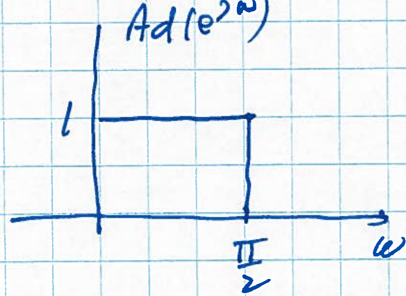
1) The ideal amplitude response, $A_d(e^{j\omega})$, for a halfband filter is



$$f_c = \frac{\frac{\pi}{2}}{2\pi} = \frac{1}{4} \text{ cycles/sample}$$

For a Kaiser window, determine β and M

$$A = \cancel{-20\log_{10}\delta} = 60\,\text{dB}.$$

Since $A > 50$

$$\beta = 0.1102(A - 8.7) = 5.65$$

$$M = \frac{A - 8}{2.285(0.1 \times 2\pi)} = \cancel{39.7 \to 74} \quad 36.2 \to 36$$

Note M must be even

Determine $h[n]$

$$h_d[n] = 2 f_c \operatorname{sinc}\left(2 f_c\left(n - \frac{M}{2}\right)\right)$$

$$h[n] = h_d[n]\, h_w[n]$$

where $h_w[n]$ is a Kaiser window, which can be calculated in MATLAB using

hw = kaiser(M+1, beta)

Question 1

Contents

Define initial values

```
transition_width = 0.1*2*pi;
fc=0.25;
A=60;
```

Generate the response

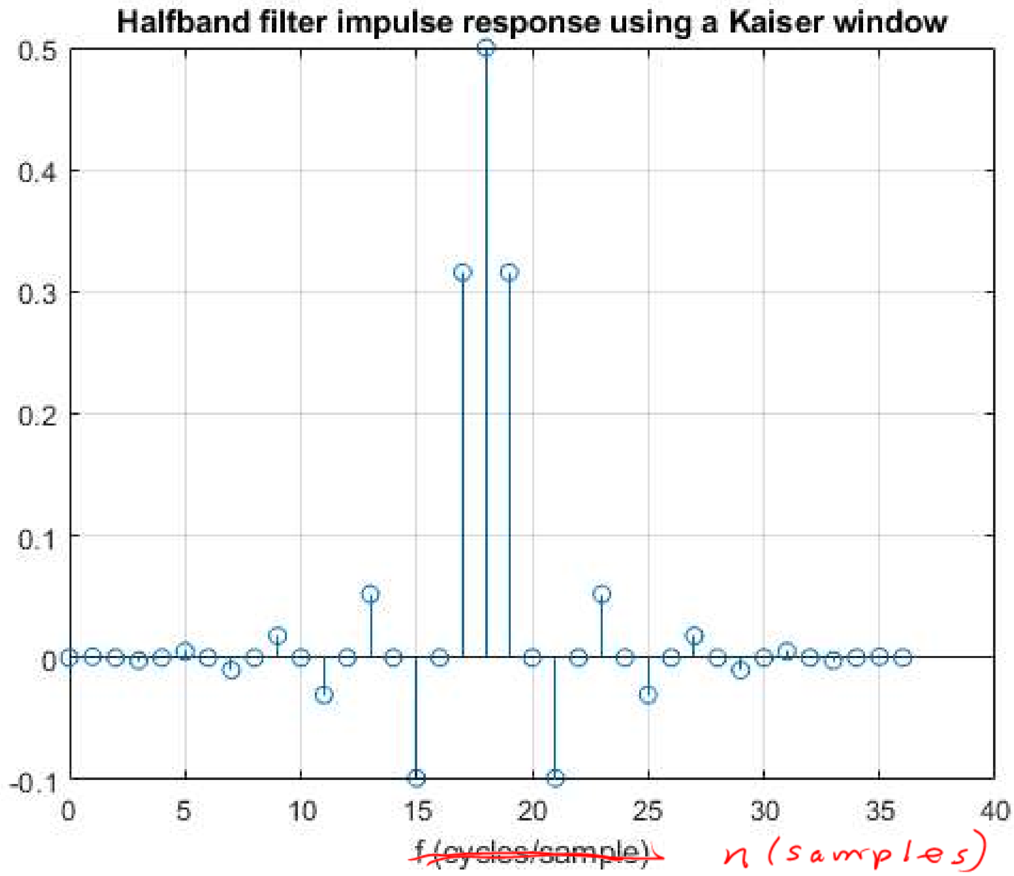
```
M=(A-8)/(2.285*(transition_width))
M=round(M)   %recall M must be even
n=0:M;
beta = .1102*(A-8.7)
if A<=50
display(['A=',num2str(A),', an incorrect expression was used for beta'])
end
win=kaiser(M+1,beta);
hd=2*fc*sinc(2*fc*(n-M/2));
h=hd.*win.';
```

```
M =

  36.219068012159987


M =

    36


beta =

   5.653260000000000
```

Plot impulse response

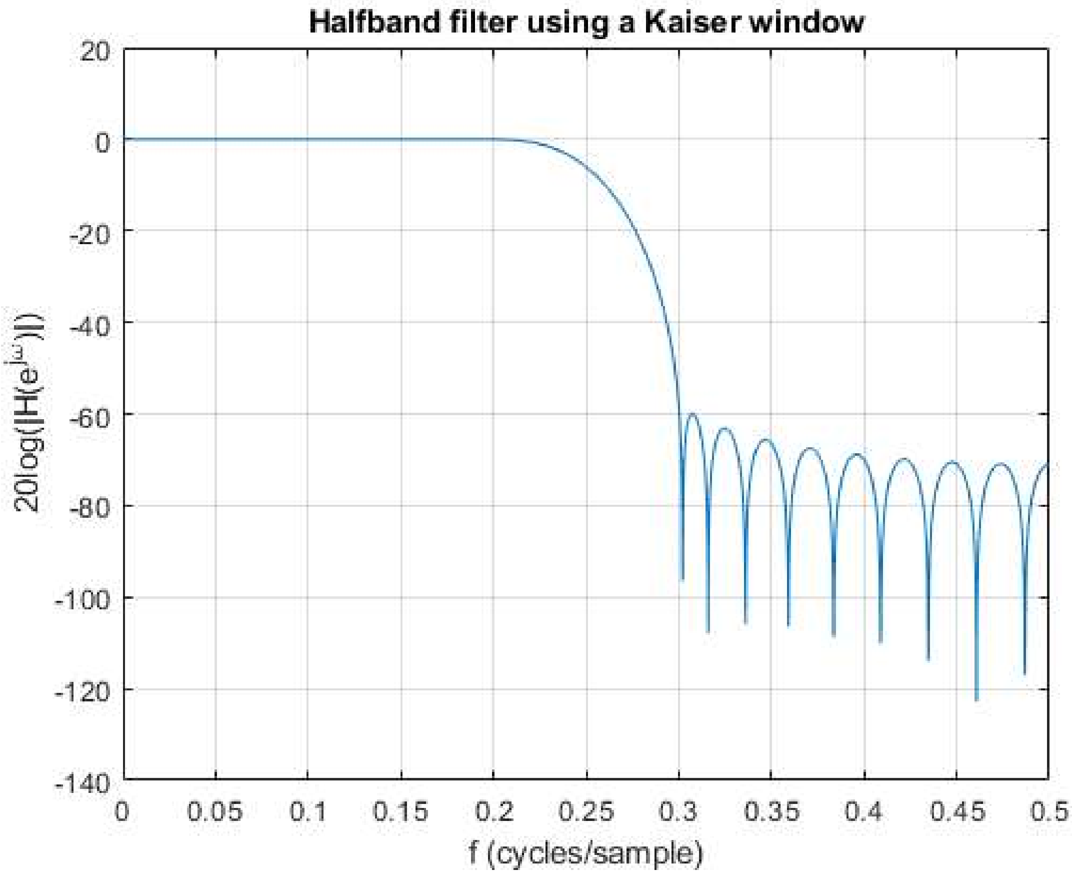
```
figure(1)
stem(n,h),grid
```

```
title('Halfband filter impulse response using a Kaiser window')
xlabel('f (cycles/sample)')
```



Plot response

```
[H,w]=freqz(h,1,[0:.001:pi]);
figure(2)
clf
plot(w/(2*pi),20*log10(abs(H)))
title('Halfband filter using a Kaiser window')
xlabel('f (cycles/sample)')
ylabel('20log(|H(e^{j\omega})|)')
grid
```



Peak Ripple

```
peak_ripple = max(abs(H))
% with M=36, this does meet spec in the passband and stopband
```

```
peak_ripple =

   1.000998277640510
```

Published with MATLAB® R2018a

3.) a) This is a raised cosine pulse so N_{sps} is the interval between zeros, $N_{sps} = 4$.

b) From the expression for $H_{rc}(F)$

$$B_{abs} = (1+\beta)F_c \ Hz$$

Normalize by dividing by the sampling frequency F_s

$$B_{abs} = (1+\beta)\frac{F_c}{F_s} = (1+\beta) f_c \text{ cycles/sample}$$

When ~~$f_c = \frac{F_c}{F_s} = \frac{1}{N_{sps}} = 0.25$~~ $f_c = \frac{1}{2N_{sps}} = \frac{1}{8}$

Then $B_{abs} =$ ~~$(1.25)(.25) = 0.3125 = \frac{5}{16}$ cycles/sample~~
$(1.25)\left(\frac{1}{8}\right) = \frac{5}{4}\left(\frac{1}{8}\right) = \frac{5}{32}$ cycles/sample

c) For a raised cosine pulse the gain must be 0.5 at f_c.

d) The raised cosine pulse has a peak value of $0.25 = \frac{1}{N_{sps}}$. This is the peak amplitude of $h_{rc}[n]$ (see the expression in the slides) for $H_{rc}(F)$ having a passband gain of 1

Question 5

Contents

Define intial values

```
beta=0.2;
Nsps=4;
span=25;
M=span*Nsps; %order of the filter
fc=1/(2*Nsps);
fp=(1-beta)*fc;
fs=(1+beta)*fc;
```

Square root raised cosine

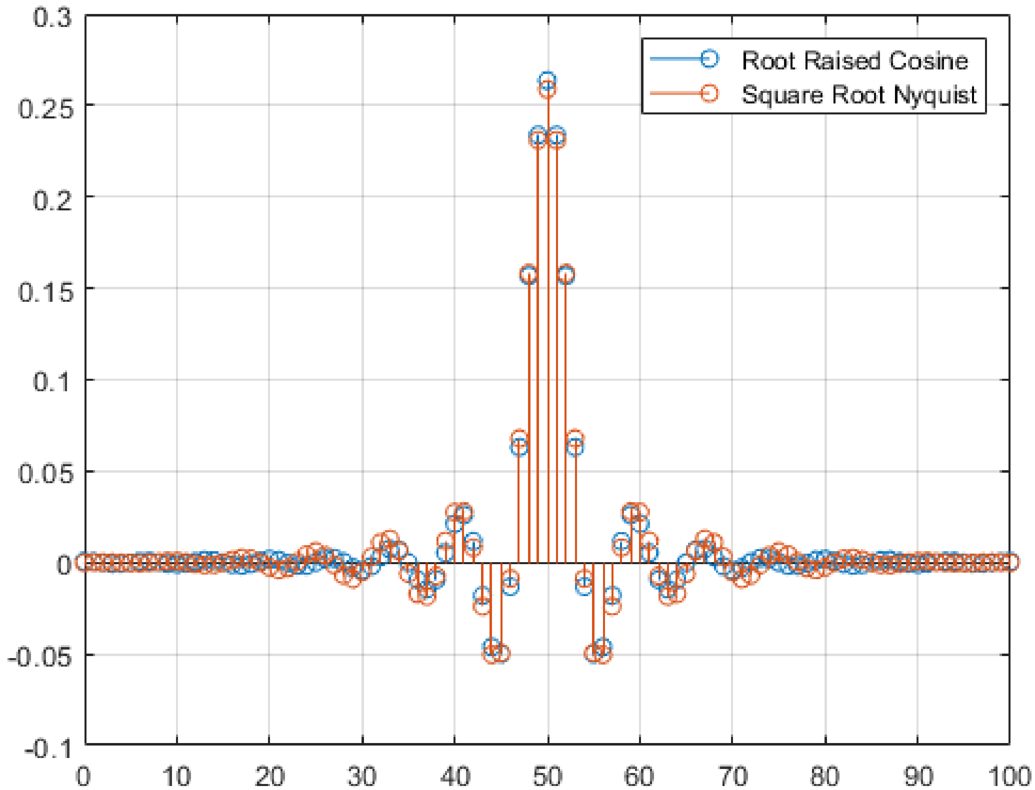
```
hrrc=rcosdesign(beta,span,Nsps);
hrrc=hrrc*(1/Nsps+(beta/Nsps)*(4/pi-1))/max(hrrc);
% Or you could scale by the DC value, try this by commenting out
% the previous statement and uncommenting the following statement
% Then examine the passband gain by expanding the y-axis.
% hrrc=hrrc/sum(hrrc); %scale by DC value
```

Square Root Nyquist Pulse

```
fb = [0 fp fc fc fs .5]*2;
a = [1 1 1/sqrt(2) 1/sqrt(2) 0 0];
wght = [2.4535 1 1];
hsrn=firpm(M,fb,a,wght);
```

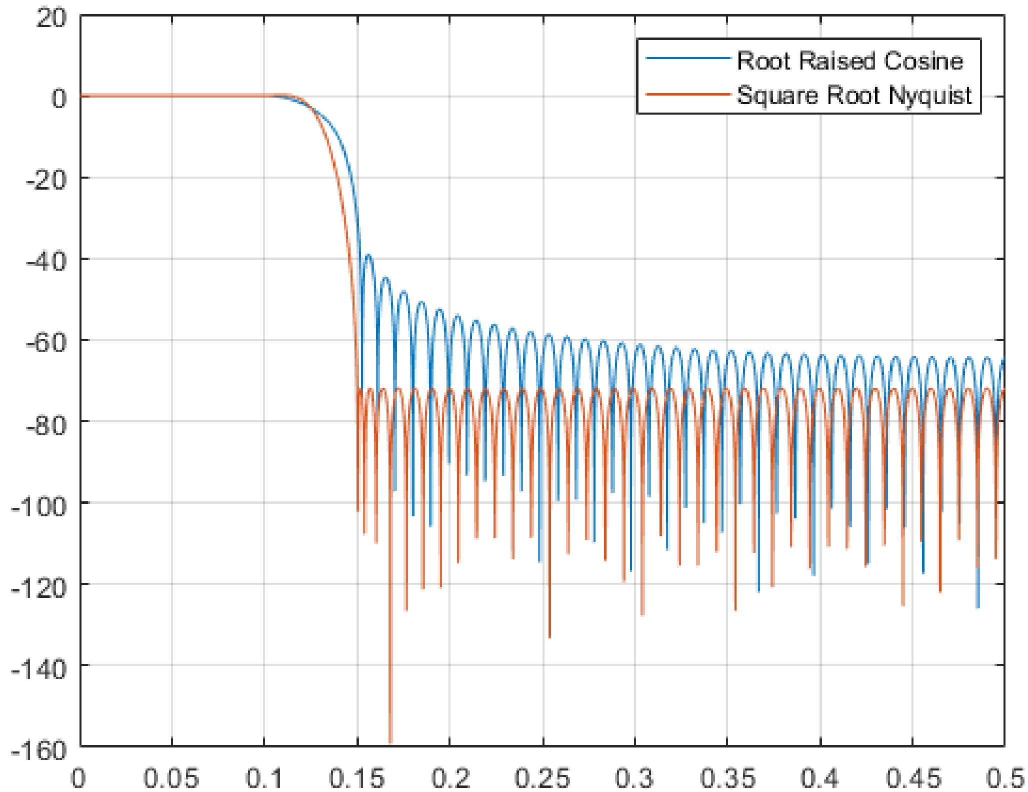
Plot Impulse Responses

```
figure
n=0:length(hrrc)-1;
stem(n,hrrc)
hold on
n=0:length(hsrn)-1;
stem(n,hsrn)
legend('Root Raised Cosine','Square Root Nyquist')
grid
```



Plot Frequency Spectrums

```
f=[0:.0001:.5];
[Hrrc,w]=freqz(hrrc,1,f*2*pi);
figure
plot(f,20*log10(abs(Hrrc)))
hold on
[Hsrn,w]=freqz(hsrn,1,f*2*pi);
plot(f,20*log10(abs(Hsrn)))
legend('Root Raised Cosine','Square Root Nyquist')
grid
```



Published with MATLAB® R2018a

Question 7

Contents

Define Intial Values

```
beta=0.2;
Nsps=4;
span=25;
M=span*Nsps; %order of the filter
fc=1/(2*Nsps);
fp=(1-beta)*fc;
fs=(1+beta)*fc;
```

Square root raised cosine

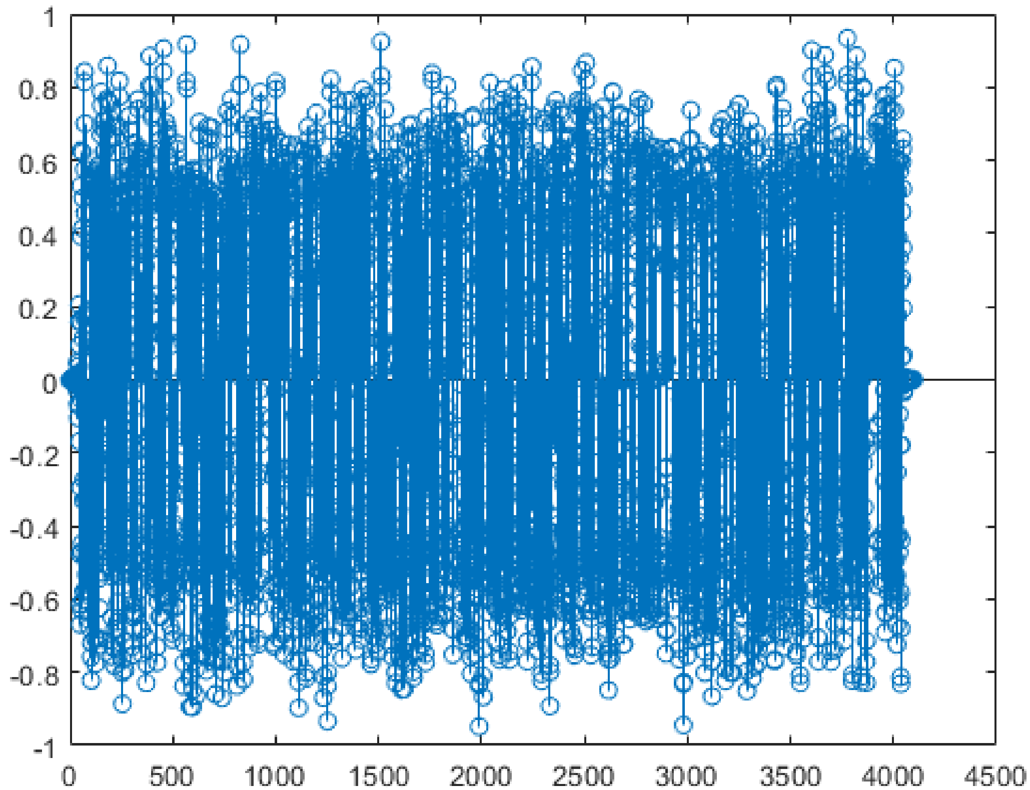
```
hrrc=rcosdesign(beta,span,Nsps);
```

Generate a Random Input Sequence

```
nbits = 1000;
input_seq = (floor(2*rand(1,nbits))-0.5)/0.5; % 1,-1
input_ps = reshape([input_seq;zeros(Nsps-1,nbits)],1,Nsps*nbits);
```

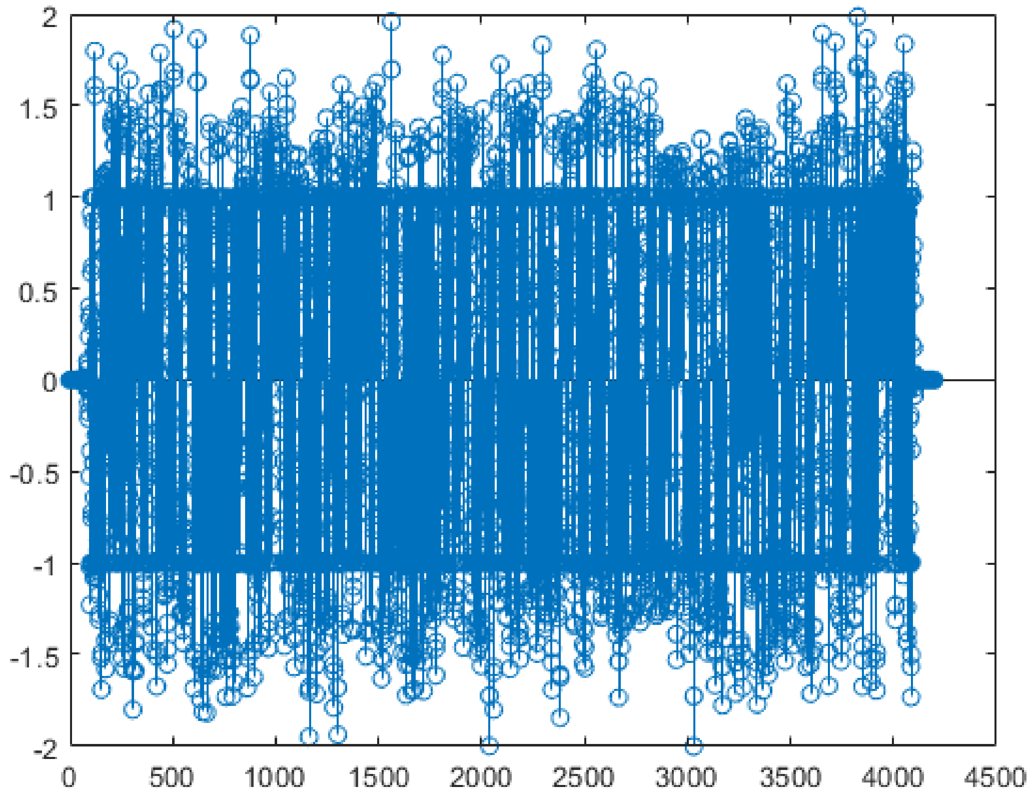
Generate and Plot the Output of the Pulse Shaping Filter

```
output_ps = conv(input_ps,hrrc);
n=0:length(output_ps)-1;
figure
stem(n,output_ps)
```



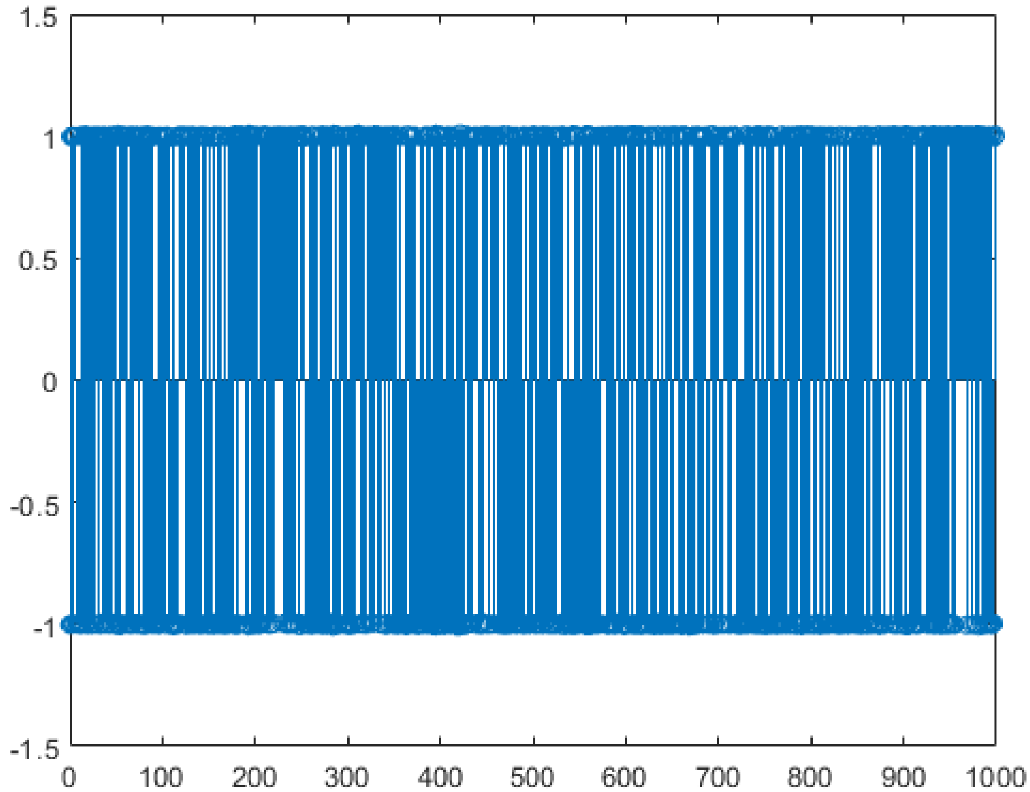
Generate and Plot the Output of the Matched Filter

```
output_mf = conv(output_ps,hrrc);
n=0:length(output_mf)-1;
figure
stem(n,output_mf)
```



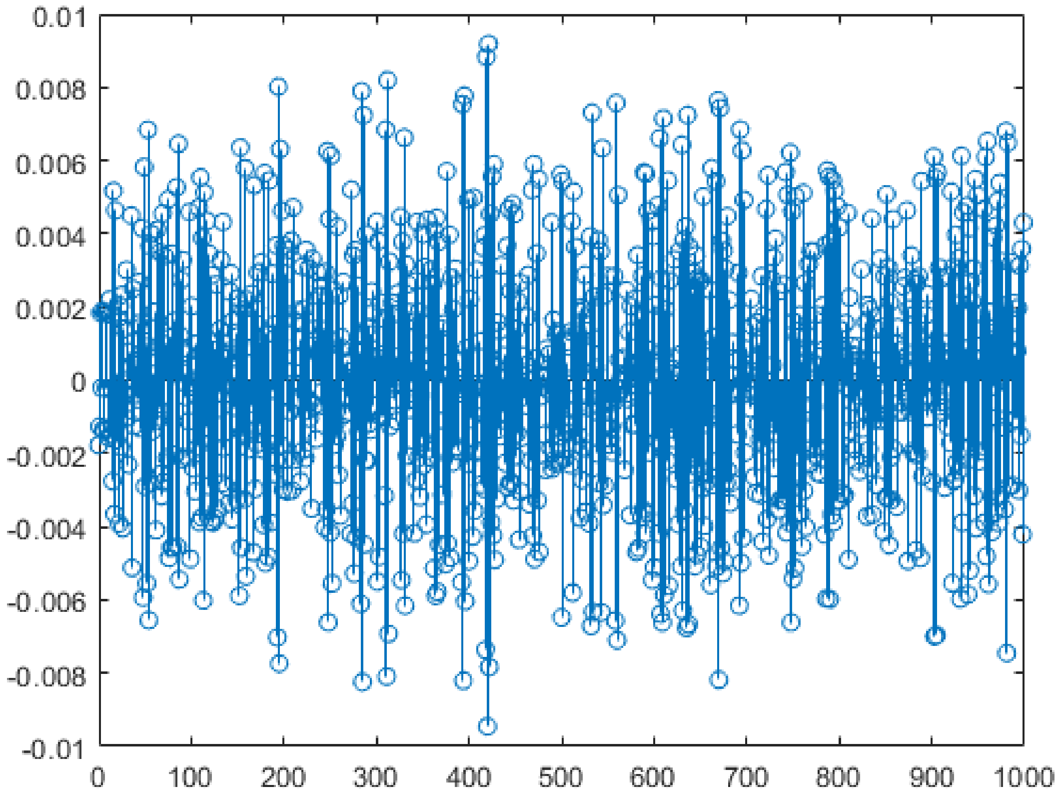
Generate and Plot the Output Sequence Samples

```
output_seq = output_mf(M+1:Nsps:end-M); % discard transients
figure
stem([0:length(output_seq)-1],output_seq)
```



Determine ISI and Plot

```
isi_seq = output_seq - input_seq;
figure
stem([0:length(isi_seq)-1],isi_seq)
```



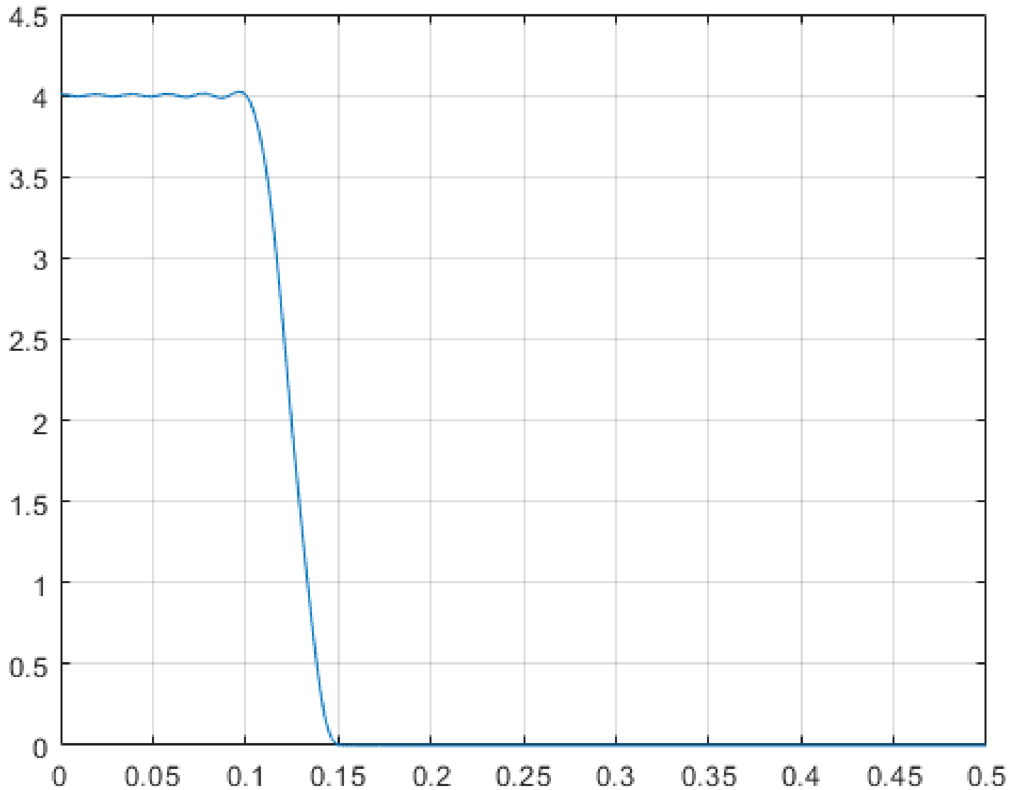
Determine RMS value of ISI

```
rms_isi = sqrt(mean(isi_seq.^2))
```

```
rms_isi =

   0.003342766145759
```

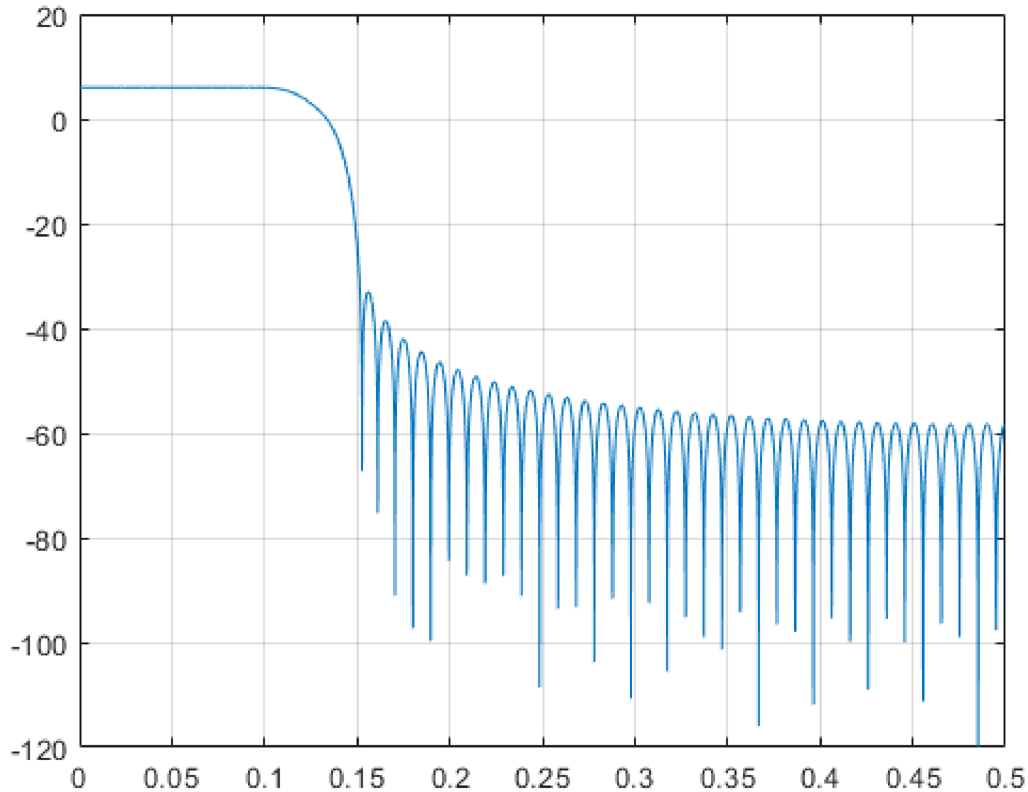
Determine and Plot the $h_{rrc} * h_{rrc}$ Magnitude Response

```
hrc = conv(hrrc,hrrc);
f = [0:.0001:.5];
[Hrc,w] = freqz(hrc,1,f*2*pi);
figure
plot(f,abs(Hrc)),grid
```



Determine and Plot the h_{rrc} Magnitude Response (dB)

```
[Hrrc,w] = freqz(hrrc,1,f*2*pi);
figure
plot(f,20*log10(abs(Hrrc))),grid
```



Published with MATLAB® R2018a

Question 9

Contents

Define Intial Values

```
beta=0.2;
Nsps=4;
span=25;
M=span*Nsps; %order of the filter
fc=1/(2*Nsps);
fp=(1-beta)*fc;
fs=(1+beta)*fc;
```

Square Root Nyquist

```
fb = [0 fp fc fc fs .5]*2;
a = [1 1 1/sqrt(2) 1/sqrt(2) 0 0];
wght = [2.4535 1 1];
% Implement approach from the Harris paper
% First initialize for the while loop
h=firpm(M,fb,a,wght); % generate the initial impulse response
cnt=0; % iteration count for interest only
hconv=conv(h,h);
hconv(M+1)=0; % zero the peak value
isi_total_prev = sum(abs(hconv(1:Nsps:end)));
isi_total = isi_total_prev;
cstep=0.005;  % step size increment for fp
c=1;
% the second inequality in the while loop keeps fp<fc
while isi_total <= isi_total_prev && c <.9*fc/fp
    isi_total_prev = isi_total;
    c=c+cstep;
    fb = [0 c*fp fc fc fs .5]*2;
    h=firpm(M,fb,a,wght);
    hconv=conv(h,h);
    hconv(M+1)=0;
    isi_total = sum(abs(hconv(1:Nsps:end)));
```

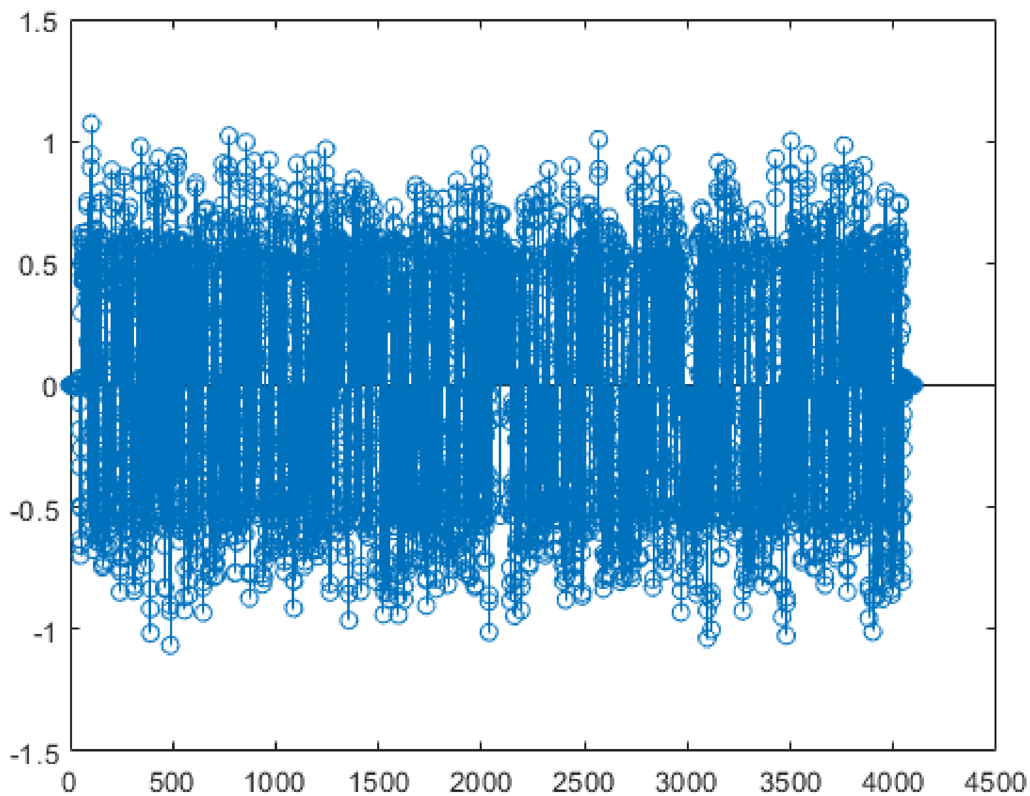
```
    cnt=cnt+1;
end
hsrn=h/sqrt(sum(h.^2)); % Normalize to unit energy
```

Generate Random Input Sequence for Pulse Shaping Filter

```
nbits = 1000;
input_seq = (floor(2*rand(1,nbits))-0.5)/0.5; % 1,-1
input_ps = reshape([input_seq;zeros(Nsps-1,nbits)],1,Nsps*nbits);
```

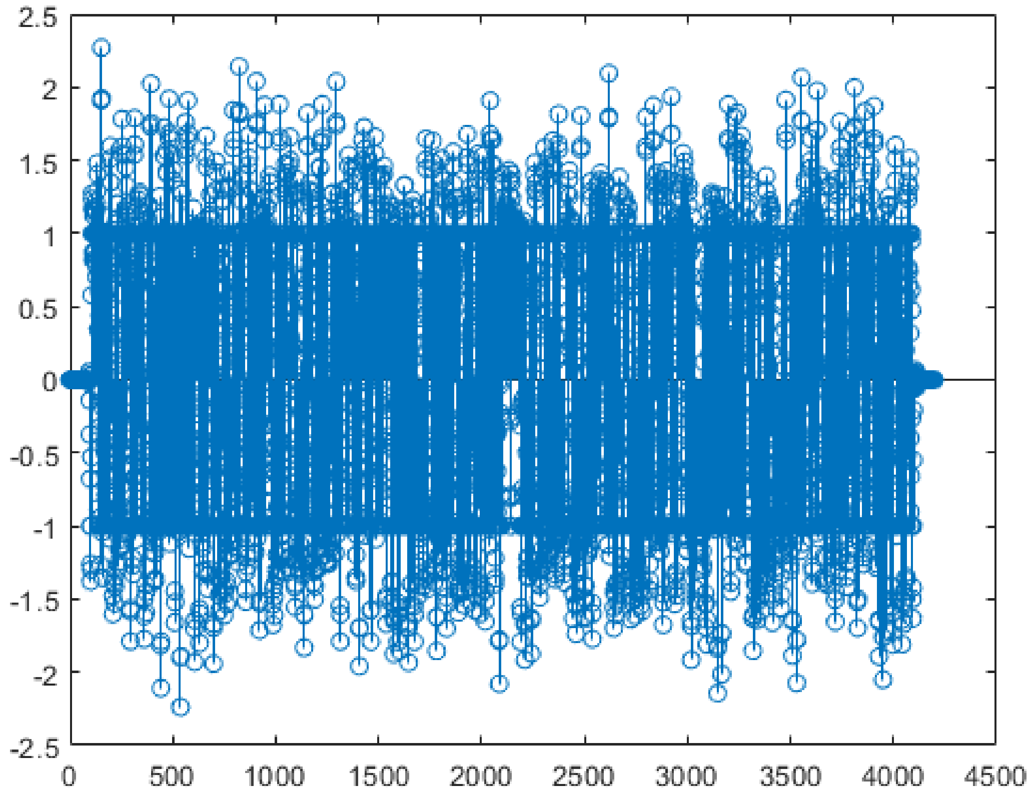
Generate Output of Pulse Shaping Filter

```
output_ps = conv(input_ps,hsrn);
n=0:length(output_ps)-1;
figure
stem(n,output_ps)
```



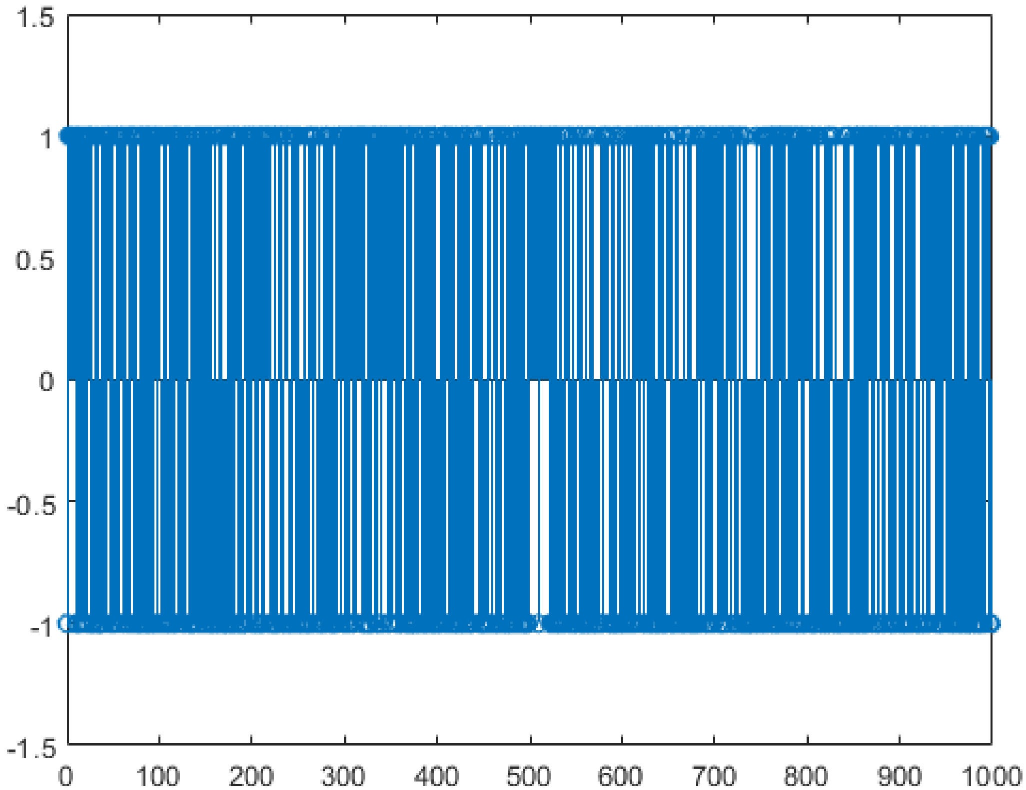
Generate Output of Matched Filter

```
output_mf = conv(output_ps,hsrn);
n=0:length(output_mf)-1;
figure
stem(n,output_mf)
```



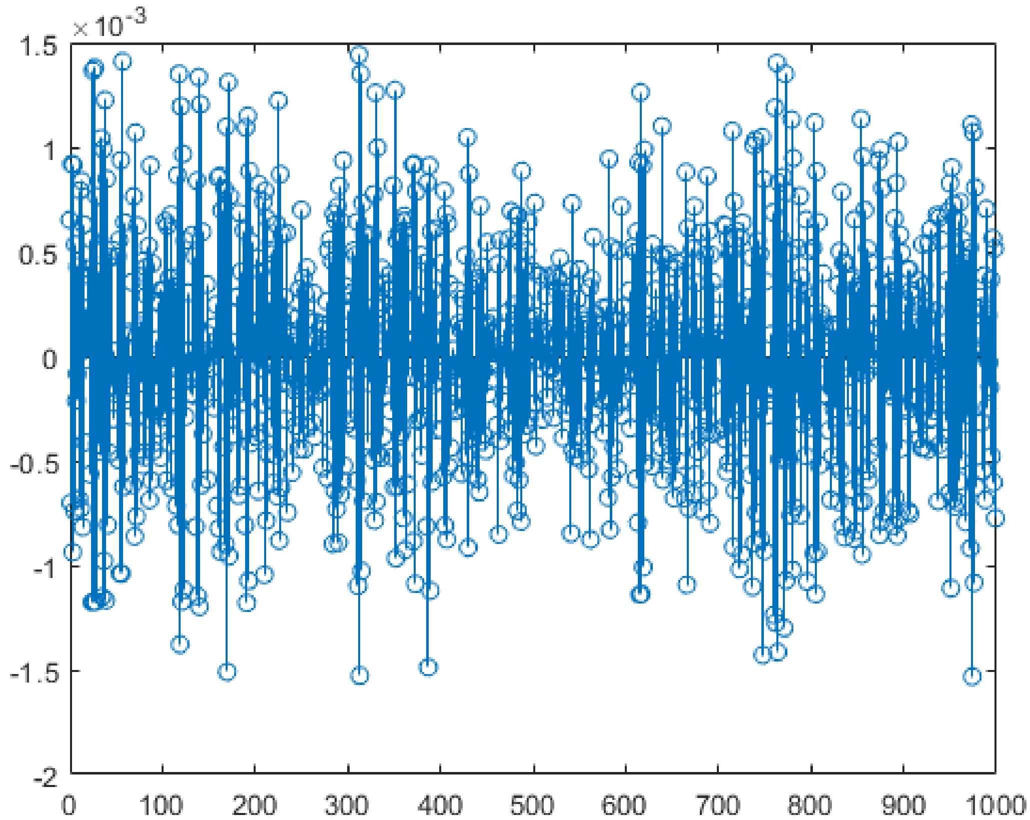
Generate and Plot Output Sequence Samples

```
output_seq = output_mf(M+1:Nsps:end-M); % discard transients
figure
stem([0:length(output_seq)-1],output_seq)
```



Determine ISI and Plot

```
isi_seq = output_seq - input_seq;
figure
stem([0:length(isi_seq)-1],isi_seq)
```



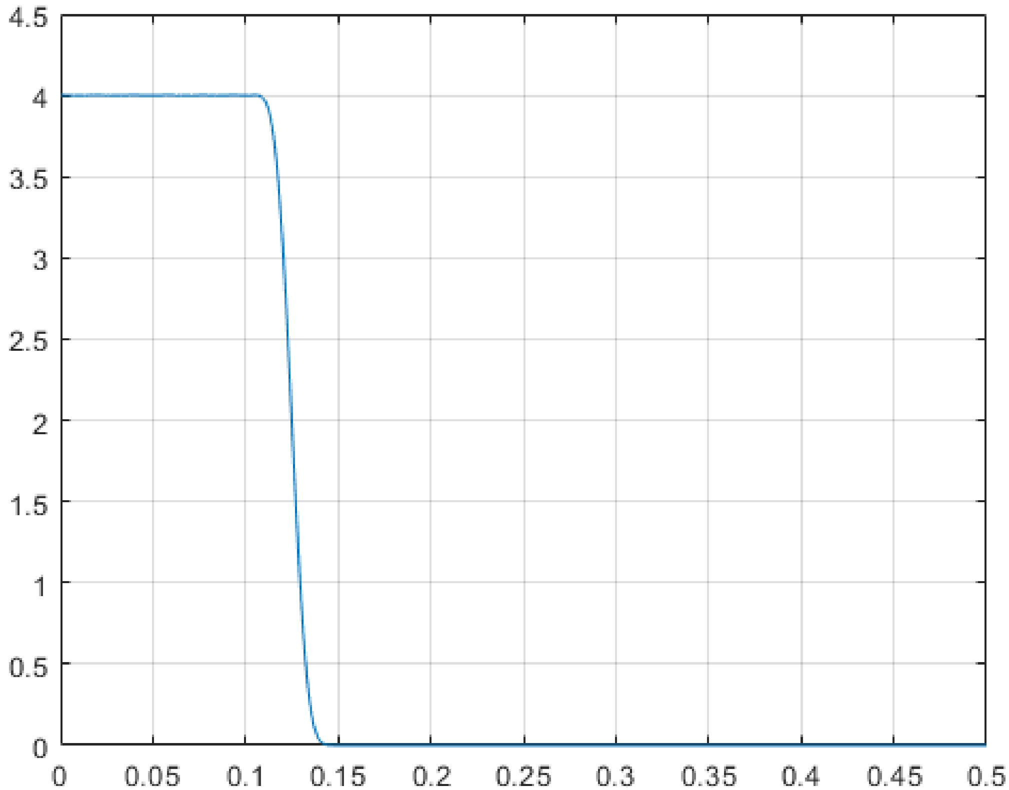
Determine RMS value of ISI

```
rms_isi = sqrt(mean(isi_seq.^2))
```

```
rms_isi =

     5.621332475800471e-04
```

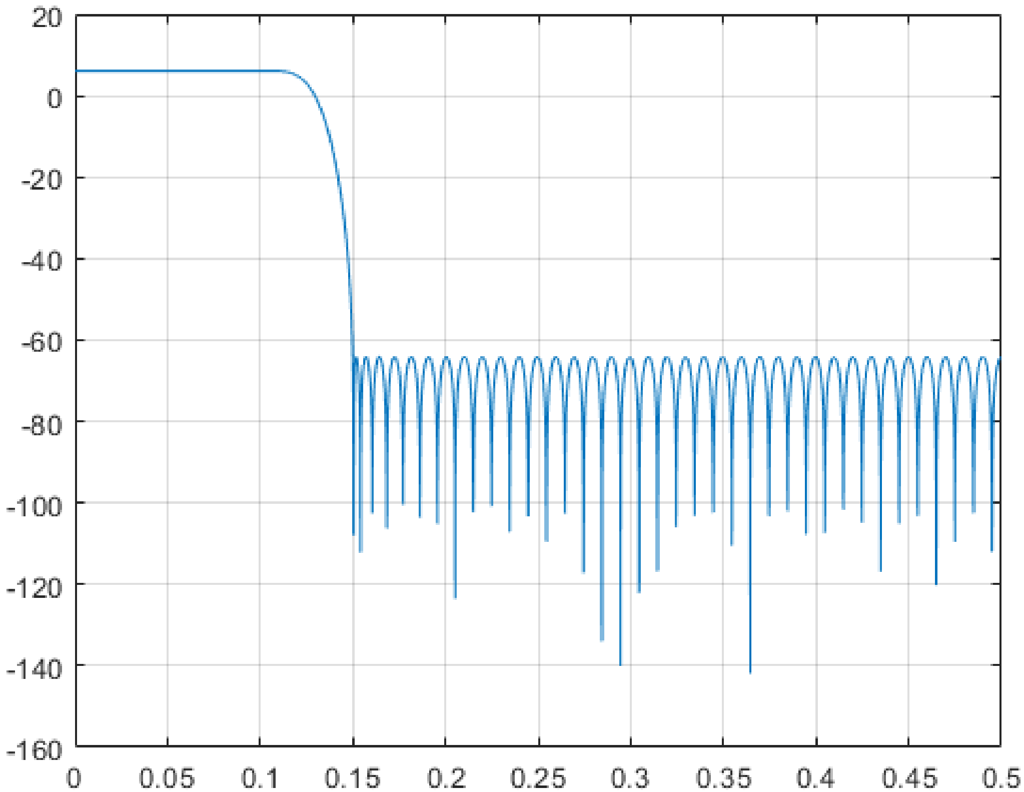
Determine and Plot the $h_{srn} * h_{srn}$ Magnitude Response

```
hn = conv(hsrn,hsrn);
f = [0:.0001:.5];
[Hn,w] = freqz(hn,1,f*2*pi);
figure
plot(f,abs(Hn)),grid
```



Determine and Plot the h_{srn} Magnitude Response (dB)

```
[Hsrn,w] = freqz(hsrn,1,f*2*pi);
figure
plot(f,20*log10(abs(Hsrn))),grid
```


20
0
-20
-40
-60
-80
-100
-120
-140
-160
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5