Pulse Shaping and Matched Filtering

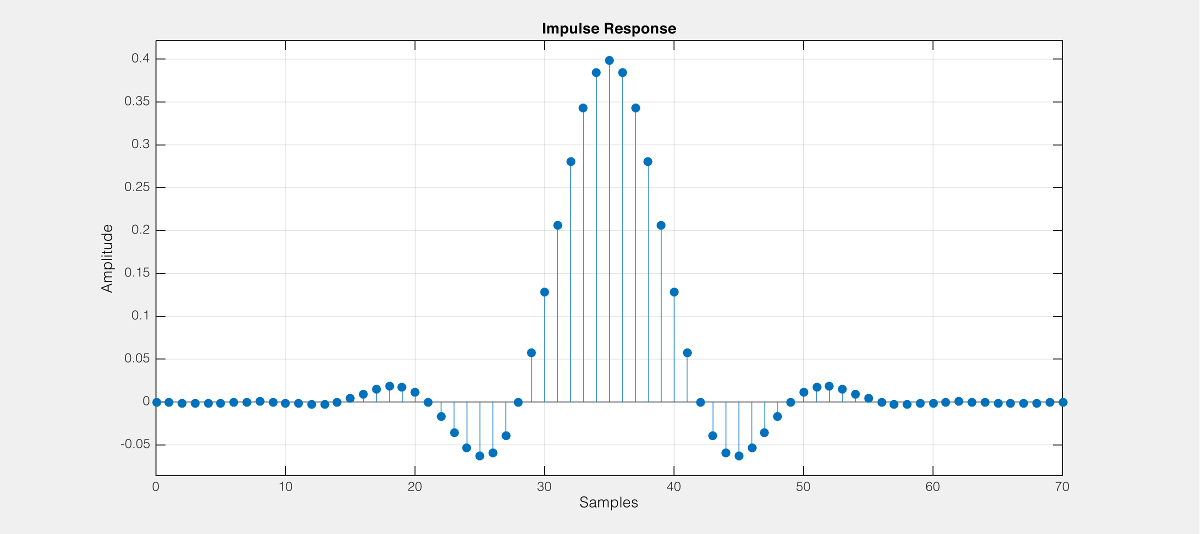
1. Generate a string of message bits.
2. Use root rasied cosine pulse p(t) as the shapig pulse, and generate the corresponding baseband signal with a fixed bit duration Tb. You may use roll-off factor as α = 0.4.
3. Simulate transmission of baseband signal via an AWGN channel
4. Apply matched filter with frequency response Pr(f) = P ∗(f) to the received signal.
5. Sample the signal at mTb and compare it against the message sequence

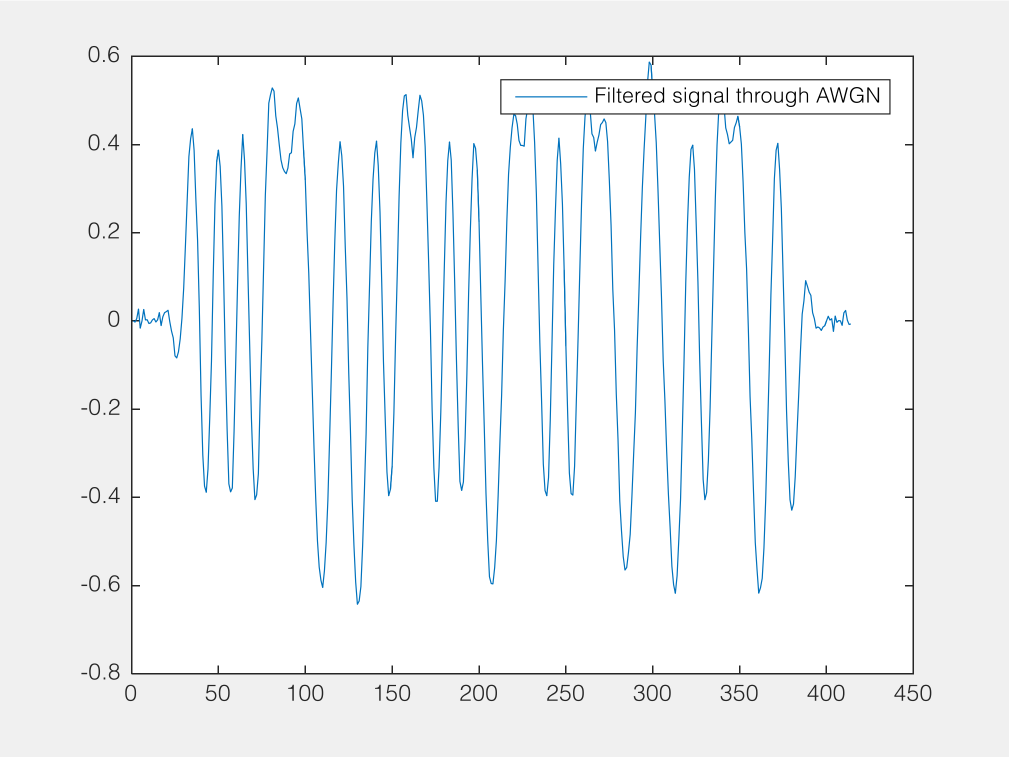
Program name: IMPL_PulseSp_MF.m

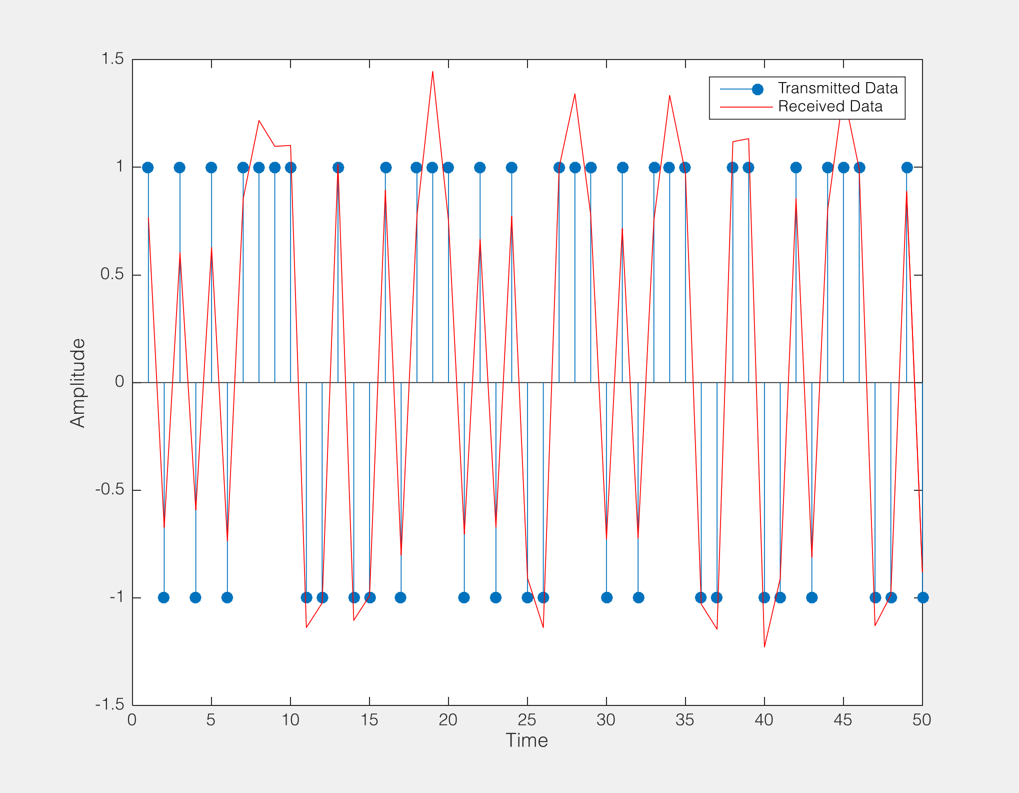
```
% Code made by Manu Prasad (IMPLearn)
% Program for pass a signal through a square-root, raised
cosine filter.
% Pulse Shaping and Matched Filtering
close all;
clear;
rolloff = 0.4;        % Rolloff factor
span = 10;            % Filter span in symbols
sps = 7;              % Samples per symbol
M = 16;               % Modulation order
k = log2(M);          % Bits per symbol

% Generate the square-root, raised cosine filter coefficients.
rctFilt = rcosdesign(rolloff, span, sps,'normal');
fvtool(rctFilt,'Analysis','impulse')

% Create a vector of bipolar data.
BP_Data = 2*randi([0 1], 50, 1) - 1;

% Upsample and filter the data for pulse shaping.
UP_s = upfirdn(BP_Data, rctFilt, sps,1);

% Using the number of bits per symbol (k)
% and the number of samples per symbol (sps),
% convert the ratio of energy per bit to noise power spectral
density (EbNo)
% to an SNR value for use by the awgn function.
EbNo = 100;
snr = EbNo + 10*log10(k) - 10*log10(sps);
filtlen = 10;         % Filter length in symbols

rxSignal = awgn(UP_s,snr,'measured');% filtering the signal
through an AWGN channel.

% Add noise.
rxSignal = rxSignal + randn(size(rxSignal))*.01;

rxFiltSignal = upfirdn(rxSignal,rctFilt,1,sps);       %
Downsample and filter
```

```
rxFiltSignal = rxFiltSignal(filtlen + 1:end - filtlen); % Account for delay

% Plotting the function
figure;
stem(BP_Data,'filled')
hold on
plot(rxFiltSignal,'r')
xlabel('Time'); ylabel('Amplitude');
legend('Transmitted Data','Received Data')
figure;
plot(rxSignal)
legend('Filtered signal through AWGN')

eyediagram(BP_Data,2);legend('Transmitted signal')
eyediagram(rxFiltSignal,2);legend('Recieved signal')
```

Output




Filtered signal through AWGN
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
50
100
150
200
250
300
350
400
450


Transmitted Data
Received Data
Amplitude
Time
1.5
1
0.5
0
-0.5
-1
-1.5
0
5
10
15
20
25
30
35
40
45
50