Title: Design of FIR FILTER using window method

AIM: To design FIR filters using the window method.

Objective: To design FIR filters using the window method using MATlab.

Write MATLAB code for lowpass, high-pass and band-pass FIR filter using Hamming window with:

- **a. Ripple Pass band=0.06**
- **b. Stop band Attenuation=0.05**
- **c. Pass band Frequency=1000Hz**
- **d. Stop band Frequency=3000Hz**
- **e. Sampling frequency=5000Hz**

and plot its magnitude and phase responses.

Description

FIR is a filter whose impulse response is of finite period, as a result of it settles to zero in finite time.

An FIR filter is a filter with no feedback in its equation. This can be an advantage because it makes an FIR filter inherently stable. Another advantage of FIR filters is the fact that they can produce linear phases. So, if an application requires linear phases, the decision is simple, an FIR filter must be used. The main drawback of a digital FIR filter is the time that it takes to execute. Since the filter has no feedback, many more coefficients are needed in the system equation to meet the same requirements that would be needed in an IIR filter. For every extra coefficient, there is an extra multiply and extra memory requirements for the DSP. For a demanding system, the speed and memory requirements to implement an FIR system can make the system unfeasible.

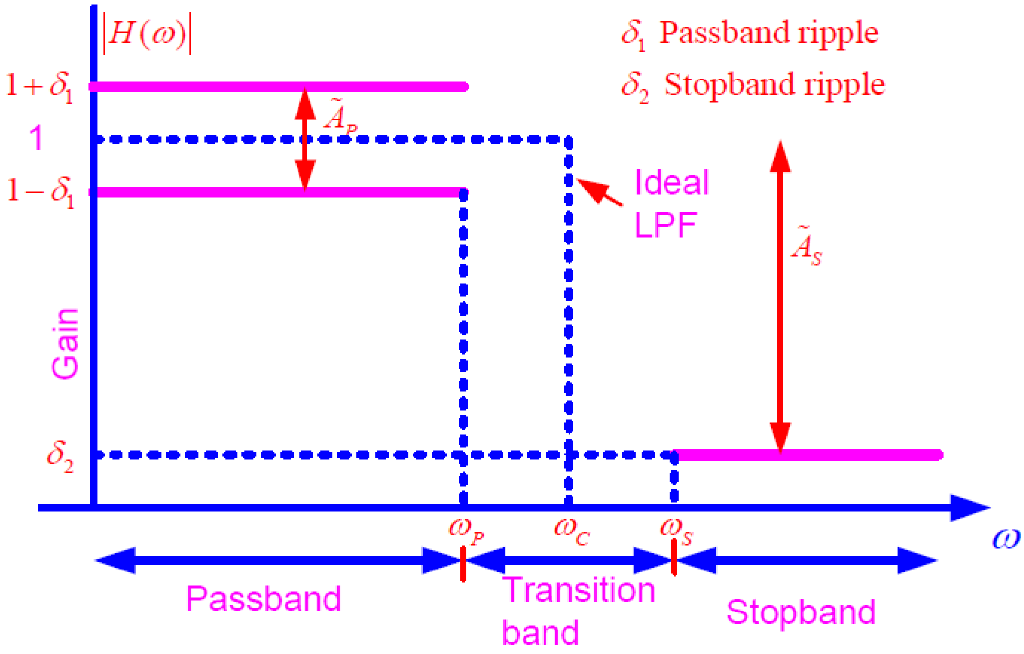
Hamming window

Hamming window function for M sample is given by

$$w(n) = \left[0.54 - 0.46\cos\left(\frac{2\pi n}{M-1}\right)\right]$$

$$M \geq \frac{8\pi}{\omega_s - \omega_p}$$

Low Pass FIR Filter Design using hamming window



The impulse response of low pass filter with group delay

$$\tau = \frac{M-1}{2}$$

Is given by

$$H_d(e^{j\omega}) = \begin{cases} e^{-j\omega\tau} & -\omega_c \le \omega \le \omega_c \\ 0 & Otherwise \end{cases}$$

By taking Inverse Fourier Transform,

$$\begin{aligned} h_d(n) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}d\omega \\ &= \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{-j\omega\tau}e^{j\omega n}d\omega \\ &= \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j\omega(n-\tau)}d\omega \end{aligned}$$

$$\begin{aligned} h_d(n) &= \frac{1}{2\pi}\left[\frac{e^{j\omega(n-\tau)}}{j(n-\tau)}\right]_{-\omega_c}^{\omega_c} \\ &= \frac{1}{2j\pi(n-\tau)}\left[e^{j\omega_c(n-\tau)} - e^{-j\omega_c(n-\tau)}\right] \\ &= \frac{1}{\pi(n-\tau)}\left[\frac{e^{j\omega_c(n-\tau)} - e^{-j\omega_c(n-\tau)}}{2j}\right] \end{aligned}$$

$$h_d(n) = \frac{sin\left[\omega_c(n-\tau)\right]}{\pi(n-\tau)}$$

Let for example M = 11

Then

for $n \neq 5$ and $\omega_c = \frac{\pi}{2}$, $\tau = \frac{M-1}{2} = 5$

$$h_d(n) = \frac{sin\left[\omega_c(n-5)\right]}{\pi(n-5)} = \frac{sin\left[\frac{\pi(n-5)}{2}\right]}{\pi(n-5)}$$

for n=5 $h_d(n) = \frac{0}{0}$. Using L Hospital's Rule

$$\lim_{\theta \to 0} \frac{\sin B\theta}{\theta} = B$$

$$\lim_{n \to 5} \frac{\sin \frac{\pi}{2}(n-5)}{\pi(n-5)} = \frac{\pi/2}{\pi} = 0.5$$

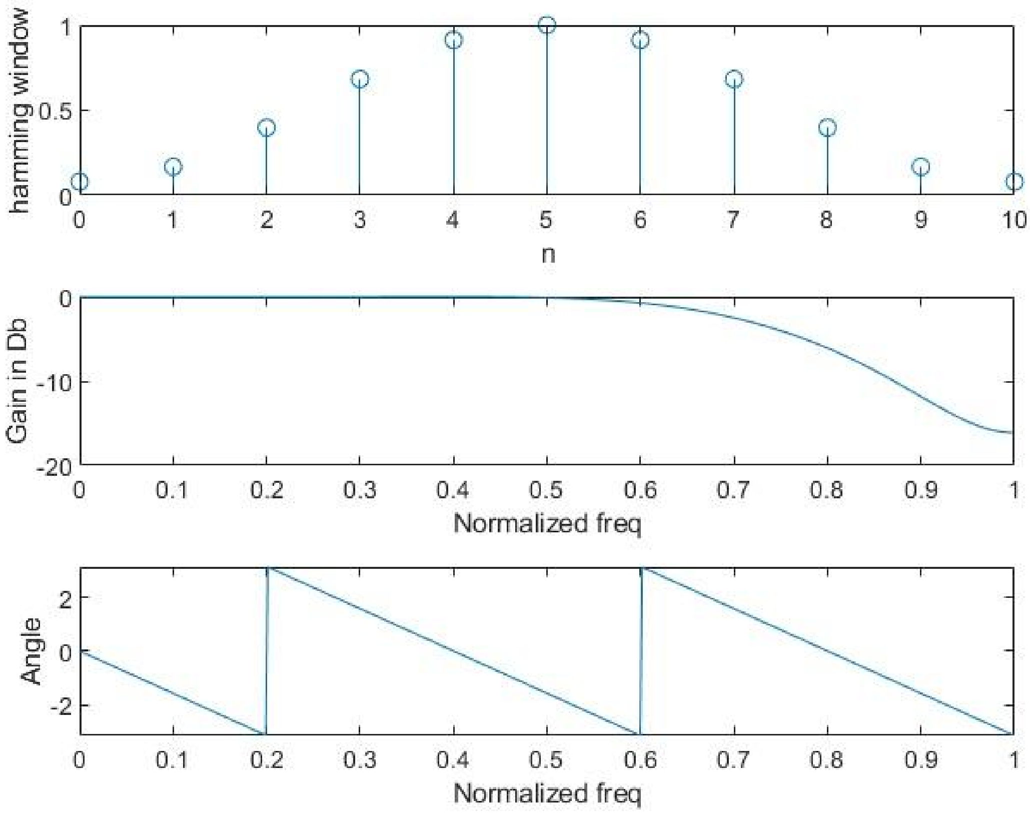
Then the FIR low pass filter is given by h(n) = hd(n) * w(n)

Matlab Program for Low pass filter using hamming window

```
rp=0.06;    % Passband ripple
rs=0.05;    % stop band ripple
fp=1000;    % Passband frequency
fs=3000;    % Stopband frequency
f= 5000;     % Sampling frequency
wp=2*pi*fp/f;
ws=2*pi*fs/f;
m = 8*pi/(ws-wp);
if(floor(m/2)==m/2)
   m = m+1;
end
n = 0:m-1;
%Hamming window
delay = (m-1)/2;
teta = 2*pi*n/(m-1);
W = 0.54 - 0.46*cos(teta);
subplot(3,1,1);
stem(n,W);
xlabel('n');
ylabel('hamming window');
%Low pass filter
p = n-((m-1)/2);
wc = (wp+ws)/2;
hd= sin(wc.*p)./(pi.*p);
hd(delay+1) = wc/pi;
h = hd.*W;
[H,w] = freqz(h);
subplot(3,1,2);
plot(w/pi,20.*log10(abs(H)));
xlabel('Normalised freq');
ylabel('Gain in Db');
subplot(3,1,3);
plot(w/pi,angle(H));
```

```
xlabel('Normalised freq');
ylabel('Angle');
```

Output of the program



High Pass FIR Filter Design using hamming window

The Impulse response of High Pass filter is given by

$$H_d(e^{j\omega}) = \begin{cases} e^{-j\omega\tau} & -\pi \leq \omega \leq -\omega_c \\ e^{-j\omega\tau} & \omega_c \leq \omega \leq -\pi \\ 0 & -\omega_c \leq \omega \leq \omega_c \end{cases}$$

Let for example M = 7 ,then the group delay

$$\tau = \frac{M-1}{2} = 3$$

$$H_d(e^{j\omega}) = \begin{cases} e^{-j\omega\tau} & -\pi \leq -\omega \leq \omega_c \\ 0 & Otherwise \end{cases}$$

Taking Inverse Fourier Transform

$$\begin{aligned} h_d(n) &= \frac{1}{\pi(n-\tau)}\left[\frac{e^{j\pi(n-\tau)} - e^{-j\pi(n-\tau)} - \left[e^{j\omega_c(n-\tau)} - e^{-j\omega_c(n-\tau)}\right]}{2j}\right] \\ &= \frac{1}{\pi(n-\tau)}[sin\pi(n-\tau) - sin\omega_c(n-\tau)] \end{aligned}$$

when $n = \tau$ using L Hospital rule

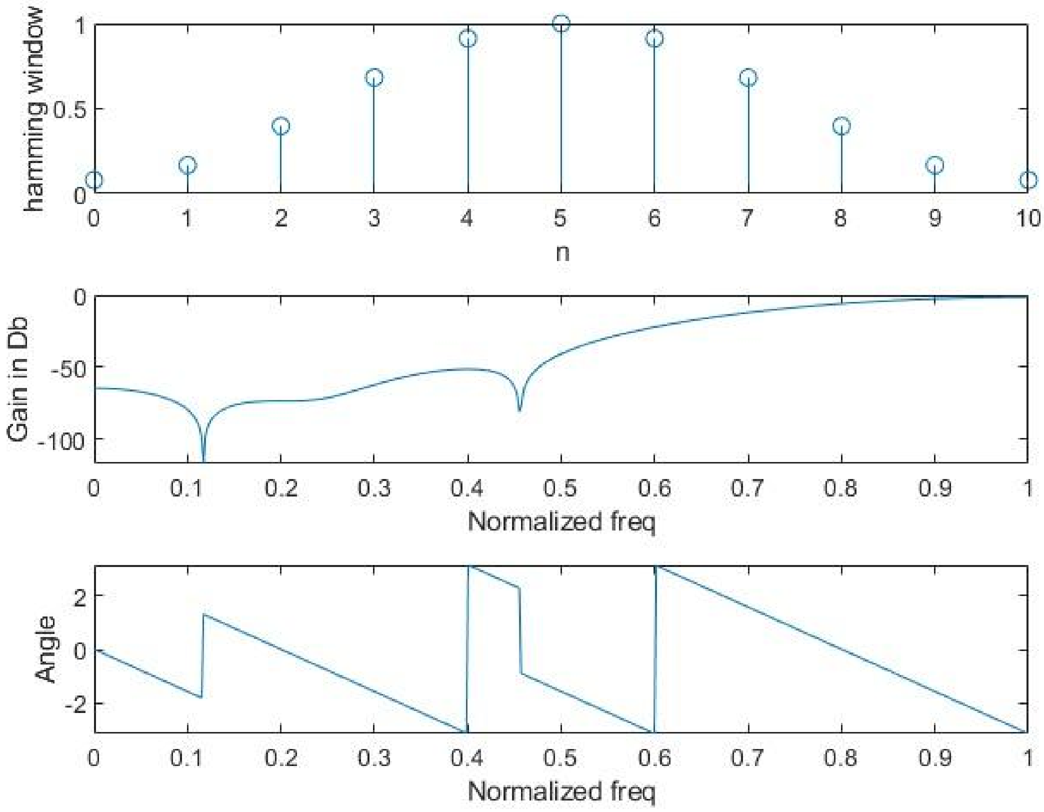
$$h_d(n) = \frac{1}{\pi}\left[\frac{sin\pi(n-3)}{(n-3)} - \frac{sin\omega_c(n-3)}{(n-3)}\right] = \frac{1}{\pi}[\pi - \omega_c]$$

Then the FIR low pass filter is given by h(n) = hd(n) * w(n)

Matlab Program for High Pass FIR Filter Design using hamming window

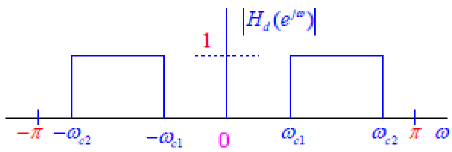
```
rp=0.06;    % Passband ripple
rs=0.05;    % stop band ripple
fp=1000;    % Passband frequency
fs=3000;    % Stopband frequency
f= 5000;     % Sampling frequency
wp=2*pi*fp/f;
ws=2*pi*fs/f;
m = 8*pi/(ws-wp);
if(floor(m/2)==m/2)
   m = m+1;
end
n = 0:m-1;
%Hamming window
teta = 2*pi*n/(m-1);
W = 0.54 - 0.46*cos(teta);
subplot(3,1,1);
stem(n,W);
xlabel('n');
ylabel('hamming window');
%High pass filter
p = n-((m-1)/2);
wc = (wp+ws)/2;
hd= (sin(pi.*p) - sin(wc.*p))./(pi.*p);
hd(delay+1) = (pi-wc)/pi;
h = hd.*W;
[H,w] = freqz(h);
subplot(3,1,2);
plot(w/pi,20.*log10(abs(H)));
xlabel('Normalised freq');
ylabel('Gain in Db');
subplot(3,1,3);
plot(w/pi,angle(H));
xlabel('Normalised freq');
ylabel('Angle');
```

Output of the High pass filter using hamming window



Band Pass FIR Filter Design using hamming window

$$H_d(\omega) = \begin{cases} e^{-j\omega\tau} & \omega_{c1} \leq |\omega_c| \leq \omega_{c_2} \\ 0 & Otherwise \end{cases}$$



$$\begin{aligned} h_d(n) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(\omega)e^{j\omega n}d\omega \\ &= \frac{1}{2\pi}\left[\int_{-\omega_{c_2}}^{-\omega_{c_1}} e^{-j\omega\tau}e^{j\omega n}d\omega + \int_{\omega_{c_1}}^{\omega_{c_2}} e^{-j\omega\tau}e^{j\omega n}d\omega\right] \\ &= \frac{1}{2\pi}\left[\int_{-\omega_{c_2}}^{-\omega_{c_1}} e^{j\omega(n-\tau)} + \int_{\omega_{c_1}}^{\omega_{c_2}} e^{j\omega(n-\tau)}d\omega\right] \\ &= \frac{1}{2\pi}\left[\left[\frac{e^{j\omega(n-\tau)}}{(n-\tau)}\right]_{-\omega_{c_2}}^{-\omega_{c_1}} + \left[\frac{e^{j\omega(n-\tau)}}{(n-\tau)}\right]_{\omega_{c_1}}^{\omega_{c_2}}\right] \\ &= \frac{sin\omega_{c_2}(n-\tau) - sin\omega_{c_1}(n-\tau)}{\pi(n-\tau)} \quad for\ \ n \neq \tau \end{aligned}$$

$$h_d(n) = \frac{sin\omega_{c_2}(n-\tau) - sin\omega_{c_1}(n-\tau)}{\pi(n-\tau)}$$
$$= \frac{\omega_{c_2} - \omega_{c_1}}{\pi} \quad for \;\; n = \tau$$

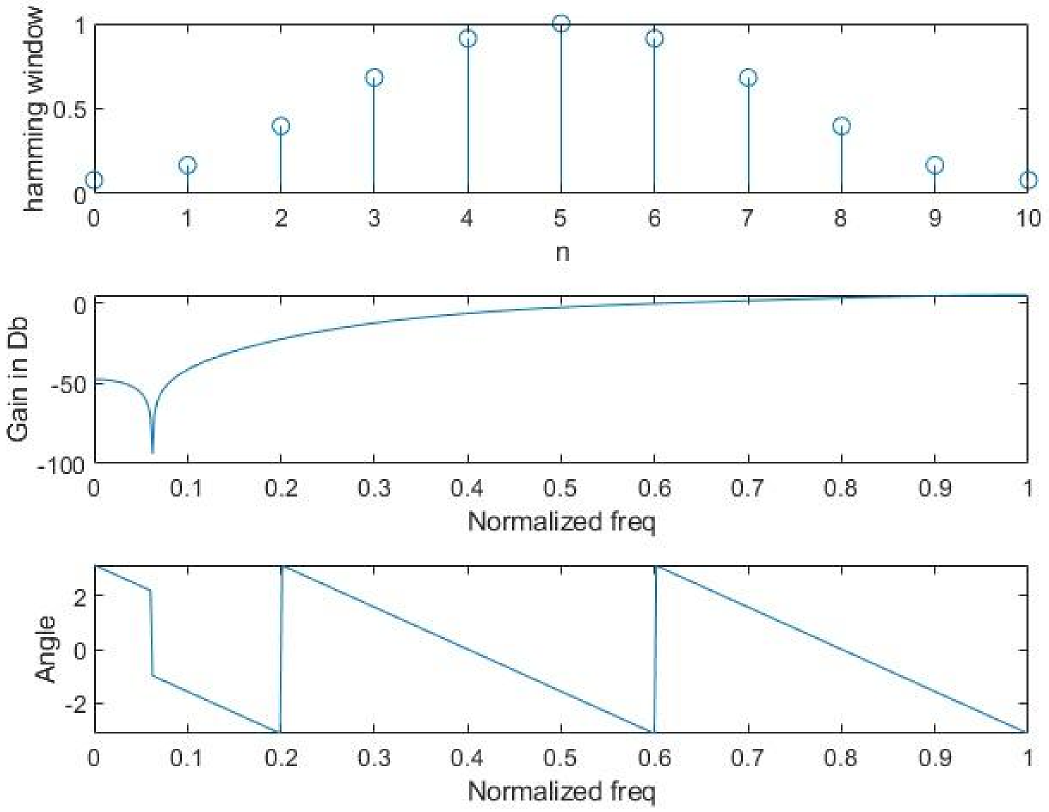
$$h_d(n) = \begin{cases} \frac{sin\omega_{c_2}(n-\tau) - sin\omega_{c_1}(n-\tau)}{\pi(n-\tau)} & for \;\; n \neq \tau \\ \frac{\omega_{c_2} - \omega_{c_1}}{\pi} & for \;\; n = \tau \end{cases}$$

Then the FIR low pass filter is given by h(n) = hd(n) * w(n)

Matlab Program for Band Pass FIR Filter Design using hamming window

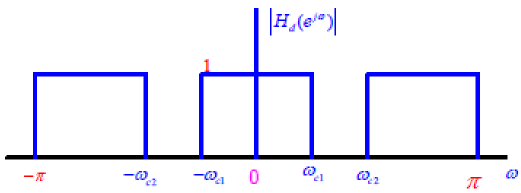
```
rp=0.06;    % Passband ripple
rs=0.05;    % stop band ripple
fp=1000;    % Passband frequency
fs=3000;    % Stopband frequency
f= 5000;     % Sampling frequency
wp=2*pi*fp/f;
ws=2*pi*fs/f;
m = 8*pi/(ws-wp);
if(floor(m/2)==m/2)
   m = m+1;
end
n = 0:m-1;
%Hamming window
teta = 2*pi*n/(m-1);
W = 0.54 - 0.46*cos(teta);
subplot(3,1,1);
stem(n,W);
xlabel('n');
ylabel('hamming window');
%Band pass filter
p = n-((m-1)/2);
hd= (sin(ws.*p) - sin(wp.*p))./(pi.*p);
hd(delay+1) = (ws-wp)/pi;
h = hd.*W;
[H,w] = freqz(h);
subplot(3,1,2);
plot(w/pi,20.*log10(abs(H)));
xlabel('Normalised freq');
ylabel('Gain in Db');
subplot(3,1,3);
plot(w/pi,angle(H));
xlabel('Normalised freq');
ylabel('Angle');
```

Output of the program Band pass filter using hamming window



Band Stop FIR Filter Design using hamming window

$$H_d(e^{j\omega}) = \begin{cases} e^{-j\omega\tau} & for \ -\pi \le \omega \le -\omega_{c2} \\ e^{-j\omega\tau} & for \ -\omega_{c1} \le \omega \le \omega_{c1} \\ e^{-j\omega\tau} & for \ \omega_{c2} \le \omega \le \pi \\ 0 & \quad for \ \omega_{c1} \le |\omega| \le \omega_{c2} \end{cases}$$



$$\begin{aligned} h_d(n) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}d\omega \\ &= \frac{1}{2\pi}[\int_{-\pi}^{-\omega_{c2}} e^{j\omega(n-\tau)}d\omega + \int_{-\omega_{c1}}^{\omega_{c1}} e^{j\omega(n-\tau)}d\omega + \int_{\omega_{c2}}^{\pi} e^{j\omega(n-\tau)}d\omega \\ &= \frac{1}{2\pi}\left[\frac{e^{j\omega(n-\tau)}}{j(n-\tau)}\right]_{-\pi}^{-\omega_{c2}} + \frac{1}{2\pi}\left[\frac{e^{j\omega(n-\tau)}}{j(n-\tau)}\right]_{-\omega_{c1}}^{\omega_{c1}} + \frac{1}{2\pi}\left[\frac{e^{j\omega(n-\tau)}}{j(n-\tau)}\right]_{\omega_{c2}}^{\pi} \\ &= \frac{1}{\pi(n-\tau)}[sin\omega_{c1}(n-\tau) + sin\pi(n-\tau - sin\omega_{c2}(n-\tau)] \end{aligned}$$

- The inverse transform of the $H_d(e^{\omega})$ is

$\tau = \frac{M-1}{2} = \frac{5-1}{2} = 2$ $\omega_{c1} = 2$ rad/sec $\omega_{c2} = 3$ rad/sec

$$h_d(n) = \frac{1}{\pi(n-2)}\left[sin2(n-2) + sin\pi(n-2) - sin3(n-2)\right] \qquad for \ n \neq 2$$

for $n = \tau$

$$h_d(n) = \frac{1}{\pi}\left[\lim_{n\to\tau}\frac{sin\omega_{c1}(n-\tau)}{(n-\tau)} + \lim_{n\to\tau}\frac{sin\pi(n-\tau)}{(n-\tau)} - \lim_{n\to\tau}\frac{sin\omega_{c2}(n-\tau)}{(n-\tau)}\right]$$

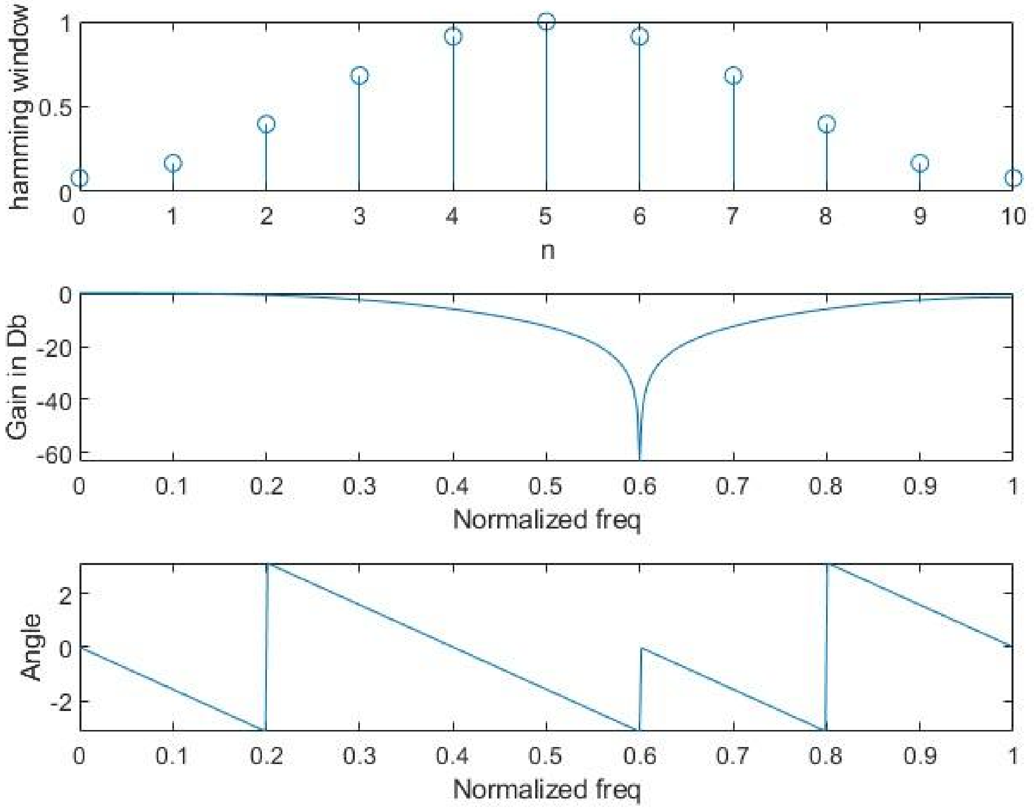
$$h_d(n) = \frac{1}{\pi}\left[\omega_{c1} + \pi - \omega_{c2}\right] = \frac{1}{\pi}\left[\pi - 1\right]$$

Then the FIR low pass filter is given by h(n) = hd(n) * w(n)

Matlab Program for BandStop FIR Filter Design using hamming window

```
rp=0.06;    % Passband ripple
rs=0.05;    % stop band ripple
fp=1000;    % Passband frequency
fs=3000;    % Stopband frequency
f= 5000;     % Sampling frequency
wp=2*pi*fp/f;
ws=2*pi*fs/f;
m = 8*pi/(ws-wp);
if(floor(m/2)==m/2)
   m = m+1;
end
n = 0:m-1;
%Hamming window
teta = 2*pi*n/(m-1);
W = 0.54 - 0.46*cos(teta);
subplot(3,1,1);
stem(n,W);
xlabel('n');
ylabel('hamming window');
%Band stop filter
delay = (m-1)/2;
p = n-((m-1)/2);
hd= (sin(pi.*p) + sin(wp.*p) - sin(ws.*p))./(pi.*p);
hd(delay+1) = (pi-ws+wp)/pi;
h = hd.*W;
[H,w] = freqz(h);
subplot(3,1,2);
plot(w/pi,20.*log10(abs(H)));
xlabel('Normalised freq');
ylabel('Gain in Db');
subplot(3,1,3);
plot(w/pi,angle(H));
xlabel('Normalised freq');
ylabel('Angle');
```

Output of the program Band stop filter using hamming window



Write MATLAB code for lowpass, high-pass and band-pass FIR filter using kaiser window with:

a. Ripple Pass band=0.04
b. Stop band Attenuation=0.05
c. Pass band Frequency=2000Hz
d. Stop band Frequency=4000Hz
e. Sampling frequency=8000Hz
f. Beta=2, 4 and plot its magnitude and phase responses.

Description

- The Kaiser window is parametric and its bandwidth as well as its sidelobe energy can be designed.
- Mainlobe bandwidth controls the transition characteristics and sidelobe energy affects the ripple characteristics.
- The Kaiser window function is given by

$$w_k(n) = \frac{I_0\left[\alpha\sqrt{1-\left(\frac{2n}{M-1}\right)^2}\right]}{I_0(\alpha)}$$

where M is the order of the filter, $I_0(x)$ is a zeroth Bessel function of the first kind

$$\begin{aligned} I_0(x) &= 1+\sum_{k=1}^{\infty}\left[\frac{1}{k!}\left(\frac{x}{2}\right)^k\right] \\ &= 1+\frac{0.25x^2}{(1!)^2}+\frac{(0.25x^2)^2}{(2!)^2}+\frac{(0.25x^2)^3}{(3!)^2}+ \end{aligned}$$

$$\begin{aligned} \alpha &= 0 && \text{if A} < 21 \\ &= 0.5842(A-21)^{0.4}+0.07886(A-21) && \text{if } 21 \leq \text{A} \leq 50 \text{ dB} \\ &= 0.1102(A-8.7)) && \text{if A} > 50 \text{ dB} \end{aligned}$$

Kaiser Window Design Equations

1. Determine ideal frequency response

$$H_d(e^{j\omega}) = \begin{cases} 1 \; for \; |\omega| \leq \omega_c \\ 0 \; for \; \omega_c \leq |\omega| \leq \pi \end{cases}$$

   where $\omega_c = \frac{1}{2}(\omega_p + \omega_s)$

2. Chose δ such that the actual passband ripple, A_p is equal to or less than the specified passband ripple $\tilde{A}_p$, and the actual minimum stopband attenuation A is equal or greater than the specified minimum stop attenuation $\tilde{A}_s$

$$\delta = min(\delta_p, \delta_s)$$

   where $\delta_p = \frac{10^{0.05\tilde{A}_p}-1}{10^{0.05\tilde{A}_p}+1}$ and $\quad \delta_s = 10^{-0.05\tilde{A}_s}$

3. The actual stopband attenuation is

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

4. The parameter α is

$$\alpha = \begin{cases} 0 & for \;\; A \leq 21 \\ 0.5842(A-21)^{0.4}+0.07886(A_a-21) & for \; 21 < A \leq 50 \\ 0.1102(A-8.7) & for \;\; A > 50 \end{cases}$$

5. The value of M is found by

$$M \geq \frac{A - 7.95}{14.36\Delta f}$$

where $\Delta f = \frac{\Delta\omega}{2\pi} = \frac{\omega_s - \omega_p}{2\pi}$ and $\Delta\omega$ is the width of transition band

6. Obtain impulse response by multiplying Kaiser window function

$$h(n) = h_d(n)w_k(n)$$

7. Obtain the causal finite impulse response
8. The system function is given by

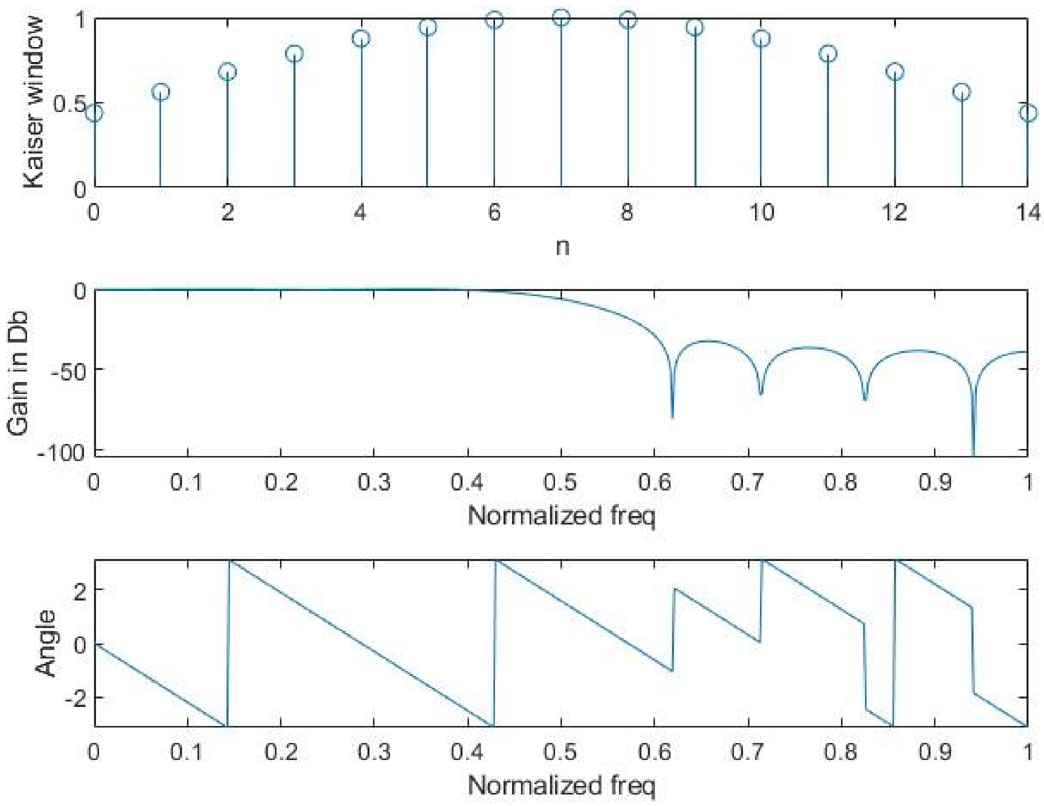
$$H(z) = \sum_{n=0}^{M-1} h(n)z^{-n}$$

Matlab Program for Low pass filter using kaiser window with beta = 2

```
rp=0.04;    % Passband ripple
rs=0.05;    % stop band ripple
fp=2000;    % Passband frequency
fs=4000;    % Stopband frequency
f= 8000;     % Sampling frequency
beta = 2;
if(rp>rs)
   A = -20*log(rs);
else
   A = -20*log(rp);
end
A = floor(A);
wp=2*pi*fp/f;
ws=2*pi*fs/f;
delfreq = (ws-wp)/(2*pi);
m =(A-7.95)/(14.36*delfreq);
m=floor(m);
if(floor(m/2)==m/2)
   m = m+1;
end
n = 0:m-1;
delay = (m-1)/2;
%Kaiser window
W = kaiser(m,beta);
subplot(3,1,1);
stem(n,W);
xlabel('n');
ylabel('Kaiser window');
%Low pass filter
```

```
p = n-((m-1)/2);
wc=(wp+ws)/2;
hd=  sin(wc.*p)./(pi.*p);
hd(delay+1) = (pi-ws+wp)/pi;
h = hd'.*W;
[H,w] = freqz(h);
subplot(3,1,2);
plot(w/pi,20.*log10(abs(H)));
xlabel('Normalised freq');
ylabel('Gain in Db');
subplot(3,1,3);
plot(w/pi,angle(H));
xlabel('Normalised freq');
ylabel('Angle');
```

Output of the program Low pass filter using kaiser window with beta = 2

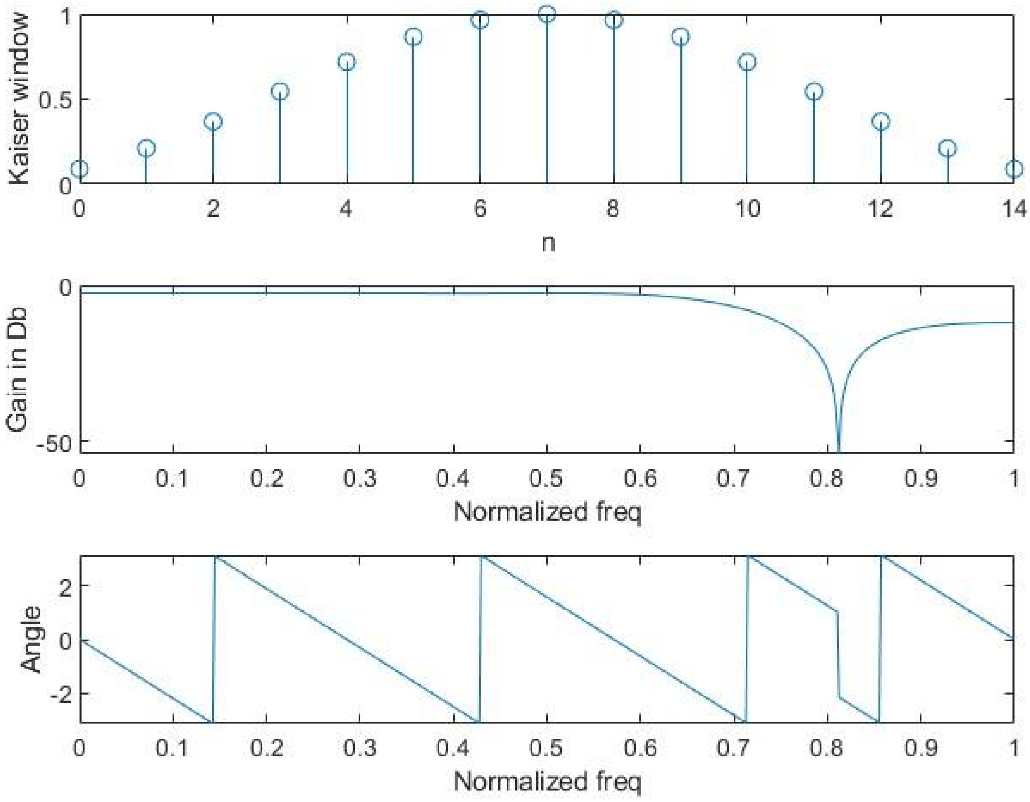


Matlab Program for Low pass filter using kaiser window with beta = 4

```
rp=0.04;    % Passband ripple
rs=0.05;    % stop band ripple
fp=2000;    % Passband frequency
fs=4000;    % Stopband frequency
f= 8000;     % Sampling frequency
beta = 4;
if(rp>rs)
   A = -20*log(rs);
else
   A = -20*log(rp);
end
```

```matlab
A = floor(A);
wp=2*pi*fp/f;
ws=2*pi*fs/f;
delfreq = (ws-wp)/(2*pi);
m =(A-7.95)/(14.36*delfreq);
m=floor(m);
if(floor(m/2)==m/2)
   m = m+1;
end
n = 0:m-1;
delay = (m-1)/2;
%Kaiser window
W = kaiser(m,beta);
subplot(3,1,1);
stem(n,W);
xlabel('n');
ylabel('Kaiser window');
%Low pass filter
p = n-((m-1)/2);
wc=(wp+ws)/2;
hd=  sin(wc.*p)./(pi.*p);
hd(delay+1) = (pi-ws+wp)/pi;
h = hd'.*W;
[H,w] = freqz(h);
subplot(3,1,2);
plot(w/pi,20.*log10(abs(H)));
xlabel('Normalised freq');
ylabel('Gain in Db');
subplot(3,1,3);
plot(w/pi,angle(H));
xlabel('Normalised freq');
ylabel('Angle');
```

Output of the program Low pass filter using kaiser window with beta = 4

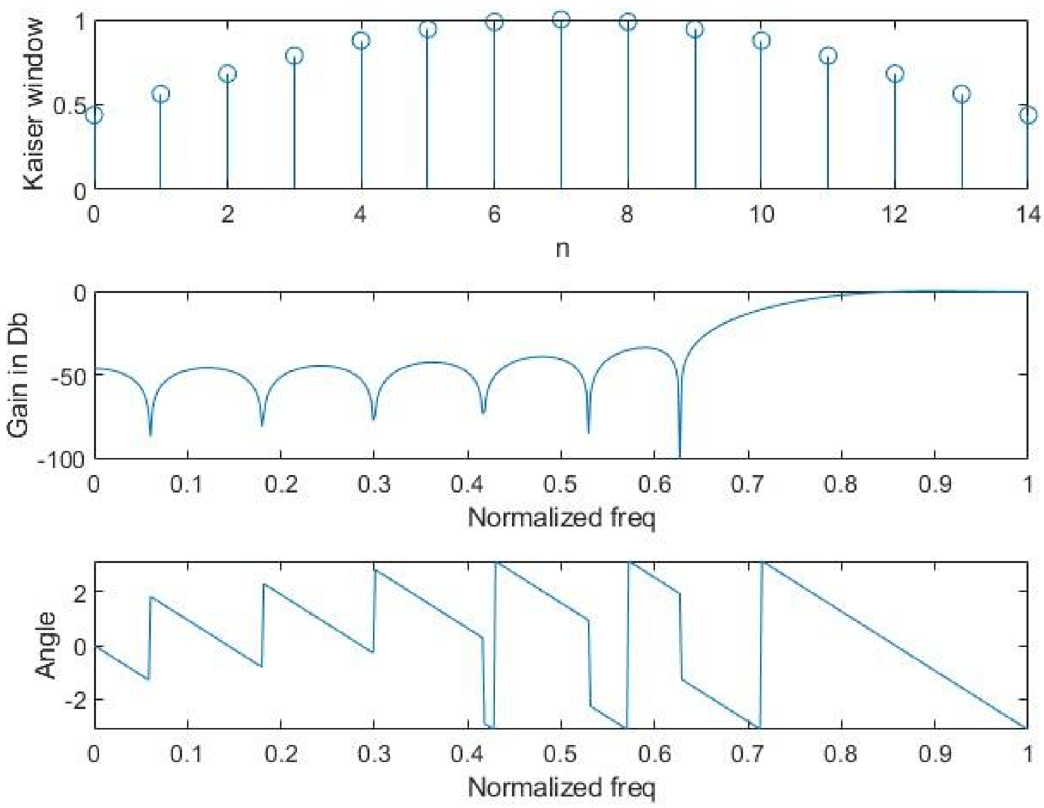


Matlab Program for High pass filter using kaiser window with beta = 2

```
rp=0.04;    % Passband ripple
rs=0.05;    % stop band ripple
fp=2000;    % Passband frequency
fs=4000;    % Stopband frequency
f= 8000;     % Sampling frequency
beta = 2;
if(rp>rs)
   A = -20*log(rs);
else
   A = -20*log(rp);
end
A = floor(A);
wp=2*pi*fp/f;
ws=2*pi*fs/f;
delfreq = (ws-wp)/(2*pi);
m =(A-7.95)/(14.36*delfreq);
m=floor(m);
if(floor(m/2)==m/2)
   m = m+1;
end
n = 0:m-1;
delay = (m-1)/2;
%Kaiser window
W = kaiser(m,beta);
subplot(3,1,1);
```

```
stem(n,W);
xlabel('n');
ylabel('Kaiser window');
%High pass filter
p = n-((m-1)/2);
wc=(wp+ws)/2;
hd= (sin(pi.*p) - sin(wc.*p))./(pi.*p);
hd(delay+1) = (pi-wc)/pi;
h = hd'.*W;
[H,w] = freqz(h);
subplot(3,1,2);
plot(w/pi,20.*log10(abs(H)));
xlabel('Normalised freq');
ylabel('Gain in Db');
subplot(3,1,3);
plot(w/pi,angle(H));
xlabel('Normalised freq');
ylabel('Angle');
```

Output of the program High pass filter using kaiser window with beta = 2

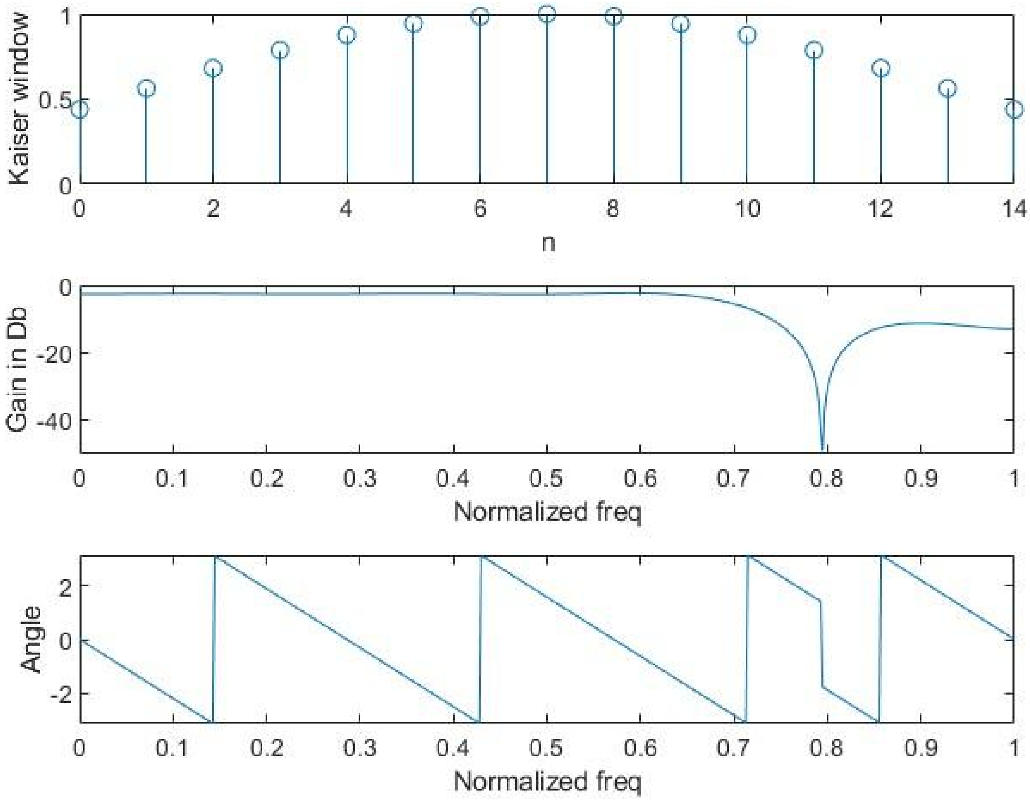


Matlab Program for High pass filter using kaiser window with beta = 4

```
rp=0.04;    % Passband ripple
rs=0.05;    % stop band ripple
fp=2000;    % Passband frequency
fs=4000;    % Stopband frequency
f= 8000;     % Sampling frequency
beta = 4;
if(rp>rs)
```

```
   A = -20*log(rs);
else
   A = -20*log(rp);
end
A = floor(A);
wp=2*pi*fp/f;
ws=2*pi*fs/f;
delfreq = (ws-wp)/(2*pi);
m =(A-7.95)/(14.36*delfreq);
m=floor(m);
if(floor(m/2)==m/2)
   m = m+1;
end
n = 0:m-1;
delay = (m-1)/2;
%Kaiser window
W = kaiser(m,beta);
subplot(3,1,1);
stem(n,W);
xlabel('n');
ylabel('Kaiser window');
%High pass filter
p = n-((m-1)/2);
wc=(wp+ws)/2;
hd= (sin(pi.*p) - sin(wc.*p))./(pi.*p);
hd(delay+1) = (pi-wc)/pi;
h = hd'.*W;
[H,w] = freqz(h);
subplot(3,1,2);
plot(w/pi,20.*log10(abs(H)));
xlabel('Normalised freq');
ylabel('Gain in Db');
subplot(3,1,3);
plot(w/pi,angle(H));
xlabel('Normalised freq');
ylabel('Angle');
```

Output of the program High pass filter using kaiser window with beta = 4

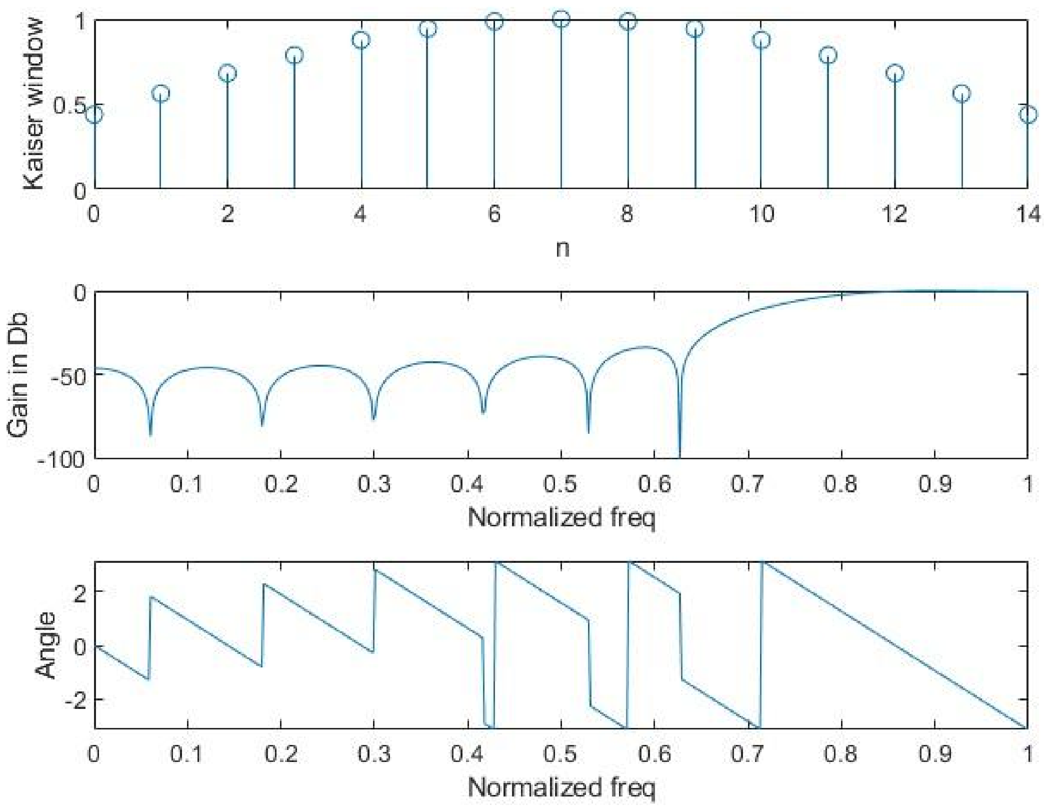


Matlab Program for Band pass filter using kaiser window with beta = 2

```
rp=0.04;    % Passband ripple
rs=0.05;    % stop band ripple
fp=2000;    % Passband frequency
fs=4000;    % Stopband frequency
f= 8000;     % Sampling frequency
beta = 2;
if(rp>rs)
   A = -20*log(rs);
else
   A = -20*log(rp);
end
A = floor(A);
wp=2*pi*fp/f;
ws=2*pi*fs/f;
delfreq = (ws-wp)/(2*pi);
m =(A-7.95)/(14.36*delfreq);
m=floor(m);
if(floor(m/2)==m/2)
   m = m+1;
end
n = 0:m-1;
delay = (m-1)/2;
%Kaiser window
W = kaiser(m,beta);
subplot(3,1,1);
```

```
stem(n,W);
xlabel('n');
ylabel('Kaiser window');
%Band pass filter
p = n-((m-1)/2);
hd= (sin(ws.*p) - sin(wp.*p))./(pi.*p);
hd(delay+1) = (ws-wp)/pi;
h = hd'.*W;
[H,w] = freqz(h);
subplot(3,1,2);
plot(w/pi,20.*log10(abs(H)));
xlabel('Normalised freq');
ylabel('Gain in Db');
subplot(3,1,3);
plot(w/pi,angle(H));
xlabel('Normalised freq');
ylabel('Angle');
```

Output of the program Band pass filter using kaiser window with beta = 2

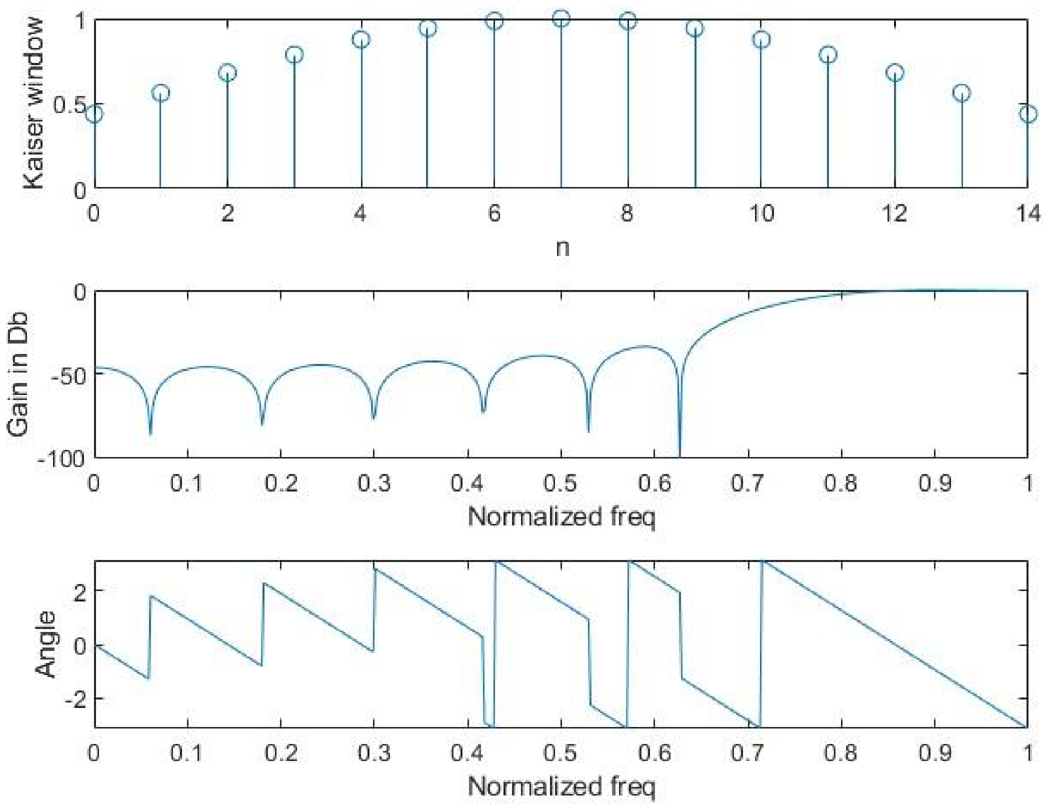


Matlab Program for Band pass filter using kaiser window with beta = 4

```
rp=0.04;    % Passband ripple
rs=0.05;    % stop band ripple
fp=2000;    % Passband frequency
fs=4000;    % Stopband frequency
f= 8000;     % Sampling frequency
beta = 4;
```

```
if(rp>rs)
   A = -20*log(rs);
else
   A = -20*log(rp);
end
A = floor(A);
wp=2*pi*fp/f;
ws=2*pi*fs/f;
delfreq = (ws-wp)/(2*pi);
m =(A-7.95)/(14.36*delfreq);
m=floor(m);
if(floor(m/2)==m/2)
   m = m+1;
end
n = 0:m-1;
delay = (m-1)/2;
%Kaiser window
W = kaiser(m,beta);
subplot(3,1,1);
stem(n,W);
xlabel('n');
ylabel('Kaiser window');
%Band pass filter
p = n-((m-1)/2);
hd= (sin(ws.*p) - sin(wp.*p))./(pi.*p);
hd(delay+1) = (ws-wp)/pi;
h = hd'.*W;
[H,w] = freqz(h);
subplot(3,1,2);
plot(w/pi,20.*log10(abs(H)));
xlabel('Normalised freq');
ylabel('Gain in Db');
subplot(3,1,3);
plot(w/pi,angle(H));
xlabel('Normalised freq');
ylabel('Angle');
```

Output of the program Band pass filter using kaiser window with beta = 4

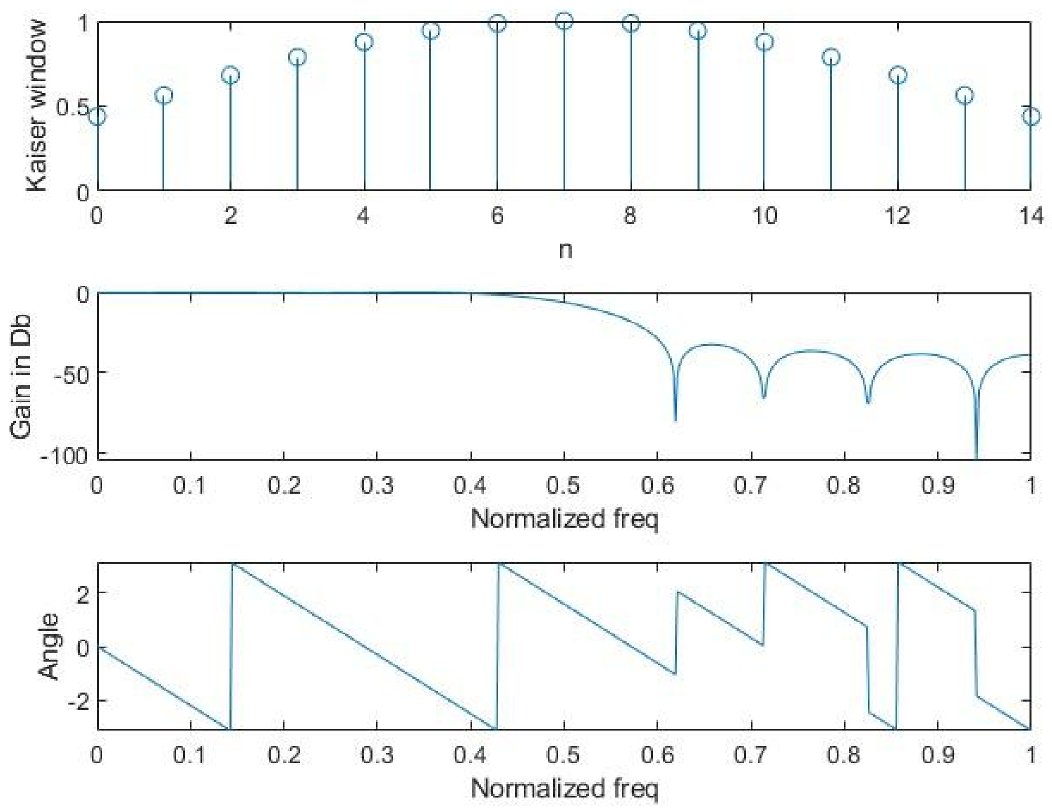


Matlab Program for Band stop filter using kaiser window with beta = 2

```
rp=0.04;    % Passband ripple
rs=0.05;    % stop band ripple
fp=2000;    % Passband frequency
fs=4000;    % Stopband frequency
f= 8000;     % Sampling frequency
beta = 2;
if(rp>rs)
   A = -20*log(rs);
else
   A = -20*log(rp);
end
A = floor(A);
wp=2*pi*fp/f;
ws=2*pi*fs/f;
delfreq = (ws-wp)/(2*pi);
m =(A-7.95)/(14.36*delfreq);
m=floor(m);
if(floor(m/2)==m/2)
   m = m+1;
end
n = 0:m-1;
delay = (m-1)/2;
%Kaiser window
```

```
W = kaiser(m,beta);
subplot(3,1,1);
stem(n,W);
xlabel('n');
ylabel('Kaiser window');
%Band pass filter
p = n-((m-1)/2);
hd= (sin(pi.*p) + sin(wp.*p) - sin(ws.*p))./(pi.*p);
hd(delay+1) = (pi-ws+wp)/pi;
h = hd'.*W;
[H,w] = freqz(h);
subplot(3,1,2);
plot(w/pi,20.*log10(abs(H)));
xlabel('Normalised freq');
ylabel('Gain in Db');
subplot(3,1,3);
plot(w/pi,angle(H));
xlabel('Normalised freq');
ylabel('Angle');
```

Output of the program Band stop filter using kaiser window with beta = 2

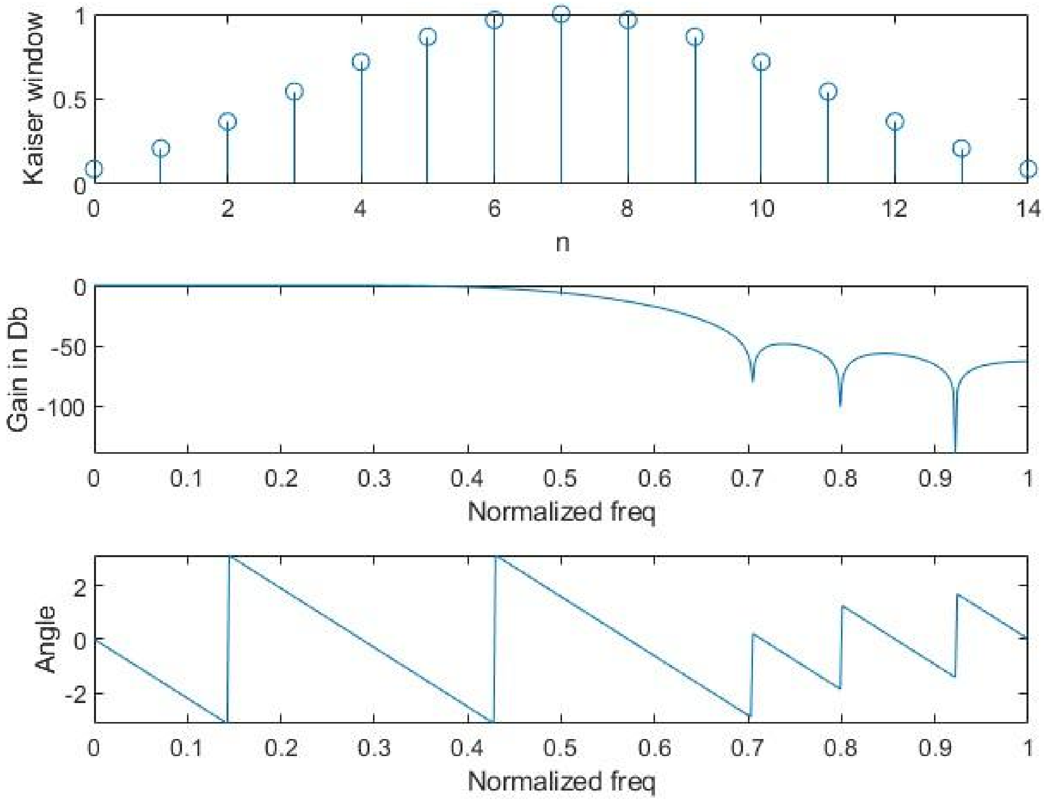


Matlab Program for Band stop filter using kaiser window with beta = 4

```
rp=0.04;    % Passband ripple
rs=0.05;    % stop band ripple
fp=2000;    % Passband frequency
fs=4000;    % Stopband frequency
f= 8000;     % Sampling frequency
```

```
beta = 4;
if(rp>rs)
   A = -20*log(rs);
else
   A = -20*log(rp);
end
A = floor(A);
wp=2*pi*fp/f;
ws=2*pi*fs/f;
delfreq = (ws-wp)/(2*pi);
m =(A-7.95)/(14.36*delfreq);
m=floor(m);
if(floor(m/2)==m/2)
   m = m+1;
end
n = 0:m-1;
delay = (m-1)/2;
%Kaiser window
W = kaiser(m,beta);
subplot(3,1,1);
stem(n,W);
xlabel('n');
ylabel('Kaiser window');
%Band pass filter
p = n-((m-1)/2);
hd= (sin(pi.*p) + sin(wp.*p) - sin(ws.*p))./(pi.*p);
hd(delay+1) = (pi-ws+wp)/pi;
h = hd'.*W;
[H,w] = freqz(h);
subplot(3,1,2);
plot(w/pi,20.*log10(abs(H)));
xlabel('Normalised freq');
ylabel('Gain in Db');
subplot(3,1,3);
plot(w/pi,angle(H));
xlabel('Normalised freq');
ylabel('Angle');
```

Output of the program Band stop filter using kaiser window with beta = 4



Result :FIR Filter is designed using Hamming and Kaiser Widow.