Assigned: 11/05/10 Due: 11/15/10

IIR Filter Design

This project will explore and compare various methods for designing IIR filters.

There are 4 classical IIR filters and their counterparts (1) Butterworth, (2) Chebyshev, (3) Chebyshev II, (4) elliptic function. They represent four different combinations of two error approximation measures. One error measure uses the Taylor series. This method equates as many of the derivatives of the desired response as possible to those of the actual response. The other approximation minimizes the difference between the desired and the actual response over a band of frequencies. The MATLAB analog filter design programs all normalize the band edge to $\Omega_0 = 1$.

1. The analog *Butterworth* filter is based on a Taylor series approximation in the frequency domain with expansions at $\Omega = 0$ and $\Omega = \infty$. This filter is also called maximally flat since it is optimal in the sense that as many derivatives as possible equal zero at $\Omega = 0$ and $\Omega = \infty$. The formula for the magnitude squared of the normalized frequency response of an Nth order analog Butterworth lowpass filter is given by

$$| H(\Omega) |^2 = \frac{1}{1 + \Omega^{2N}} \tag{1}$$

   This response is normalized so that the magnitude squared is always 1/2 at $\Omega = 1$ for any N. Replacing Ω by Ω/Ω_0 would allow an arbitrary band edge at Ω_0.

2. The analog *Chebyshev* filter has a minimum maximum error over the passband and Taylor approximation at $\Omega = \infty$. The maximum error over a band is called the Chebyshev error.

3. The analog *Chebyshev II* filter (sometimes called the *inverse Chebyshev* filter) is a Taylor series approximation at $\Omega = 0$ and has minimum Chebyshev error in stop band. This is often a more practical combination of characteristics than the usual Chebyshev filter.

4. The analog *elliptic* function filter uses a Chebyshev approximation in both the passband and the stopband. Since the elliptic function requires the evaluation of the Jacobian elliptic functions and the complete elliptic integrals the theory is considerably more complicated than the other cases. Fortunately MATLAB does all the work for you so that all are equally easy to design.

These four optimal analog filters can be transformed into optimal digital filters with the bilinear transform that is investigated in the second half of this project. The IIR filter design programs in MATLAB take care of analog filter design and bilinear transformation into the digital form automatically.

Exercises

1. **Analyze a 5th order LP IIR filter**

   Use the MATLAB command `butter` to design a 5th-order lowpass IIR filter with a passband cutoff of 0.15π. Plot the magnitude and phase frequency responses using `freqz`. Plot the pole and zero location diagram using `zplane`. Plot the significant part of the impulse response using `filter` to give around 25 output values. (Think of the filter function with the coefficients as the implementation of a system. What input to the system must you use to get the impulse response out?) Discuss briefly how the magnitude response makes sense from the pole-zero plot.

Use the MATLAB command `cheby1` to design a 5th-order lowpass IIR filter with a passband cutoff of 0.15π and a passband ripple of 0.5 dB. Plot the magnitude and the phase frequency responses, as well as the impulse response. Plot the pole and zero location diagram. Discuss briefly how the magnitude response makes sense from the pole-zero plot.

Use the MATLAB command `cheby2` to design a 5th-order lowpass IIR filter with a passband cutoff of 0.15π and a maximum stopband ripple 30 dB below the passband response. (Hint: You may need to use `cheb2ord` in a trial-and-error fashion to meet the stopband requirement and the passband cutoff. Note that `cheb2ord` returns the stopband frequency that must be used in `cheby2`.) Plot the magnitude and the phase frequency responses, as well as the impulse response. Plot the pole and zero location diagram. Discuss briefly how the magnitude response makes sense from the pole-zero plot.

Use the MATLAB command `ellip` to design a 5th-order lowpass IIR filter with a passband cutoff of 0.15π, a passband ripple of 0.5 dB, and a maximum stopband ripple 30 dB below the passband response. Plot the magnitude and the phase frequency responses, as well as the impulse response. Plot the pole and zero location diagram. Discuss briefly how the magnitude response makes sense from the pole-zero plot.

In what way(s) can the performance of these filters be compared since their specifications are not the same?

Do the impulse responses approximate the impulse response of an ideal lowpass filter? Explain.

2. **Compare the order of the four designs**

   The filtering specifications for a particular job have a passband ripple of 0.5 dB, passband edge of 0.15π, stopband edge of 0.6π, and a stopband ripple below 30 dB. What order Butterworth, Chebyshev, Chebyshev II, and elliptic filters will meet these requirements? Use the `buttord`, `cheb1ord`, `cheb2ord`, and `ellipord` commands. Why does the elliptic filter have the lowest order?

3. **Design a lowpass filter for an audio signal**

   Read in the audio file (right-click the link) `ftp://ftp.eng.auburn.edu/pub/sjreeves/classes/doorbell.au` using `auread`. Using the `fft` function, determine the frequencies of the upper and lower tone of the doorbell. Design a filter such that the lower tone is attenuated no more than 3 dB while the upper tone is attenuated at least 15 dB. Design all four types of filter. Listen to the filtered signal for each case. Compare the results for each of the filters in terms of sound and filter order.

Write a **short** report describing your findings following my format instructions. The text should be no more than two pages of 12-point type with 1.5 line spacing, not including plots. The report should contain a concise description of your results. **Include all plots you were required to generate. Include the plots as small as remains legible *within* the text of the report, not at the end.** *Be sure to answer all questions.*

NOTE: All out-of-class work is to be done independently. Sharing of programming tips and discussing general concepts is ok. Collaborating on experiments or code-writing is not. Any such collaboration on these assignments will be considered an act of dishonesty and will be treated accordingly.
For further help:

- MATLAB Primer
- MATLAB Help Desk