General Stopband Filter Design Report

ECE580 Make-up Project 2: Filter Design
by Arne Bostrom, Rick Crispo and Erin Sullivan
November, 30th 2012

Problem Statement

The purpose of this project is to find the transfer fucntion H(s) of an equal-ripple passband/general purpose stopband lowpass filter. The requirement for the passband attenuation between DC and 1MHz is 0.1dB. The piece-wise constant stopband requirements are over 65dB between 2.13MHz and 3MHz and over 40dB above 3MHz. From these requirements the project seeks to find the optimum location of transmission zeros (loss poles), so that the margins (D_i) between the actual and specified stopband loss responses is the largest possible.

Technology Overview

One of the leading software packages in filter design is the Fitler Design and Analysis Tool, or fdatool, found in MATLAB. This tool is used to quickly design very basic filters. However, given the non-standard requirements of this project, the fdatool in MATLAB will not be sufficient.

The pole-placer algorithm is a great alternative to the fdatool. This algorithm, which is a tool bar add on created at Linköping University [1], is based on the work of B.R. Smith and G.C Temes [2]. The algorithm uses transformed variables, which separates the filter passband and stopband by mapping different parts of the -axis in the s-plane to the real- and imaginary axis in the Zplane.
The relation between the Z-plane and the s-plane is

Eqn.1 : The Relationship between the Z- and S-plane

where ω_A and ω_B are the lower and upper limits of the passband. The algorithm maps transformed variables, such as clustered zeros, further apart. This is particularly valuable because poles in the passband tend to cluster close to the transition region. Compared to the standard MATLAB library (the Signal Processing Toolbox) which uses rational functions in polynomial form, the transformation used by the Pole-Place algorithm enables simpler computational methods and therefore the synthesis of higher order designs.

Another important aspect of this algorithm is that it sets transmission zeros based upon an initial placement input and iterates their position until the margins are *all* as large as possible. As illustrated in Fig.1, this happens when D1 and D2 margin is as close in value to each other as possible.

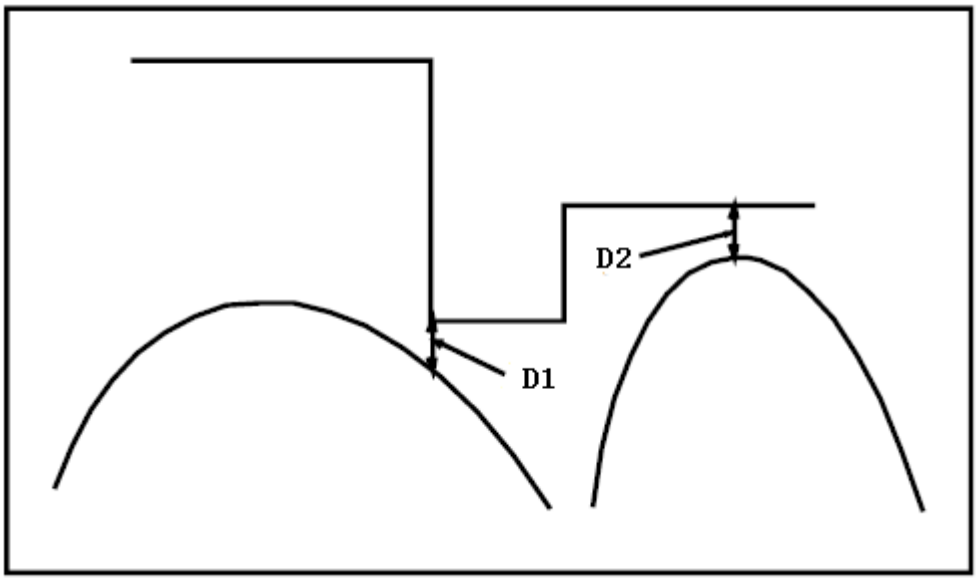

Fig.1: Margin Measurement

For this project, the "POLE_PLACE_LP_EQ_S" algorithm was used based on the given specifications. This algorithm inputs the commanded frequencies and loss requirements and returns the zeros, poles and scaling constant for a lowpass filter with equiripple passband.

Design Results

Applying this algorithm to the design of an equal-ripple/general purpose stopband lowpass filter, then, there are four margins of interest, D1 through D4, which are illustrated in Fig. 2.

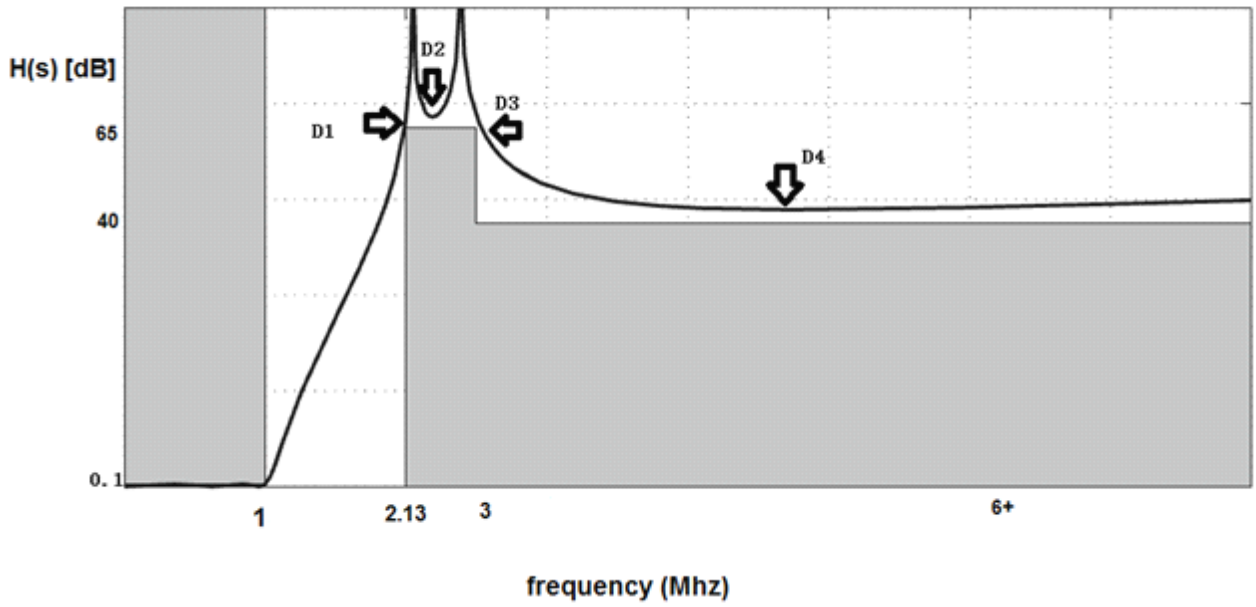

Fig. 2: The Four Margins

The design specification calls for the transmission zeros to be placed such that the margin between the actual and specified stopband loss is the largest possible. The design goal therefore is to make D1, D2, D3, and D4 equal, or close enough to equal so that they are all as large as possible.

Results

The inputs to the pole-placer algorithm are as follows:

While the passband ripple requirement was 0.1 dB, it was found that if Amax was set to 0.1, the actual ripple was higher than the requirement. Thus, Amax was reduced to 0.085dB.

Given the inputs above, the pole placer algorithm was then utilized to find the optimal poles, zeros and the gain of the transfer function, which are given below. Also, Fig. 4 shows a plot of the poles and zeros of the transfer function on the real and imaginary axis

The Poles of the Transfer function

The Zeros of the Transfer Function

The Gain of the Transfer Function

Thus, the Transfer Function is:

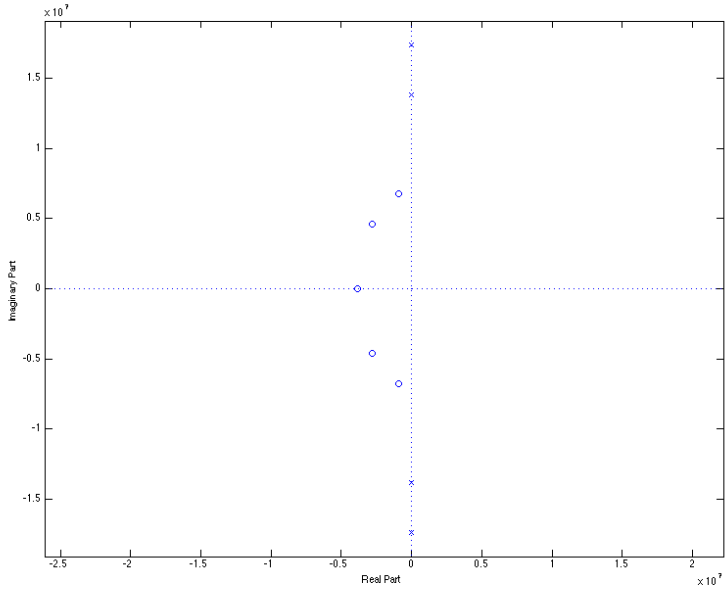

Fig.4: Pole/Zero Plot

From this result, the attenuation can be plotted as a function of frequency and can be seen in Fig. 5. Also, Fig. 6 shows a close up of the ripple in the passband from DC to 1MHz. From this attenuation plot, the four margins of interest are found and displayed after Fig. 6. It can be observed that D1, D2 and D3 are all equally maximized, while D4 is much greater than the other three. Therefore, the optimal filter has been created for this project. The MATLAB code for this project can be seen in the appendix section after the references

section.

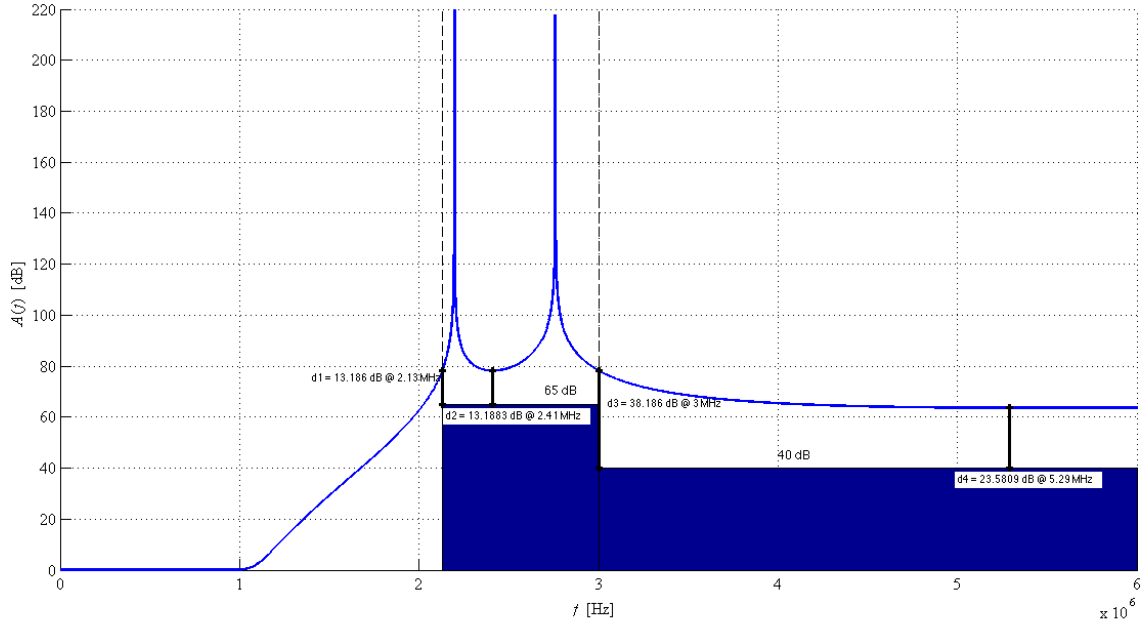

Fig. 5: Attenuation plot of the Transfer Function

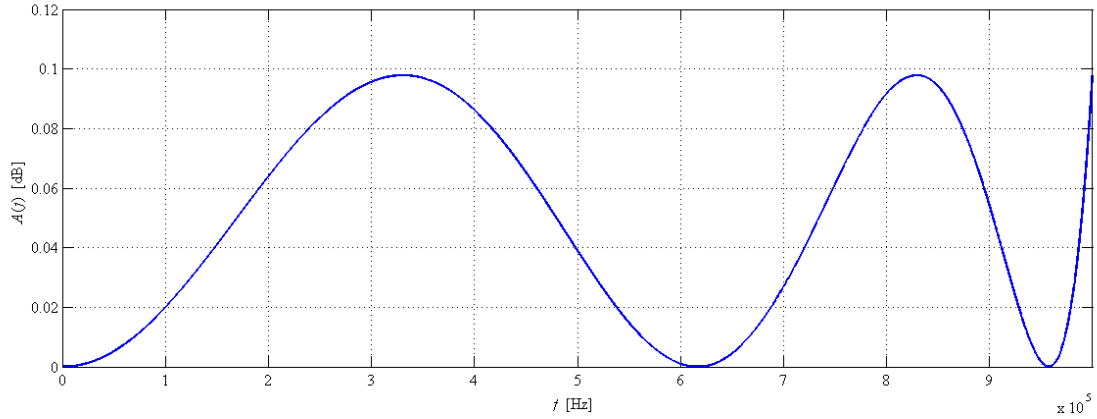

Fig.6: Passband Ripple Plot

Result of the four Margins

Verification of Algorithm

To verify that the pole placer algorithm actually produced the optimal filter given the break frequencies, a sweep of the upper bound (f_{upper}) and lower bound (f_{lower}) of the break frequencies (f_{step}=[f_{lower} f_{upper}]) were carried out using MATLAB. The lower bound frequency was varied from 2MHz to 2.2MHz and the upper bound frequency was varied from 2.7MHz to 3.3MHz. Both were incremented by 10kHz steps. Fig. 7 shows the comparison of the optimal break frequencies (fstep=[2.13MHz 3MHz]) in green to that of the lower bound break frequencies (fstep=[2.0MHz 2.7MHz]) in blue. Notice how d_1 and d_2 are significantly larger than d_3. This shows that not all margins are optimized.

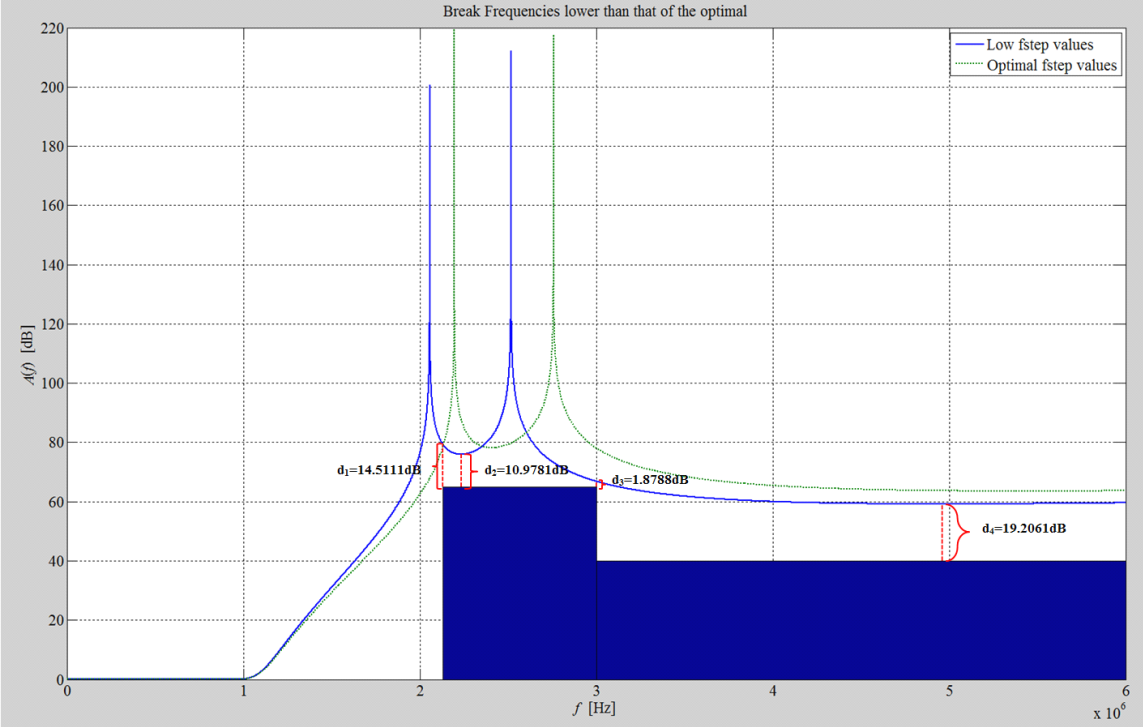

Fig. 7: Comparative plot of the optimal break frequencies in green (fstep=[2.13MHz 3MHz]) and lower break frequencies in blue (fstep=[2.0MHz 2.7MHz]).

The same can be said for break frequencies that are higher than the optimal. Fig. 8 shows the comparison of the optimal break frequencies shown in green to that of the higher bound frequencies (fstep=[2.2MHz 3.3MHz]). From this graph, d_3 and d_2 are now significantly greater than d_1. Again, the margins are not optimized. These two graphs prove that break frequencies at 2.13MHz and 3MHz do in fact produce the optimal filter.

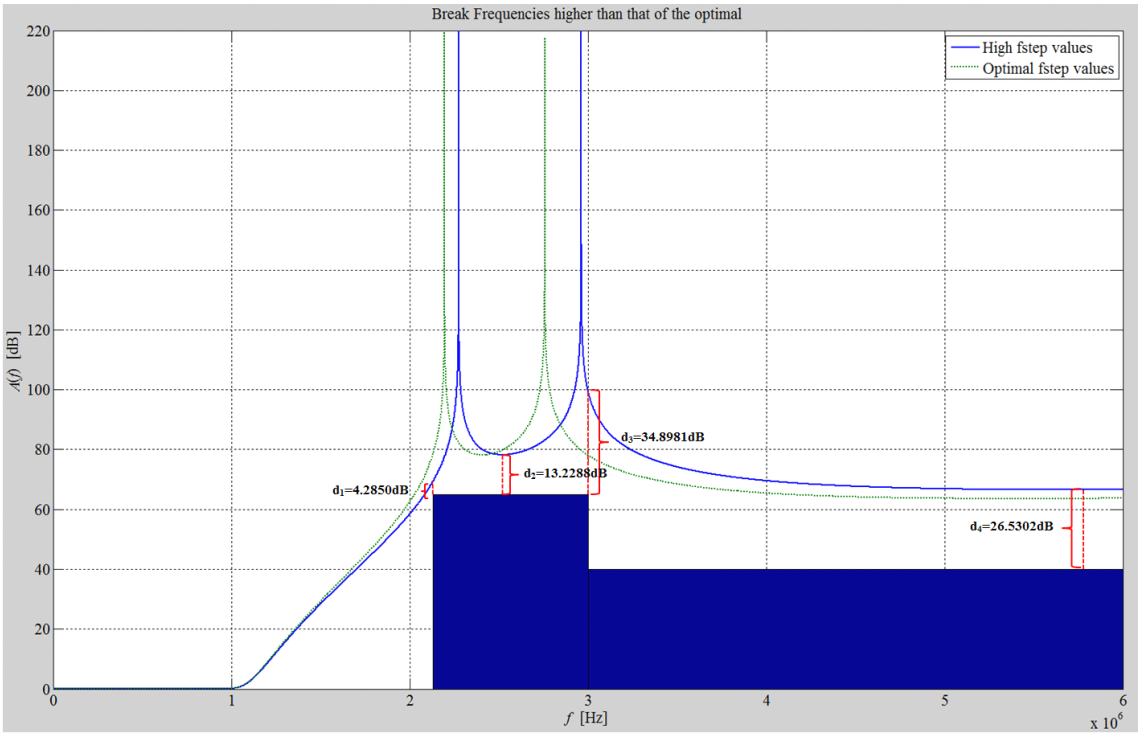

Fig. 8: Comparative plot of the optimal break frequencies in green (fstep=[2.13MHz 3MHz]) and higher break frequencies in blue (fstep=[2.2MHz 3.3MHz]).

References

[1] **Per Löwenborg, Oscar Gustafsson, and Lars Wanhammar.** "Filter Design Using MATLAB." Department of Electrical Engineering, Linköping University, Linköping Sweden, 1999.

[2] **B.R. Smith and G.C. Temes.** "An Iterative Approximation Procedure for Automatic Filter Design". *IEEE Trans. Circuit Theory*, Vol. 12, pp. 107-112, Mar. 1965.