EE-3220 LABORATORY

Week 5

Filter Design and Filter Response

Goal – Articulate the difference in system requirements between and IIR and FIR filter given identical design specifications. Illustrate the steady state response of a digital filter to sinusoidal input sequences.

Materials - Laptop computer with MATLAB

Overview: Design and analysis are two separate engineering functions. With analysis, engineers are provided a solution or equation and asked to compute how this solution will behave based on known inputs. Design is the reverse of the analysis process.

During the design process, engineers are provided specifications and tasked with developing a solution that meets the specifications. In design, there can be multiple solutions that meet the given specifications and having a clear understanding of the differences between solutions is very important.

In this lab, you will be designing digital filters. There are varieties of tools to help engineers design filters. For example, Texas Instruments FilterPro is an excellent application for designing active analog filters. MATLAB provides numerous tools for designing digital filters.

Below is a partial list of MATLAB functions (there are many others!). To see more details on these functions and their usage, type “help” followed by the function name (e.g., “help freqz”).

filterbuilder – a GUI based filter design object

freqz(b,a) - Calculates and plots the frequency response of the filter corresponding to coefficient vectors a and b (in powers of z^{-1} form).

fvtool(b,a) - Launches the Filter Visualization Tool that analyzes digital filters.

filter(b,a,x) - Filters the data in vector x with the filter corresponding to coefficient vectors a and b (in powers of z^{-1} form).

A few other helpful functions:

roots(c) - Computes the roots of the polynomial whose coefficients are the elements of the vector c.

poly(v) - Computes the coefficients of the polynomial whose roots are the elements of v.

Example 1. Low Pass FIR Filter:

Let’s use **filterbuilder** to design a low pass FIR filter with direct form realization that has the following specifications. The design will be done using normalized (0 to 1) frequency units, a pass frequency equal to 0.3 π rads/sample, and a stop band frequency at 0.8 π rads/sample. The amplitude should have unity gain in the passband and attenuate the signal by 60dB in the stop band frequencies.

1. *View the filter response and verify the specifications. Copy and place this in your submittal.*

The "b" coefficients of the difference equation for the FIR filter are stored in the "object" instance Hlp and can be found with the following command:

```
>>b = Hlp.Numerator
```

The "a" coefficient is assumed to be 1 for the FIR filter, therefore:

```
>>a = 1
```

The filter coefficients can also be viewed in the "Filter Visualization Tool" (the window that opened when you selected "View Filter Response". Click the "Filter coefficients" icon to see a listing of the coefficients. These can be copied into Matlab for analysis.

Let's find the impulse response function, h(n) for the filter, and the DTFT, H(exp(jw)), for the normalized filter:

```
>>[d,n] = impseq(0,0,60)
>>h = filter(b,a,d)
```

2. *What is the order of this filter? That is, how many delay elements are required to implement this filter?*
3. *View the stem plot of the impulse response function, h(n), for this filter. Copy and place this in your submittal. Cleary label all axes.*

```
>> f = linspace(0,1,4000);
>> w = pi*f;
>> for k = 1:4000,disp(k),H(k)=exp(-n*i*w(k))*h';end
>> plot(f, 20*log10(abs(H)))
```

4. *Label and set the axis for the magnitude plot of the DTFT, H(). Note that the horizontal axis is in π- radians/sample. Copy and place this in your submittal and provide a comparison to the plot obtained directly from the filterbuilder GUI.*

The phase plot also provides information on how the filter processes the signal. Filters with "linear" phase will maintain proper phase relationship between sinusoidal inputs. We will discuss this in more depth later. We can plot the phase of the filter with the following:

```
>>plot(f, angle(H))
```

5. *Label and set the axis for the phase plot of the DFT, H(). Note that the horizontal axis is in π-radians/sample and the vertical axis has the units of radians. Copy and place this in your submittal.*

6. *Is the phase a linear function of the frequency w? Explain.*

Example 2. Low Pass IIR Filter using 2nd order sections (sos)

In the design process, it is often easier to divide a large problem into smaller problems that are more manageable. This is called the “divide-and-conquer” methodology.

When designing IIR filters, the problem is typically divided into numerous 2nd order sub-filters and the resulting filter is a cascaded series of the sub-filters. For example, consider a 6th order filter that consists of three cascaded 2nd order sections. The resulting impulse response and transfer functions are:

$h(n) = h_1(n) \circledast h_2(n) \circledast h_3(n)$

and

$H(\exp(jw)) = H_1(\exp(jw))H_2(\exp(jw))H_3(\exp(jw))$,

respectively.

IIR filters designed using the filterbuilder application in MATLAB return a matrix that contains the coefficients for the second order sections in each row of the ***Hlp.sosMatrix*** variable. Therefore, the number of columns will always be six and the number of rows will equal the number of second order sections. The gain for each stage is provided in ***Hlp.scaleValues***

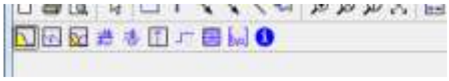
Use the filterbuilder application to design a filter with the same specifications as listed for Example 1, but now use an IIR topology.

Now, launch the Filter Visualization Tool using the fvtool() function:
>> fvtool(Hlp)

Note: You may need to do “fvtool(Hlp2)”, depending on whether the FIR filter from Example 1 is still in your Matlab workspace. You can specify a name using the “Save variable as:” option in filterbuilder.

You may also simply click “View Filter Response” from the filterbuilder window. This launches “fvtool” with the current filter parameters.

Experiment with the options in the toolbar shown below.



This tool provides most of the functionality you will need to analyze your designs. Using fvtool():

7. *Provide a plot of the magnitude of the transfer function*
8. *Provide a plot of the phase response*
9. *Provide the a and b coefficients.*
10. *Provide a plot of the impulse response function*

Matlab's **fdatool** provides much of the same functionality as filterbuilder and fvtool, but integrated into one GUI. Launch by typing "fdatool".

Consider the following design example. Use fdatool to experiment with FIR and IIR designs. For this design, we will use frequencies specified in Hz rather than normalized frequencies.

Design Example:

11. *Design a bandpass filter that will be used to filter a sampled sequence where the sampling frequency is 48kHz. The filter should have a passband from 330Hz to 3000Hz. The stop band shall be to 0 to 200Hz and greater than 4000Hz, with greater than 20dB of attenuation in the stop bands. Provide the following:*
    - *Compare the number of coefficients and filter order in the FIR and IIR designs. Why might you choose one versus the other?*
    - *Provide plots of the magnitude response for both filters. Compare them. Why might you choose one versus the other?*
    - *Provide plots of the phase response for both filters. Compare them. Why might you choose one versus the other?*

Submittal

Your submittal is due ***Monday, April 15th at 3PM***. Please submit to Dr. Kelnhofer during the normal lecture period. The submittal should include:

Course-Section:
Lab:
Name:
Date:

Provide a one or two paragraph Summary, and provide answers in a sentence format to the questions listed in italics and include all the plots that were requested.