EE-3220 LABORATORY
Week 5
Filter Design and Filter Response

Goal – Explain the difference in system requirements between and IIR and FIR filters 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 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 begin with specifications and then develop a solution that meets them. There are almost always multiple solutions that meet the specifications and having a clear understanding of the differences between solutions is necessary in order to make a rational choice among them.

In this lab, you will design digital filters. There are many tools to help design filters. For example, Texas Instruments' WEBENCH Filter Designer 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. To see more details type "help" or "doc" 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 difference equation coefficient vectors a and b.
>
> 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 the difference equation described by difference equation coefficient vectors a and b (which are also the coefficients of powers of z^{-1} in the z-transform form).

Some 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

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 frequency units (0 to 1, with 1 corresponding to a digital frequency of π rad/sample or fs/2 Hz), a pass frequency equal to 0.2 π rad/sample, and a stop band frequency at 0.7 π rad/sample. The amplitude should have unity gain in the passband and attenuate the signal by 40 dB in the stop band frequencies.

1. *After entering the specifications, click the "View 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 Hlp and can be retrieved with the following statement:

```
b = Hlp.Numerator;
```

To see what's in this object, type Hlp or double click on it in the Workspace pane. The a_0 coefficient is 1 for an FIR filter, therefore:

```
a = 1; % not used here, but needed when calling filter()
```

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.

Since this is an FIR filter, the impulse response, h, is equal to b.

```
h = b;
```

2. *What is the order of this filter? That is, what is the maximum delay of the previous values of x and y used?*
3. *View the stem plot of the impulse response function, h(n), for this filter. Place this in your submittal. Clearly label all axes.*

Let's find the DTFT, $H(e^{j\omega})$, for the filter.

```
f = linspace(0,1,4000); % normalized frequency
omega = pi*f; % digital frequency
n = 0:length(h)-1;
for k = 1:length(f)
      H(k) = exp(-1j*omega(k)*n)*h';
end
figure, plot(f, 20*log10(abs(H)))
ax=axis; ax(3:4)=[-120 5]; axis(ax) % override y-axis (dB) auto-range
```

4. *Label and set the axis for the magnitude plot of the DTFT, H(). Note that the horizontal axis is in π-radians/sample. Place this in your submittal and compare it 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 (they will have constant delay regardless of frequency). Consider a linear phase system where a 10 Hz wave goes through 360° in 0.1 s and was delayed by a phase φ. In this system a 20 Hz wave will have gone through 720° in the same amount of time; its phase is twice that of the 10 Hz wave, therefore its phase delay must also be doubled (2φ) in order for the peaks of both waves to retain their time relationship. We can plot the phase of the filter with the following:

```
figure, plot(f, unwrap(angle(H))) % figure creates a new plot window
```

5. *Read the documentation for unwrap and describe its purpose in your own words. You might find it helpful to generate the above plot without unwrap for comparison.*
6. *Label and set the axis for the phase plot of the DTFT, H(). Note that the horizontal axis is in π-radians/sample and the vertical axis has the units of radians. Place this in your submittal.*
7. *Is the phase a linear function of the frequency omega? 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:

$h(n) = h_1(n) * h_2(n) * h_3(n)$

and

$H(e^{j\omega}) = H_1(e^{j\omega})\, H_2(e^{j\omega}) H_3(e^{j\omega})$,

respectively.

IIR filters designed using filterbuilder are returned in a structure, typically named ***Hlp*** for a lowpass filter. The gain for each stage is provided in ***Hlp.scaleValues***. Coefficients for the second order sections are in each row of the ***Hlp.sosMatrix*** matrix. See the documentation for the sosMatrix layout.

- Type ***doc*** to open the interactive help system.
- Search for ***sosMatrix***.
- Review the options and select ***sos*** since it looks promising.
- The documentation you need isn’t here, but find a link to documentation for ***tf2sos***, which does document sosMatrix.

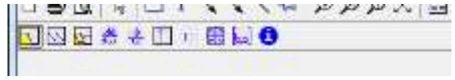
Use filterbuilder to design a filter with the same specifications as listed for Example 1, but now use an IIR topology.

Click “View Filter Response” from the filterbuilder window to launch “fvtool” with the current filter parameters. As an alternative, you may launch the tool from the prompt after clicking Apply (just creates filter in your workspace) or Ok (also closes the design tool):

```
fvtool(Hlp)
```

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

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():

8. *Provide a plot of the magnitude of the transfer function*
9. *Provide a plot of the phase response*
10. *Provide the a and b coefficients (also available in Hlp.sosMatrix).*
11. *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".

Work through 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

12. *Design a bandpass filter that will be used to filter a sampled sequence where the sampling frequency is 44.1 kHz. The filter should have a passband from 330 Hz to 3000 Hz. The stop bands shall be (a) 0 to 225 Hz and (b) greater than 4000 Hz, with greater than 20 dB of attenuation in the stop bands (you will enter 20; the tool understands that this is a lower bound).*
    Note: *some of the options require you to specify an order; it is possible to specify an order that is too low to accommodate the specifications. Start by experimenting with design methods that allow you to select "Minimum order," which will calculate the minimum order required to meet the specification for you.*
    Provide the following:
    - *Compare the number of coefficients and filter order in the FIR and IIR designs. Be sure to clearly state which design method you used (e.g., IIR: Butterworth). 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

The submittal should follow all instructions on the provided grading checklist and cover sheet.

The standard summary section is not required this week.

Additional Problems

Take a break from additional problems this week.