```
Output = 'WashU McKelvey School of Engineering'

% PURPOSE:  Processes an input signal into ascii
%           16-QAM, HRRC pulse
%
% INPUTS:
%   r:   received signal, 16-QAM HRRC pulse
%
% OUTPUTS:
%   output: the estimated signal in ascii, a char array
%   sym:    the estimated signal data as passed to ESE471lab6b

function [output, sym] = ESE471lab6(r)
    %% processing
    %r(nT) = s

    %Demodulation
    t = 0:length(r)-1;
    M = 16;
    T_CARRIER = 4; %4 samples / cycle
    r_dm(1,:) = r.*sqrt(2).*cos(t*2*pi/T_CARRIER);
    r_dm(2,:) =-r.*sqrt(2).*sin(t*2*pi/T_CARRIER);

    %matched filter
    SAMPLE_RATE = 8; %Ts: 8 samples / signal
    N = SAMPLE_RATE;
    ENERGY_PER_SYMBOL = 40;
    A = sqrt(ENERGY_PER_SYMBOL/10);
    alpha = 0.3;
    Lp = 8;
    n = -Lp*N:Lp*N;
    p = 1/sqrt(N)*(sin(pi*(1-alpha)*n/N) + 4*alpha*n/N.*cos(pi*(1+alpha)*n/N)) ...
                ./ (pi*n/N.*(1-(4*alpha*n/N).^2));
    p(n==0) = 1/sqrt(N)*(1-alpha+4*alpha/pi);
    p(ismember(n,[-N/(4*alpha), N/(4*alpha)])) = 1/sqrt(N)*alpha/sqrt(2)*...
        ((1+2/pi)*sin(pi/4/alpha)+(1-2/pi)*cos(pi/4/alpha));

    h = p(end:-1:1);
    x2(1,:) = conv(h,r_dm(1,:));
    x2(2,:) = conv(h,r_dm(2,:));

    %time synch
    SYNCH_DELAY=(2*Lp-1)*N+7;
    x2 = x2(:,SYNCH_DELAY+1:end-SYNCH_DELAY);

    figure(4)
    subplot(2,1,1)
    plot_eye_diagram(x2(1,:), N, 1);
    title('In Phase')
    subplot(2,1,2)
    plot_eye_diagram(x2(2,:), N, 1);
    title('Quadrature')

    %downsample
    r_hat(1,:) = x2(1,SAMPLE_RATE+1:SAMPLE_RATE:end);
    r_hat(2,:) = x2(2,SAMPLE_RATE+1:SAMPLE_RATE:end);


    %bit decisions
    a_hat   = round((r_hat/A/2+sqrt(M)-1)/2);
    sym     = b2gray(a_hat(1,:))*4 + b2gray(3-a_hat(2,:));
    sym_bin = de2bi(sym,log2(M),'left-msb')'; % express it as a binary vector

    %postprocessing
    try
        output = binvector2str(sym_bin(:)');
    catch e
```

```
        warning(e.message) %using a try/catch here makes the program still plot diagnostics even 
when the message fails to process
        output = '';
    end
    %% plotting

    figure(2)
    clf;
    subplot(3,2,1)
    plot(r)
    hold on
    title('r(nT)','Interpreter','Latex')
    ylabel('Volts')
    xlabel('Time (s)')

    subplot(3,2,2)
    plot(p)
    title('p(t)','Interpreter','Latex')
    ylabel('Volts')
    xlabel('Time (s)')

    subplot(3,2,3)
    plot(x2(1,:))
    hold on
    plot(x2(2,:))
    t = (1+SAMPLE_RATE):SAMPLE_RATE:length(x2);
    plot(t,r_hat(1,:),'o')
    plot(t,r_hat(2,:),'o')

    title('x(nT) vs x(kTs) (downsampled)','Interpreter','Latex')
    ylabel('Watts')
    xlabel('Time (s)')
    legend("x_0(nT)","x_1(nT)","x_0(kTs)","x_1(kTs)")

    subplot(3,2,4)
    plot(r_hat(1,:),r_hat(2,:),'o')
    xlim([min(r_hat(1,:))-1,max(r_hat(1,:))+1])
    ylim([min(r_hat(2,:))-1,max(r_hat(2,:))+1])
    title('Signal Space Projections of Symbol Estimates $$\hat{r}$$','Interpreter','Latex')
    xlabel('Amplitude (times A)')
    legend('x(kTs)','location','best')

    subplot(3,2,5)
    plot(sym,'o-')
    title('$$\hat{sym}(kTs)$$','Interpreter','Latex')
    ylabel('A')
    xlabel('k')

    subplot(3,2,6)
    histogram(sym)
    title('Symbol appearances in $$\hat{sym}(kTs)$$','Interpreter','Latex')
    ylabel('Number of Appearances')
    xlabel('Symbol')
end
```

```matlab
% PURPOSE:  Generates a test signal for ESE471lab6, with diagnostic plots
%           16-QAM, HRRC pulse
%
% INPUTS:
%   data:   data vector, base 16
%
% OUTPUTS:
%   s:      signal vector

function s = ESE471lab6b(data)
    %uses Communications Systems Toolbox (gray2bin);
    %data = [0 15 1 12 2 9 6];
    %16-QAM
    M=16;

    a_k_b = b2gray([fix(data/log2(M));
             mod(data,log2(M))]);
    a_k   = 2*[a_k_b(1,:);3-a_k_b(2,:)]-sqrt(M)+1;

    %upsample
    SAMPLE_RATE = 8; %8 samples/bit
    a_k_ex(1,:) = upsample(a_k(1,:),SAMPLE_RATE,SAMPLE_RATE-1);
    a_k_ex(2,:) = upsample(a_k(2,:),SAMPLE_RATE,SAMPLE_RATE-1);

    %modulate
    SAMPLE_RATE = 8; %Ts: 8 samples / signal
    N = SAMPLE_RATE;
    ENERGY_PER_SYMBOL = 40;
    A = sqrt(ENERGY_PER_SYMBOL/10);
    alpha = 0.3;
    Lp = 8;
    n = -Lp*N:Lp*N;
    p = 1/sqrt(N)*(sin(pi*(1-alpha)*n/N) + 4*alpha*n/N.*cos(pi*(1+alpha)*n/N)) ...
            ./ (pi*n/N.*(1-(4*alpha*n/N).^2));
    p(n==0) = 1/sqrt(N)*(1-alpha+4*alpha/pi);
    p(ismember(n,[-N/(4*alpha), N/(4*alpha)])) = 1/sqrt(N)*alpha/sqrt(2)*...
        ((1+2/pi)*sin(pi/4/alpha)+(1-2/pi)*cos(pi/4/alpha));
    p = A*p;

    signal(1,:) = conv(a_k_ex(1,:),p);
    signal(2,:) = conv(a_k_ex(2,:),p);

    %time_desync
    TIME_SYNCH = 0;
    signal_ex = [zeros(2,TIME_SYNCH) signal];
    t = 0:length(signal_ex)-1;
    T_CARRIER = 4; %4 samples / cycle
    signal_mod(1,:) = signal_ex(1,:).*sqrt(2).*cos(t*2*pi/T_CARRIER);
    signal_mod(2,:) =-signal_ex(2,:).*sqrt(2).*sin(t*2*pi/T_CARRIER);

    s = signal_mod(1,:)+signal_mod(2,:);

    figure(3)
    clf
    subplot(3,2,1)
    plot(data,'-o')
    legend('data(t)')
    title('$$a_k(t)$$','Interpreter','Latex')

    subplot(3,2,2)
    plot(n,p)
    title('p(t)','Interpreter','Latex')

    subplot(3,2,3)
    plot(signal_ex(1,:))
    hold on
    plot(signal_ex(2,:))
    t = (1:length(a_k))*SAMPLE_RATE+Lp*N;
    plot(t,a_k(1,:),'o')
    plot(t,a_k(2,:),'o')
    title('$$s_{unmodulated}(t)$$','Interpreter','Latex')
    legend('s_0(t)','s_1(t)','a_0(k)','a_1(k)')

    subplot(3,2,4)
    plot(a_k(1,:),a_k(2,:),'o')
    xlim([min(a_k(1,:))-1,max(a_k(1,:))+1])
    ylim([min(a_k(2,:))-1,max(a_k(2,:))+1])
    title('Signal Space Projections of Symbol Estimates s','Interpreter','Latex')
    xlabel('Amplitude (times A)')
    legend('x(kTs)')

    subplot(3,2,5)
    plot(s)
    title('s(t)','Interpreter','Latex')

    subplot(3,2,6)
    histogram(data)
    title('Symbol appearances in s(kTs)','Interpreter','Latex')
    ylabel('Number of Appearances')
    xlabel('Symbol')
end
```

```
function gray = b2gray(bin)
    gray = bitxor(bin,fix(bin/2));
end
```

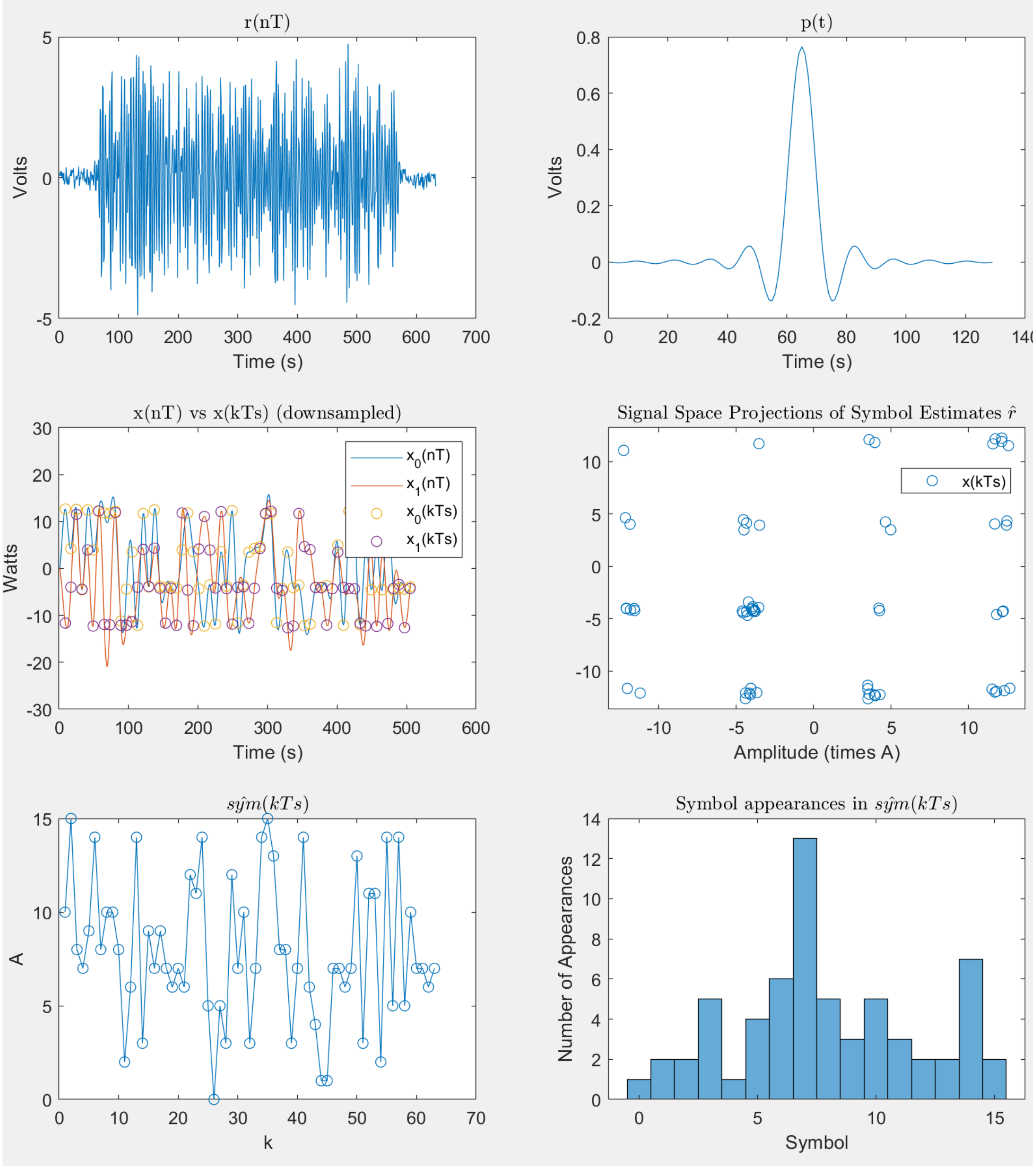
```
%Currently broken for array inputs, using b2gray instead for length 2 gray code as it is the same
function bin = gray2b(gray)
    mask = fix(gray/2);
    bin = gray;
    while mask ~= 0
        bin = bitxor(bin,mask);
        mask = fix(mask/2);
    end
end
```

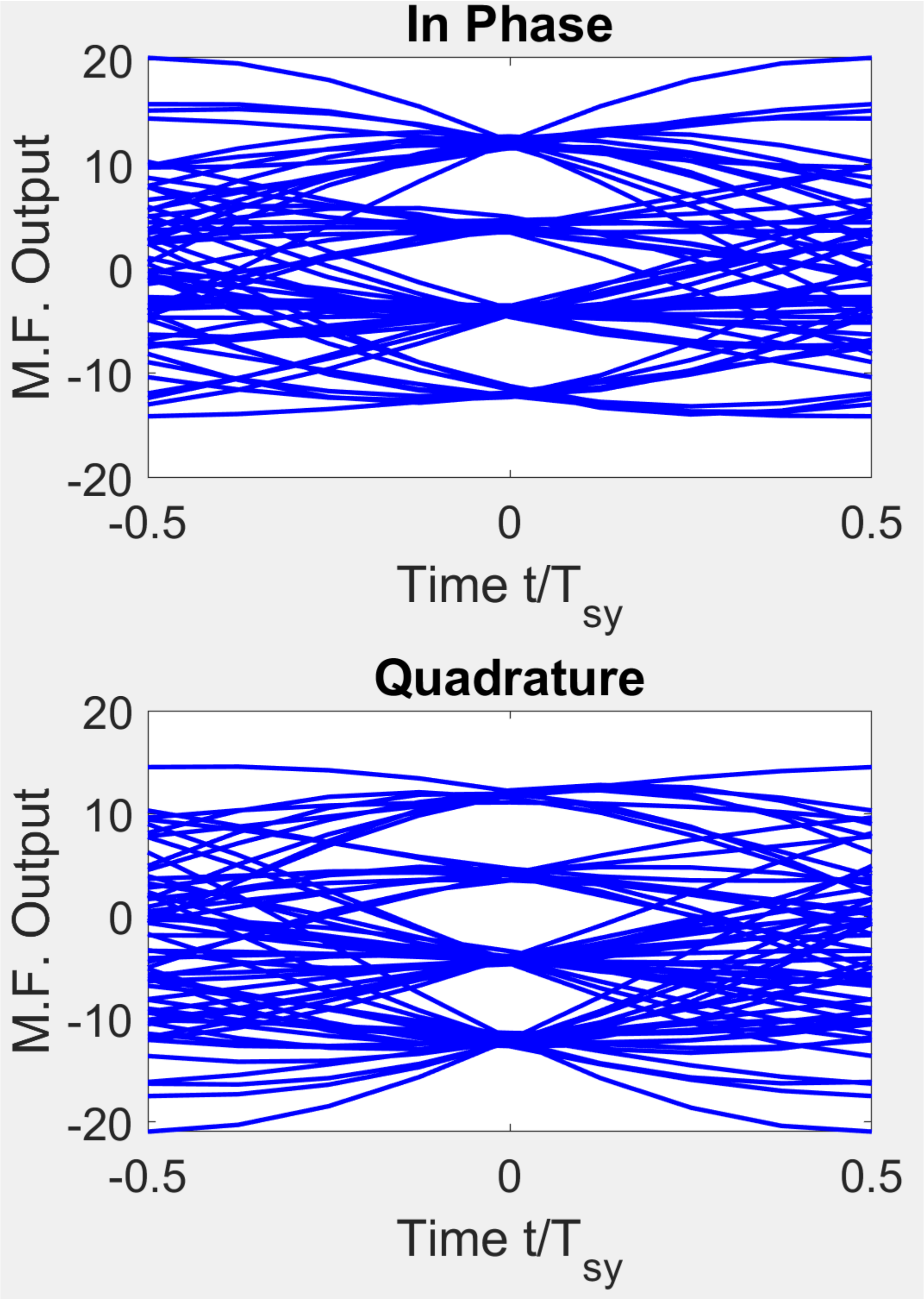
```
% PURPOSE:  Plot an eye diagram of a signal
%
% INPUTS:
%   y_s:    vector of signal samples out of the matched filter
%   N:      the number of samples per symbol.  Assumes that time 0 is
%           at sample y_s(1).  If not, you must send in an offset integer.
%   offset: the number of samples at the start of y_s to ignore
%
% OUTPUTS:  h, the handle to the lines plotted


function h = plot_eye_diagram(y_s, N, offset)

        if ~exist('offset','var'),
            offset = 0;
        end

        start_indices = (floor(N/2):N:(length(y_s)-N-offset)) + offset;
        time_vals     = (0:N)./N - 0.5;
        %figure; %Commented out to allow subplots
        for i=1:length(start_indices),
            h(i) = plot(time_vals, y_s(start_indices(i):start_indices(i)+N), 'b-');
            hold on;
        end
        set(h,'LineWidth',2)
        set(gca,'FontSize',20)
        xlabel('Time t/T_s_y')
        ylabel('M.F. Output')
```


r(nT)
Volts
5
0
-5
0
100
200
300
400
500
600
700
Time (s)
p(t)
Volts
0.8
0.6
0.4
0.2
0
-0.2
0
20
40
60
80
100
120
140
Time (s)
x(nT) vs x(kTs) (downsampled)
Watts
30
20
10
0
-10
-20
-30
0
100
200
300
400
500
600
Time (s)
x_0(nT)
x_1(nT)
x_0(kTs)
x_1(kTs)
Signal Space Projections of Symbol Estimates $\hat{r}$
x(kTs)
10
5
0
-5
-10
-10
-5
0
5
10
Amplitude (times A)
$s\hat{y}m(kTs)$
A
15
10
5
0
0
10
20
30
40
50
60
70
k
Symbol appearances in $s\hat{y}m(kTs)$
Number of Appearances
14
12
10
8
6
4
2
0
0
5
10
15
Symbol


In Phase
20
10
0
-10
-20
M.F. Output
-0.5
0
0.5
Time t/T_{sy}
Quadrature
20
10
0
-10
-20
M.F. Output
-0.5
0
0.5
Time t/T_{sy}