UNIVERSITY OF MORATUWA, SRI LANKA

Faculty of Engineering

Department of Electronic and Telecommunication Engineering

Semester 3 (Intake 2020)

EN2063—SIGNALS AND SYSTEMS

Filter Design Project

Name: A.A.H. Pramuditha

Index No.: 200476P

1. FIR digital bandpass filter design using Kaiser Window method.

When using the Kaiser window method to design FIR (Finite Impulse Response) digital bandpass filter, we need to specify two main parameters. They are,

1. Shape parameter (β)
2. Length of the window (N+1; N is the order)

Design Parameters:

- Maximum passband ripple (˜A_p) = 0.14 dB
- Minimum stopband attenuation (˜A_a) = 57 dB
- Lower passband edge (Ω p) = 1000 rad/s
- Upper passband edge (Ω p) = 1500 rad/s
- Lower stopband edge (Ω s) = 700 rad/s
- Upper stopband edge (Ω s) = 1700 rad/s
- Sampling frequency (Ω sm) = 4200 rad/s

Design steps:

1. Determining a delta (δ) value which is allowable for passband ripples and stopband attenuations.

$$\delta\,(s) = 10^{-0.05Aa}$$

$$\delta\,(p) = \frac{10^{0.05Ap}-1}{10^{0.05Ap}+1}$$

$$\delta = \min\{\delta\,(s), \delta\,(p)\}$$

2. Calculating the actual attenuation based on this new (δ) value.

$$A = -20\log_{10}(\delta)$$

3. Calculating the parameter (β) based on the following conditions to determine the shape of the window.

$$\beta = 0.1102\,(A - 8.7) \text{ if } A > 50$$

$$\beta = 0.5842\ (A - 21)^{0.4} + 0.07886\ (A - 21) \text{ if } 21 <= A <= 50$$

$$\beta = 0 \text{ if } A<21$$

4. Calculating the length (M) of the filter.

$$M = \frac{A-8}{2.285\Delta\omega}$$ where $\Delta\omega$ is the transition bandwidth. ($\Delta\omega = \omega p - \omega s$)

5. Calculating the FIR filter coefficients using following equations.

$$h(n) = h_d(n) \times \omega(n)$$

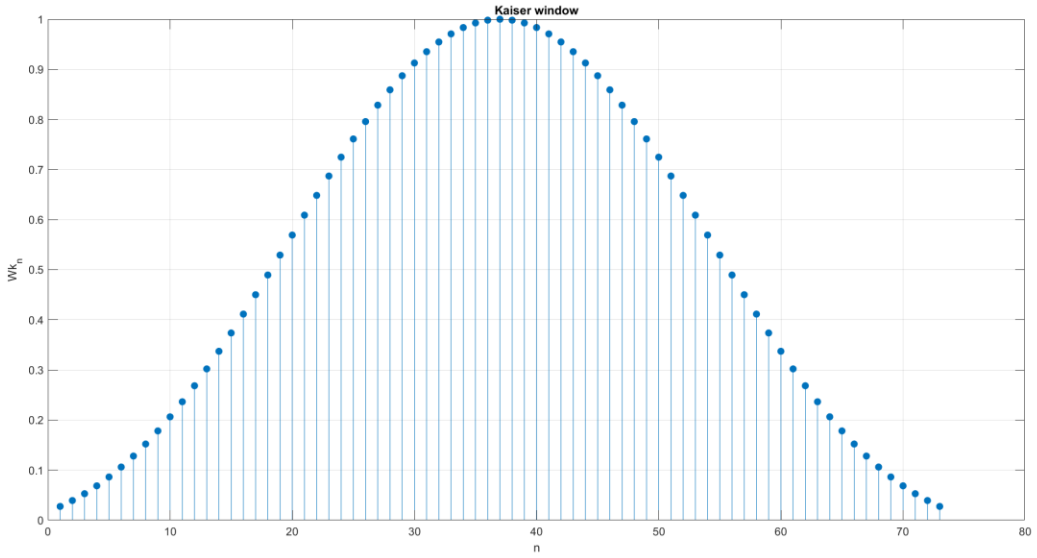
where $h_d(n)$ is the ideal impulse response of the filter and ω_n is the Kaiser window function.

$$\omega_k(n) = \frac{Io[\beta\left(\left(1-\left(\frac{n-\alpha}{\alpha}\right)^2\right)\right)^{\frac{1}{2}}]}{Io(\beta)}$$; 0<= n <= M, where $\alpha = M/2$.

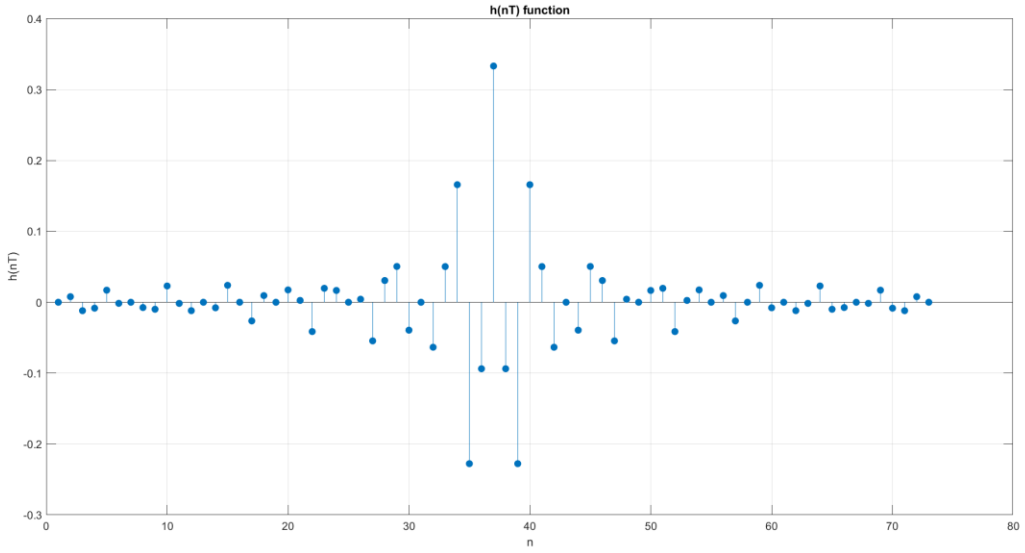
$$\omega_k(n) = 0; \text{ otherwise}$$

$$h(n) = \frac{\sin\left(n-\left(\frac{M-1}{2}\right)\right)}{\pi\left(n-\left(\frac{M-1}{2}\right)\right)}$$; $n \neq \frac{M-1}{2}$ and

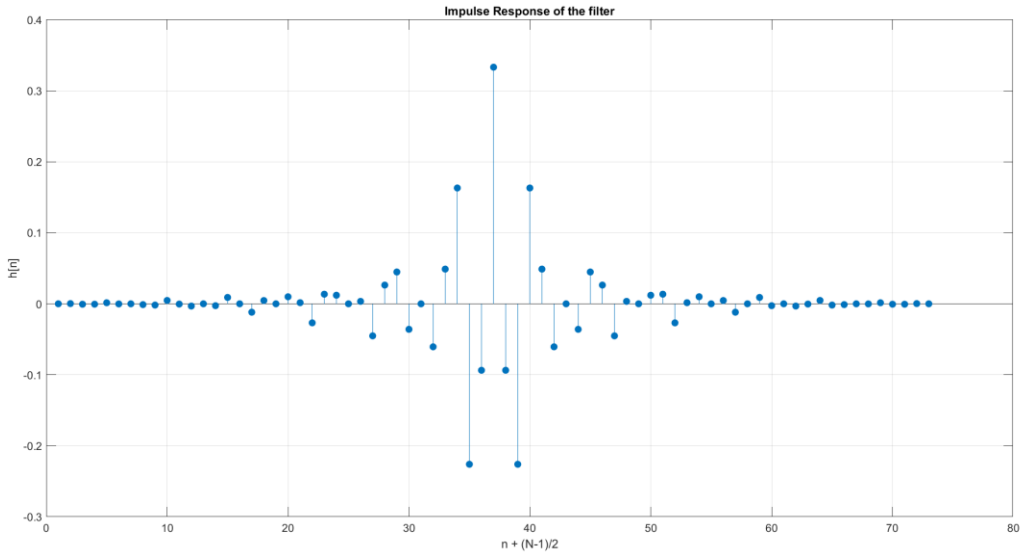
$$h(n) = \frac{\omega c}{\pi}$$; $n = \frac{M-1}{2}$



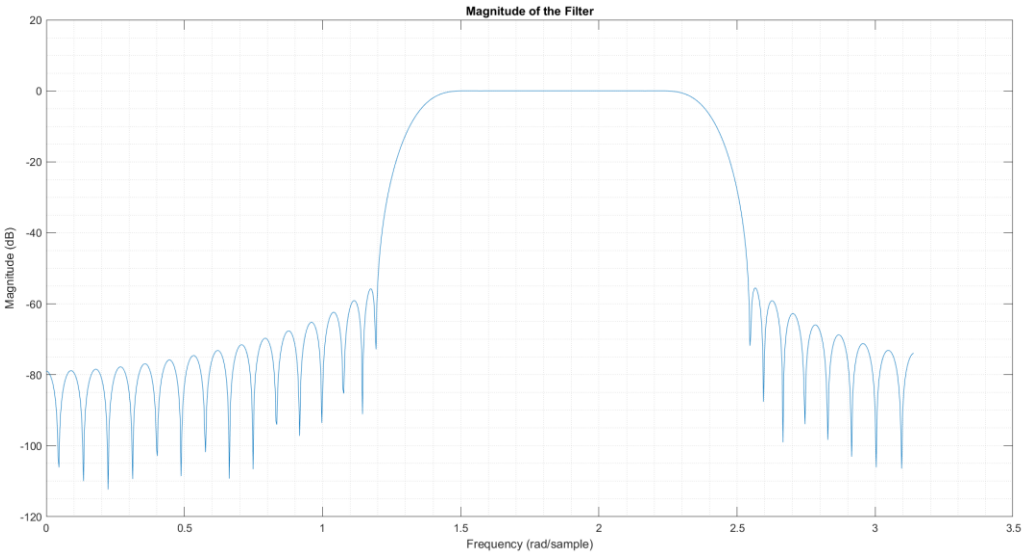
Kaiser Window Function



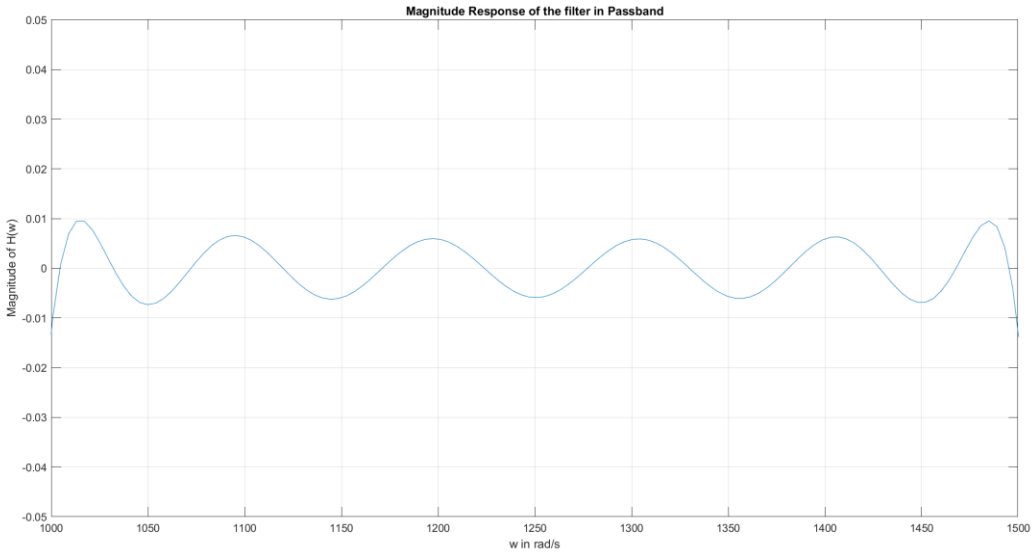
Ideal Impulse Response Function



Impulse Response of FIR Bandpass Filter



Magnitude Response of FIR Filter



Magnitude Response of FIR Filter in Passband. (ωp1 ≤ ω ≤ ωp2)

2. IIR digital bandpass filter design using Chebyshev type II approximation.

Design Parameters:

- Maximum passband ripple (˜A_p) = 0.14 dB
- Minimum stopband attenuation (˜A_a) = 57 dB
- Lower passband edge (Ω p) = 1000 rad/s
- Upper passband edge (Ω p) = 1500 rad/s
- Lower stopband edge (Ω s) = 700 rad/s
- Upper stopband edge (Ω s) = 1700 rad/s
- Sampling frequency (Ω sm) = 4600 rad/s

When designing an IIR (infinite impulse response) filter, first we need to figure out what design parameters our analog filter should have. To calculate this, we use the frequency pre-warping technique.

After that, we design our analog filter according to those specifications and synthesize the frequency response in the Laplace domain.

Then we use the bilinear transformation method to transform it to the Z domain.

Transfer function coefficients:

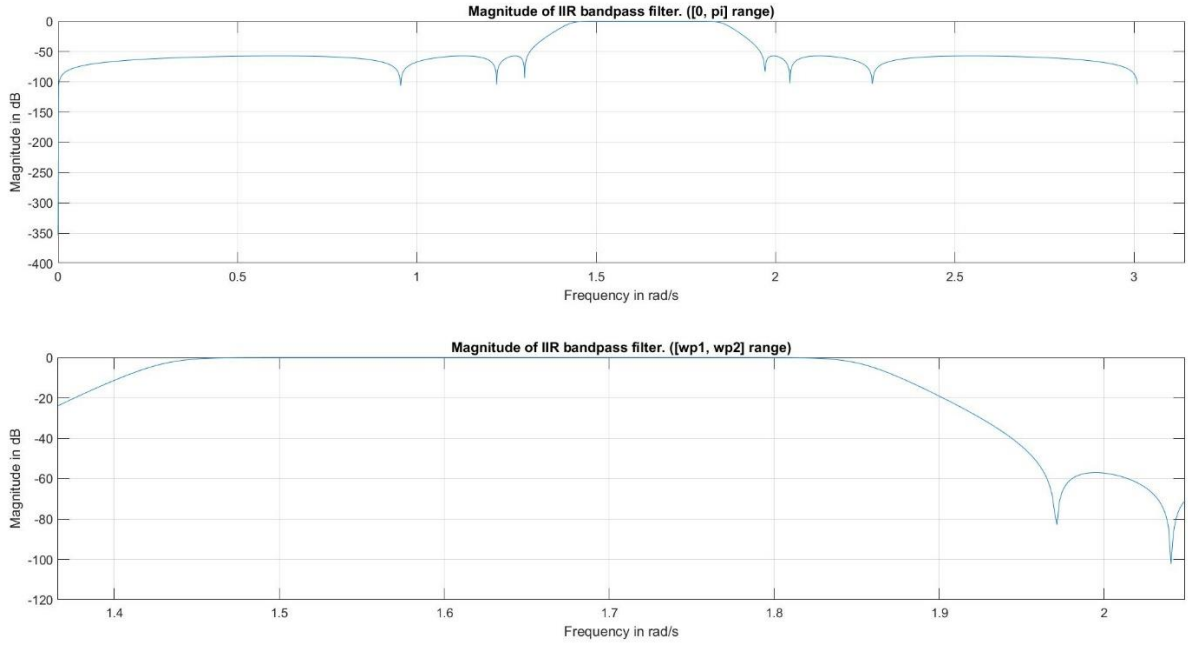
Numerator Coefficients	Denominator Coefficients
0.0027	1.0000
0.0036	1.7149
0.0082	6.1325
0.0083	7.7468
0.0118	15.0151
0.0080	14.4812
0.0052	19.3596
-0.0000	14.3035
-0.0052	14.2743
-0.0080	7.8576
-0.0118	6.0077
-0.0083	2.2712
-0.0082	1.3242
-0.0036	0.2691
-0.0027	0.1154

- Zeros of the system:

  0.0000 + 1.0257i
  0.0000 - 1.0257i
  0.0000 + 1.2790i
  0.0000 - 1.2790i
  0.0000 + 2.3048i
  0.0000 - 2.3048i

- Poles of the system:

  0, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, -0.0000, 0.0000, -0.0000, 6.6983, 0



Magnitude Responses of IIR Filter

3. Comparison between FIR and IIR filters.

Based on the given specifications, the order of the IIR filter is 7 and the order of the FIR filter is 73.

The number of multiplications and the number of additions done by the FIR filter are 74 and 73, respectively. But the number of multiplications and the number of additions done by the IIR filter are 15 and 14.

So, we can say that IIR filters can be implemented with a lesser number of computations than FIR filters for a given set of specifications.

4. Appendix

FIR Filter Implementation in MATLAB.

```
%%Pramuditha A.A.H. (200476P)
%%EN2063 - Signals and Systems
%%Digital Filter Design - Semester 03

clc;
close all;

%%Specifying design parameters.
Ap = 0.14;          %maximum passband ripple in dB.
As = 57;            %minimum stopband attenuation in dB.
wp1 = 1000;         %lower passband edge.
wp2 = 1500;         %upper passband edge.
wa1 = 700;          %lower stopband edge.
wa2 = 1700;         %upper stopband edge.
ws = 4200;          %sampling frequency.
T = 2*pi/ws;        %Sampling period.

%%Selcting the transition bandwidth and cutoff frequencies.
tran_bw = min(wp1-wa1, wa2-wp2);
wc1 = wp1 - tran_bw/2;
wc2 = wp2 + tran_bw/2;

%%Defining delta value using Ap and As.
delta_p = ((10^(0.05*Ap))-1)/((10^(0.05*Ap))+1);
delta_s = 10^(-0.05*As);
delta = min(delta_p, delta_s);

%%Calculating the actual stopband attenuation of the
filter.
Atten = -20*log10(delta);

%%Designing kaiser window function. Determining alpha
parameter.
if Atten<=21
   alpha = 0;
elseif Atten<=50
   alpha = (0.5842*((Atten-21)^0.4)) + (0.07886*(Atten-
21));
else
   alpha = 0.1102*(Atten - 8.7);
```

```
end

%%Selecting parameter D based on actual attenuation.
if Atten<=21
   D = 0.9222;
else
   D = (Atten - 7.95)/14.36;
end

%%Selecting the proper odd value for N which satisfies the
inequality
%%N>=(ws*D/tran_bw)+1.
N = ceil((ws*D/tran_bw)+1);
if mod(N,2) == 0
   N=N+1;
end

%%Forming Kaiser window wk[n].
Wk_n = zeros(N,1);
for n = -(N-1)/2:(N-1)/2
   beta = alpha * (1 - (2*n/(N-1))^2)^0.5;
   numerator = Bessel_Func(beta);
   denominator = Bessel_Func(alpha);
   Wk_n(n+(N-1)/2+1) = numerator/denominator;
end

%%Plotting Kaiser window.
stem(Wk_n, 'filled');
title('Kaiser window');
xlabel('n');
ylabel('Wk_n');
grid on;

%%Generating h(nT).
h_n = zeros(N,1);
h_n(38) = (2/ws)*(wc2-wc1);
for n = -(N-1)/2:(N-1)/2
   if n==0
      h_n(n+(N-1)/2+1) = (2/ws)*(wc2-wc1);
   else
      h_n(n+(N-1)/2+1) = (1/(n*pi)) * (sin(wc2*n*T) -
sin(wc1*n*T));
   end
end
```

```
figure;

%%Plotting h(nT) function.
stem(h_n, 'filled');
title('h(nT) function');
xlabel('n');
ylabel('h(nT)');
grid on;

%%Getting the final digital filter.
filter = h_n.*Wk_n;

%%Plotting the impulse response of the filter.
figure
stem(filter, 'filled');
title('Impulse Response of the filter');
xlabel('n + (N-1)/2');
ylabel('h[n]');
grid on;

%%Magnitude Response of the Filter
figure;
[H, f] = freqz(filter, 1, 1024, ws);
plot(f.*(2*pi/ws),(mag2db(abs(H))))
title('Magnitude of the FIR filter')
xlabel('Frequency')
ylabel('Magnitude')
grid on

%%Plotting the magnitude response of the passband.
[Amplitude, H] = freqz(filter);
analog_fr = H*ws/(2*pi);
amp_db = 20*log10(abs(Amplitude));
figure
plot(analog_fr, amp_db);
axis([wp1 wp2 -0.05 0.05]);
title('Magnitude Response of the filter in Passband');
xlabel('w in rad/s');
ylabel('Magnitude of H(w)');
grid on;

%%Introducing Bessel Function to obtain zeroth order for
modified bessel
%%function of first kind.
```

```
function [value] = Bessel_Func(x)
    k=1;
    value=0;
    i = 10;
    while (i>10^(-6))
        i = (((x/2)^k)/(factorial(k)))^2;
        value = value + i;
        k = k+1;
    end

    value = value+1;
end
```

IIR Filter Implementation in MATLAB.

```
clc;
close all;
clear;

%Specifying design parameters.
Ap = 0.14;       %Maximum passband ripple.
As = 57;         %Minimum stopband attenuation.
wp1 = 1000;      %Lower passband edge.
wp2 = 1500;      %Upper passband edge.
ws1 = 700;       %Lower stopband edge.
ws2 = 1700;      %Upper stopband edge.
fs = 4600;       %Sampling frequency. For this particular
filter I had to use 4800 rad/s as my sampling rate.

T = 2*pi/fs;     %Sampling period.

%Using prewarping technique to determine the design
parameters for the
%analog filter.
Wp1 = 2*(1/T)*tan(wp1*T/2);
Wp2 = 2*(1/T)*tan(wp2*T/2);
Ws1 = 2*(1/T)*tan(ws1*T/2);
Ws2 = 2*(1/T)*tan(ws2*T/2);

%Returning filter's order and stopband cutoff frequency
[n,Ws] = cheb2ord([Wp1 Wp2]/fs,[Ws1 Ws2]/fs, Ap, As);

disp('The order of the filter:');
```

```
disp(n)

%Returning poles, zeros and gain of nth order chebyshev
type 2 filter.
[z, p, k] = cheb2ap(n, As);

%State space representation.
[A, B, C, D] = zp2ss(z, p, k);

%Converting continuous-time state-space filter prototype to
a bandpass
%filter.
[At, Bt, Ct, Dt] = lp2bp(A, B, C, D, sqrt(Wp1*Wp2), Wp2-
Wp1);

%Converting state space representation into a transfer
function.
[p, q] = ss2tf(At, Bt, Ct, Dt);

%Getting the Laplase doamin frequency response.
%[h, w] = freqs(p, q, 2048);

%Implementing bilinear transformation to convert it to Z-
doamin.
[Ad, Bd, Cd, Dd] = bilinear(At, Bt, Ct, Dt, (1/T));

filter = ss2sos(Ad, Bd, Cd, Dd);
[b,a] = sos2tf(filter);

filter = tf(b,a);
%Extracting the oefficients of the transfer function
[num,den] = tfdata(filter);
filtord(b,a)
celldisp(num);
celldisp(den);
%[hd,f] = freqz(b,a,W,2);

%Getting frequency response of the digital filter in z-
domain.
[hd, fd] = freqz(ss2sos(Ad, Bd, Cd, Dd), 2048, (1/T));

x = fd*4*pi^2/4800;
```

```
%Plotting the magnitude of the IIR bandpass filter in
[0,pi] range.
subplot(2, 1, 1)
plot((x), 20*log10(abs(hd)))
xlim([0, pi])
title('Magnitude of IIR bandpass filter. ([0, pi] range)')
xlabel('Frequency in rad/s')
ylabel('Magnitude in dB')
grid

%Plotting the magnitude of the IIR bandpass filter in
[wp1,wp2] range.
subplot(2, 1, 2)
plot(x, 20*log10(abs(hd)))
xlim([wp1,wp2]*T)
title('Magnitude of IIR bandpass filter. ([wp1, wp2]
range)')
xlabel('Frequency in rad/s')
ylabel('Magnitude in dB')
grid
```