Lab 7
Filter Transfer Functions in Matlab

In this lab you will use Matlab to examine several different filter types.

Pre-lab

1. If you have access to Matlab read the documentation for the function that will be used in this lab.
2. Depending on your Matlab access and skills you can carry out all the work for this lab before the lab meets.

Pole-Zero Plots

Plot the poles and zeros of several different filter transfer functions in the s-plane.

1. Write a program which produces a pole-zero plot of the transfer function for the following types of filters: Butterworth, Chebyshev Type 1, Elliptic, and Bessel.
2. The functions `buttap`, `cheb1ap`, `besselap` and `ellipap` give the complex poles and zeros of the filters (use help to find out about these functions). For each filter it is necessary to specify the order N. For the Chebyshev Type 1 and Elliptic, you also have to specify the pass band ripple R_p. For the Elliptic, you need to specify the stop band ripple R_S.
3. Use the `plot` function with specified character type (`x` or `o`) to generate pole-zero plots for the filters. Use the `hold(on/off)` function to hold one plot (say the poles) while overlaying the zeroes on the plot.
4. Document (print for lab book and discuss) the pole placement of the three filter types for the case of six poles, 3 dB pass band ripple, and (for the Elliptic filter) 80 dB of stop band ripple. Label the pole-zero plot using the `title` command. What is the effect on the poles and zeros of changing the ripple of the Chebyshev and Elliptic filters?

Frequency Response

Expand your program to plot the frequency response $H(j\omega)$ for the four different types of filters.

5. This is done using the `poly` function to convert the pole values into the coefficients of the denominator polynomial and the zero values into the coefficients of the numerator polynomial. You could also use the `zp2tf` function to convert the zero-pole representation to the polynomial representation of the transfer function. Generate polynomials

for the denominator and numerator of $H(s)$, then use the function `freqs` to obtain $H(j\omega)$. To do this, you will need to generate a vector of ω values. This is best done using the `logspace` function which gives equally spaced values for a Bode plot. 100 to 200 values should suffice. For choosing the range of ω values, note from your pole-zero plots that the filter functions are normalized to have ω_0 values of unity.

6. Plot the magnitude (using `abs`) of $H(j\omega)$ vs. frequency on a log-log plot, using the `loglog` function. Use the `hold` function to plot the magnitude vs. frequency of all filters on one plot.

7. Plot the magnitude (using `abs`) and phase (using `angle`) of $H(j\omega)$ vs. frequency on a semi-log plot, using `semilogx` function. It may be helpful to take out jumps of 2 pi using the `unwrap` function, and to convert the phase values from radians to degrees. You can pause between successive plots in your program using the `pause` function.

8. Compare your plots with those on p. 276 of Horowitz and Hill.

9. Check that the Chebyshev filters are "equiripple" (best done using a linear magnitude plot) and that the ripple values are as advertised. How many ripples are there in the pass band of a six-pole filter? What is the effect of the zeroes in the Elliptic filter?

Step and Impulse Responses

Expand your program (or write a new program) to plot the step and impulse response of the different filters.

10. Recall from linear system theory that the the impulse response $h(t)$ and the step response $g(t)$ are inverse Laplace transform of the system function $H(s)$:

$$h(t) = L^{-1}\left(H(s)\right) \qquad g(t) = L^{-1}\left(\frac{H(s)}{s}\right)$$

11. Use the MATLAB functions `step` and `impulse` to plot the step and impulse responses for your 6-pole filters. Briefly comment on differences in rise times, overshoot, etc. Which filter gives the best reproduction of a step?