Lab 8 Report

Analog Filters

Alex Hirzel

Submitted to Yang Liu for EE4252

Due November 15, 2012[1]

Michigan Technological University

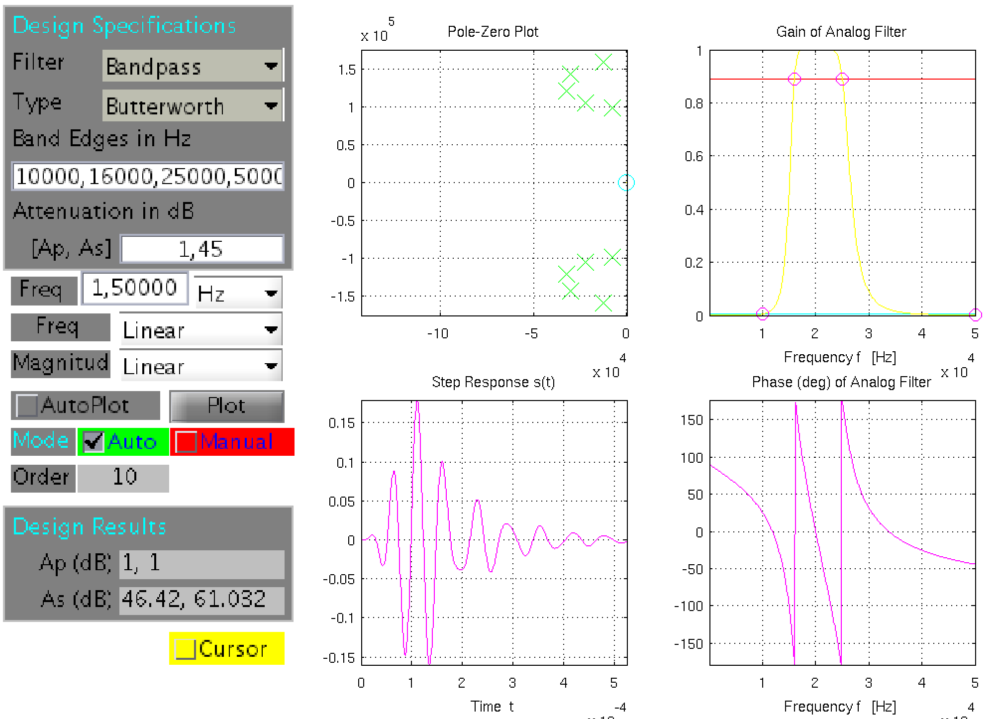
[1]postponed to November 29, 2012 due to academic schedule

Project 1: Basic Analog Filter Design

This project creates and compares four implementations of a band-pass filter. Butterworth, Chebyshev I/II and elliptic design techniques are compared. All designs are completed using `afdgui` and by hand.

Butterworth filter

(a) This filter was designed using `afdgui` with band edges of $[f_1, f_2, f_3, f_4] = [10000, 16000, 25000, 50000]$ and attenuation specifications $[A_p, A_s] = [1, 45]$. A screenshot of `afdgui` is shown below.



(b) As seen in Listing 2, the orders of the numerator and denominator of the final band-pass filter are 5 and 10, respectively.

(c) The low-pass prototype should be designed with order 5 (because $2 \cdot 5$ poles will appear in the resulting denominator after transforming the LPP to a band-pass.)

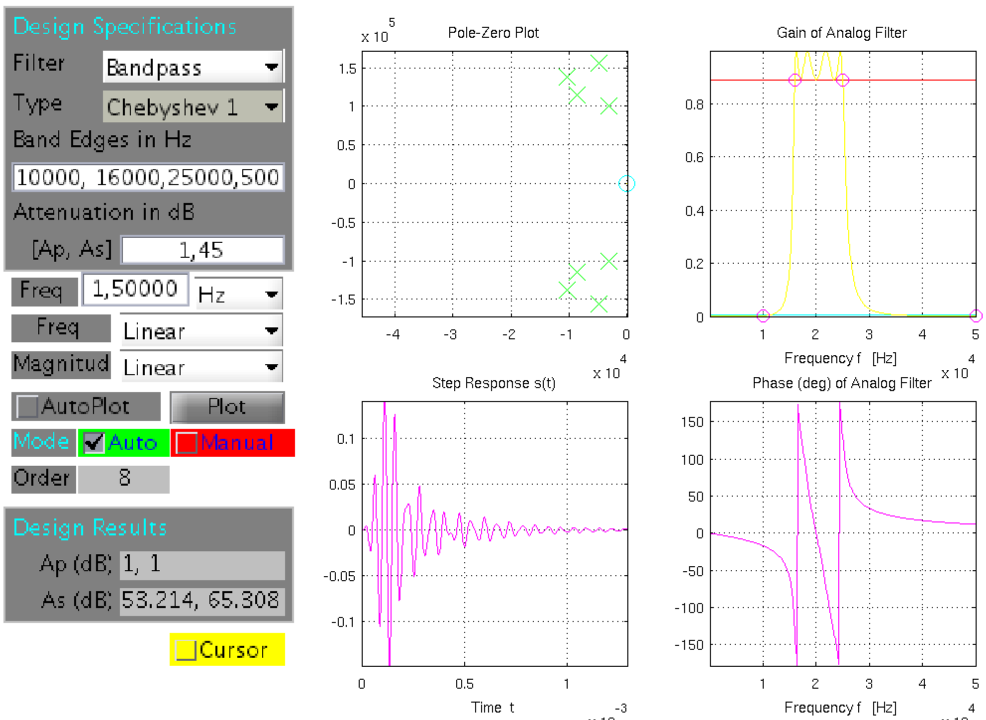
(d) As seen in Listing 2, the orders of the numerator and denominator of the low-pass prototype are 0 and 5, respectively. The final band-pass filter is this low-pass prototype transformed by a function involving s^2, so the order of the final band-pass filter is expected to be 10.

(e) The pole-zero plot is shown in Figure 1(a). The pole locations for the low-pass prototype appear to lie equi-spaced on the left half of a circle. (This is expected.) There are no zeroes in the prototype.

(g) Yes, the transfer functions match as seen in Listing 2.

Chebyshev I filter

(a) This filter was designed using `afdgui` with band edges of $[f_1, f_2, f_3, f_4] = [10000, 16000, 25000, 50000]$ and attenuation specifications $[A_p, A_s] = [1, 45]$. A screenshot of `afdgui` is shown below.



(b) As seen in Listing 2, the orders of the numerator and denominator of the final band-pass filter are 4 and 8, respectively.

(c) The low-pass prototype should be designed with order 4 (because $2 \cdot 4$ poles will appear in the resulting denominator after transforming the LPP to a band-pass.)

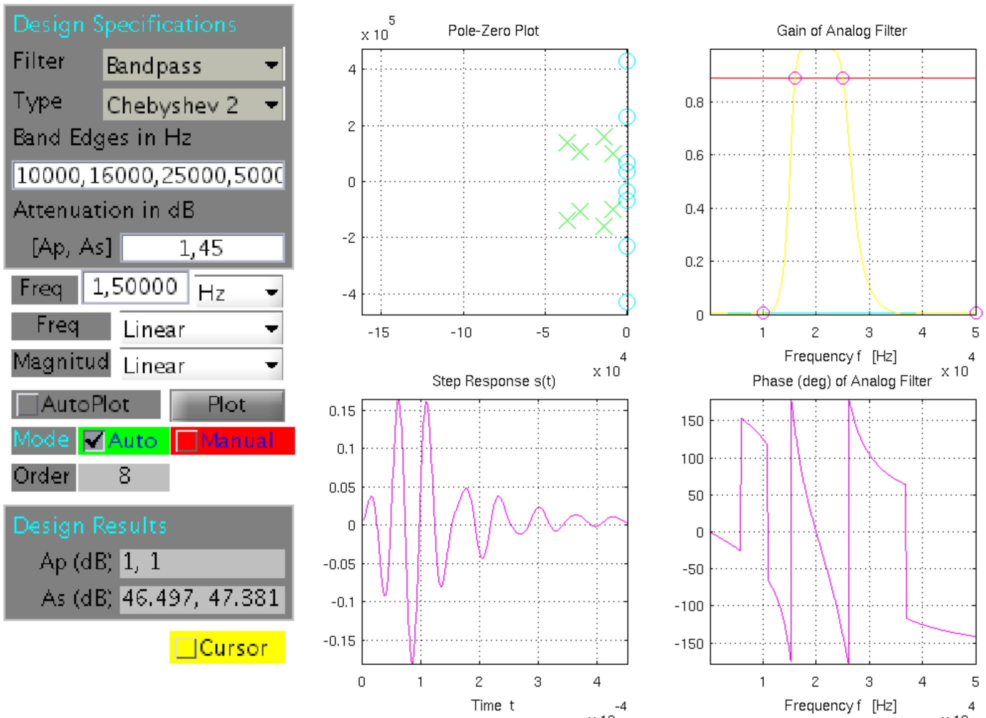
(d) As seen in Listing 2, the orders of the numerator and denominator of the low-pass prototype are 0 and 4, respectively. The final band-pass filter is this low-pass prototype transformed by a function involving s^2, so the order of the final band-pass filter is expected to be 8.

(e) The pole-zero plot is shown in Figure 1(b). The pole locations for the low-pass prototype appear to lie equi-spaced on the left half of an ellipse. (This is expected.) There are no zeroes in the prototype.

(g) Yes, the transfer functions match as seen in Listing 2.

Chebyshev II filter

(a) This filter was designed using **`afdgui`** with band edges of $[f_1, f_2, f_3, f_4] = [10000, 16000, 25000, 50000]$ and attenuation specifications $[A_p, A_s] = [1, 45]$. A screenshot of **`afdgui`** is shown below.



(b) As seen in Listing 2, the orders of the numerator and denominator of the final band-pass filter are 4 and 8, respectively.

(c) The low-pass prototype should be designed with order 4 (because $2 \cdot 4$ poles will appear in the resulting denominator after transforming the LPP to a band-pass.)

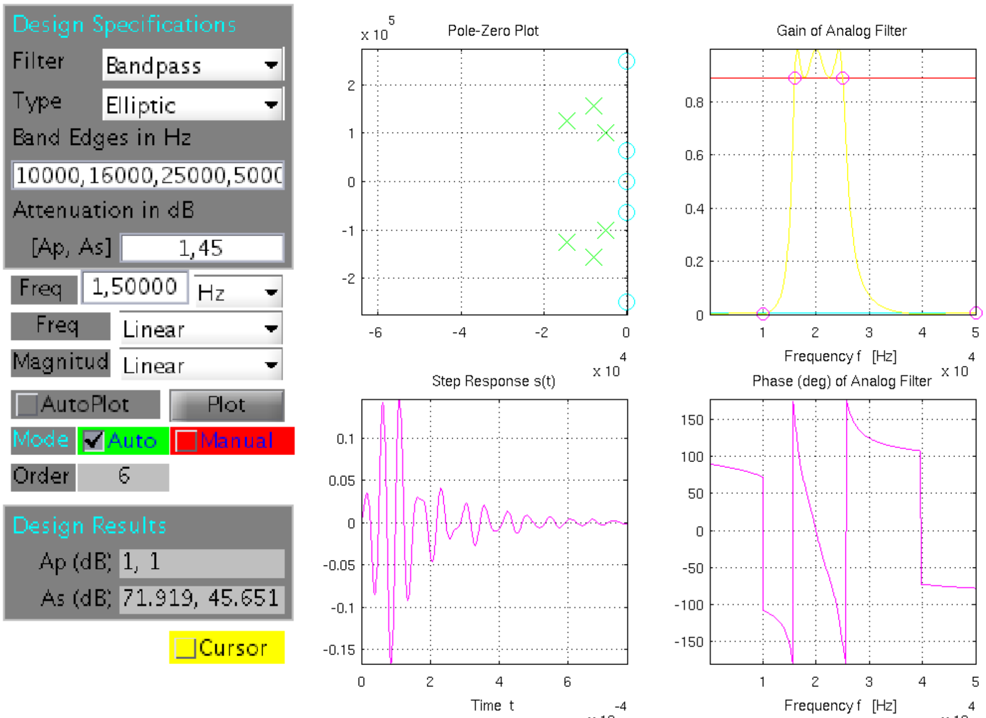
(d) As seen in Listing 2, the orders of the numerator and denominator of the low-pass prototype are 0 and 4, respectively. The final band-pass filter is this low-pass prototype transformed by a function involving s^2, so the order of the final band-pass filter is expected to be 8.

(e) The pole-zero plot is shown in Figure 1(c). The pole locations for the low-pass prototype appear to lie equi-spaced on the left half of an ellipse. (This is expected.) The zeroes are all on the $j\omega$ axis.

(g) Yes, the transfer functions match as seen in Listing 2.

Elliptic filter

(a) This filter was designed using **afdgui** with band edges of $[f_1, f_2, f_3, f_4] = [10000, 16000, 25000, 50000]$ and attenuation specifications $[A_p, A_s] = [1, 45]$. A screenshot of **afdgui** is shown below.



(b) As seen in Listing 2, the orders of the numerator and denominator of the final band-pass filter are 5 and 6, respectively.

(c) The low-pass prototype should be designed with order 3 (because $2 \cdot 3$ poles will appear in the resulting denominator after transforming the LPP to a band-pass.)

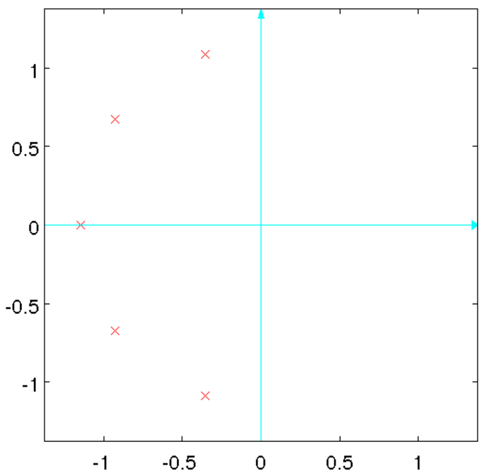
(d) As seen in Listing 2, the orders of the numerator and denominator of the low-pass prototype are 2 and 3, respectively. The final band-pass filter is this low-pass prototype transformed by a function involving s^2, so the order of the final band-pass filter is expected to be 6.

(e) The pole-zero plot is shown in Figure 1(d). The pole locations for the low-pass prototype appear to lie equi-spaced on the left half of an ellipse. (This is expected.) The zeroes are all on the $j\omega$ axis.

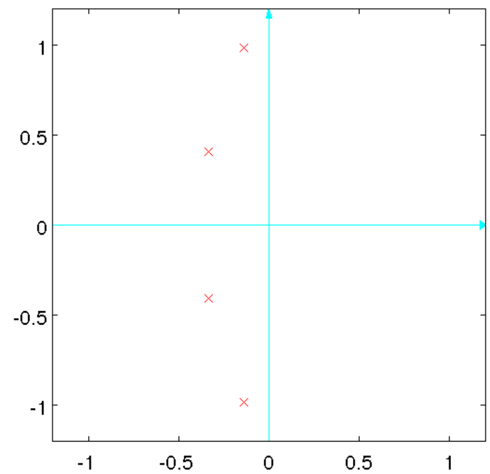
(g) Yes, the transfer functions match as seen in Listing 2.

Summary

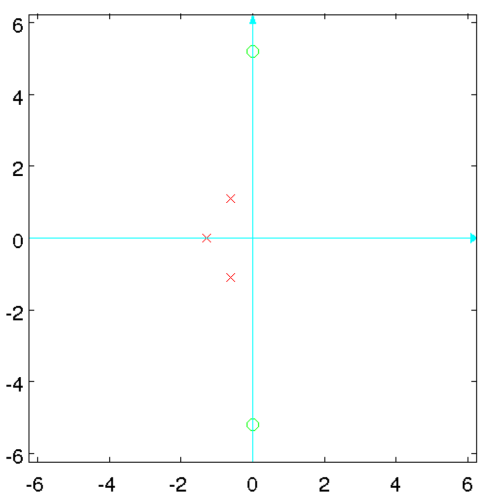
(k) The elliptic filter ends up with the smallest order for these specs (order 3). This is expected, because the elliptic filter has the largest compromise: ripple in both the passband and stopband. The designer is able to exploit this tolerance of ripple to reduce the filter order, whereas the monotonic Butterworth filter has a higher order but the (sometimes) useful property that there is no ripple.



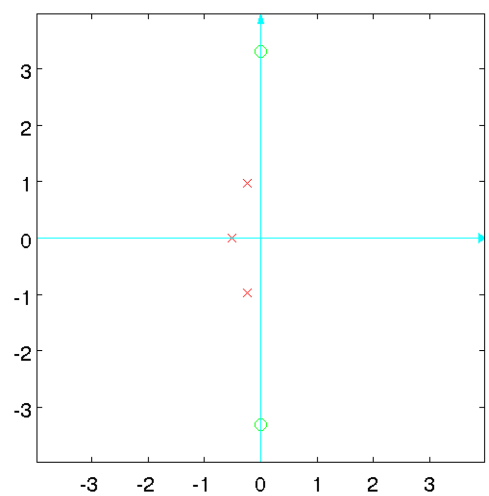
(a) Pole-zero plot of Butterworth LPP.



(b) Pole-zero plot of Chevyshev I LPP.



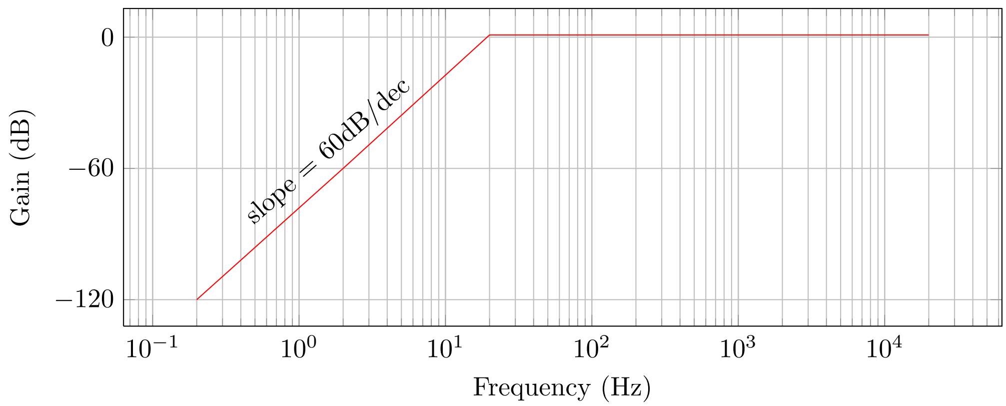
(c) Pole-zero plot of Chebyshev II LPP.



(d) Pole-zero plot of elliptic LPP.

Project 2: Subsonic Filters

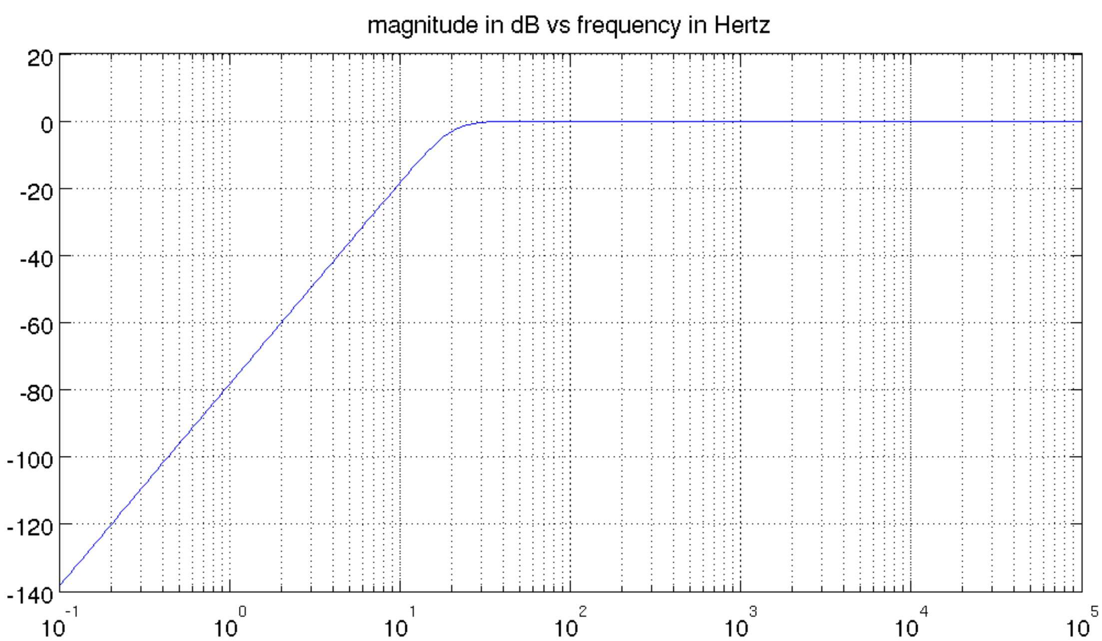
The filter to be designed for this project would have a Bode plot of the following general form:



The Butterworth filter design will require the use of a third-order system (because Butterworth filters inherently decay at −20dB/dec). The functions `lpp` and `lp2af` were used to arrive at the final filter design:

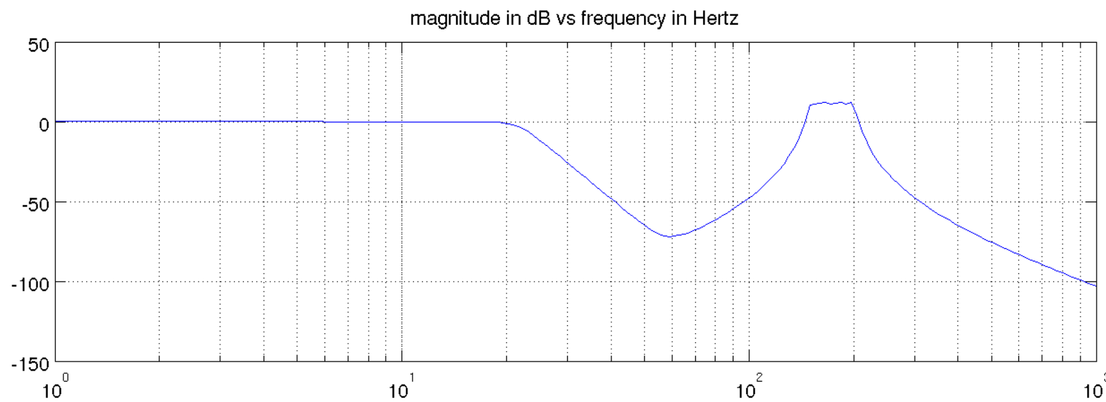
$$\boxed{H(s) = \frac{s^3}{s^3 + 251.1s^2 + 31530s + 1980000}}$$

which has a frequency response as shown below:



Project 3: A Multi-Band Filter

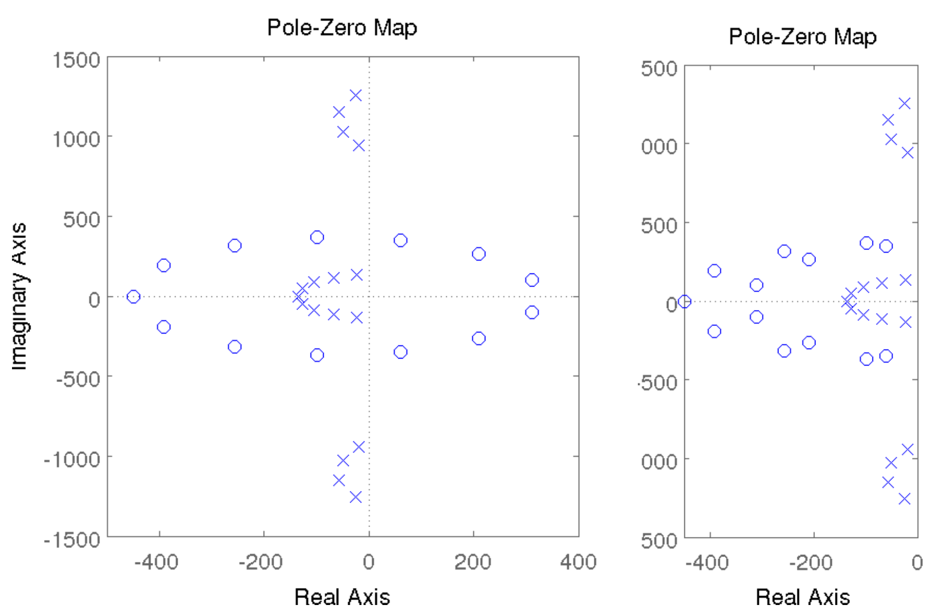
(a) For Passband 1, a Butterworth filter must be used due to the ripple requirements. A Chebyshev I filter can be used for Passband 2 because pass-band ripples are permitted. The final filter was designed in MATLAB using `afd` and has the following magnitude spectrum:



(b) The maximum attenuation in the range $[0, 400]$Hz is about 72dB at $\approx$ 60Hz.

Frequency	20	40	100	150	200	300
Magnitude	0.999988	48.325129	47.800160	-11.000000	-11.000000	47.802299

(c) The filter as designed is not stable (there are zeroes in the right-half plane as shown below in Figure 1(a)). The designed filter is *not* minimum phase, however, and using `minphase` to transform the filter results in a stable filter (shown in Figure 1(b)).

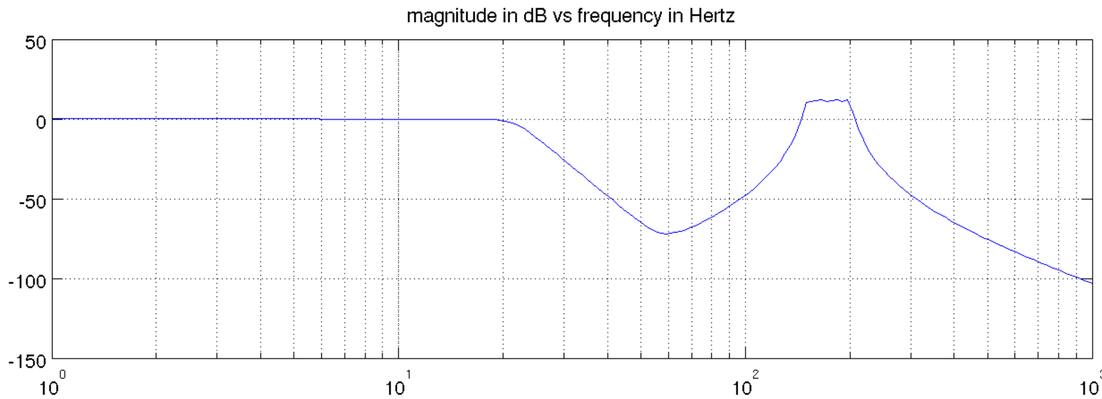


(a) Non-minimum phase.

(b) Minimum phase.

Figure 1: Pole-zero plots of filter designed for Project 3.

(d) Overplotting shows that $H(s) == H_{\mathrm{M}}(s)$. Shown below is $H_{\mathrm{M}}(s)$.



Appendix: MATLAB Source Code

What follows is a listing of the MATLAB source code (Listing 1) and the text output of this code (Listing 2) used to generate the figures and other information presented in this report.

Listing 1: The MATLAB script used for this report, `Lab08_ahirzel.m`.

```
addpath ../../ClassWorkspace;
delete 'generated/diary.txt'; diary 'generated/diary.txt'; diary on


% Project 1: Basic Analog Filter Design %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% FOR EACH:
%afdgui % make sure to export the data as bw_*
%mysaveas('p1a-bw', 8, 8);

load bw_export
[N, D] = lpp('bw', 5, 1); plotpz(N, D, 's'); mysaveas('p1a-bw-pz', 4, 4);
[Nbp, Dbp] = lp2af('bp', N, D, 20000*2*pi, 9000*2*pi);
disp('Project 1: Butterworth LPP by hand:'); tf(N, D)
disp('Project 1: Butterworth TFs generated by afdgui and by hand:')
tf(bw_Numaf, bw_Denaf)
tf(Nbp, Dbp)

load c1_export
[N, D] = lpp('c1', 4, 1); plotpz(N, D, 's'); mysaveas('p1a-c1-pz', 4, 4);
[Nbp, Dbp] = lp2af('bp', N, D, 20000*2*pi, 9000*2*pi);
disp('Project 1: Chebyshev I LPP by hand:'); tf(N, D)
disp('Project 1: Chebyshev I TFs generated by afdgui and by hand:')
tf(c1_Numaf, c1_Denaf)
tf(Nbp, Dbp)

load c2_export
[N, D] = lpp('c2', 3, [1 45]); plotpz(N, D, 's'); mysaveas('p1a-c2-pz', 4, 4);
[Nbp, Dbp] = lp2af('bp', N, D, 20000*2*pi, 9000*2*pi);
disp('Project 1: Chebyshev II LPP by hand:'); tf(N, D)
disp('Project 1: Chebyshev II TFs generated by afdgui and by hand:')
tf(c2_Numaf, c2_Denaf)
tf(Nbp, Dbp)

load el_export
[N, D] = lpp('el', 3, [1 45]); plotpz(N, D, 's'); mysaveas('p1a-el-pz', 4, 4);
[Nbp, Dbp] = lp2af('bp', N, D, 20000*2*pi, 9000*2*pi);
disp('Project 1: Elliptic LPP by hand:'); tf(N, D)
disp('Project 1: Elliptic TFs generated by afdgui and by hand:')
tf(el_Numaf, el_Denaf)
tf(Nbp, Dbp)


% Project 2: Subsonic filters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[Nlp, Dlp] = lpp('bw', 3, 3);
[N, D]     = lp2af('hp', Nlp, Dlp, 20*2*pi);
tfplot('s', N, D, [0.2, 20000], 1, 1); mysaveas('p2-final-bode', 8, 4);
tf(N, D)


% Project 3: A Multi-Band Filter %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[N1, D1, Nlpp1, Dlpp1] = afd('bw', 'lp', [1 45], 20, 40);
[N2, D2, Nlpp2, Dlpp2] = afd('c1', 'bp', [1 50], [150 200], [100 300]);

HN = conv(N1, D2) + conv(10^(12/20) .* N2, D1);
HD = conv(D1, D2);
H = @(s) polyval(HN, s) ./ polyval(HD, s);

for f = [20 40 100 150 200 300 60 400]
    fprintf('%u,%f\n', f, -20*log10(abs(H(1j*2*pi*f))));
end

pzmap(tf(HN,  HD));  mysaveas('p3-pz', 4, 4);
tfplot('s', HN, HD, [1, 400], 1, 1); mysaveas('p3-bode', 10, 3);
[HNm, HDm] = minphase('s', HN, HD);
tfplot('s', HNm, HDm, [1, 400], 1, 1); mysaveas('p3-bode-minphase', 10, 3);
pzmap(tf(HNm, HDm)); mysaveas('p3-pz-minphase', 2, 4);

diary off
```

Listing 2: The output of listing 1, `diary.txt`.

```
Project 1: Butterworth LPP by hand:

Transfer function:
                         1.965
-------------------------------------------------------
s^5 + 3.704 s^4 + 6.861 s^3 + 7.853 s^2 + 5.556 s + 1.965

Project 1: Butterworth TFs generated by afdgui and by hand:

Transfer function:

                              1.136e24 s^5
------------------------------------------------------------------------
s^10 + 2.095e05 s^9 + 1.009e11 s^8 + 1.465e16 s^7 + 3.59e21 s^6

        + 3.594e26 s^5 + 5.669e31 s^4 + 3.654e36 s^3 + 3.973e41 s^2

                                                    + 1.303e46 s + 9.82e50



Transfer function:

                              1.136e24 s^5
------------------------------------------------------------------------
s^10 + 2.095e05 s^9 + 1.009e11 s^8 + 1.465e16 s^7 + 3.59e21 s^6

        + 3.594e26 s^5 + 5.669e31 s^4 + 3.654e36 s^3 + 3.973e41 s^2

                                                    + 1.303e46 s + 9.82e50



Project 1: Chebyshev I LPP by hand:

Transfer function:
                    0.2457
----------------------------------------------
s^4 + 0.9528 s^3 + 1.454 s^2 + 0.7426 s + 0.2756

Project 1: Chebyshev I TFs generated by afdgui and by hand:

Transfer function:

                              2.512e18 s^4
------------------------------------------------------------------------
s^8 + 5.388e04 s^7 + 6.781e10 s^6 + 2.687e15 s^5 + 1.646e21 s^4

                     + 4.243e25 s^3 + 1.691e31 s^2 + 2.122e35 s + 6.218e40



Transfer function:

                              2.512e18 s^4
------------------------------------------------------------------------
s^8 + 5.388e04 s^7 + 6.781e10 s^6 + 2.687e15 s^5 + 1.646e21 s^4

                     + 4.243e25 s^3 + 1.691e31 s^2 + 2.122e35 s + 6.218e40


Project 1: Chebyshev II LPP by hand:

Transfer function:
       0.07581 s^2 + 2.041
---------------------------------
s^3 + 2.513 s^2 + 3.154 s + 2.041

Project 1: Chebyshev II TFs generated by afdgui and by hand:

Transfer function:

   0.005623 s^8 + 1.379e09 s^6 + 6.407e19 s^4 + 3.44e29 s^2 + 3.497e38
------------------------------------------------------------------------
s^8 + 1.778e05 s^7 + 7.897e10 s^6 + 9.252e15 s^5 + 2.019e21 s^4

                         + 1.461e26 s^3 + 1.969e31 s^2 + 7e35 s + 6.218e40
```

```
Transfer function:

                   4287 s^5 + 5.045e14 s^3 + 1.069e24 s
--------------------------------------------------------------------------
s^6 + 1.421e05 s^5 + 5.746e10 s^4 + 4.857e15 s^3 + 9.074e20 s^2

                                                   + 3.543e25 s + 3.938e30


Project 1: Elliptic LPP by hand:

Transfer function:
       0.04715 s^2 + 0.5152
-----------------------------------
s^3 + 0.9812 s^2 + 1.242 s + 0.5152

Project 1: Elliptic TFs generated by afdgui and by hand:

Transfer function:

                   2666 s^5 + 1.774e14 s^3 + 6.649e23 s
--------------------------------------------------------------------------
s^6 + 5.548e04 s^5 + 5.135e10 s^4 + 1.845e15 s^3 + 8.108e20 s^2

                                                   + 1.384e25 s + 3.938e30



Transfer function:

                   2666 s^5 + 1.774e14 s^3 + 6.649e23 s
--------------------------------------------------------------------------
s^6 + 5.548e04 s^5 + 5.135e10 s^4 + 1.845e15 s^3 + 8.108e20 s^2

                                                   + 1.384e25 s + 3.938e30



Transfer function:
                 s^3
--------------------------------------
s^3 + 251.1 s^2 + 3.153e04 s + 1.98e06

20,0.999988
40,48.325129
100,47.800160
150,-11.000000
200,-11.000000
300,47.802299
60,71.956852
400,65.019183
```