Designing Filters Using Matlab for the Real World

EE431
Lecture 23 notes

Lecture portion

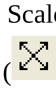
Types of FIR/IIR filters (note: all can be LP, HP, BP, BS)

1. FIR filter types
   a. Equiripple
      i. specify acceptable ripple in passband frequency
      ii. specify minimum attenuation in stopband frequency
   b. There are many, less-commonly-used FIR filters (e.g. least-squares), but I've never heard of a problem that the equiripple design method could not solve
   c. All these FIR filters are exactly linear phase, but the price paid is that they require a much higher filter order (and therefore, amount of processing power) than the IIR filters for the same specifications
2. IIR filter types
   a. Butterworth
      i. Pro: No ripples in passband or stopband ("monotonically decreasing"). Fairly close to linear phase.
      ii. Con: Requires a relatively high order for an IIR filter (but better than FIR).
   b. Chebyshev I: Don't use – use Chebyshev II instead
   c. Chebyshev II
      i. Pro: No passband ripple. Requires lower order than Butterworth or FIR.
      ii. Con: Stopband ripple. Not close to being linear phase.
   d. Elliptic
      i. Pro: extremely low order required. (e.g. 5^{th} order elliptic can be as efficient as 40^{th} order FIR).
      ii. Con: Ripple in both passband and stopbands. Highly non-linear phase.

filterDesigner brief demonstration

Hints to use filterDesigner in Matlab

1. Start filterDesigner by typing “filterDesigner” in the command window.

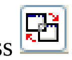
2. Design flow concept:
   a. Specify the filter in the lower half of the screen from left to right, then press "Design Filter"
   b. View its performance (e.g. gain, phase, group delay) in the upper window (choose what to view using the Analysis menu option). Set axis types (e.g. dB/linear) by right-clicking axis. Scale axis by clicking a magnify button  (for both axes, x, or y-only) or full-screen ().

3. To force frequency response graphs to use "normalized" units, i.e. zero to 1 (times π) , rather than Hz: right-click frequency axis and choose "normalized frequency".

4. When designing FIR filters, an option called "Density Factor" appears. It refers to how accurately the program will design the equiripple filter's coefficients. Use "20" for all designs; this is accurate yet fast. If the ripples don’t look equal, recomputed with a higher density number.

5. To get a printout of a graph or to embed it within a Word document

a. Design the filter and press the "Design Filter" button
b. Open just the upper graph window in its own window by pressing 
c. Select the graph to view using the Analysis menu
d. Either File → Print or Edit → Copy Figure

6. To compare responses of different filters
   a. Design the first filter
   b. Open up a graph window by pressing , and in the new window press  (tooltip: Set link mode to "add") to accumulate graphs.
   c. In the main filterDesigner window, design the second filter. This new one and the first one are both displayed in the stand-alone graph window.

7. To see the filter coefficients (i.e. H(z))
   a. **Right-click “structure” in the upper-left corner group box titled “Current filter information”. Select "Convert to single section" if available**.
   b. To view coefficients, menu Analysis → Filter Coefficients
   c. To create Matlab vector variables "Num" and "Den" (note capitalization) with the coefficients, menu File → Export, workspace, export as coefficients. (For FIR filters, only Num is created since Den=1).

In-Class Portion

With one partner:
1. Design a 20th order FIR lowpass filter used in a system with a 10kHz sampling frequency. The filter should pass frequencies lower than 2kHz but block frequencies greater than 2.5kHz. Use the help screen (from the filterDesigner window, choose help from the menu, then “filterDesigner help”).
2. (The "weights" should be set to 1 for both bands. They address the fact that FIR filters have ripple in both their stopband and passband. If both weights are the same, the ripples in each band are the same. Weighting the passband more heavily than the stopband will result in greater stopband ripples and reduced passband ripples).
3. Get a printout of the magnitude of the filter's transfer function (the "Magnitude Response") in dB and a list of the filter coefficients. You may either leave the class or begin on the homework assignment once you get checked off.

Out-of-Class Homework

Admin: this is a mini-lab. Turn in only one assignment per lab group. As with lab reports, you may not collaborate outside your lab group.

Hot research area: A rapidly-growing area of research deals with the construction of a 2D mat of sensors that are placed over a subject's head. As the subject thinks different thoughts, different signals are recorded in different areas of the brain, and show up as different measured signals on the sensor mat. Subjects have already learned to navigate through video games, controlling their motion only by thoughts. Two heavily-funded goals are to create systems that allow quadriplegics to walk (controlling remote motors on prosthetic legs via thought) and pilots to control air-to-air missile firing mechanisms faster (eliminating the ~0.2s of time it takes for a thought to be converted into a nerve impulse that is transmitted to hand motion).

A problem with reading the sensor values is that the transduced signals are very weak, on the order of 10μV. The scalp is high-impedance, so 60Hz noise is easily capacitively-coupled and contaminates the EEG (electroencephalogram, the brain's signal).

Using only the person(s) in your lab group for assistance,

1. Filter Design: Use Matlab's filterDesigner to design two different lowpass filters to be used to eliminate the 60Hz contamination in this system. Assume the system uses a 160Hz sampling frequency. The goal is to allow the EEG frequencies below 45 Hz to pass unimpeded (with at most a 1dB variance in this passband from the ideal 0dB gain), but eliminate 99% of the 60Hz (i.e. have a gain of at least -40dB at 60Hz).
   a. Design a FIR filter to accomplish this.
      i. What is its order?
      ii. What is its H(z)? (cut and paste the numerator vector)

   b. Design an elliptic filter to accomplish this.
      i. What is its order?
      ii. What is its H(z)? (cut and paste its numerator and denominator vectors)

2. Filter Analysis. Test the performance of each of your filters in Matlab by creating a mock EEG signal with 60Hz noise and see the effects of your filter upon it. Note: to make the four plots below more easily comparable, change the axis scale (help "axis") to make each horizontal axis go from 0 to 1 and each vertical axis go from -10 to 10.
   a. In Matlab (regular Matlab, not filterDesigner), create a signal sampled at 160Hz of a 16Hz squarewave of ±1V that is exactly 1 second long. Plot the signal as the top plot of a 4 row x 1 column subplot. Title it "Orignal EEG signal".
   b. Contaminate the EEG signal by adding to it a 60Hz, ± 10V sinusoid (10x higher amplitude than the EEG signal; the original signal will be unnoticeable). Make it 1 second long. Plot the combined signal and title it "EEG with 60Hz noise".
   c. Plot the signal after filtering with the FIR filter. (Hint: Matlab's "filter" command). Plot the signal and title it "EEG signal after FIR processing".
   d. Plot the signal after filtering with the IIR filter. Plot the signal and title it "EEG signal after IIR processing".

3. a. What are the benefits/drawbacks of each of the two filters you designed for this application?

   b. Both the FIR and IIR processed signals should have similar frequency content in the frequency domain, but as you would expect, the FIR-processed signal will look more similar to the original signal than the IIR-processed signal. Is this just because the IIR signal was a lower order? (To be sure you answer this question correctly, try designing an IIR filter of the same order as the FIR filter in filterDesigner and see!).