SSY130 - Hand in Problem 2

Alfred Aronsson

December 1, 2023

Student ID: 20000729, Secret Key: 'Swinub'

1

```
function h = gen_filter()
    % Define parameters
    Dt = 1; % Sampling interval
    fs = 1/Dt; % Sampling frequency (Hz)
    passband_freq = 0.05/(fs/2); % Normalized passband frequency (Hz)
    stopband_freq = 0.1/(fs/2); % Normalized stopband frequency (Hz)


    N = 60; % +1 Filter order
    freqs = [0, passband_freq, stopband_freq, 1];
    mags = [0, passband_freq*pi, 0, 0];

    h = firpm(N, freqs, mags, 'differentiator');
end

funs.gen_filter = @gen_filter;
```

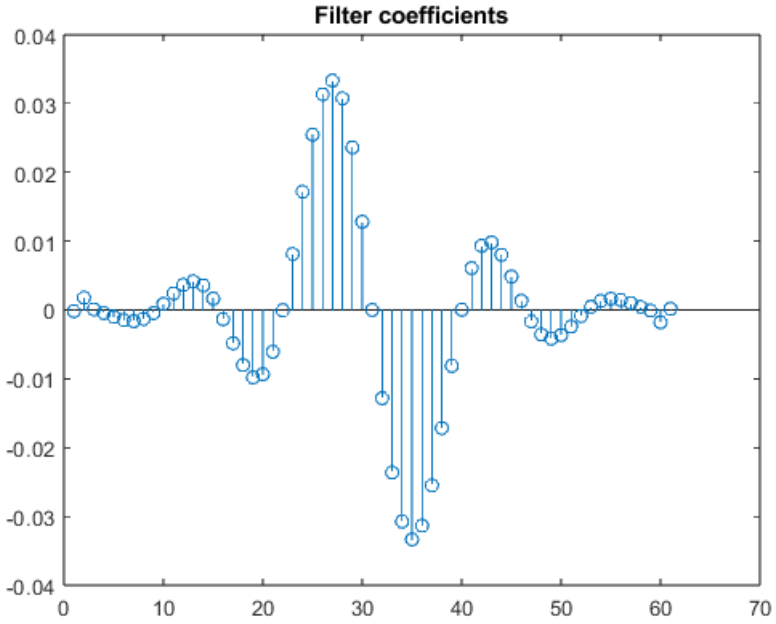
This is the code I wrote to generate the FIR-filter. $Dt = 1$ is given in the problem definition, as is the passband frequency and the stopband frequency. The difference is that I have normalized them. The problem definition also states that the filter should have 61 filter coefficients and that in Matlab the convention is that a filter of order N has $N + 1$ filter coefficients. Therefore I assigned N to be 60. Next I defined the list freqs, which is a list that specifies the frequency band edges of the filter. 0 corresponds to the starting frequency, passbandfreq corresponds to the passband frequency, stopbandfreq corresponds to the stopband frequency and 1 corresponds to the Nyquist frequency (since the band frequencies are already normalized). The list mags is a lists that specifies the magnitudes of the filter for each frequency in freq. We know that:

$$|H(f)| = \frac{f}{f_s} \cdot \frac{f_s}{2} \cdot 2\pi = f\pi, \text{ for } f \leq 0.05$$
$$|H(f)| = 0, \text{ for } f > 0.1$$

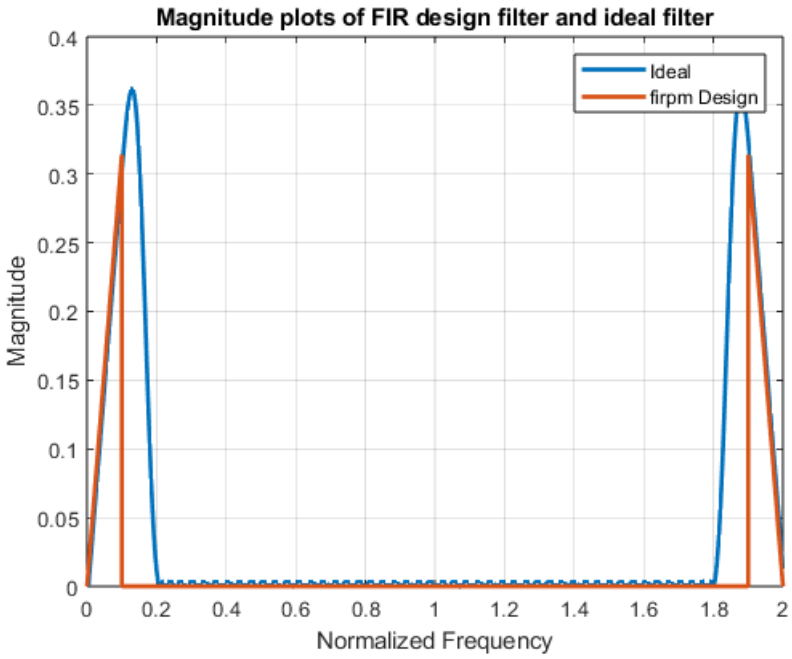
From this we can determine that the amplitude for the starting frequency will be 0, and $f_{pb}\pi$ for the passband frequency, and 0 for the stopband and Nyquist frequency. As sugested in the problem description i have also included 'differntiator' as a last argument in firpm to ensure that the filter has a phase of $+90°$.

2

In the figure below we find the stem-plot of the filter coefficients of the FIR filter designed in the previous question:



And in the next figure the frequency response of the designed FIR filter and an ideal filter will be shown:



It should be noted that the frequencies are plotted using a frequency that is normalized with the Nyquist frequency.

3

All filters phase shift the signal that is being filtered. In the time domain this phase shift corresponds to time delay. To determine the time delay of the filter we can consult the lecture notes, which on page 47 states that the formula for calculating the time delay of a symmetric FIR filter is the following:

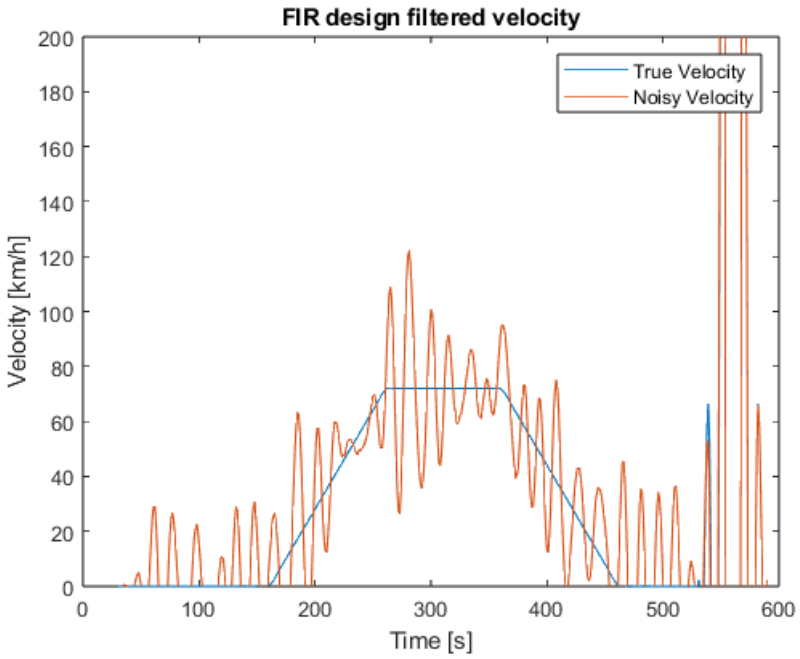
$$\text{Delay} = \Delta t(M-1)/2, \text{ where } M \text{ is the filter length.}$$

Our filter has 61 coefficients and and has as such filter length 61. It is also known that our Δt is one second. The time delay is calculated to be:

$$1 \cdot (61-1)/2 = 30s$$

4

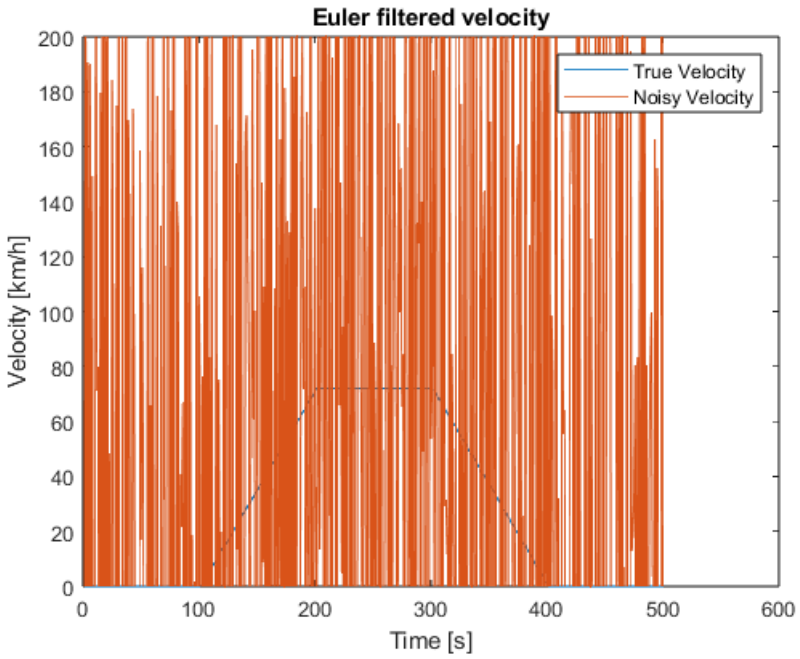
In the figure below the velocities corresponding to **true_position** and **noisy_position** are plotted:



The reason for the aggressive oscillations at the end of the plot is that the filtered data and the filter has different lengths. The data has a length of 500 whilst the filter has a length of 61. This means that the filter can not be neatly convoluted with the data. As a result of this the data is appended with a bunch of zeros to make up for the difference in length. This however leads to that the filter "thinks" that the appended zeros are part of the actual data. As the last real data point is 4000, it is an enormous shift to the appended values which are zeros. The filter interprets this as noise and tries to suppress it, which is why the oscillations occur.

5

When filtering the data through the Euler filter defined in the problem description instead of the FIR filter we get the following results:



The Euler filter is, as stated in the problem definition, a trivial filter. As such it can't suppress noisy signals well at all, which can be seen. It can however filter the true signal well. It can be noted that since the Euler signal has a length of two, the data does not have to be appended with zeros. Which means that the Euler filter does not produce the enormous oscillations that the FIR filter produces.

6

The maximum speed of the vehicle for the noisy and true signal can be found by using the following Matlab commands:

```
max(abs(diff(true_position)))*3.6
max(abs(diff(noisy_position)))*3.6
```

Which grants the results:

$$v_{max,true} = 72 \text{ km/h}$$
$$v_{max,noisy} = 805 \text{ km/h}$$