EE 324 LAB 9

Digital IIR Filter Design

In this lab, you will learn how to design infinite-impulse-response (IIR) digital filters by digital conversion of analog filters. This method starts off with the s-domain transfer function of an analog filter (prototype). The digital filter is obtained as a "discrete approximation" of the analog filter. Such a discrete approximation can be found by one of the transformations studied in Lab 5. In this lab, you will use the bilinear/trapezoidal/Tustin approximation.

Prelab:

Goal is to design a 4^{th} -order IIR Butterworth BPF with center frequency 100 Hz and bandwidth 40 Hz.

1. Choose an appropriate sampling rate (e.g., 300 Hz) and determine the sample period.

2. Pre-warp the center, lower, and upper cutoff frequencies using,
$$\omega_p = \frac{2}{T}\tan(\frac{T}{2}\omega)$$

3. Create a normalized analog 4^{th}-order Butterworth LPF (normalized filter will have cutoff of 1 rad/sec), and apply LP to BP transform, using the prewrapped frequencies in step 3.

4. Apply bilinear transform to create a digital filter, $H(z)$.

5. Plot frequency response of both BPFs, $H(s)$ and $H(z)$.

Laboratory Assignment:

Our goal is to design a bandpass IIR digital filter with the specifications defined in Figure 1.

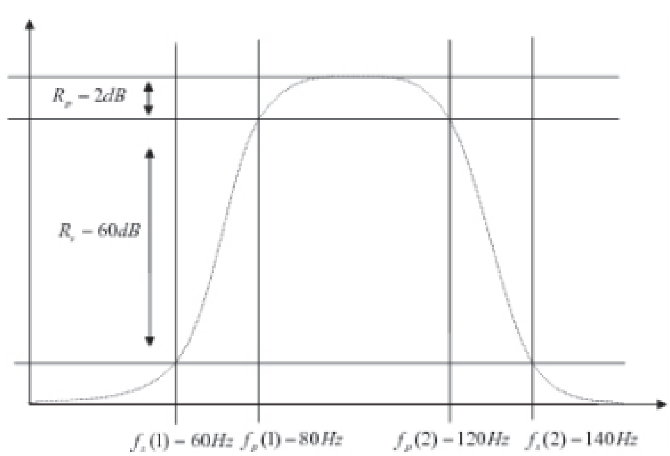

Figure 1 Filter Specifications

1. Using MATLAB, design an analog Chebyshev filter with the given specifications (as in Lab 7), and report the resulting transfer function (in Laplace representation).

2. The next step in digital filter design is to choose an appropriate sampling frequency, f_s. For this lab, set $f_s = \frac{1}{T} = 300$ Hz. (This works well from Nyquist requirement.)

3. Apply a bilinear transformation (using `c2d`) to obtain a discrete approximation of the analog filter.

4. Using the function `freqz`, look at the frequency characteristics of the filter, and report the attenuation at the "frequencies" corresponding to [60, 80, 120, 140] Hz. [Recall that in the discrete-time Fourier transform (DTFT) domain, the Ω value of π corresponds to the maximum frequency (= half the sampling frequency $\frac{f_s}{2}$) in the continuous-time Fourier domain.]

5. You will note that the discretized filter fails to meet the required specifications. This is due to the *frequency warping* introduced by the bilinear transformation.
   For a frequency ω, the bilinear mapping transforms it to: $\omega' = \frac{2}{T}\tan^{-1}(\frac{T}{2}\omega)$.

6. Perform *frequency pre-warping*, i.e. modify the significant frequencies of the analog prototype [60, 80, 100, 120, 140] such that a frequency value ω will become:
$$\omega_p = \frac{2}{T}\tan(\frac{T}{2}\omega)$$

7. Design a new Chebyshev prototype, with the new specifications, and apply the bilinear transformation again to obtain a new digital filter.

8. Verify the specifications of the new digital filter.

9. Design a digital filter with the specifications of Figure 1 directly, using the MATLAB commands `cheb1ord` and `cheby1` (but without mentioning the argument `'s'`). Note that the frequencies used as the arguments of `cheby1` should be normalized to the Nyquist frequency. Visualize its frequency characteristics and compare them to the filter you designed by "manual" pre-warping.

Save your code for use in the next labs. Also, include it in your report, along with the rest of the reporting.