EECS 455: Solutions to Problem Set 3

1. (a) Consider a complex sequence of length N=10;

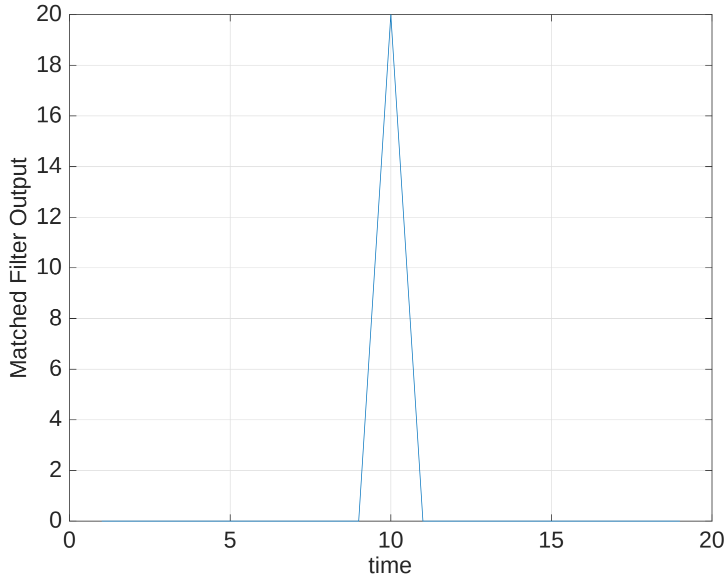
$$x = (-1-j, 1+j, 1+j, -1+j, 1+j, -1+j, 1+j, 1-j, 1-j, -1+j)$$

Now consider a matched filter $h(n) = x^*(N-n)$ Determine the output of the matched filter when the input is x. That is determine y where

$$y(n) = \Re\left[\sum_{l=0}^{N-1} h(n-l)x(l)\right]$$

and $\Re(w)$ is the real part of w. That is, plot the result.

Solution:

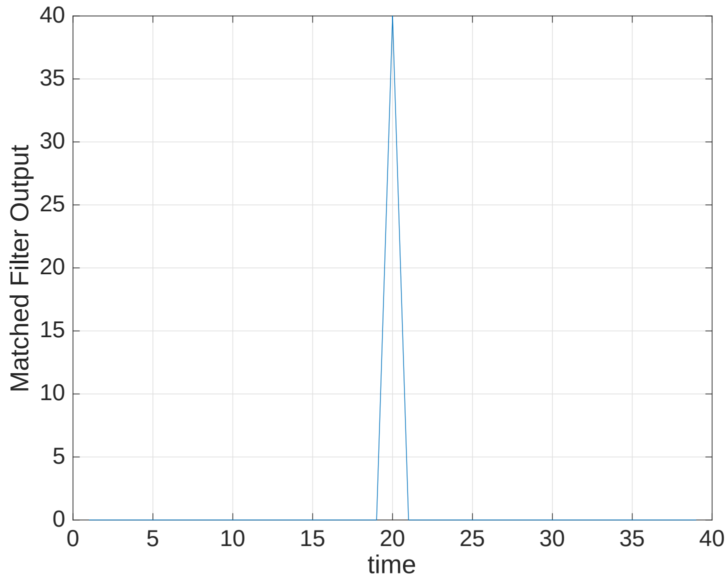


(b) Consider the sequence of length 20

$$\begin{aligned}
\Re(u) &= [\Re(x) \quad +\Im(x)] \\
\Im(u) &= [\Re(x) \quad -\Im(x)] \\
u &= \Re(u) + j\Im(u)
\end{aligned}$$

That is, the first half of the real part of u is the real part of x. The second half of the real part of u is the imaginary part of x. Similarly for the imaginary part of u. Determine the output of a filter matched to the sequence u. Plot the result.

Solution:




```
xr=[-1 +1 +1 -1 +1 -1 +1 +1 +1 -1];
xi=[-1 +1 +1 +1 +1 +1 +1 -1 -1 +1];
x=xr+j*xi;
h=fliplr(conj(x));
z=real(conv(h,x));
figure(1)
plot(z)
grid on
set(gca,'FontSize',16)
xlabel('time','FontSize',16)
ylabel('Matched Filter Output','FontSize',16)

u= [xr xi]+j*[xr -xi];
h=fliplr(conj(u));
z=real(conv(h,u));
figure(2)
plot(z)
grid on
xlabel('time','FontSize',16)
ylabel('Matched Filter Output','FontSize',16)
set(gca,'FontSize',16)
```

2. (a) Data is transmitted using a rectangular pulse with amplitude A or $-A$ and duration T. The receiver filters the signal with a filter impulse resonse

$$h(t) = e^{-\alpha t}u(t).$$

The filter output is sampled at time T.

Solution: The filter output is found using convolution.

$$\begin{aligned}
y(T) &= \int_{\infty}^{\infty} h(T-\tau)x(\tau)d\tau \\
&= A\int_0^T h(T-\tau)d\tau \\
&= A\int_0^T e^{-\alpha(T-\tau)}d\tau \\
&= \frac{A}{\alpha}e^{-\alpha(T-\tau)}|_0^T \\
&= \frac{A}{\alpha}[e^{-\alpha(0)} - e^{-\alpha(T)}] \\
&= \frac{A}{\alpha}[1-e^{-\alpha(T)}]
\end{aligned}$$

(b) White Gaussian noise with power spectral density $N_0/2$ is the input to an RC filter with impulse response

$$h(t) = e^{-\alpha t}u(t)$$

where $u(t)$ is one for $t > 0$ and is 0 otherwise. Find the variance of the noise at the output of the filter.

Solution: The variance of the noise at the output is

$$\begin{aligned}
\sigma^2 &= \frac{N_0}{2}\int_{-\infty}^{\infty} h^2(t)dt \\
&= \frac{N_0}{2}\int_0^{\infty} e^{-2\alpha t}dt \\
&= \frac{N_0}{2}\left[-\frac{1}{2\alpha}e^{-2\alpha t}\right]_0^{\infty} \\
&= \frac{N_0}{4\alpha}
\end{aligned}$$

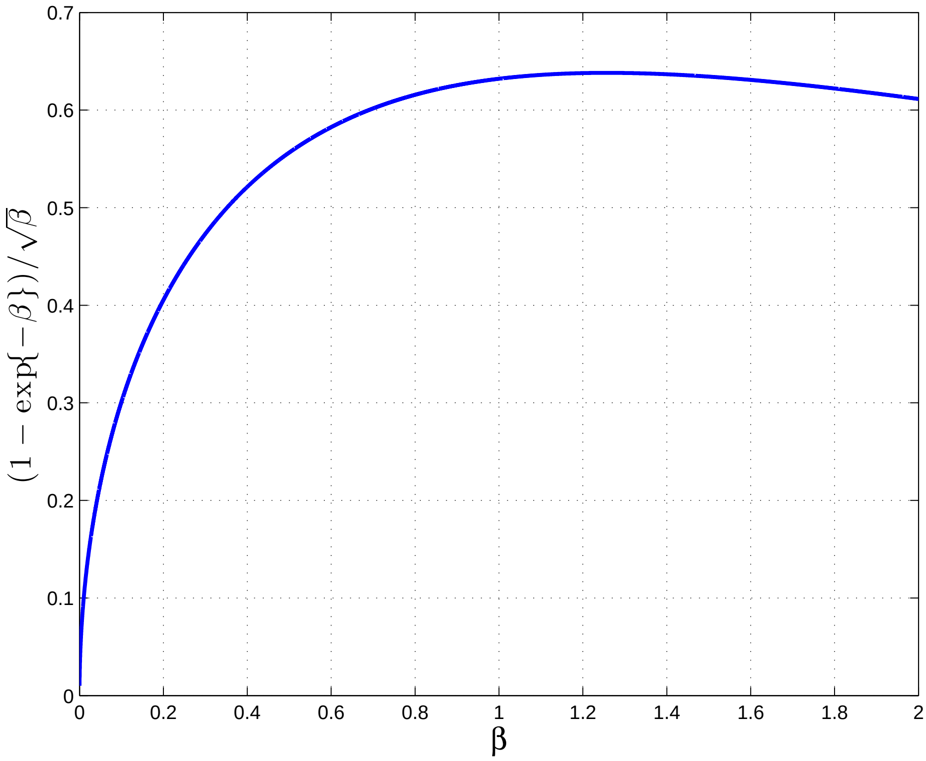
(c) Let $\beta = \alpha T$. Find the value of β that maximizes $Y(T)/\sigma$, the signal-to-noise ratio. (You probably need to do this numerically, e.g. Matlab). Find the resulting signal-to-noise ratio for the best choise of β. Express your answer in terms of the energy of the signal and the noise power spectral density.

Solution:

The signal-to-noise ratio is

$$\begin{aligned}
Y(T)/\sigma &= \frac{A(1-e^{-\alpha T})/\alpha}{\sqrt{N_0/(4\alpha)}} \\
&= \frac{A(1-e^{-\alpha T})}{\sqrt{\alpha N_0/4}}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{2A\sqrt{T}}{\sqrt{N_0}} \frac{(1-e^{-\alpha T})}{\sqrt{\alpha T}} \\
&= \frac{2A\sqrt{T}}{\sqrt{N_0}} \left[\frac{(1-e^{-\beta})}{\sqrt{\beta}} \right]
\end{aligned}
$$



The maximum occurs when $\beta = 1.256$. For this value of β the signal-to-noise ratio is

$$
\begin{aligned}
Y(T)/\sigma &= \frac{2A\sqrt{T}}{\sqrt{N_0}} 0.638 \\
&= \sqrt{2}\frac{\sqrt{2A^2T}}{\sqrt{N_0}} 0.638 \\
&= 0.902 \frac{\sqrt{2E}}{\sqrt{N_0}} \\
&= 0.902 \sqrt{\frac{2E}{N_0}}
\end{aligned}
$$

3. This problem uses the same signal sets as the first homework (problem 2). A first signal set with $M = 16$ signals in two dimensions that can transmit 4 bits of information has the following signals.

$$
s_0 = A(-3,-3)
$$

$$
\begin{aligned}
s_1 &= A(-3,-1)\\
s_2 &= A(-3,+3)\\
s_3 &= A(-3,+1)\\
s_4 &= A(-1,-3)\\
s_5 &= A(-1,-1)\\
s_6 &= A(-1,+3)\\
s_7 &= A(-1,+1)\\
s_8 &= A(+1,-3)\\
s_9 &= A(+1,-1)\\
s_{10} &= A(+1,+3)\\
s_{11} &= A(+1,+1)\\
s_{12} &= A(+3,-3)\\
s_{13} &= A(+3,-1)\\
s_{14} &= A(+3,+3)\\
s_{15} &= A(+3,+1)
\end{aligned}
$$

(a) Simulate the probability of error for this signal set as a function of the signal-to-noise ratio (E_b/N_0). That is, plot the probability of choosing the wrong transmitted signal at the receiver as a function of the signal-to-noise ratio. Consider signal-to-noise ratios that yield an error probabilty between 0.0001 and 1.

(b) Repeat part a for the signal set below.

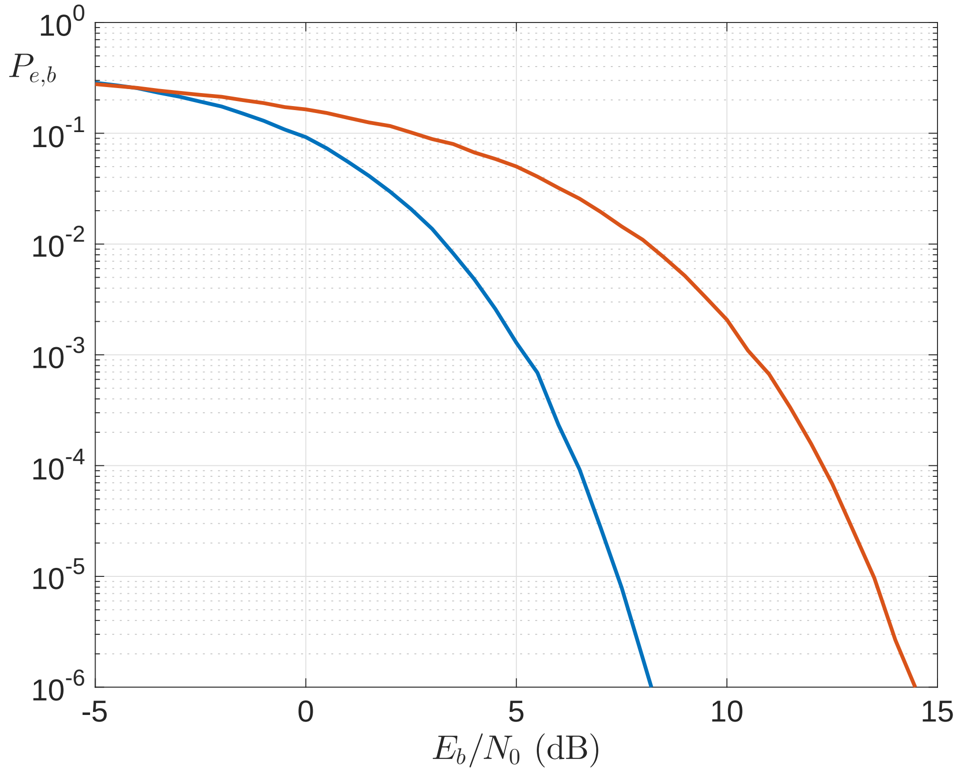
$$
\begin{aligned}
s_0 &= A(+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1)\\
s_1 &= A(+1,-1,+1,-1,+1,-1,+1,-1,+1,-1,+1,-1,+1,-1,+1,-1)\\
s_2 &= A(+1,+1,-1,-1,+1,+1,-1,-1,+1,+1,-1,-1,+1,+1,-1,-1)\\
s_3 &= A(+1,-1,-1,+1,+1,-1,-1,+1,+1,-1,-1,+1,+1,-1,-1,+1)\\
s_4 &= A(+1,+1,+1,+1,-1,-1,-1,-1,+1,+1,+1,+1,-1,-1,-1,-1)\\
s_5 &= A(+1,-1,+1,-1,-1,+1,-1,+1,+1,-1,+1,-1,-1,+1,-1,+1)\\
s_6 &= A(+1,+1,-1,-1,-1,-1,+1,+1,+1,+1,-1,-1,-1,-1,+1,+1)\\
s_7 &= A(+1,-1,-1,+1,-1,+1,+1,-1,+1,-1,-1,+1,-1,+1,+1,-1)\\
s_8 &= A(+1,+1,+1,+1,+1,+1,+1,+1,-1,-1,-1,-1,-1,-1,-1,-1)\\
s_9 &= A(+1,-1,+1,-1,+1,-1,+1,-1,-1,+1,-1,+1,-1,+1,-1,+1)\\
s_{10} &= A(+1,+1,-1,-1,+1,+1,-1,-1,-1,-1,+1,+1,-1,-1,+1,+1)\\
s_{11} &= A(+1,-1,-1,+1,+1,-1,-1,+1,-1,+1,+1,-1,-1,+1,+1,-1)\\
s_{12} &= A(+1,+1,+1,+1,-1,-1,-1,-1,-1,-1,-1,-1,+1,+1,+1,+1)\\
s_{13} &= A(+1,-1,+1,-1,-1,+1,-1,+1,-1,+1,-1,+1,+1,-1,+1,-1)\\
s_{14} &= A(+1,+1,-1,-1,-1,-1,+1,+1,-1,-1,+1,+1,+1,+1,-1,-1)
\end{aligned}
$$

$$s_{15} = A(+1,-1,-1,+1,-1,+1,+1,-1,-1,+1,+1,-1,+1,-1,-1,+1)$$

[These signals are written into a Matlab file `HW01_Problem2.m` on Canvas].

(c) Determine the improvement in signal-to-noise ratio required for an error probability of 0.0001 by using the smaller rate (bits/dimension) of the signal set in part (b) compared to part (a).

Solution:



The signal-to-noise ratio required for BER 0.0001 is lower for 16-ary orthogonal compared to 16-QAM by about 5.6dB.

```
clear all
hold on
A=1;
s213_signal_set1
N=length(s(1,:));
E=sum(sum(abs(s).^2))/16;
Eb=E/4;
    tic
for j1=1:29
```

```
    EbN0dB(j1)=(j1-1)/2;                 % Change values of EbN0 (dB)
    EbN0=10.^(EbN0dB(j1)/10);            % Convert to non-dB units
    N0=Eb/EbN0;                          % Determine N0 from Eb and Eb/
    sigma=sqrt(N0/2);                    % Sqrt of variance
    biterror=0;
    n=0;                                 % Number of iterations
    while (biterror < 100)               % Do this loop till you count
        data=randi(16,1);                % Choose a signal at random
        r=s(data,:)+sigma*randn(1,N);    % Generate received signal
        dmin=10000;                      % Initial distance to nearest
        for k=1:16
            d=sum((r-s(k,:).^2));                    % Find signal clos
            if (d<dmin) info=k; dmin=d; end
        end
        if (info ~= data)
            bhat=de2bi(info-1,4);          % Determine the bits at the
            b=de2bi(data-1,4);           % Determine the bits at the tr
            biterror =biterror+sum(abs(b-bhat));  % Count the number
        end
        n=n+1;
        if (mod(n,100000)==0)
           SNR=EbN0dB(j1)
           n
           biterror
        end
    end

    Pe(j1)=biterror/(4*n);
    [EbN0dB(j1),Pe(j1)*10000]

    semilogy(EbN0dB,Pe,'LineWidth',2)
    grid on
    xlabel('$E_b/N_0$ (dB)','FontSize',16,'Interpreter','Latex')
    ylabel('$P_{e,b}$', 'FontSize',16,'Interpreter','Latex','Rotation
    set(gca,'FontSize',16)
    axis([0 14 1e-6 1])
    pause(0.1)
end
    toc
    %clear all
%=======================================================================
hold on
clear all
A=1;
```

```
s213_signal_set2
N=length(s(1,:));
E=sum(sum(abs(s).^2))/16;
Eb=E/4;
    tic
for j1=1:17

    EbN0dB(j1)=(j1-1)/2;                 % Change values of EbN0 (dB)
    EbN0=10.^(EbN0dB(j1)/10);            % Convert to non-dB units
    N0=Eb/EbN0;                          % Determine N0 from Eb and Eb/
    sigma=sqrt(N0/2);                    % Sqrt of variance
    biterror=0;
    n=0;                                 % Number of iterations
    while (biterror < 100)               % Do this loop till you count
        data=randi(16,1);                % Choose a signal at random
        r=s(data,:)+sigma*randn(1,N);    % Generate received signal
        dmin=10000;                      % Initial distance to nearest
        for k=1:16
            d=sum((r-s(k,:).^2));                    % Find signal clos
            if (d<dmin) info=k; dmin=d; end
        end
        if (info ~= data)
            bhat=de2bi(info-1,4);        % Determine the bits at the
            b=de2bi(data-1,4);           % Determine the bits at the tr
            biterror =biterror+sum(abs(b-bhat));  % Count the number
        end
        n=n+1;
        if (mod(n,100000)==0)
           SNR=EbN0dB(j1)
           n
           biterror
        end
    end

    Pe(j1)=biterror/(4*n);
    [EbN0dB(j1),Pe(j1)*10000]

    semilogy(EbN0dB,Pe,'LineWidth',2)
    grid on
    xlabel('$E_b/N_0$ (dB)','FontSize',16,'Interpreter','Latex')
    ylabel('$P_{e,b}$', 'FontSize',16,'Interpreter','Latex','Rotation
    set(gca,'FontSize',16)
    axis([0 14 1e-6 1])
    pause(0.1)
end
```

```
toc
```

4. This problem uses the same signal sets as the first homework (problem 2). A first signal set with $M = 16$ signals in two dimensions that can transmit 4 bits of information has the following signals.

$$
\begin{aligned}
s_0 &= A(-3,-3) \\
s_1 &= A(-3,-1) \\
s_2 &= A(-3,+3) \\
s_3 &= A(-3,+1) \\
s_4 &= A(-1,-3) \\
s_5 &= A(-1,-1) \\
s_6 &= A(-1,+3) \\
s_7 &= A(-1,+1) \\
s_8 &= A(+1,-3) \\
s_9 &= A(+1,-1) \\
s_{10} &= A(+1,+3) \\
s_{11} &= A(+1,+1) \\
s_{12} &= A(+3,-3) \\
s_{13} &= A(+3,-1) \\
s_{14} &= A(+3,+3) \\
s_{15} &= A(+3,+1)
\end{aligned}
$$

```
clear all
A=1;
signal_set1
N=length(s(1,:));
E=sum(sum(abs(s).^2))/16;
Eb=E/4;
for j1=1:29
    EbN0dB(j1)=(j1-1)/2;                % Change values of EbN0 (dB)
    EbN0=10.^(EbN0dB(j1)/10);           % Convert to non-dB units
    N0=Eb/EbN0;                         % Determine N0 from Eb and Eb/N0
    sigma=sqrt(N0/2);                   % Sqrt of variance
    biterror=0;
    n=0;                                % Number of iterations
    while (biterror < 100)              % Do this loop till you count a certa
        data=randi(16,1);               % Choose a signal at random
        r=s(data,:)+sigma*randn(1,N);   % Generate received signal


%==============================================================================
%                    YOU FILL IN THIS PART.
%==============================================================================
%  a)  FIND THE SIGNAL CLOSEST TO THE RECEIVED SIGNAL
%  b)  FIND THE BITS (I called this bhat) ASSOCIATED WITH THAT SIGNAL
%  c)  FIND THE BITS (I called this b) ASSOCIATED WITH THE ACTUAL TRANSMITTE
%  d)  DETERMINE HOW MANY BITS ARE IN ERROR
%==============================================================================

        biterror =biterror+sum(abs(b-bhat));  % Count the number of errors
        n=n+1;
    end

    Pe(j1)=biterror/(4*n);

end
semilogy(EbN0dB,Pe,'LineWidth',2)
grid on
xlabel('$E_b/N_0$ (dB)','FontSize',16,'Interpreter','Latex')
ylabel('$P_{e,b}$', 'FontSize',16,'Interpreter','Latex','Rotation',0)
set(gca,'FontSize',16)
axis([0 14 1e-6 1])
```

signal_set1.m

```
%  First Signal Set
s( 1,:) = A *[-3, -3];
s( 2,:) = A *[-3, -1];
s( 3,:) = A *[-3, +3];
s( 4,:) = A *[-3, +1];
%==================================================================
s( 5,:) = A *[-1, -3];
s( 6,:) = A *[-1, -1];
s( 7,:) = A *[-1, +3];
s( 8,:) = A *[-1, +1];
%==================================================================
s( 9,:) = A *[+1, -3];
s(10,:) = A *[+1, -1];
s(11,:) = A *[+1, +3];
s(12,:) = A *[+1, +1];
%==================================================================
s(13,:) = A *[+3, -3];
s(14,:) = A *[+3, -1];
s(15,:) = A *[+3, +3];
s(16,:) = A *[+3, +1];
%==================================================================
```