Communication System Lab 4 Part 2

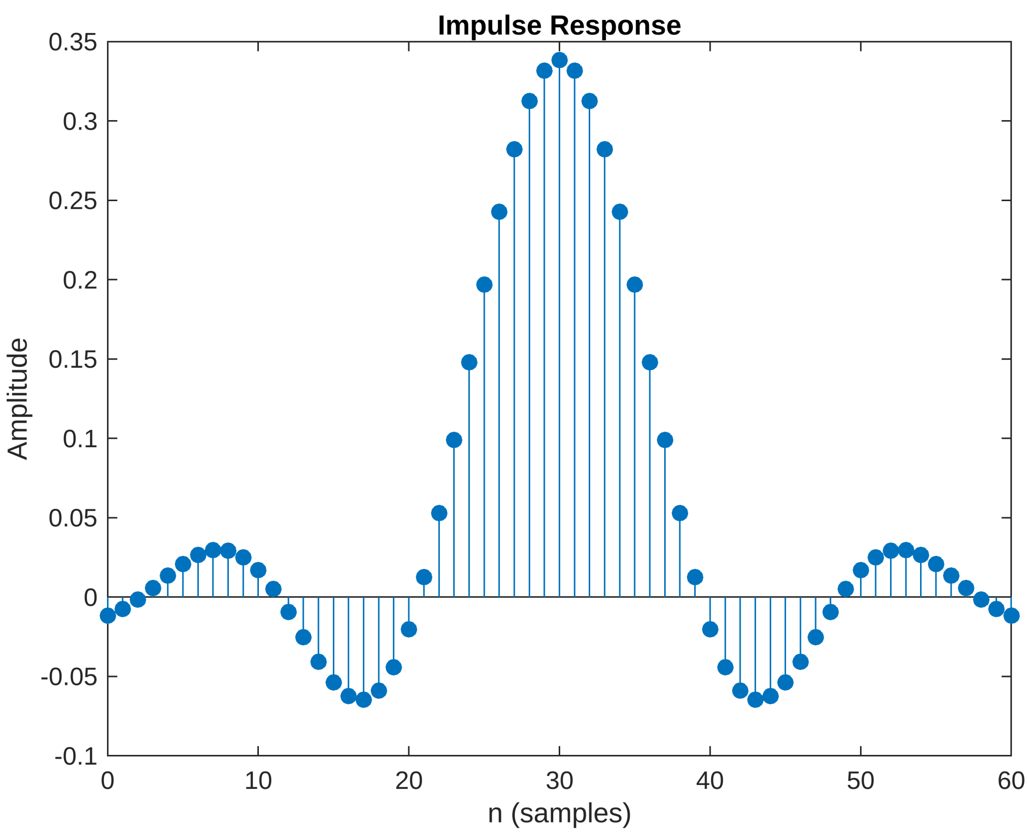
By: Saurabh Kumar (SC22B146)

End to end system with SRRC

1. Generate the line code brc(t) corresponding to a random sequence of 100 bits for Tb = 0.1s using the SRRC pulse shape (suitably truncated as in the case of sinc and raised cosine done earlier).

```
% paramters
Tb = 0.1;
fs = 100;
beta = 0.25;
N = 100;

brc = rcosdesign(beta, 6, Tb*fs,'sqrt');
impz(brc);
```

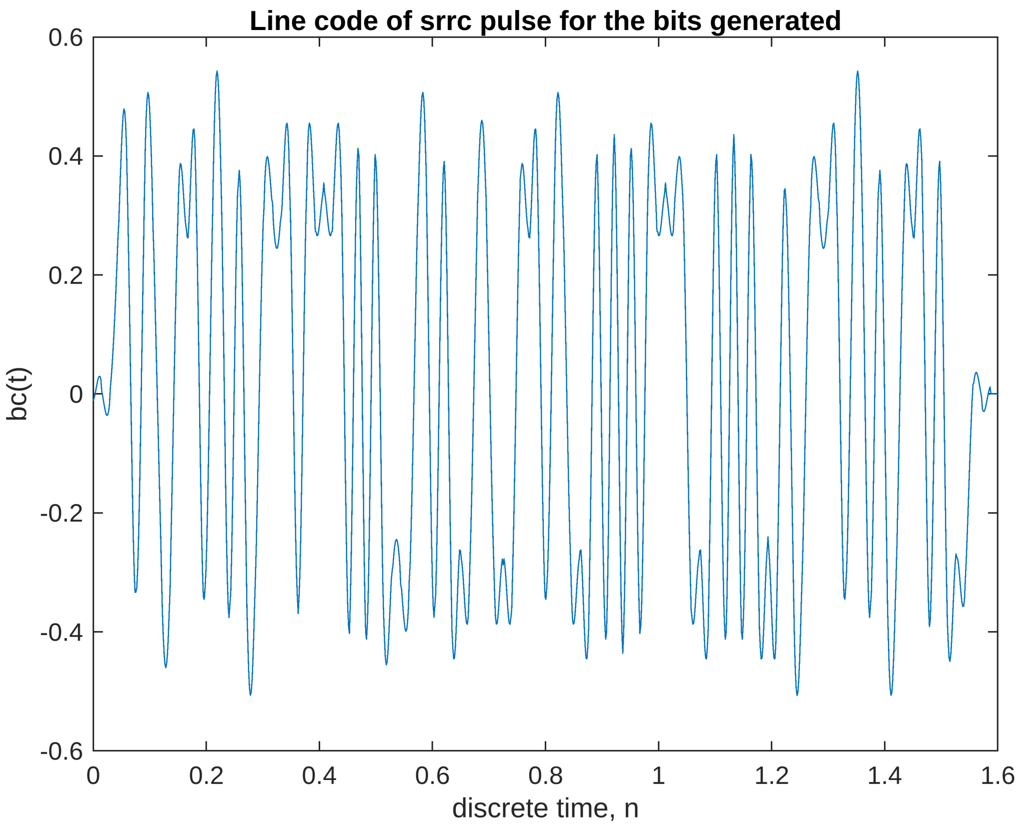



```
% source bits
rng("default");
source = rand(1, N) > 0.5;
source = double(source);
source(source==0) = -1;
disp(['Source bits: ', num2str(source)]);
```

```
Source bits: 1  1 -1  1  1 -1 -1  1  1  1 -1  1  1 -1  1 -1 -1  1  1  1  1 -1  1  1  1  1  1 -1  1 -1  1 -
```

```
sig_len = N*Tb+6*Tb;
signal_brc = zeros(1,sig_len*fs);
for i=1:(length(source))
    signal_brc(round((i-1)*Tb*fs+1):round((i+5)*Tb*fs+1)) =
signal_brc(round((i-1)*Tb*fs+1):round((i+6-1)*Tb*fs+1)) + source(i)*brc;
end

t = linspace(0,16*Tb,length(signal_brc));
figure;
plot(t,signal_brc);
title("Line code of srrc pulse for the bits generated");
xlabel("discrete time, n");
ylabel("bc(t)");
```

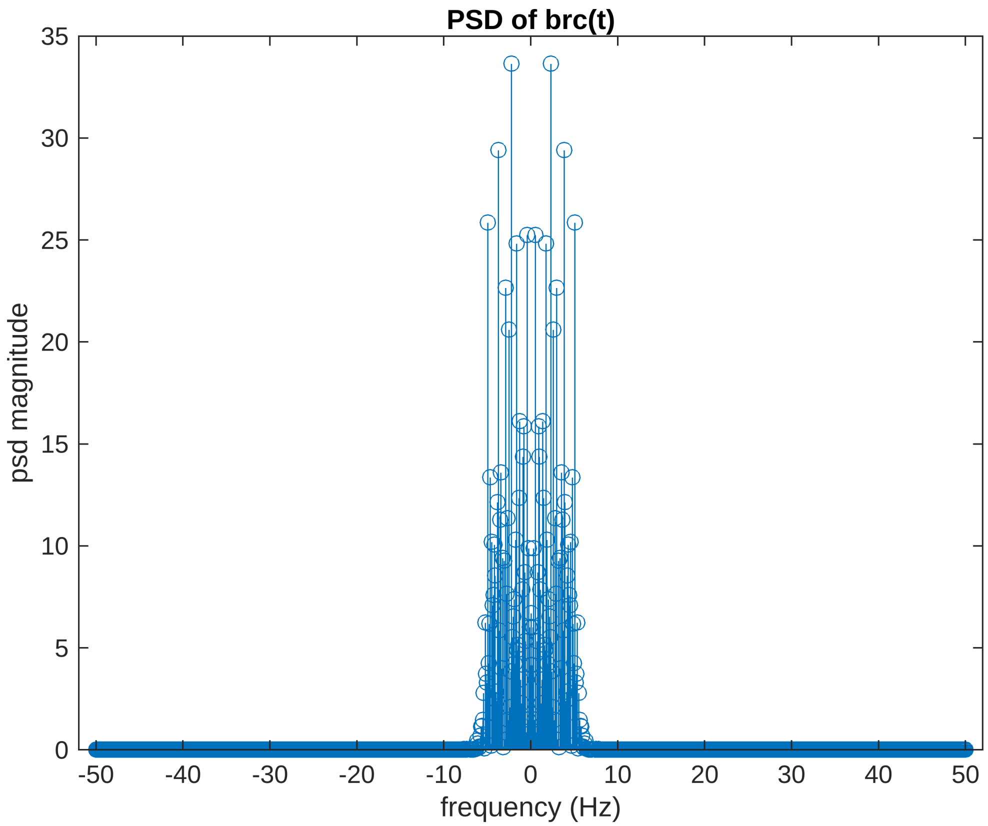


Inference: The pulse is generated using the built-in function 'rcosdesign' and the line code is plotted for 100 bits.

2. Plot the PSD as well as the eye diagram for brc(t). Record the plots as well as your observations. Compare with the PSD and eye diagram for sinc and raised cosine pulses designed for the same Tb.

```
% psd
psd_brc = fftshift((abs(fft(signal_brc)).^2)/N);
f = linspace(-fs/2,fs/2,length(psd_brc));
figure;
```

```
stem(f,psd_brc);
title("PSD of brc(t)");
xlabel("frequency (Hz)");
ylabel("psd magnitude");
```




```
% eye diagram
eyediagram(signal_brc, 2*Tb*fs, 1, round(Tb*fs/2));
```

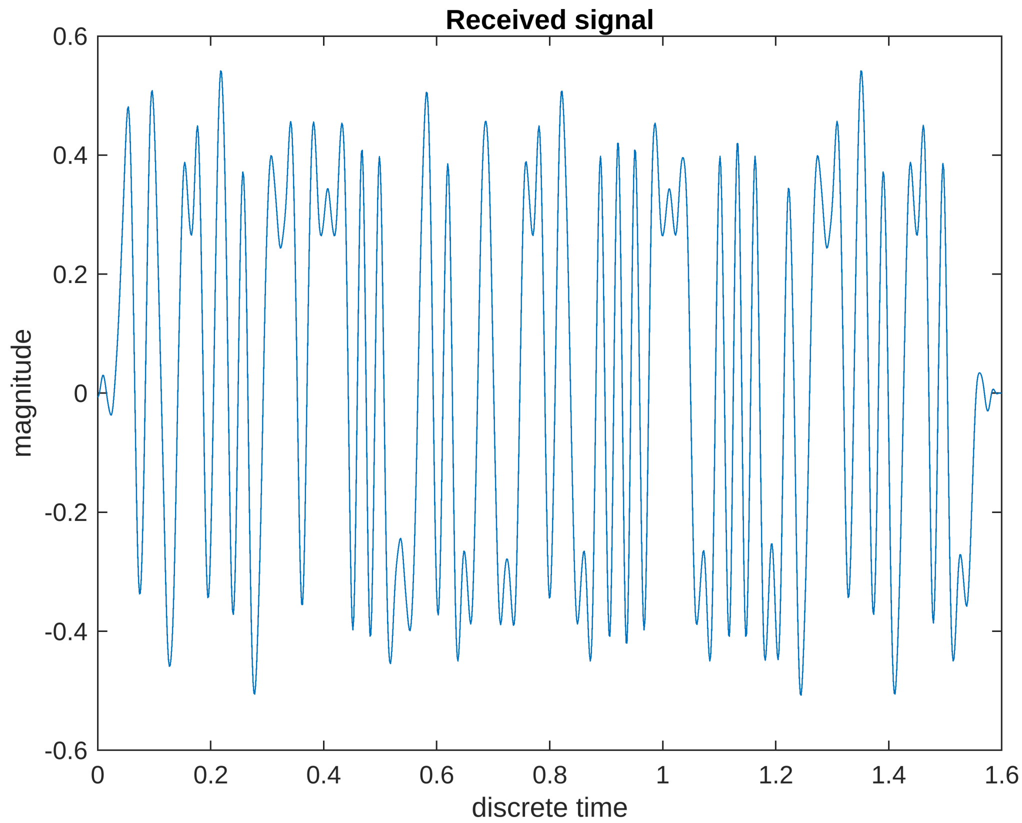
3. Simulate transmission of brc(t) through a channel which is a low pass filter with passband edge frequency of 10Hz and unit gain in the passband (implemented using the function developed in this labsheet).

```
function y = channel_with_no_plot(x,fs,fc,g)
    h = firpm(75, [0 fc fc+10, fs/2]/(fs/2), [g g 0 0]);
    y = conv(x, h, 'same');
end

% received signal
channel_fs = 100; channel_fc = 10; channel_g = 1;
received_signal =
channel_with_no_plot(signal_brc,channel_fs,channel_fc,channel_g);

t = linspace(0,16*Tb,length(signal_brc));
figure;
plot(t,received_signal);
title("Received signal");
```

```
xlabel("discrete time");
ylabel("magnitude");
```

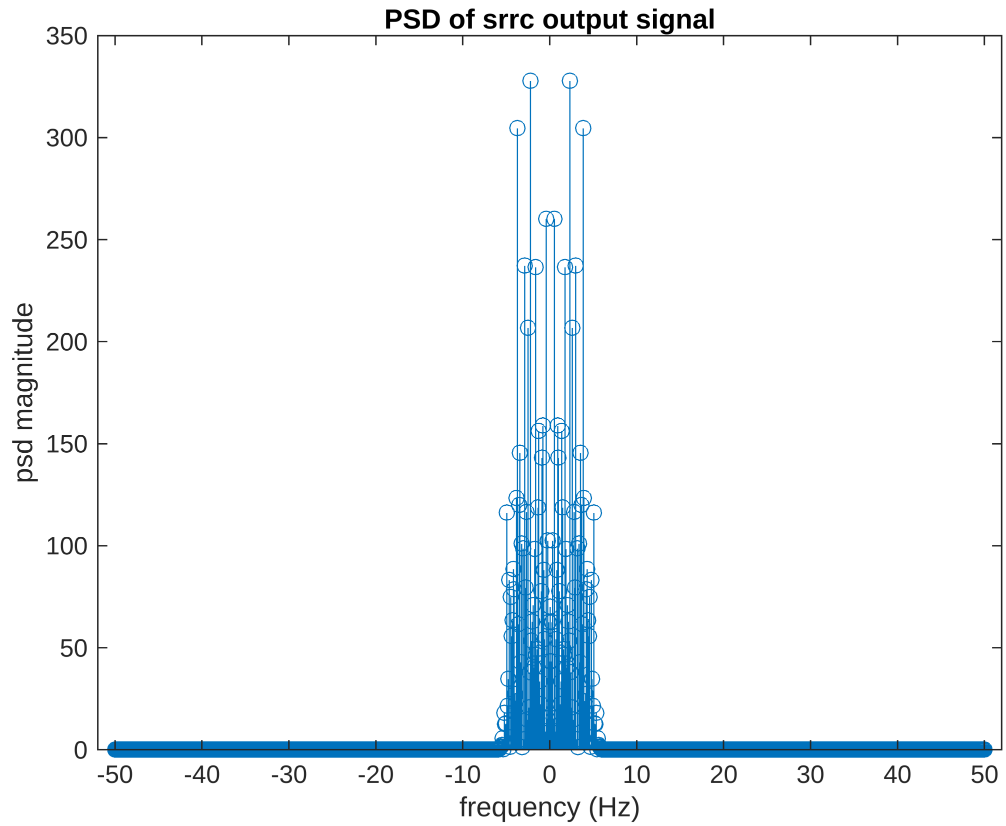


4. Implement a receive filter which has a SRRC impulse response which is matched to the pulse shape using for generating brc(t). Filter the channel output using this SRRC receive filter. Plot the PSD as well as the eye diagram of the output of the receive filter. Record the plots as well as your observations. Compare with the PSD and eye diagram for sinc and raised cosine pulses designed for the same Tb.

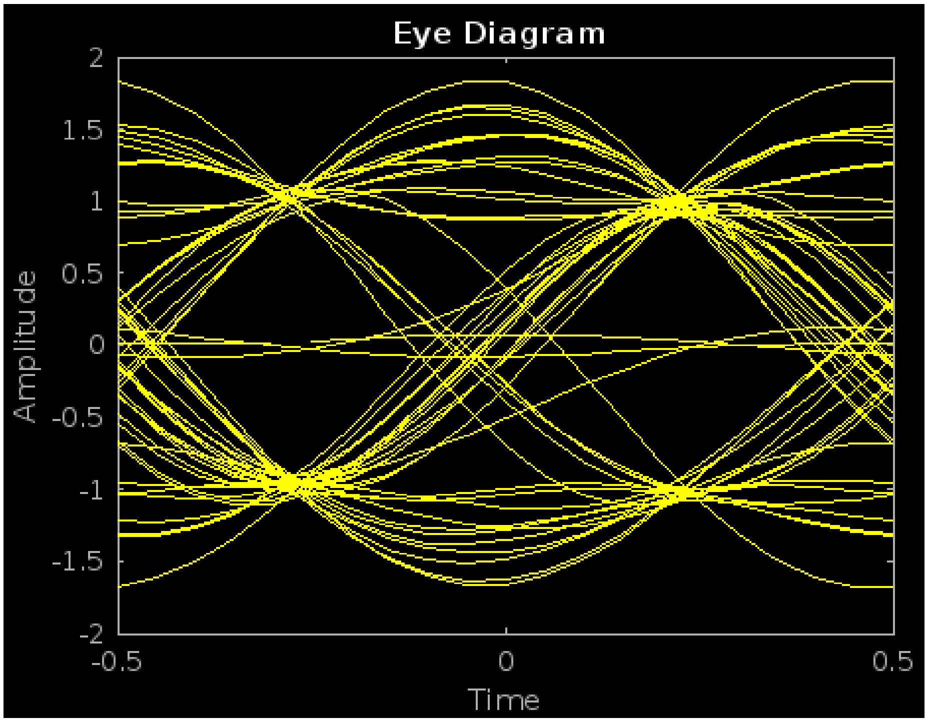
```
% matched filter and filtered signal
h_srrc = rcosdesign(beta, 6, Tb*fs,'sqrt');
filtered_signal = conv(received_signal, h_srrc, 'same');

% psd
psd_srrc = fftshift((abs(fft(filtered_signal)).^2)/N);

f = linspace(-fs/2,fs/2,length(psd_srrc));
figure;
stem(f,psd_srrc);
title("PSD of srrc output signal");
xlabel("frequency (Hz)");
ylabel("psd magnitude");
```




```
% eye diagram
eyediagram(filtered_signal, 2*Tb*fs, 1, round(Tb*fs/2));
```



Inference: The signal is passed through the channel and the matched filter specified, and the output psd and the eye diagram is plotted. The eye diagram is more clear and wider than the sinc and raised cosine linecodes, indicated a higher signal-to-noise ratio.