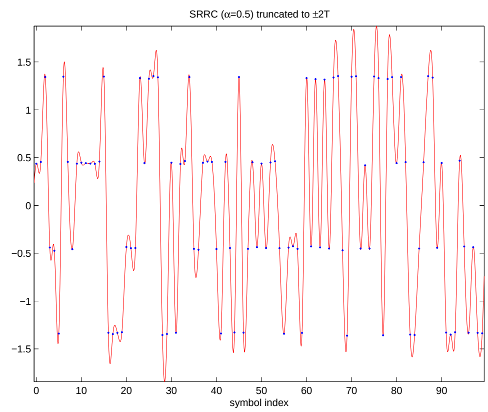
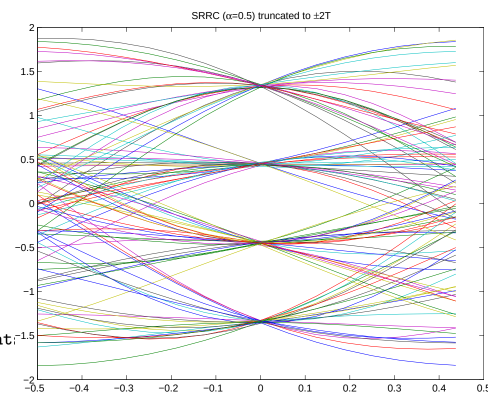
HOMEWORK SOLUTIONS #7

1. The MATLAB code for 4-PAM transmission, along with a plot of the output signal and the eye diagram, appears below.

```
% design SRRC
P = 16; % oversampling factor
alpha = 0.5; % SRRC rolloff param
D = 2; % truncation to [-DT,DT]
g = srrc(D,alpha,P); % SRRC pulse
Ng = length(g); % pulse length

% generate symbols
N = 100; % # symbols
M = 4; % alphabet size
sig2a = 1; % symbol variance
a = pam(N,M,sig2a);  % symbol sequence

% pulse-amplitude modulate
a_up = zeros(1,N*P);
a_up(1:P:end) = a;  % upsampled symbols
m = conv(a_up,g); % PAM

% matched-filter demodulate
y_up = conv(m,g); % use SRRC again

% remove causal filtering delay
dly = (Ng-1)/2+(Ng-1)/2;% delay due to pulses
y_up = y_up([1:P*N]+dly);% remove delay
y = y_up(1:P:P*N); % downsample

% plot received signal
figure(1)
plot([0:P*N-1]/P,y_up,'r',[0:N-1],y,'.');
axis('tight')
title(['SRRC (\alpha=',num2str(alpha),...
       ') truncated to \pm',num2str(D),'T'])
xlabel(['symbol index'])

% plot eye diagram
figure(2)
Y_up = reshape(y_up(P/2+[1:P*(N-1)]),P,N-1); % extract N-1 segment
plot([0:P-1]/P-1/2,Y_up) % superimpose segments
title(['SRRC (\alpha=',num2str(alpha),...
       ') truncated to \pm',num2str(D),'T'])
xlabel(['relative symbol index'])
```





In the plots above, where $\alpha = 0.5$, it can be seen that the recovered symbols $y[m]$ match the transmitted symbols $a[m]$ closely (though not perfectly); the eye is seen to be "open." The plots on the next page show that the eye is nearly closed when $\alpha = 0.25$, while there is a near-perfect match between $y[m]$ and $a[m]$ when $\alpha = 1$.

We can understand the effect of α as follows. When α is large, the pulse $g(t)$ decays quickly, so that truncation does not affect it too much. When α is small, the pulse $g(t)$ decays slowly, so that truncating it chops off much of the sidelobes. With significant alteration, the combined pulse $p(t) = g(t) * q(t) = g(t) * g(t)$ is far from Nyquist, and thus inter-symbol interference results. Fig. 3 illustrates the effect.

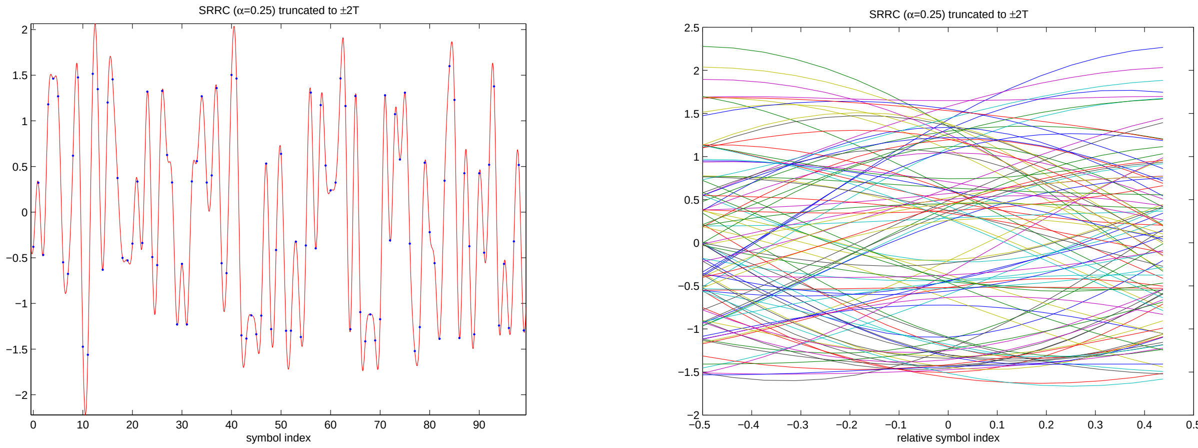


Figure 1: Plots for SRRC rolloff parameter $\alpha = 0.25$.

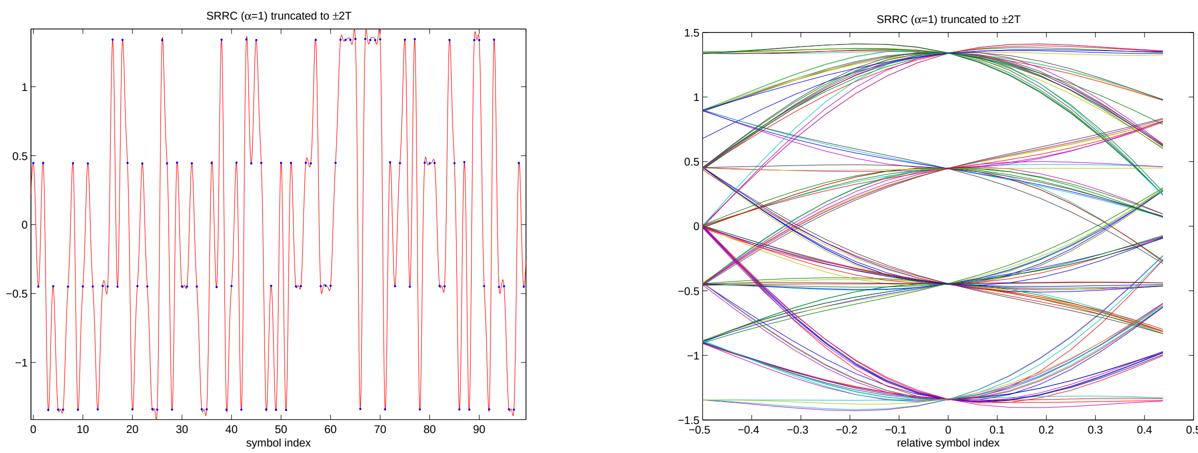


Figure 2: Plots for SRRC rolloff parameter $\alpha = 1.0$.

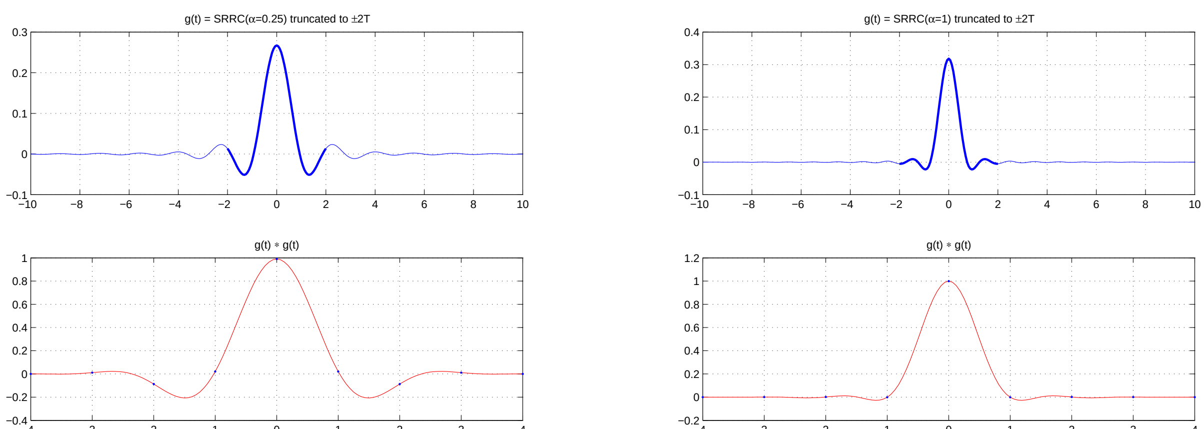
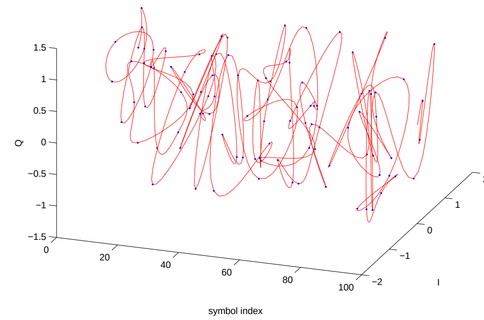
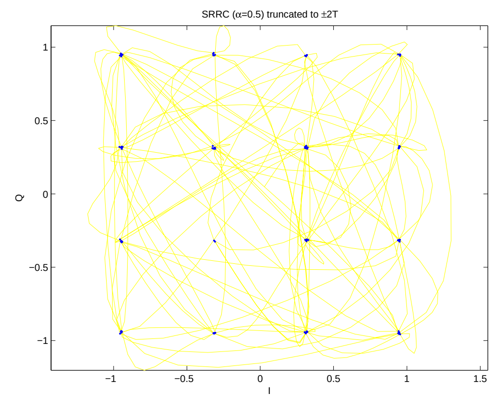


Figure 3: Truncated SRRC pulse and corresponding combined pulse $p(t) = g(t) * g(t)$ for $\alpha = 0.25$ (left) and $\alpha = 1$ (right). For zero ISI, we need $p(nT) = \delta[n]$, which is clearly not happening when $\alpha = 0.25$.

2. The MATLAB code for 16-QAM transmission, along with a plot of the output signal and the constellation diagram, appears below.

```
% design SRRC
P = 8; % oversampling factor
alpha = 0.5; % SRRC rolloff param
D = 2; % truncation to [-DT,DT]
g = srrc(D,alpha,P); % SRRC pulse
Ng = length(g);

% generate symbols
N = 100;  % # symbols
M = 4; % (sqrt) alphabet size
sig2a = 1; % symbol variance
a = qam(N,M,sig2a);  % symbol sequence

% pulse-amplitude modulate
a_up = zeros(1,N*P);
a_up(1:P:end) = a;  % upsampled symbols
m = conv(a_up,g); % PAM

% matched-filter demodulate
y_up = conv(m,g); % use SRRC again

% remove causal filtering delay
k = [1:P*N]; % desired time indices
dly = (Ng-1)/2+(Ng-1)/2;% delay due to pulses
y_up = y_up(k+dly); % remove delay
y = y_up(1:P:end); % downsample

% plot received signal
figure(1)
plot3([0:P*N-1]/P,real(y_up),imag(y_up),...
'r',[0:N-1],real(y),imag(y),'.');
xlabel('symbol index');
ylabel('I'); zlabel('Q');
view(20,30); % to see full trajectory
%also try view(0,90), view(0,0), view(90,0)

% plot constellation diagram
figure(2)
plot(real(y_up),imag(y_up),'y',real(y),imag(y),'.');
xlabel('I'); ylabel('Q');
title(['SRRC (\alpha=',num2str(alpha),...
        ') truncated to \pm',num2str(D),'T'])
axis('equal');
```





It is interesting to note that applying `view(0,90)` to the 3D output signal plot yields a view of the in-phase signal $y_\mathrm{I}(t)$, while `view(0,0)` yields a view of the quadrature signal $y_\mathrm{Q}(t)$, while `view(0,0)` yields a view of the constellation diagram:

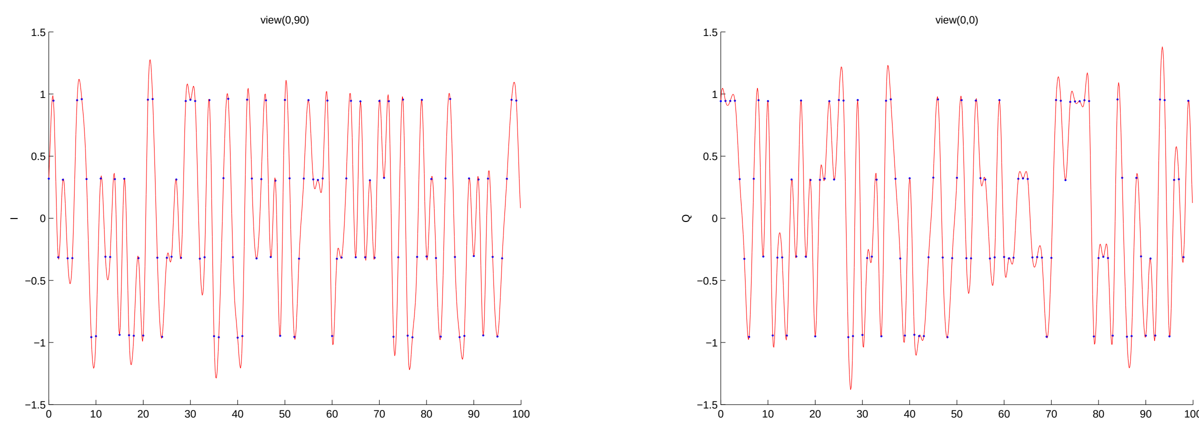


Figure 4: Different views of the output $y(t)$ show the in-phase $y_\mathrm{I}(t)$ and quadrature $y_\mathrm{Q}(t)$ components.

The clusters are reasonably tight for $\alpha = 0.5$. Further experiments show that the clusters are not at all tight for $\alpha - 0.25$ and extremely tight for $\alpha = 1.0$, due to the ISI-inducing properties of truncated SRRC pulses discussed in the previous problem.

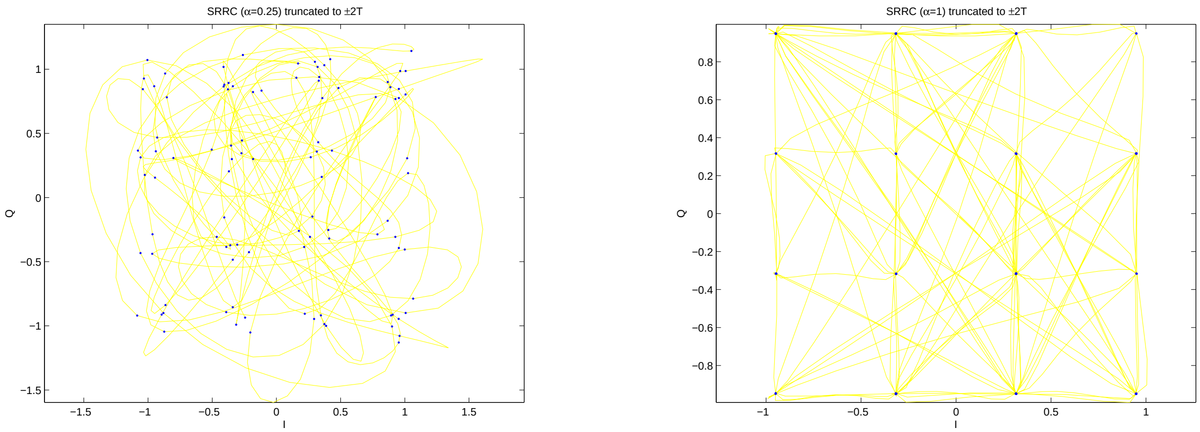
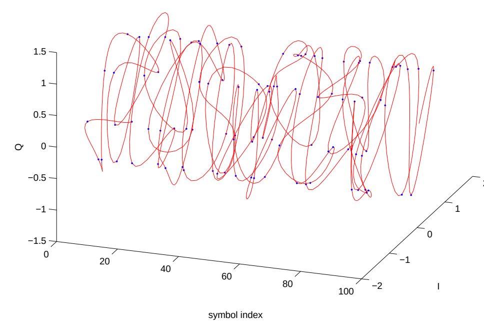
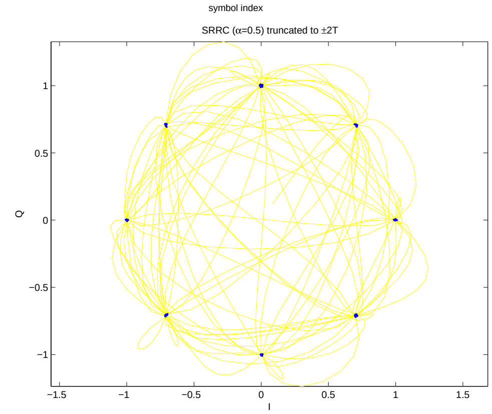


Figure 5: QAM constellation diagrams for $\alpha = 0.25$ (left) and $\alpha = 1$ (right).

3. The MATLAB code and plots for 8-PSK appear below.

```
% design SRRC
P = 8; % oversampling factor
alpha = 1.0; % SRRC rolloff param
D = 2; % truncation to [-DT,DT]
g = srrc(D,alpha,P); % SRRC pulse
Ng = length(g);

% generate symbols
N = 100;  % # symbols
M = 8; % alphabet size
sig2a = 1; % symbol variance
a = psk(N,M,sig2a);  % symbol sequence

% pulse-amplitude modulate
a_up = zeros(1,N*P);
a_up(1:P:end) = a;  % upsampled symbols
m = conv(a_up,g); % PAM

% matched-filter demodulate
y_up = conv(m,g); % use SRRC again

% remove causal filtering delay
k = [1:P*N]; % desired time indices
dly = (Ng-1)/2+(Ng-1)/2;% delay due to pulses
y_up = y_up(k+dly); % remove delay
y = y_up(1:P:end); % downsample

% plot received signal
figure(1)
plot3([0:P*N-1]/P,real(y_up),imag(y_up),...
'r',[0:N-1],real(y),imag(y),'.');
xlabel('symbol index');
ylabel('I'); zlabel('Q');
view(20,30); % to see full trajectory
%also try view(0,90), view(0,0), view(90,0)

% plot constellation diagram
figure(2)
plot(real(y_up),imag(y_up),'y',real(y),imag(y),'.');
xlabel('I'); ylabel('Q');
title(['SRRC (\alpha=',num2str(alpha),...
        ') truncated to \pm',num2str(D),'T'])
axis('equal');
```





For each value of α, the cluster-size was the same as that for QAM (as seen below).

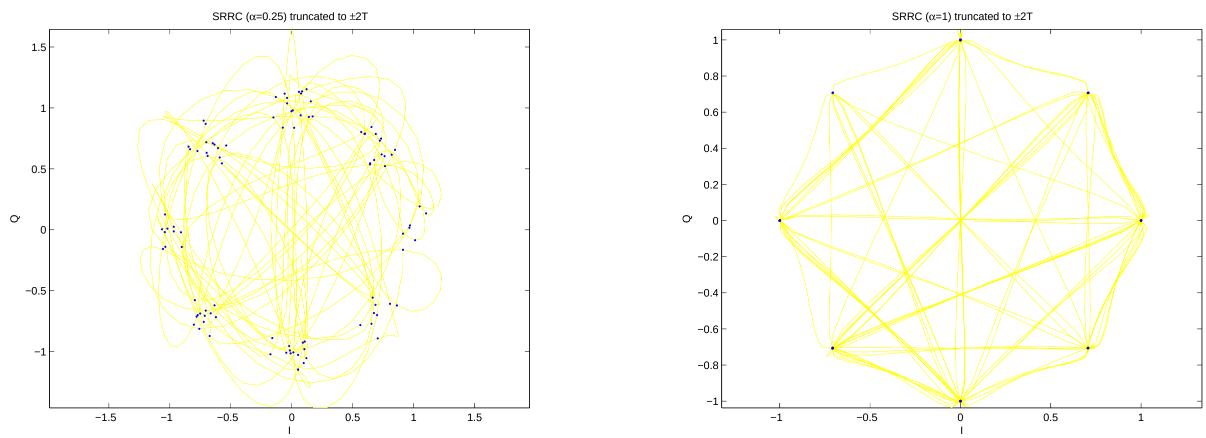


Figure 6: PSK constellation diagrams for $\alpha = 0.25$ (left) and $\alpha = 1$ (right).