Digital FIR Filter Design

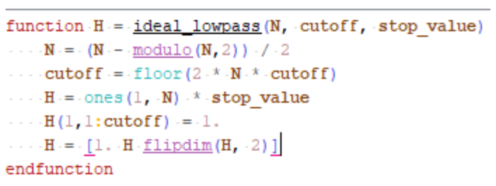
Aidar Ahmetshin

April 2020

1 Task I

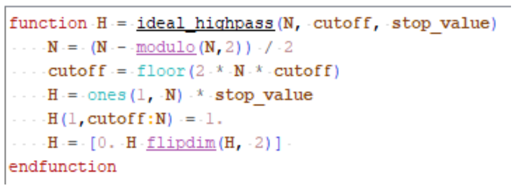
1.1 Steps

First thing that we should do it implement low pass and high pass filters or together it will be band pass FIR filter that will reduce noise in low and high frequencies.



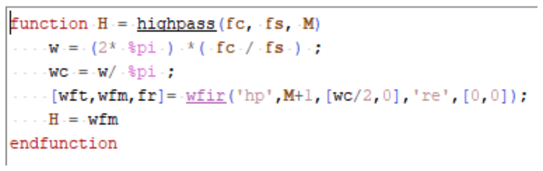
```
function H = ideal_lowpass(N, cutoff, stop_value)
    N = (N - modulo(N,2)) / 2
    cutoff = floor(2 * N * cutoff)
    H = ones(1, N) * stop_value
    H(1,1:cutoff) = 1.
    H = [1. H flipdim(H, 2)]
endfunction
```

Figure 1: Low-pass filter



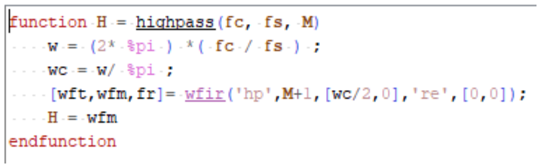
```
function H = ideal_highpass(N, cutoff, stop_value)
    N = (N - modulo(N,2)) / 2
    cutoff = floor(2 * N * cutoff)
    H = ones(1, N) * stop_value
    H(1,cutoff:N) = 1.
    H = [0. H flipdim(H, 2)]
endfunction
```

Figure 2: High-pass filter



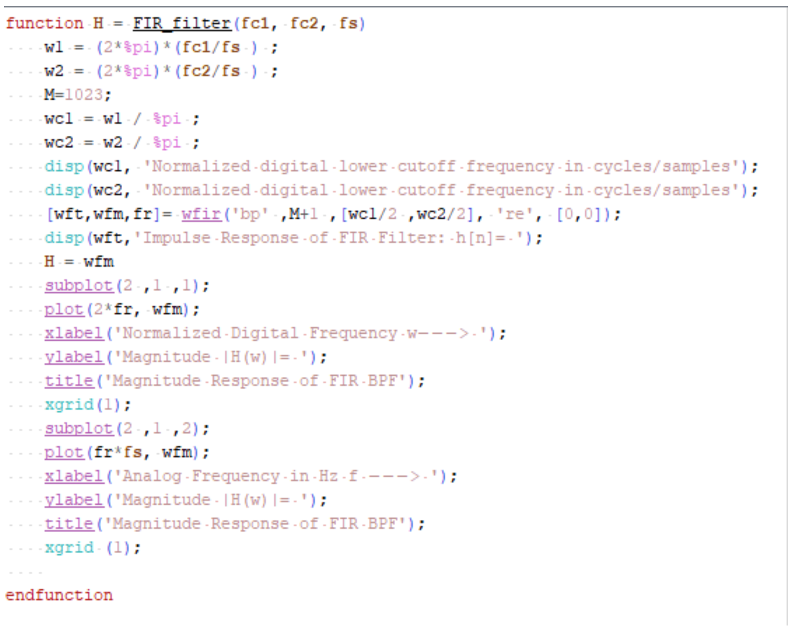
```
function H = highpass(fc, fs, M)
    w = (2* %pi ) *( fc / fs ) ;
    wc = w/ %pi ;
    [wft,wfm,fr]= wfir('hp',M+1,[wc/2,0],'re',[0,0]);
    H = wfm
endfunction
```

Figure 3: High-pass filter



```
function H = highpass(fc, fs, M)
    w = (2* %pi ) *( fc / fs ) ;
    wc = w/ %pi ;
    [wft,wfm,fr]= wfir('hp',M+1,[wc/2,0],'re',[0,0]);
    H = wfm
endfunction
```

Figure 4: Low-pass filter



```
function H = FIR_filter(fc1, fc2, fs)
    w1 = (2*%pi)*(fc1/fs ) ;
    w2 = (2*%pi)*(fc2/fs ) ;
    M=1023;
    wc1 = w1 / %pi ;
    wc2 = w2 / %pi ;
    disp(wc1, 'Normalized digital lower cutoff frequency in cycles/samples');
    disp(wc2, 'Normalized digital lower cutoff frequency in cycles/samples');
    [wft,wfm,fr]= wfir('bp' ,M+1 ,[wc1/2 ,wc2/2], 're', [0,0]);
    disp(wft,'Impulse Response of FIR Filter: h[n]= ');
    H = wfm
    subplot(2 ,1 ,1);
    plot(2*fr, wfm);
    xlabel('Normalized Digital Frequency w---> ');
    ylabel('Magnitude |H(w)|= ');
    title('Magnitude Response of FIR BPF');
    xgrid(1);
    subplot(2 ,1 ,2);
    plot(fr*fs, wfm);
    xlabel('Analog Frequency in Hz f ---> ');
    ylabel('Magnitude |H(w)|= ');
    title('Magnitude Response of FIR BPF');
    xgrid (1);
    
endfunction
```

Figure 5: Band-pass filter

We have two opportunities how to design: 1) do it like it was suggested in work description 2) use special feature from scilab called wfir - function which makes linear-phase, FIR low-pass, band-pass, high-pass, and stop-band filters using the windowing technique.

Second thing, that we should pay attention to shift our filter frequency response. The fact that the impulse response does not decay to 0 make this FIR filter impractical. The common solution is to apply a window function.

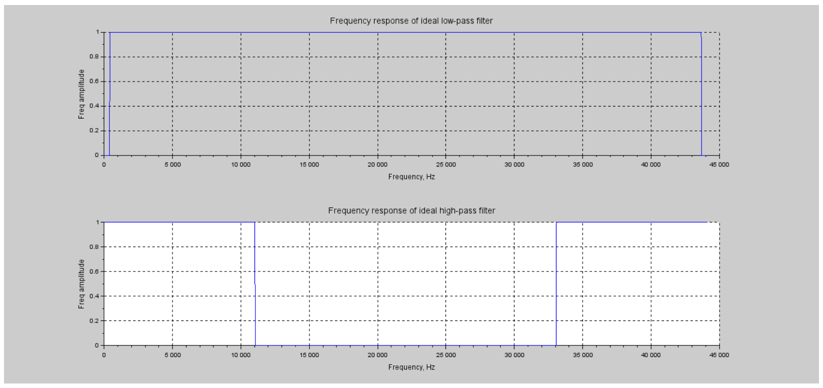


Figure 6: Filters

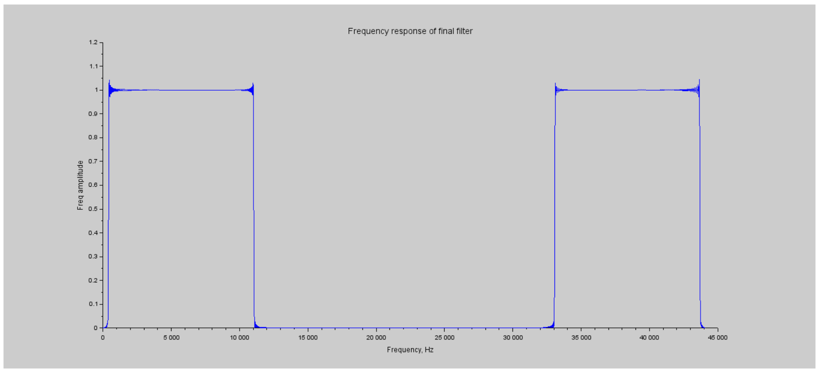


Figure 7: Filter response

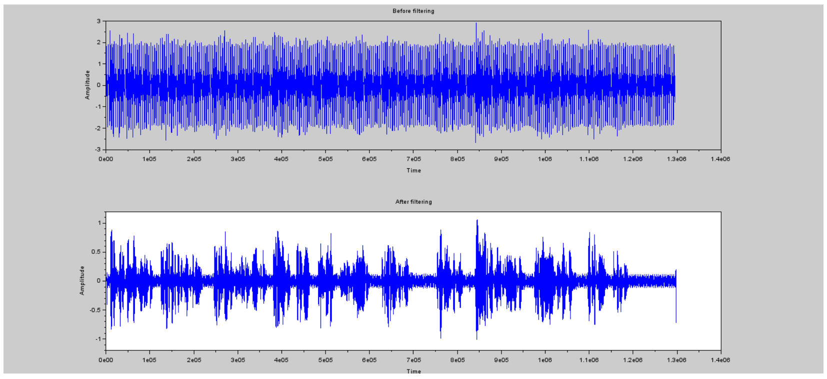


Figure 8: Results

1.2 Results

2 Task II

2.1 Steps

We should generate inverse filter for some IRC or we can called it effect, that based on itself, will help us remove some noise added by IRC. Steps for creating inverse filter are: 1)analyze in which domain——limits lays interested us 'not useful data'. Than we should shift it and apply window function to decay to zero.

2.2 Results

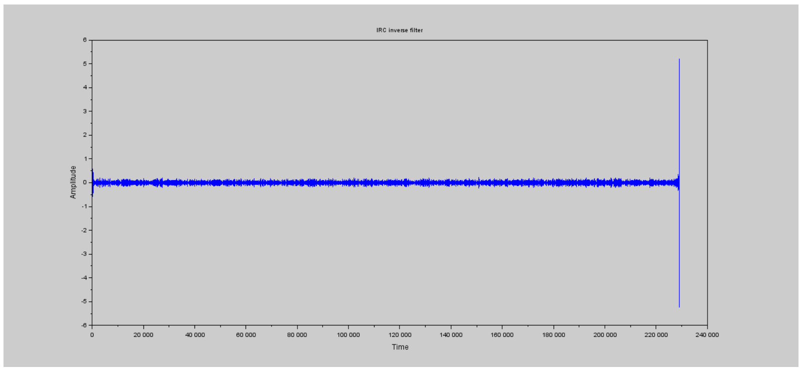


Figure 9: Filter for my IRC

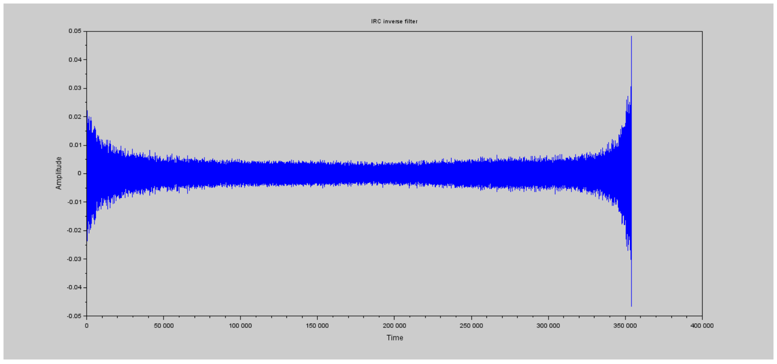


Figure 10: Filter for Marble hall

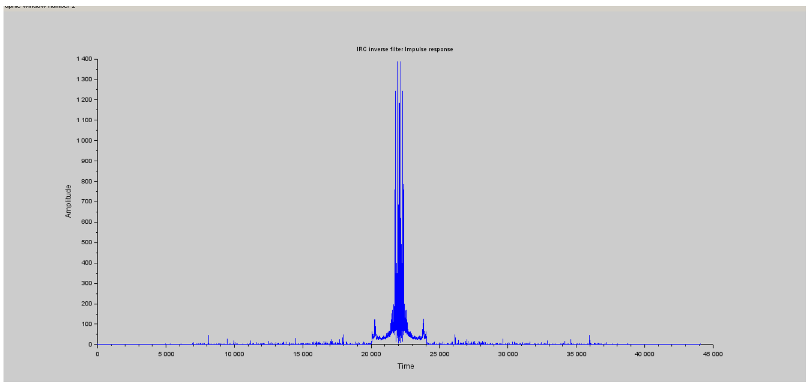


Figure 11: IRC inverse filter Impulse response

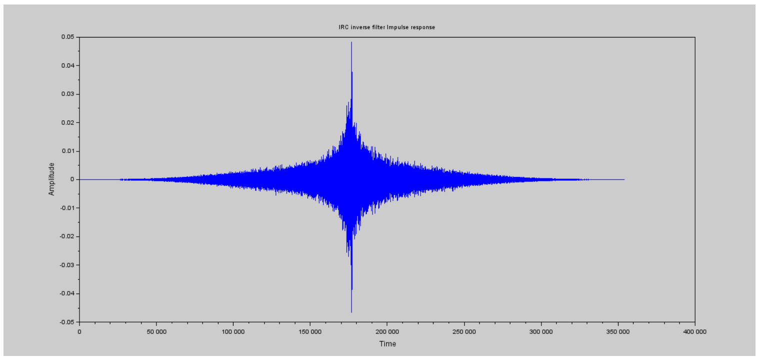


Figure 12: Filter after applying window

3 References

Git Hub repo