Cairo University
Faculty of Engineering

Department of Computer Engineering

ELC 325B – Spring 2023

Digital Communications

Assignment #2

Submitted to

Dr. Mai

Dr. Hala

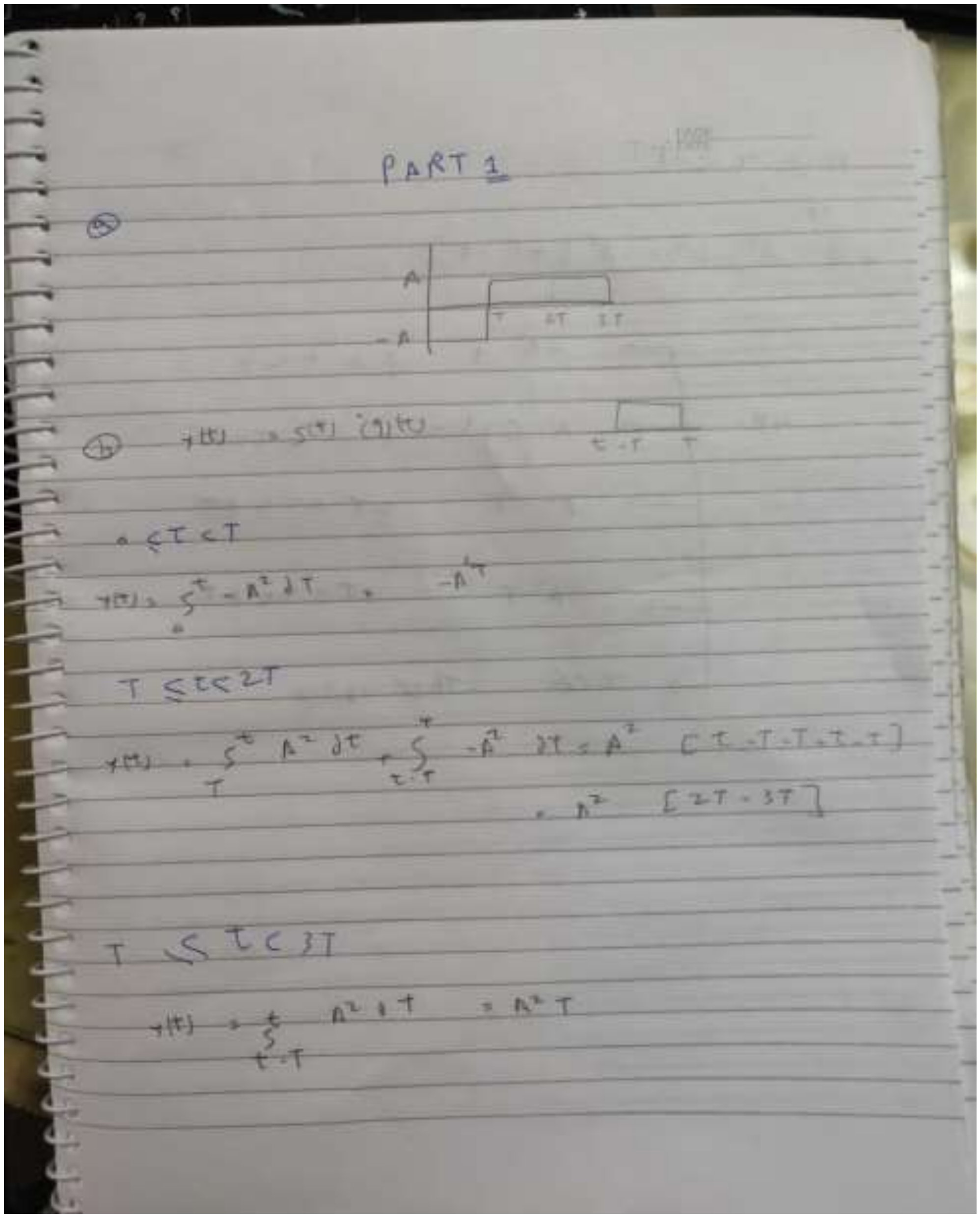
Eng. Mohamed Khaled

Submitted by

Name	Sec	BN
Nour Ziad	2	31
Eslam Ashraf	1	13

Contents

Part 1:



PART 1

(a)

(b) $y(t) = s(\tau)\, g(t)$

$0 \leq t < T$

$y(t) = \int_0^t -A^2\, d\tau = -A^2 t$

$T \leq t < 2T$

$y(t) = \int_T^t A^2\, d\tau + \int_{t-T}^{T} -A^2\, d\tau = A^2 \left[t - T - T + t - T \right]$

$= A^2 \left[2t - 3T \right]$

$T \leq t < 3T$

$y(t) = \int_{t-T}^{t} A^2\, d\tau = A^2 T$

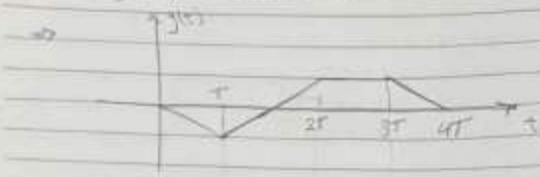
$3T \le t < 4T$

$\int_{t=T}^{3T} A^2 \, dt = A^2 [4T - t]$

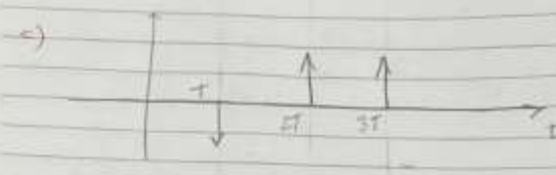
$$y(t) = \begin{cases} -A^2 t & 0 \le t < T \\ A^2 (2t - 3T) & T \le t < 2T \\ A^2 T & 2T \le t < 3T \\ A^2 T & 3T \le t < 4T \\ \text{zero} & \text{otherwise} \end{cases}$$

applying superposition & summation

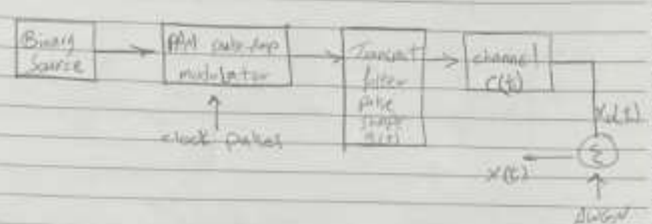
=>



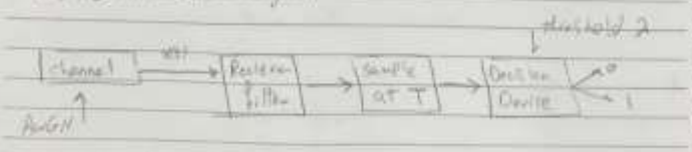
c)



d) transmitter Block diagram



e) Receiver Block Diagram



Part 2:

Derive the probability of error in the three mentioned cases.

DATE
PAGE

Part ②

a) h(t) is a matched filter with unit energy

$$h(t) = g(T-t) = A\, rect\left(\frac{T/2 - t}{T}\right)$$

$$r(t) = g(t) + w(t)$$

$$y(t) = r(t) * h(t)$$

$$= g(t)*h(t) + w(t)*h(t)$$ refer to part ① b

$$\mu_y = E(y(T)) = E(g_o(T)) + E(n(T))$$

$$\mu_y = E(\mp A^2 T + n(T)) = \mp A^2 + E(n(T))$$

$$E(n(T)) = E\left(\int w(t) \cdot \underbrace{h(T-t)}_{\to g(t)}\, dt\right)$$

$$= E\left(\int_0^T \mp A\, w(t)\, dt\right) = \int_0^T \mp A\, E(w(t))\, dt = 0$$

$$\therefore \mu_y = \mp A^2 T$$

$$\sigma_y^2 = Var(y(t)) = E(g_o(T) + n(T))$$

$$\sigma_y^2 = Var(n(T)) = E(n^2(T)) - E(\underbrace{n(T)}_{\to 0})^2$$

$$= \int_{-\infty}^{\infty} S_n(f)\, df$$

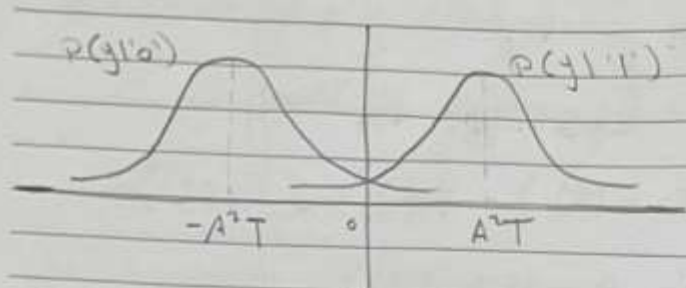
$$\sigma_y^2 = \frac{N_0}{2}\int_{-\infty}^{\infty} |H(f)|^2\, df = \frac{N_0}{2}\int_0^T |h(t)|^2\, dt$$

$$= \frac{N_0}{2} * A^2 T$$

$$p(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y-\mu_y)^2}{2\sigma^2}}$$

$$p(y\,|\,'0') = \frac{1}{\sqrt{\pi N_0 A^2 T}} e^{\frac{-(y+A^2T)^2}{N_0 A^2 T}}$$

$$p(y\,|\,'1') = \frac{1}{\sqrt{\pi N_0 A^2 T}} e^{\frac{-(y-A^2T)^2}{N_0 A^2 T}}$$



$$p(e\,|\,'0') = \int_0^\infty \frac{1}{\sqrt{\pi N_0 A^2 T}} e^{\frac{-(y+A^2T)^2}{N_0 A^2 T}} dy$$

Let $Z = \frac{y + A^2 T}{\sqrt{N_0 A^2 T}} \Rightarrow dZ = \frac{dy}{\sqrt{N_0 A^2 T}}$

$$p(e\,|\,'0') = \int_{\frac{A^2T}{\sqrt{N_0A^2T}}}^{\infty} \frac{1}{\sqrt{\pi}} e^{-Z^2} dZ = \frac{1}{2}$$

$$P(e) = p(e\,|\,'1')\,p('1') + p(e\,|\,'0')\,p('0')$$

$$p('1') = p('0') = 0.5$$

$$P(e) = 2\,p(e\,|\,'0') \times 0.5 = p(e\,|\,'0')$$

$$p(e) = \frac{1}{2}\,\mathrm{erfc}\left(\frac{A^2T}{\sqrt{N_o A^2 T}}\right)$$

$$@\ A = T = 1 \quad , \quad p(e) = \frac{1}{2}\,\mathrm{erfc}\left(\frac{1}{\sqrt{N_o}}\right)$$

b) $h(t)$ not exist

$$y(t) = r(t) * h(t) = r(t) * \delta(t)$$

$$y(T) = g(T) + w(T) = \mp A + w(T)$$

$$E(y(T)) = E(\mp A + w(T)) = \mp A + \underbrace{E(w(T))}_{0}$$

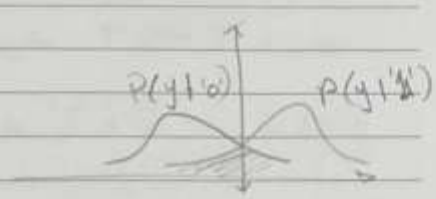
$$\mu_y = \begin{cases} -A \\ A \end{cases}$$

$$\sigma_y^2 = var(y(T)) = var(\mp A + w(T)) = var(w(T)) = N_o/2$$

$$p(y|'0') = \frac{1}{\sqrt{N_o \pi}}\, e^{\frac{-(y+A)^2}{N_o}} = N\left(A, \frac{N_o}{2}\right)$$

$$p(y|'1') = \frac{1}{\sqrt{N_o \pi}}\, e^{\frac{-(y-A)^2}{N_o}} = N\left(-A, \frac{N_o}{2}\right)$$

$$@\ \lambda = 0 \quad , \quad p('0') = p('1')$$



DATE
PAGE

$$P(e|'0') = \int_0^{\infty} p(y|'0')\,dy$$

$$= \int_0^{\infty} \frac{1}{\sqrt{N_0 \pi}} e^{\frac{-(y+A)^2}{N_0}}\,dy$$

let $Z = \frac{y+A}{\sqrt{N_0}} \rightarrow dZ = \frac{dy}{\sqrt{N_0}}$

$$P(e|'0') = \int_{A/\sqrt{N_0}}^{\infty} \frac{1}{\sqrt{\pi N_0}} e^{-Z^2} \times \sqrt{N_0}\,dZ$$

$$= \frac{1}{2} \operatorname{erfc}\left(\frac{A}{\sqrt{N_0}}\right)$$

$P(e) = 2P(e|'0') \times 0.5$

$$P(e|'0') = \frac{1}{2} \operatorname{erfc}\left(\frac{1}{\sqrt{N_0}}\right)$$

c) $h(t) = \sqrt{3}\ t$

$y(t) = r(t) * h(t) = g_o(t) + n(t)$

$g_o(t) = g(t) * h(t) = \int_{-\infty}^{\infty} g(\tau) \cdot h(t-\tau)\,d\tau$

$\rightarrow$ $t < 0 \Rightarrow g_o(t) = 0$ no intersection

$\rightarrow$ $0 < t \leq T \Rightarrow g_o(t) = \int_0^t \sqrt{3} A \tau\,d\tau = \frac{\sqrt{3}}{2} A t^2$

$$y(t) = \begin{cases} -\frac{\sqrt{3}}{2} A T^2 + n(T) \\ \frac{\sqrt{3}}{2} A T^2 + n(T) \end{cases}$$

$$\mu_y = E(y(T)) = E(g_o(T)) + E(n(T))$$

$$= E\left(\mp \frac{\sqrt{3}}{2} AT^2 + n(T)\right) = \mp \frac{\sqrt{3}}{2} AT^2 + E(n(T))$$

$$E(n(T)) = E\left(\int w(\tau)\, g(\tau)\, d\tau\right)$$

$$= \int_0^T \mp A\, E(w(\tau))\, d\tau = 0$$

$$\mu_y = \begin{cases} -\frac{\sqrt{3}}{2} AT^2 \\ \frac{\sqrt{3}}{2} AT^2 \end{cases}$$

$$\sigma_y^2 = var(y(T)) = E(g_o(T) + n(T)) = E\left(\mp \frac{\sqrt{3}}{2} AT^2 + n(T)\right)$$

$$\sigma_y^2 = var(n(T)) = E(n^2(T)) = \int_{-\infty}^{\infty} S_n(f)\, df$$

$$\sigma_y^2 = \frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2 df = \frac{N_0}{2} \int_0^T |h(t)|^2 dt$$

$$\sigma_y^2 = \frac{N_0}{2} \int_0^T 3t^2\, dt = \frac{N_0}{2} T^3$$

$$p(y) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(y-\mu)^2}{2\sigma^2}}$$

$$p(y|'0') = \frac{1}{\sqrt{\pi N_0 T^3}}\, e^{-\frac{(y+\frac{\sqrt{3}}{2}AT^2)}{N_0 T^3}}$$

$$p(y|'1') = \frac{1}{\sqrt{\pi N_0 T^3}}\, e^{-\frac{(y-\frac{\sqrt{3}}{2}AT^2)}{N_0 T^3}}$$

$$p(e|'0') = \int_0^{\infty} \frac{1}{\sqrt{\pi N_0 T^3}}\, e^{-\frac{(y+\frac{\sqrt{3}}{2}AT^2)}{N_0 T^3}}$$

By substitution

$$P(e|'0') = \int_{\frac{\frac{\sqrt{T}}{2}AT^2}{\sqrt{N_0 T^3}}}^{\infty} \frac{1}{\sqrt{\pi N_0 T^3}} e^{-z} \sqrt{N_0 T^3}\, dz$$

$$= \frac{1}{2} \operatorname{erfc}\left(\frac{\frac{\sqrt{T}}{2} AT^2}{\sqrt{N_0 T^3}} \right)$$

assume $P('0') = P('1')$

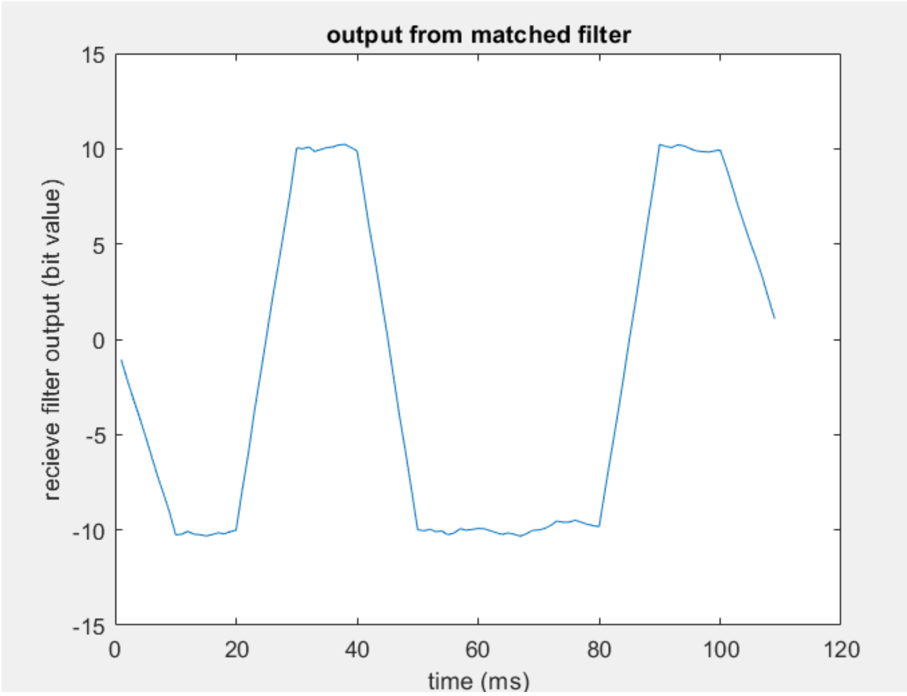
$$\therefore P(e) = \frac{1}{2} \operatorname{erfc}\left(\frac{\frac{\sqrt{T}}{2} AT^2}{\sqrt{N_0 T^3}} \right)$$

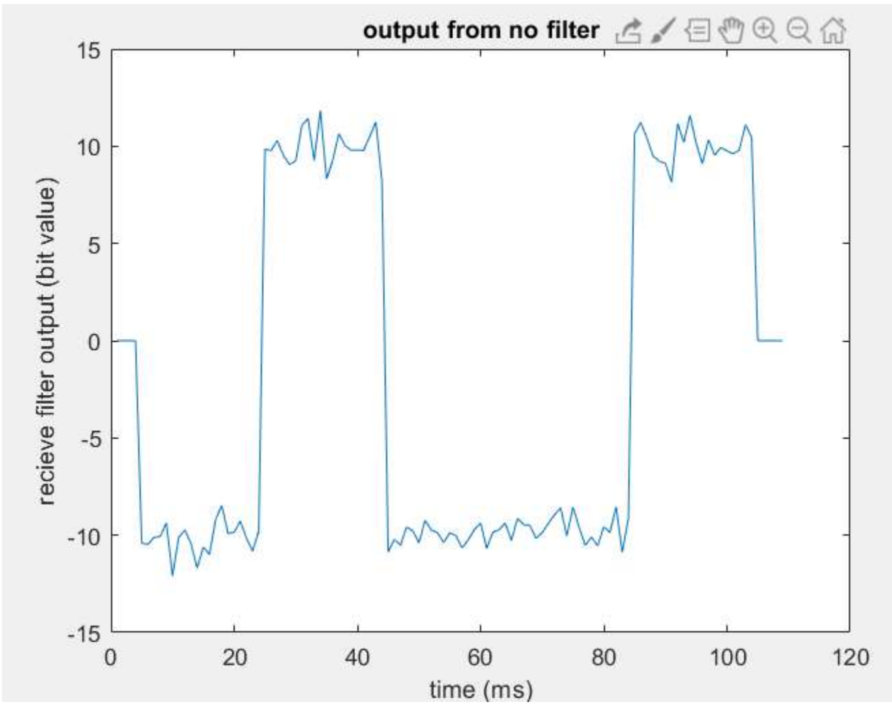
@ $A = 1$, $T = 1$ $\Rightarrow P(e) = \frac{1}{2} \operatorname{erfc}\left(\frac{\frac{\sqrt{T}}{2}}{\sqrt{N_0}} \right)$

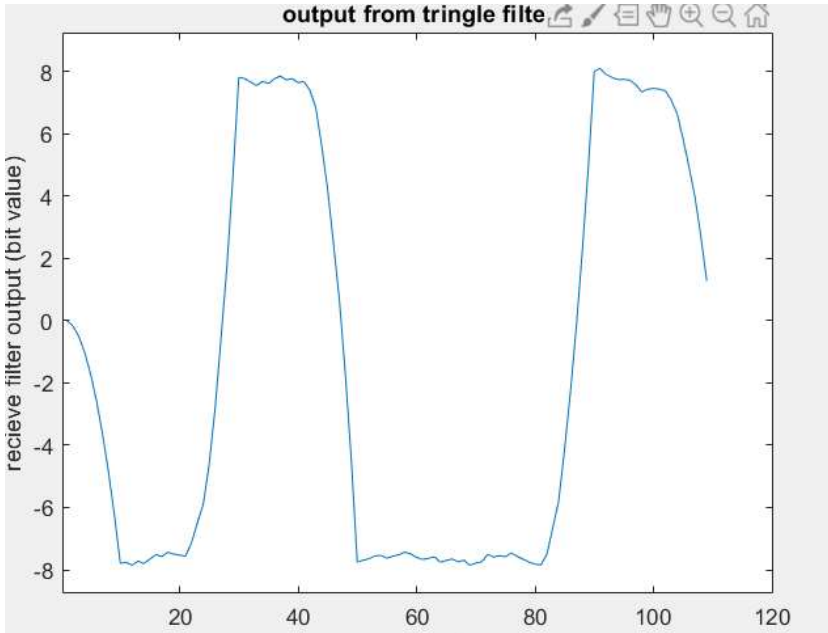
Scanned with CamScanner

Comment:

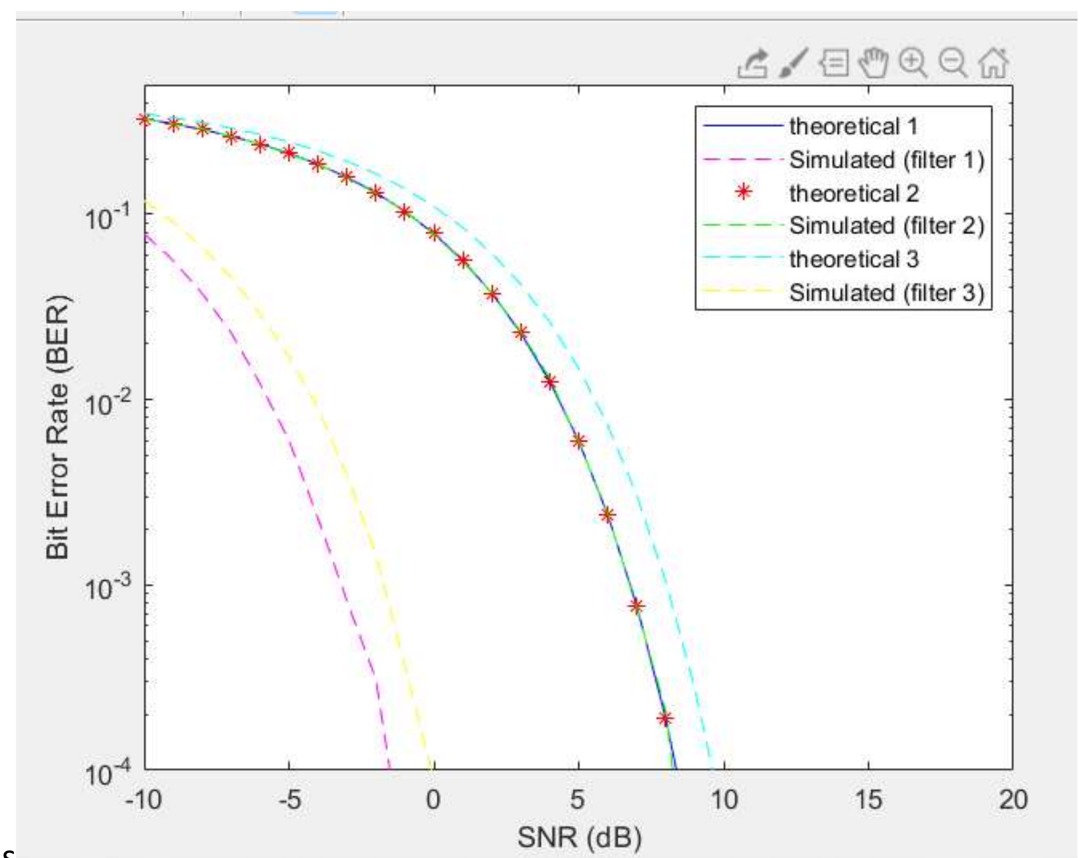
3. Plot the output of the receive filter for the three mentioned cases
For input: 0011000011







4.**On the same figure, plot the theoretical and simulated Bit Error Rate (BER) Vs E/No (where E is the average symbol energy) for the three mentioned cases. Take E/No to be in the range -10 dB: 20:dB. (Use a semilogy plot)**



S

5.Is the BER an increasing or a decreasing function of E/No? Why?

BER is decreasing function of $\boldsymbol{E/No}$

because by increasing **E/$\boldsymbol{No}$** we increase the transmitted energy related to noise energy. The transmitted signal was not.

significantly impacted by noise. but if noise energy close to transmitted energy then noise easily destroy the signal

6. Which case has the lowest BER? Why?

matched filter has the lowest BER.

because of multiplying high values in signal by high values in filter than in thresholding step then you can easily detect it is high value,

the same thing to low values

but other filters don't multiply signal values with suitable values.

Code:

```
%% constants
num_bits = 10;
num_of_samples = 10;


%% generate random bits
random_Indices = randperm(num_bits, num_bits/2);
array= ones(num_bits,1);
array(random_Indices)=-1;
% For input: 0011000011
array(1)=-1;
array(2)=-1;
array(3)=1;
array(4)=1;
array(5)=-1;
array(6)=-1;
array(7)=-1;
array(8)=-1;
array(9)=1;
array(10)=1;
% range of SNR
snr_range = -10:1:20;
%% receive with matched filter
filter_matched = ones(1,num_of_samples);
energy=1;

for i = 1:length(snr_range)
  snr = 10 ^(snr_range(i)/10);
  signal_with_noise_1=noise(num_bits,num_of_samples,array,energy/(2.0*snr));

[filtered_samples_1,filtered_bits_1]=applay_filter(num_bits,num_of_samples,signal_with_noise_1,filter_matched);
end
% ploting
out_1 = reshape(filtered_bits_1, [], 1);
figure(1)
t1 = 1:(length(out_1));
plot(t1,  out_1)
```

```
title('output from matched filter')
xlabel('time (ms)')
ylabel('recieve filter output (bit value)')

%% receive with no filter
filter_no_filter = zeros(1,num_of_samples);
filter_no_filter(num_of_samples/2)=10;
for i = 1:length(snr_range)
  % conert to value fro db
  snr = 10 ^(snr_range(i)/10);
  signal_with_noise_2=noise(num_bits,num_of_samples,array,energy/(2.0*snr));

[filtered_samples_2,filtered_bits_2]=applay_filter(num_bits,num_of_samples,signal_with
_noise_2,filter_no_filter);
end
% ploting
out_2 = reshape(filtered_bits_2, [], 1);
figure(2)
t2 = 1:(length(out_2));
plot(t2,  out_2)
title('output from no filter')
xlabel('time (ms)')
ylabel('recieve filter output (bit value)')

% receive with tringle
t3 = 0:1: 10 - 1;
filter_tringle =(1.732 / 10) * t3;
for i = 1:length(snr_range)
  snr = 10 ^(snr_range(i)/10);
  signal_with_noise_3=noise(num_bits,num_of_samples,array,energy/(2.0*snr));

[filtered_samples_3,filtered_bits_3]=applay_filter(num_bits,num_of_samples,signal_with
_noise_3,filter_tringle);
end
% ploting
out_3 = reshape(filtered_bits_3, [], 1);
figure(3)
t3 = 1:(length(out_3));
plot(t3,  out_3)
title('output from tringle filter')
```

```
xlabel('time (ms)')
ylabel('recieve filter output (bit value)')
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
num_bits_2 = 100000; % siganl 10^5
random_Indices_2 = randperm(num_bits_2, 1);
array_2= ones(num_bits_2,1);
array_2(random_Indices_2)=-1;
BER_experimental_1=zeros(length(snr_range),1);
BER_theortical_1 =zeros(length(snr_range),1);
BER_experimental_2=zeros(length(snr_range),1);
BER_theortical_2 =zeros(length(snr_range),1);
BER_experimental_3=zeros(length(snr_range),1);
BER_theortical_3 =zeros(length(snr_range),1);
for i = 1:length(snr_range)
   snr = 10 ^(snr_range(i)/10);
   signal_with_noise=noise(num_bits_2,num_of_samples,array_2,energy/(2.0*snr));

[filtered_samples_1,filtered_bits_1]=applay_filter(num_bits_2,num_of_samples,signal_with_noise,filter_matched);

[filtered_samples_2,filtered_bits_2]=applay_filter(num_bits_2,num_of_samples,signal_with_noise,filter_no_filter);

[signal_with_noise_3,filtered_bits_3]=applay_filter(num_bits_2,num_of_samples,signal_with_noise,filter_tringle);
   BER_experimental_1(i)=simulated_BER(array_2,filtered_samples_1);
   BER_experimental_2(i)=simulated_BER(array_2,filtered_samples_2);
   BER_experimental_3(i)=simulated_BER(array_2,signal_with_noise_3);
   BER_theortical_1(i)=0.5*erfc(sqrt(snr));
   BER_theortical_2(i)=0.5*erfc(sqrt(snr));
   BER_theortical_3(i)=0.5*erfc((sqrt(3)/(2)*sqrt(snr)));
end

figure(4)

semilogy(snr_range, BER_theortical_1, 'b-')
hold on
semilogy(snr_range, BER_experimental_1, 'm--')
```

```
% xlabel('SNR (dB)')
% ylabel('Bit Error Rate (BER)')
% legend('theoretical 1', 'Simulated (filter 1)')
% figure(5)

hold on
semilogy(snr_range, BER_theortical_2, 'r*')
hold on
semilogy(snr_range, BER_experimental_2,'g--')
% xlabel('SNR (dB)')
% ylabel('Bit Error Rate (BER)')
% legend('theoretical 2', 'Simulated (filter 2)')

% figure(6)
hold on
semilogy(snr_range, BER_theortical_3, 'c--')
hold on
semilogy(snr_range, BER_experimental_3,'y--')
hold off
ylim([10^-4 0.5])
xlim([-10 20])
xlabel('SNR (dB)')
ylabel('Bit Error Rate (BER)')
legend('theoretical 1','Simulated (filter 1)','theoretical 2',  'Simulated (filter 2)','theoretical
3','Simulated (filter 3)')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% utilts
%% add noise to signal
% it is expanding generated_bits so each sample change to train of pulses to
discretization the pulse
% then adding gussian noise for expanded array.
function [signal_with_noise] = noise(num_of_bits,num_of_samples,indx,sigma)
  X = normrnd(0, sqrt(sigma), [1,num_of_bits*num_of_samples]);
  signal_with_noise = ones(length(indx), num_of_samples);
  for i= 1:(length(indx))
      signal_with_noise(i, :) =signal_with_noise(i, :)* indx(i);
      signal_with_noise(i, :) = signal_with_noise(i, :)+X((num_of_samples)*(i-
1)+1:(num_of_samples)*(i));
```

```
  end
end
%% applay Filter
% it calculates convolution for received_filter and each sample
function [result_sampled,result] =
applay_filter(num_bits,num_of_samples,signal_with_noise,filter)
   result_sampled = zeros(num_bits,1);
   signal_with_noise_reshape=reshape(signal_with_noise.',1,[]);
   result = conv(signal_with_noise_reshape, filter);
   for i=0:(num_bits-1)
      result_sampled(i+1) = result((num_of_samples - 1) + num_of_samples * i+1);
   end
end
%% calcute simulated BER
% it calculates convolution for received_filter and each sample
function [BER_value] = simulated_BER(true_values,result)
   received_samples = ones(size(true_values));
   received_samples = received_samples + (-2 * (result < 0));
   error_probability = sum(received_samples ~= true_values);
   BER_value = error_probability / size(true_values, 1);
end

% ?Q5:
% BER is decreasing function of ?/??
% because by increasing ?/?? we increase the transmitted energy related to noise energy. The transmitted signal was not
% significantly impacted by noise. but if noise energy close to transmitted energy then noise easily destroy the signal

% Q6:
% matched filter has the lowest BER
% because of multipling high values in signal by high values in filter than in thrsholding step then you can easly detect it is high value,
% the same thing to low values
% but other filter doesn't multply signal values with suitable values
```