ECE 4213/5213
Homework 9

Fall 2023 Dr. Havlicek

Work the Projects and Questions in Chapter 7 of the course laboratory manual.

For your report, use the file LABEX7.doc from the course web site.

Work these questions only:

1. Q7.1
2. Q7.5 - Q7.9
3. Q7.13
4. Q7.14
5. Q7.16
6. Q7.20
7. Q7.22
8. Q7.23
9. Q7.25

Clarifications:

1. In designing practical digital FIR and IIR filters, you will often use formulas and/or tables. They all involve approximations. Different authors make the approximations slightly differently. So there are many similar but slightly different versions of the formulas and tables available.

   When you are doing a design, you must be careful not to *mix* formulas and/or tables that are based on different versions of the approximations. For example, if you are doing a design using formulas and tables from the notes, then don't mix them with formulas from the Lab manual, from the textbook, or from another book... unless you are *sure* that all of the formulas are based on the same versions of the approximations!

   In particular:

   - For doing FIR window designs, the formulas in equations (7.27) through (7.29) on page 114 of the Lab manual **do not exactly match** the similar versions that are found on page 10.14 of the course notes and **do not match** the similar versions that are found in Section 7.5 of the textbook.

     For working this homework assignment, **use the versions that are in the Lab manual**.

     For the Final Exam, **use the versions that are in the course notes**.

- The formula for estimating the order of a Kaiser window that is given in (7.37) on page 115 of the Lab manual does not match the formula on page 10.20 of the course notes (which is the same as Eq. (7.76) on page 542 of the textbook).

  Again, for this homework assignment use the version that is in the Lab manual. For the Final Exam, use the version that is in the notes.

2. In (7.5) and (7.6) on p. 110 of the Lab manual, the units of Ω_p and Ω_s are rad/sec, while those for F_T are samples/sec. Thus, the units of ω_p and ω_s are

$$\frac{\text{rad/sec}}{\text{sample/sec}} = \frac{\text{rad}}{\text{sample}}.$$

   But the frequencies W_p and W_s that you need for Matlab routines like `buttord` are "normalized" so that the Nyquist frequency is mapped to a "normalized" frequency of ONE.

   So to call routines like `buttord`, the W_p that you need can be obtained from the ω_p in (7.5) by:

   - divide by 2π to convert from rad/sample to Hz/sample.
   - multiply by 2 to convert the Nyquist rate, given by π rad/sample $= 0.5$ Hz/sample, into ONE on the normalized frequency scale.

   In other words, the overall conversion for taking ω_p in (7.5) to the W_p that you need for routines like `buttord` is to divide by π.

   For example, in Q7.1 I've got $F_T = 40$ kHz, $F_p = 4$ kHz, and $F_s = 8$ kHz. Plugging into (7.5) I get $\omega_p = 0.2\pi$ rad/sample and dividing by π I get $W_p = 0.2$ for calling `buttord`.

3. If you look carefully at program P7_1 on p. 119 of the Lab manual, in the third line you have $W_p = [0.3\ 0.7]$. This is a pair of passband edge frequencies for the bandstop filter. Let's call this version of W_p the "Book" version.

   Now, if you look in the file P7_1.m available on the course web site, you see instead on the third line $W_p = [0.2\ 0.8]$. Let's call this the "Program" version of W_p.

   For a bandstop filter, the Book version is more stringent... it allows transition bands that are only 0.1 units of normalized frequency in width. The Program version is less stringent, since it allows for transition bands that are 0.2 units of normalized frequency in width.

   Using either version, the program will work and it will design a Butterworth bandstop filter that meets the specification. For the Book version (more stringent), you will get N1=9 and the order of the filter is 18. For the Program version (less stringent), you will get N1=5 and the order of the filter is 10.

   To save typing, LET'S ALL GO WITH THE PROGRAM VERSION $W_p = [0.2\ 0.8]$. This means your numerator and denominator will have only six nonzero terms (instead of 10).

4. In problem Q7.7 on page 120 of the Lab manual it says to modify program P7_1 to design a Type 2 Chebyshev filter, but in the Lab report shell file it says "Type 1

Chebyshev." The Lab manual is CORRECT and the report file is INCORRECT. You should modify the program to design a Type 2 Chebyshev highpass filter.

For Q7.7 (and for the rest of the designs in this assignment), you may find it helpful to add horizontal and vertical lines to the magnitude response graph that show the required specifications. This will make it easier to answer questions about whether or not the obtained filter design meets the spec. If you add these lines, then you can use the zoom feature of the Matlab figure window to carefully examine the transitions and verify if the filter did or didn't meet the spec.

For Q7.7, I added the following code to my program to draw the lines:

```
% Add lines to the plot to help determine if the spec was met.
hold on;
tmpY = -60:65/511:5;
tmpX = ones(1,length(tmpY))*Wp;
plot(tmpX,tmpY,'r-');    % vertical line at passband edge freq
tmpX = ones(1,length(tmpY))*Ws;
plot(tmpX,tmpY,'g-');    % vertical line at stopband edge freq
tmpY = ones(1,length(w))*(-Rp);
plot(w/pi,tmpY,'r-');    % horizontal line at Rp
tmpY = ones(1,length(w))*(-Rs);
plot(w/pi,tmpY,'g-');    % horizontal line at Rs
```

5. Question Q7.9 is a little but frustrating because it's hard to figure out what is being asked. Here's some help on that.

   First:

   - read the question and all the material that goes with it in the text of the Lab manual.
   - read the Matlab help for `sinc`.
   - read the page that comes up when you type "doc sinc" at the Matlab command prompt.

   At this point you are probably confused.

   We are talking about an ideal lowpass filter with a cutoff frequency of 0.4π rad/sample. That means the fundamental period of the frequency response is a BOXCAR that's ONE from -0.4π to 0.4π and zero outside of that interval.

   Look at a table of DTFT pairs like the one on the "useful formulas" handout from the course web site. The tenth pair in that table says that the inverse DTFT of BOXCAR is SINC. Here, you have $W = 0.4\pi$. From this entry, the impulse response of our ideal lowpass filter with cutoff frequency 0.4π is given by

$$h[n] = \frac{\sin(0.4\pi n)}{\pi n}.$$

   Now, according to the Matlab help and "doc sinc," the Matlab sinc function is

$$\texttt{sinc(x)} = \frac{\sin(\pi x)}{\pi x}.$$

So, in terms of the Matlab `sinc` function, our ideal impulse response can be written as

```
h[n] = 0.4 * sinc(0.4*n)
```

to get

$$h[n] = \frac{0.4\sin[\pi(0.4n)]}{\pi(0.4n)} = \frac{\sin(0.4\pi n)}{\pi n}. \qquad \checkmark$$

So you can get the length-81 zero-phase lowpass filter impulse response and frequency response magnitude plot like this:

```
n = -40:40; % makes the length 81
hn_81 = 0.4*sinc(0.4*n); % length-81 version of h[n]
%
% Plot the DTFT magnitude with 1024 equally spaced samples on unit circle.
%  plot against Matlab normalized frequency that goes from 0 to 1.
omega = 0:pi/1023:pi;    % radian frequency vector
W = omega/pi;            % Matlab normalized freq vector
Hz_81 = abs(freqz(hn_81,1,omega));
plot(W,Hz_81);grid;
```

Note: for observing the Gibbs phenomena, you need to use a linear plot as just demonstrated, not a log plot like the Mitra supplied `gain` function gives you.

To get the other lengths, you need to shorten the impulse response by truncating more samples from both ends of `hn_81`. You can do this by extracting CENTERED subarrays from `hn_81` like this:

```
hn_61 = hn_81(11:71) % drops 10 samples from each end
```

Even though you are shortening the length of $h[n]$, you should still use the same 1024-point "omega" vector to evaluate the frequency response and the same 1024-point "W" vector to plot it.

Comparing the frequency response magnitude plots for all the lengths, you should see that decreasing the order reduces the number of ripples and makes the transition band wider. In other words, a higher order filter achieves a sharper transition with more ripples. But the peak of the largest ripple is never reduced no matter what the order is. This is the reason that so much work was done to develop windows like the Hamming window and Kaiser window that have better properties than the rectangular "boxcar" window.

6. Question Q7.16 asks you to repeat the FIR filter order estimation problem of Q7.13, but using the function `firpmord` instead of `kaiord.m`. Here's the point: `kaiord.m` implements the general purpose Kaiser formula for esimating the order of an FIR filter as given on page 10.2 of the notes. The function `firpmord` is also for estimating the order of an FIR filter, but it's specialized for the Parks-McClellan algorithm that is briefly described on page 10.24 of the notes. So the point of this question is really just for you to see that different formulas give you different estimates for the order. If you were going to use the Parks-McClellan algorithm to design the filter, then you

would want to use `firpmord` to estimate the order (you could then call `firpm` to do the design in Matlab).

Question Q7.16 then asks you to compare the order estimate obtained with `firpmord` to what you got in Q7.13 (using the general Kaiser formula in `kaiord.m`) and in Q7.15 using `kaiserord` (which is the specialized formula for doing a Kaiser window design). The **problem** with this is that **you weren't asked to do question Q7.15!** So just compare your result to what you got in Q7.13.

7. You should use the Mitra-supplied function `kaiord.m` to estimate the filter order for **both** Q7.13 and Q7.20.

   In the file LABEX7.doc it says to use the Matlab signal processing toolbox routine `kaiserord` to estimate the order for Q7.20. But that's INCORRECT if Q7.13 and Q7.20 are going to be consistent and comparable. So use `kaiord.m` for estimating the order in both Q7.13 and Q7.20.

   Here's an explanation of the difference between `kaiord` and `kaiserord` and why you should use `kaiord` for both Q7.13 and Q7.20:

   In these two questions, you are being asked to perform a window-based FIR design with order estimated by the general Kaiser formula given in (7.7) on p. 110 of the Lab manual and on p. 10.2 of the course notes. That's what `kaiord.m` does.

   The function `kaiserord` is different: it is specifically for an FIR windowed design using a Kaiser window, e.g., (7.34) on p. 115 of the Lab manual and also on p. 10.19 of the notes.

   Now, it is true that you could use a Kaiser window to design a filter that would meet the spec. But that's not what questions Q7.13 and Q7.20 are asking you to do.

   For my solution on Q7.20, I used `kaiord` to estimate the order and then called `h = fir1(N,Wn)` to do an FIR design using the default window choice (Hamming window). But this filter did not meet the spec (it failed in both the passband and the stopband). To meet the spec, I had to increase the order up to $N = 66$.

8. For Q7.22, again use `kaiord` to estimate the order (do **not** use `remezord`, even though that's what LABEX7.doc tells you to do – `remezord` is obsolete).

   This should give you $N = 46$, which will not meet the spec. Then, you can increase the order by one to $N = 47$ and you will get a filter that does meet the spec.

   Note: for doing Parks-McClellan FIR filter design with `firpm`, the routine `firpmord` will give you a better estimate of the order than `kaiord`. For this problem, using the call

   ```
   [n,fo,mo,w] = firpmord([2000 2500],[1 0],[0.005 0.005],10000)
   ```

   instead of `kaiord` would have estimated the order correctly as $N = 47$ and would also have given you all the parameters that are needed to call `firpm`. The reason I asked you to use `kaiord` instead for this problem was to get you to go through the exercise of setting up the parameters for `firpm` manually without using `firpmord`.

9. Q7.23 tells you to use formulas (7.36) and (7.37) from page 115 of the Lab manual to estimate β and the filter order, so do that. But in the future you will probably prefer

to instead call the signal processing toolbox function `kaiserord` to estimate them for you. It will give a better estimate of the order.

But for this problem, you will need to implement formulas (7.35) and (7.37) in your Matlab program. Here's how I implemented (7.37):

```
if As > 21
    N = ceil((As-7.95)*2/(14.36*(abs(Wp-Ws)))+1)
else
    N = ceil(0.9222*2/abs(Wp-Ws)+1)
end
```

The only thing that's a little bit tricky about that is understanding why I multiplied by 2 in the numerator. Here's the reason: Δf in the formula is given by the difference between the passband edge frequency and stop band edge frequency. In Matlab, those are `Wp` and `Ws`. But in Matlab, `Wp` and `Ws` are specified in normalized frequency – which is radians per sample divided by π. Now, (7.37) needs for Δf to be in units of Hz. Normally, to convert radian frequency to Hz you divide by 2π. But `Wp` and `Ws` in Matlab are *already* divided by π. So you just need to divide them by 2. Since `abs(Wp-Ws)` appears on the *bottom* in (7.37), I implemented that by multiplying the numerator by 2 instead.

Submit this assignment electronically on Canvas.

DUE: 12/4/2023 11:59 PM