\Omega=0}?$$

- (d) If the continuous-time system is a bandstop filter, which method(s) will result in a discrete-time bandstop filter?
- (e) Suppose that  $H_1(z)$ ,  $H_2(z)$ , and  $H(z)$  are transformed versions of  $H_{c1}(s)$ ,  $H_{c2}(s)$ , and  $H_c(s)$ , respectively. Which design method(s) will guarantee that  $H(z) = H_1(z)H_2(z)$  whenever  $H_c(s) = H_{c1}(s)H_{c2}(s)$ ?
- (f) Suppose that  $H_1(z)$ ,  $H_2(z)$ , and  $H(z)$  are transformed versions of  $H_{c1}(s)$ ,  $H_{c2}(s)$ , and  $H_c(s)$ , respectively. Which design method(s) will guarantee that  $H(z) = H_1(z) + H_2(z)$  whenever  $H_c(s) = H_{c1}(s) + H_{c2}(s)$ ?
- (g) Assume that two continuous-time system functions satisfy the condition

$$\frac{H_{c1}(j\Omega)}{H_{c2}(j\Omega)} = \begin{cases} e^{-j\pi/2}, & \Omega > 0, \\ e^{j\pi/2}, & \Omega < 0. \end{cases}$$

If  $H_1(z)$  and  $H_2(z)$  are transformed versions of  $H_{c1}(s)$  and  $H_{c2}(s)$ , respectively, which design method(s) will result in discrete-time systems such that

$$\frac{H_1(e^{j\omega})}{H_2(e^{j\omega})} = \begin{cases} e^{-j\pi/2}, & 0 < \omega < \pi, \\ e^{j\pi/2}, & -\pi < \omega < 0? \end{cases}$$

(Such systems are called “90-degree phase splitters.”)

- 7.31.** Suppose that we have used the Parks–McClellan algorithm to design a causal FIR linear-phase lowpass filter. The system function of this system is denoted  $H(z)$ . The *length* of the impulse response is 25 samples, i.e.,  $h[n] = 0$  for  $n < 0$  and for  $n > 24$ , and  $h[0] \neq 0$ . The desired response and weighting function used were

$$H_d(e^{j\omega}) = \begin{cases} 1 & |\omega| \leq 0.3\pi \\ 0 & 0.4\pi \leq |\omega| \leq \pi \end{cases} \quad W(e^{j\omega}) = \begin{cases} 1 & |\omega| \leq 0.3\pi \\ 2 & 0.4\pi \leq |\omega| \leq \pi. \end{cases}$$

In each case below, determine whether the statement is true or false or that insufficient information is given. Justify your conclusions.

- (a)  $h[n + 12] = h[12 - n]$  or  $h[n + 12] = -h[12 - n]$  for  $-\infty < n < \infty$ .
- (b) The system has a stable and causal inverse.
- (c) We know that  $H(-1) = 0$ .
- (d) The maximum weighted approximation error is the same in all approximation bands.
- (e) If  $z_0$  is a zero of  $H(z)$ , then  $1/z_0$  is a pole of  $H(z)$ .
- (f) The system can be implemented by a network (flow graph) that has no feedback paths.
- (g) The group delay is equal to 24 for  $0 < \omega < \pi$ .

- (h) If the coefficients of the system function are quantized to 10 bits each, the system is still optimum in the Chebyshev sense for the original desired response and weighting function.
- (i) If the coefficients of the system function are quantized to 10 bits each, the system is still guaranteed to be a linear-phase filter.
- (j) If the coefficients of the system function are quantized to 10 bits each, the system may become unstable.

7.32. You are required to design an FIR filter,  $h[n]$ , with the following magnitude specifications:

- Passband edge:  $\omega_p = \pi/100$ .
- Stopband edge:  $\omega_s = \pi/50$ .
- Maximum stopband gain:  $\delta_s \leq -60$  dB relative to passband.

It is suggested that you try using a Kaiser window. The Kaiser window design rules for shape parameter  $\beta$  and filter length  $M$  are provided in Section 7.5.3.

(a) What values of  $\beta$  and  $M$  are necessary to meet the required specifications?

You show the resulting filter to your boss, and he is unsatisfied. He asks you to reduce the computations required for the filter. You bring in a consultant who suggests that you design the filter as a cascade of two stages:  $h'[n] = p[n] * q[n]$ . To design  $p[n]$  he suggests first designing a filter,  $g[n]$ , with passband edge  $\omega'_p = 10\omega_p$ , stopband edge  $\omega'_s = 10\omega_s$  and stopband gain  $\delta'_s = \delta_s$ . The filter  $p[n]$  is then obtained by expanding  $g[n]$  by a factor of 10:

$$p[n] = \begin{cases} g[n/10], & \text{when } n/10 \text{ is an integer,} \\ 0, & \text{otherwise.} \end{cases}$$

- (b) What values of  $\beta'$  and  $M'$  are necessary to meet the required specifications for  $g[n]$ ?
- (c) Sketch  $P(e^{j\omega})$  from  $\omega = 0$  to  $\omega = \pi/4$ . You do not need to draw the exact shape of the frequency response; instead, you should show which regions of the frequency response are near 0 dB, and which regions are at or below  $-60$  dB. Label all band edges in your sketch.
- (d) What specifications should be used in designing  $q[n]$  to guarantee that  $h'[n] = p[n] * q[n]$  meets or exceeds the original requirements? Specify the passband edge,  $\omega''_p$ , stopband edge,  $\omega''_s$ , and stopband attenuation,  $\delta''_s$ , required for  $q[n]$ .
- (e) What values of  $\beta''$  and  $M''$  are necessary to meet the required specifications for  $q[n]$ ? How many nonzero samples will  $h'[n] = q[n] * p[n]$  have?
- (f) The filter  $h'[n]$  from parts (b)–(e) is implemented by first directly convolving the input with  $q[n]$  and then directly convolving the results with  $p[n]$ . The filter  $h[n]$  from part (a) is implemented by directly convolving the input with  $h[n]$ . Which of these two implementations requires fewer multiplications? Explain. Note: you should not count multiplications by 0 as an operation.

7.37. Consider the two-stage system shown in Figure P7.37 for interpolating a sequence  $x[n] = x_c(nT)$  to a sampling rate that is 15 times as high as the input sampling rate; i.e., we desire  $y[n] = x_c(nT/15)$ .



Figure P7.37

Assume that the input sequence  $x[n] = x_c(nT)$  was obtained by sampling a band-limited continuous-time signal whose Fourier transform satisfies the following condition:  $|X_c(j\Omega)| = 0$  for  $|\Omega| \geq 2\pi(3600)$ . Assume that the original sampling period was  $T = 1/8000$ .

- Make a sketch of the Fourier transform  $X_c(j\Omega)$  of a "typical" bandlimited input signal and the corresponding discrete-time Fourier transforms  $X(e^{j\omega})$  and  $X_e(e^{j\omega})$ .
- To implement the interpolation system, we must, of course, use nonideal filters. Use your plot of  $X_e(e^{j\omega})$  obtained in part (a) to determine the passband and stopband cutoff frequencies ( $\omega_{p1}$  and  $\omega_{s1}$ ) required to preserve the original band of frequencies essentially unmodified while significantly attenuating the images of the baseband spectrum. (That is, we desire that  $w[n] \approx x_c(nT/3)$ .) Assuming that this can be achieved with passband approximation error  $\delta_1 = 0.005$  (for filter passband gain of 1) and stopband approximation error  $\delta_2 = 0.01$ , plot the specifications for the design of the filter  $H_1(e^{j\omega})$  for  $-\pi \leq \omega \leq \pi$ .
- Assuming that  $w[n] = x_c(nT/3)$ , make a sketch of  $W_e(e^{j\omega})$  and use it to determine the passband and stopband cutoff frequencies  $\omega_{p2}$  and  $\omega_{s2}$  required for the second filter.
- Use the formula of Eq. (7.117) to determine the filter orders  $M_1$  and  $M_2$  for Parks–McClellan filters that have the passband and stopband cutoff frequencies determined in parts (b) and (c) with  $\delta_1 = 0.005$  and  $\delta_2 = 0.01$  for both filters.
- Determine how many multiplications are required to compute 15 samples of the output for this case.

**7.38.** The system of Figure 7.2 is used to perform filtering of continuous-time signals with a digital filter. The sampling rate of the C/D and D/C converters is  $f_s = 1/T = 10,000$  samples/sec.

A Kaiser window  $w_K[n]$  of length  $M + 1 = 23$  and  $\beta = 3.395$  is used to design a linear-phase lowpass filter with frequency response  $H_{lp}(e^{j\omega})$ . When used in the system of Figure 7.1 so that  $H(e^{j\omega}) = H_{lp}(e^{j\omega})$ , the overall effective frequency response (from input  $x_d(t)$  to output  $y_d(t)$ ) meets the following specifications:

$$\begin{aligned} 0.99 \leq |H_{\text{eff}}(j\Omega)| \leq 1.01, & \quad 0 \leq |\Omega| \leq 2\pi(2000) \\ |H_{\text{eff}}(j\Omega)| \leq 0.01 & \quad 2\pi(3000) \leq |\Omega| \leq 2\pi(5000). \end{aligned}$$

- The linear phase of the FIR filter introduces a time delay  $t_d$ . Find the time delay through the system (in milliseconds).
- Now a highpass filter is designed with the *same* Kaiser window by applying it to the ideal impulse response  $h_d[n]$  whose corresponding frequency response is

$$H_d(e^{j\omega}) = \begin{cases} 0 & |\omega| < 0.25\pi \\ 2e^{-j\omega n_d} & 0.25\pi < |\omega| \leq \pi. \end{cases}$$

That is, a linear-phase FIR highpass filter with impulse response  $h_{hp}[n] = w_K[n]h_d[n]$  and frequency response  $H_{hp}(e^{j\omega})$  was obtained by multiplying  $h_d[n]$  by the same Kaiser window  $w_K[n]$  that was used to design the first mentioned lowpass filter. The resulting FIR highpass discrete-time filter meets a set of specifications of the following form:

$$\begin{aligned} |H_{hp}(e^{j\omega})| \leq \delta_1 & \quad 0 \leq |\omega| \leq \omega_1 \\ G - \delta_2 \leq |H_{hp}(e^{j\omega})| \leq G + \delta_2 & \quad \omega_2 \leq |\omega| \leq \pi \end{aligned}$$

Use information from the lowpass filter specifications to determine the values of  $\omega_1$ ,  $\omega_2$ ,  $\delta_1$ ,  $\delta_2$ , and  $G$ .

## MATLAB Questions

### 1.

To see what kind of problems might be involved in a simple frequency-sampled filter design, let us design a linear phase FIR lowpass filter, with cutoff frequency  $\omega_c = 0.42\pi$ . The desired filter length is  $N = 32$ .

- Find the sampled frequency response  $H[k]$  (don't forget to get the phase right!). Then, find the inverse DFT time sequence  $h[n]$ ,  $n = 0, 1, \dots, 31$ .
- Based on  $h[n]$ , evaluate with **freqz** and sub-plot 1024 points of the desired frequency response (magnitude as well as phase) and the actual frequency response (magnitude and phase). Comment on the matches/mismatches.
- Repeat the design, but this time use one transition sample. Experiment with the value of the transition sample until you get a maximally flat stopband. Again, plot the frequency response of the desired and actual filters.
- Repeat the design (without the transition sample) using the **remez** algorithm and a transition bandwidth of  $0.05\pi$ . Compare the results of all filters.

### 2.

This problem explores the design of linear phase FIR filters. A linear phase FIR filter is to be applied to an audio signal sampled at 16 kHz. The filter is lowpass and has a passband from 0 to 3250 Hz and a stopband from 4000 to 8000 Hz. The passband ripple is to be less than 0.1 dB and the stopband attenuation should be greater than 60 dB. Several filter design techniques will be assessed.

#### Notes:

- The Matlab routine **fir1** can be used to generate window-designed FIR filters. You will use this routine with an *explicitly* specified window. The routine will be called as **fir1(N,Wn,Win)**, where **N** is the filter order (careful, the filter length is **N+1**), **Wn** is the filter cutoff (**Wn** is normalized such than 1 is the half sampling frequency), and **Win** the the **N+1** element window. For some of the designs, you will have to choose **N** yourself.
  - The rectangular window can be generated with the **rectwin** (or **ones**) function.
  - The Hamming window can be generated with the **hamming** function.
  - The Kaiser window can be generated with the **kaiser** function. The Kaiser window has a parameter **beta**. An *estimate* for **beta** and **N** can be found using the routine **kaiserord**.
- A least-squares FIR design can be done with the routine **firls**. You will have to iterate on the filter length to get a filter which meets the specifications. You will also use a weighting function to change the relative deviations for the passband and stopband.

- For the equiripple design routine `remez`, the routine `remezord` can be used to estimate the order needed to meet the specifications.
- For each design, you must plot the filter response on a dB scale, with the scale chosen such that one can verify that whether passband and stopband specifications are met. You will need two plots, one with a scale appropriate for the passband and another with a scale appropriate for the stopband. A routine to plot the response on two different scales is as follows.

```
function PlotFilter (f, b, Fs)

B = freqz (b, 1, f, Fs);
BdB = 20 * log10 (abs(B));

subplot (2, 1, 1);
plot (f, BdB);
axis ([f(1), f(end), -80, +5]);

subplot (2, 1, 2);
plot (f, BdB);
axis ([f(1), f(end), -2, 2]);

return
```

### Assignment:

1. Convert the specifications (band edges, and deviations) given in terms of the 16 kHz sampling rate to a specifications on a normalized frequency scale. That frequency scale should be the one used by `fir1`, i.e. `Wn=1` corresponds to the half sampling frequency. For later use, convert the dB specifications to deviation specifications.
2. Explain why the documentation for `fir1` indicates that `N` must be even for highpass or bandpass filters.
3. Try designing the filter with `N=100` using a rectangular window. Can the specifications be met by increasing `N`? Explain.
4. Try designing the filter with `N=100` using a Hamming window. Can the specifications be met by increasing `N`? Explain.
5. Design the filter using a Kaiser window. Estimate the value of `beta` and `N`. Generate the Kaiser window and plug it into `fir1`. Check whether your design meets the specifications. If it does not, increment `N`, until it does. If it exceeds the specifications, decrease `N`, until you get the smallest value of `N` which meets the specifications.
6. Design the filter using the routine `firls`. Use weighting to best accommodate the passband and stopband requirements. What is the minimum number of coefficients of a filter which meets the specifications?
7. Design the filter using the routine `remez`. What is the minimum number of coefficients of a filter which meets the specifications?
8. Which design method gives the shortest filter meeting the specifications.