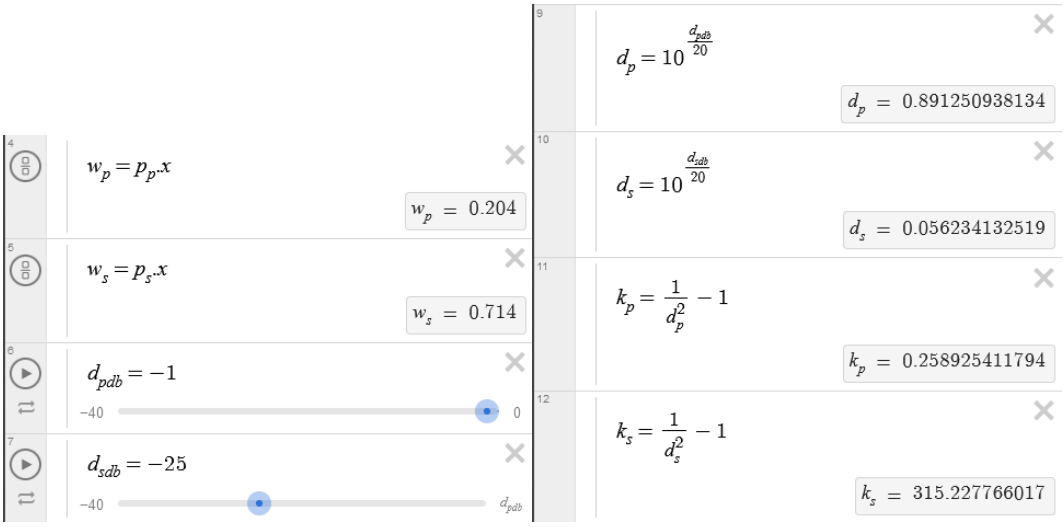
Signal Processing Report

Problem Description

We are tasked with designing a low-pass filter to filter high-frequency noise from a signal. This process includes choosing $\delta_s, \delta_p, \omega_s, \omega_p$ to calculate the order N, and ω_c. It is given that $\delta_s = -25dB \; and \; \delta_p = -1dB$. Additionally, $N \leq 10$. These values of N and ω_c should then be used to calculate A and B for a difference equation. These values for A and B should be used to rerun the microcontroller code to test the results.

Filter Design Calculations

Because I did not want to risk the possibility of getting a value of N above 10, I implemented the calculations in Desmos. I made https://www.desmos.com/calculator/mkug4xf6ef. This way I could adjust ω_s *and* ω_p until I get good values of N and ω_c. I chose my value of ω_p such that the signal at 10Hz is just in the pass-band and I chose the value of ω_s the be large enough that N is low but not so much that the signal at 203Hz isn't filtered. The 10Hz signal occurs at $2\pi\frac{10}{500} \rightarrow w = 0.125$ and the 203Hz signal occurs at $2\pi\frac{203}{500} \rightarrow w = 0.812$. As you can see my value for ω_p is 0.204 and a bit larger than 0.125. Finally, my value for ω_s is 0.714 and a bit smaller than 0.812.



N = 3, Oc = 0.2555

A & B Coefficients

I implemented this using code so I could calculate the values for A and B on the fly without tedious work on paper. This is in case my initial values don't filter the noise well. I followed the MATLAB code given and used it to extract the values for A and B.

I will note that for some reason my c2d() function provides slightly different results than from the instructions. My tf() function provides the same continuous polynomial ratio but, when converting to a digital version, the denominator is the same as shown but my numerator is slightly different. This can be due to many things, but the results are still mostly correct.

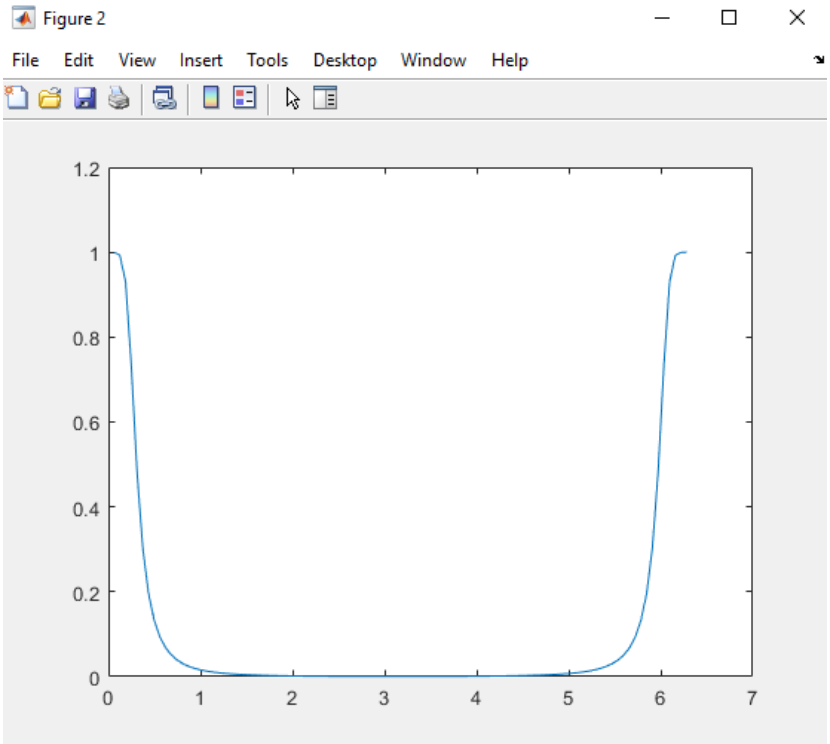
When N is large, the A and B lists still filters as expected, but the magnitude of the resulting signal is increased. This might be due to precision errors or due to the different results from c2d() on my computer. Either way, I decided to keep N within a reasonable range (N<4) to have reasonable results. I tried multiple ways to allow N to be large and to have it still work, but to no avail.

I'll include my MATLAB file in case c2d() works differently on your computer.

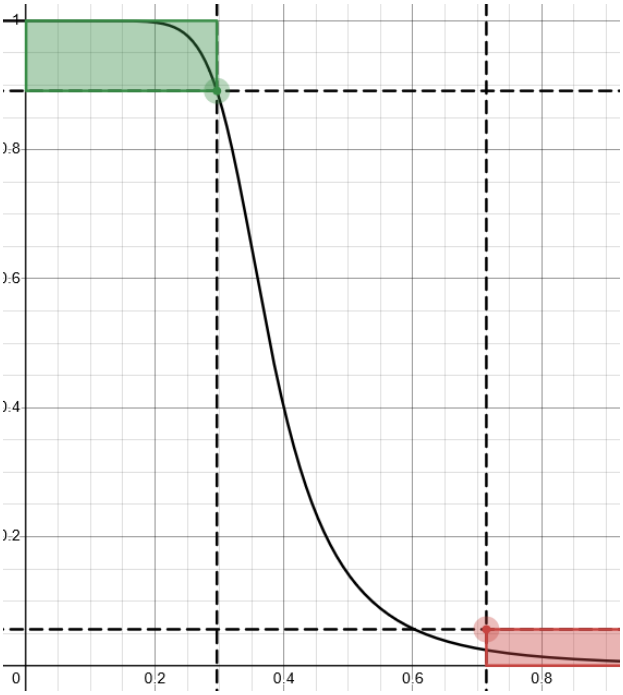
A = 1.0000 -2.4918 2.1046 -0.5999

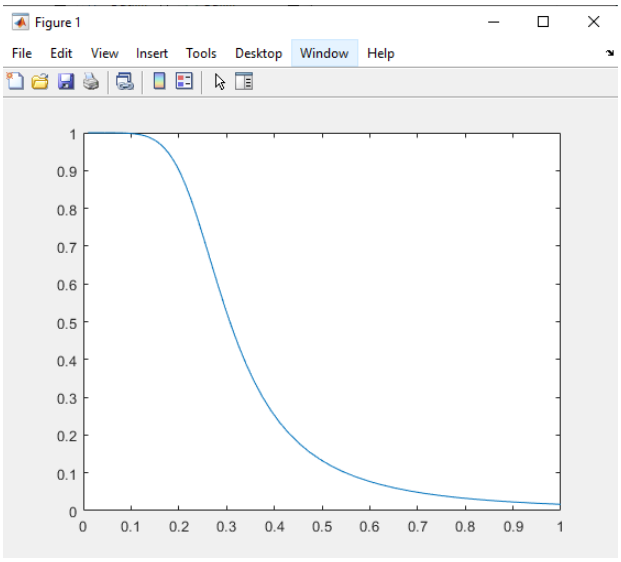
B = 0 0.0024 0.0086 0.0019

Abs(H) Plot



My Desmos code shows the Butterworth function with the filter boxes overlapped.





Main.c

Same as previous assignment but with different values for A and B. File is included in the zip.

Microcontroller Plots