
```
clear;
clc;
% Generate a binary sequence with length of 100
L = 100;
os = round(rand(1,L));
os(os==0) = -1;

% Plot both the original signal
figure(1)
subplot(311)
set(gcf,'position', [250 200 1200 600]);
stem(os)
grid on;
title("The original signal");
xlabel("time (s)")
ylabel("voltage (V)")
% Define the symbol transmission rate
Rb = 1e4;
Fs = 1e5;
% Upsample the sequence to make symbols
N = Fs/Rb;
os_upsampled = upsample(os, N);
% Let the Nyquist rate be 5kBaud and 20kBaud
Rs1 = 5e3;
Rs2 = 20e3;
% Generate the corresponding roll-off systems
Fs = 1e5;
span = 20;
alpha = 0.5;
sps1 = Fs/Rs1;
sps2 = Fs/Rs2;
h1 = rcosdesign(alpha, span, sps1);
h2 = rcosdesign(alpha, span, sps2);
% Signal and systems convolution
rs1 = conv(os_upsampled, h1, 'same');
rs2 = conv(os_upsampled, h2, 'same');

% Downsample the receive signal
rs1_downsampled = downsample(rs1, N);
rs2_downsampled = downsample(rs2, N);
% Perform sample decision and convert it into waveform
Vd = 0;
rs1_decision = [];
rs2_decision = [];
for i=1:length(rs1_downsampled)
    if rs1_downsampled(i)>=Vd
        rs1_decision = [rs1_decision, ones(1, 1)];
    else
        rs1_decision = [rs1_decision, -ones(1, 1)];
    end
    if rs2_downsampled(i)>=Vd
        rs2_decision = [rs2_decision, ones(1, 1)];
    end
end
```

```

        else
            rs2_decision = [rs2_decision, -ones(1, 1)];
        end
    end
end
% Plot the received signal1
subplot(312)
stem(rs1_decision)
grid on;
title("The original signal");
xlabel("time (s)")
ylabel("voltage (V)")
% Plot the received signal2
subplot(313)
stem(rs2_decision)
grid on;
title("The original signal");
xlabel("time (s)")
ylabel("voltage (V)")

figure
subplot(211)
plot(rs1(1:30*sps2))
hold on
plot(os_upsampled(1:30*sps2)/2)
subplot(212)
plot(rs2(1:30*sps2))
hold on
plot(os_upsampled(1:30*sps2)/2)

```

Published with MATLAB® R2021a