Cairo University
Faculty of Engineering

Department of Computer Engineering

# ELC 325B – Spring 2023

## Digital Communications

# Assignment #2

## Matched Filter

**Submitted to**

Eng. Mohamed Khaled

**Submitted by**

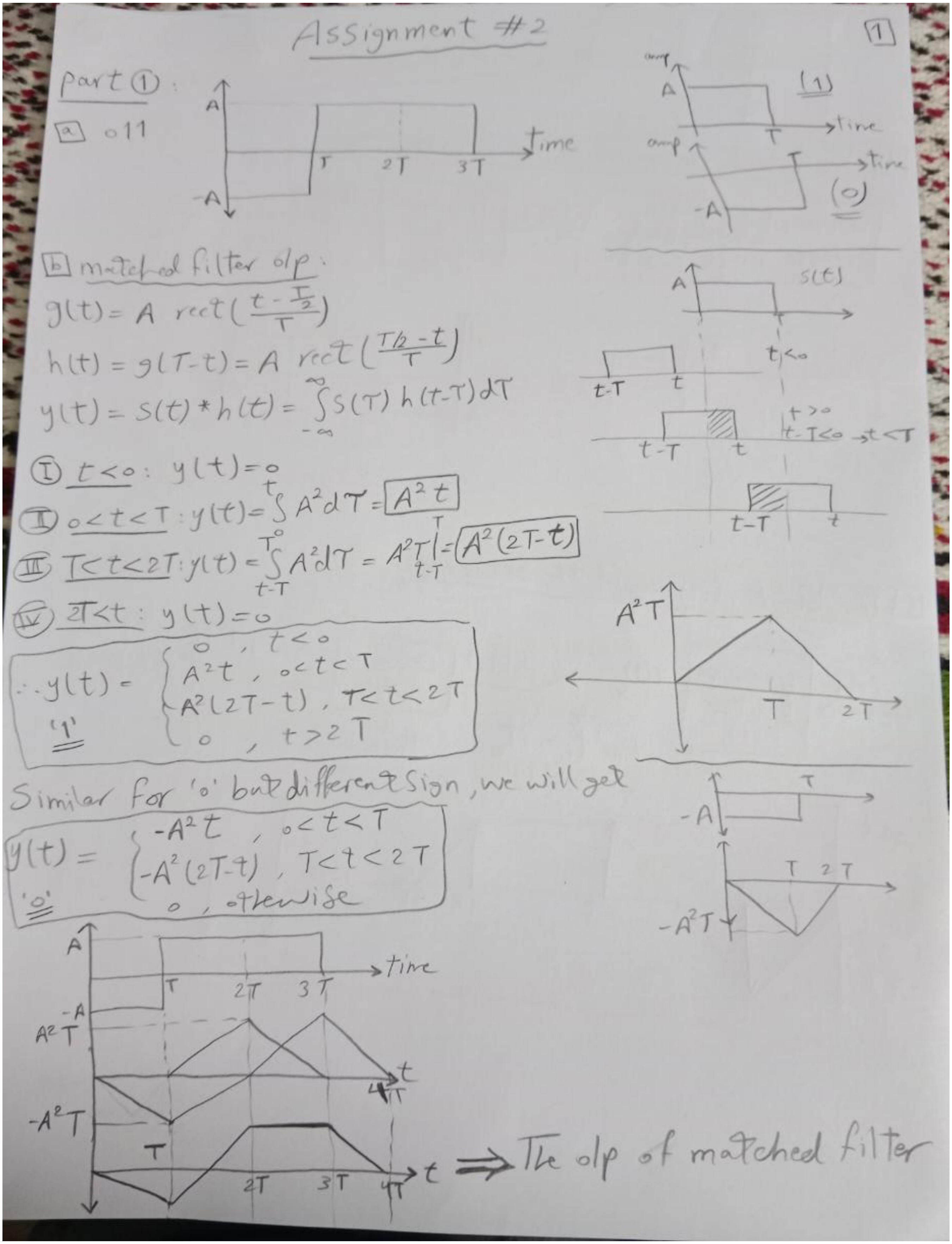
| Name | Sec | BN |
|---|---|---|
| محمد وليد فتحي | 2 | 20 |
| كريم محمود كمال | 2 | 10 |

# Contents



# Figures

## **Part 1: Solution of the question:**



Assignment #2 1

part ①:

a 011

b matched filter o/p:

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

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

$y(t) = s(t) * h(t) = \int_{-\infty}^{\infty} s(\tau)\, h(t-\tau)\, d\tau$

Ⅰ $t<0$: $y(t)=0$

Ⅱ $0<t<T$: $y(t)=\int_{0}^{t} A^2 d\tau = \boxed{A^2 t}$

Ⅲ $T<t<2T$: $y(t)=\int_{t-T}^{T} A^2 d\tau = A^2\tau\Big|_{t-T}^{T} = \boxed{A^2(2T-t)}$

Ⅳ $2T<t$: $y(t)=0$

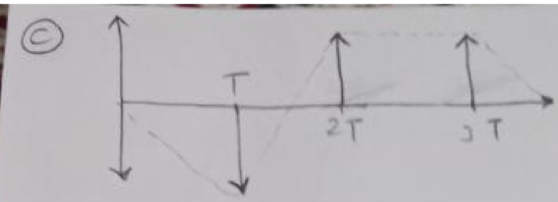
$$\therefore y(t)_{\text{'1'}} = \begin{cases} 0, & t<0 \\ A^2 t, & 0<t<T \\ A^2(2T-t), & T<t<2T \\ 0, & t>2T \end{cases}$$

Similar for '0' but different sign, we will get

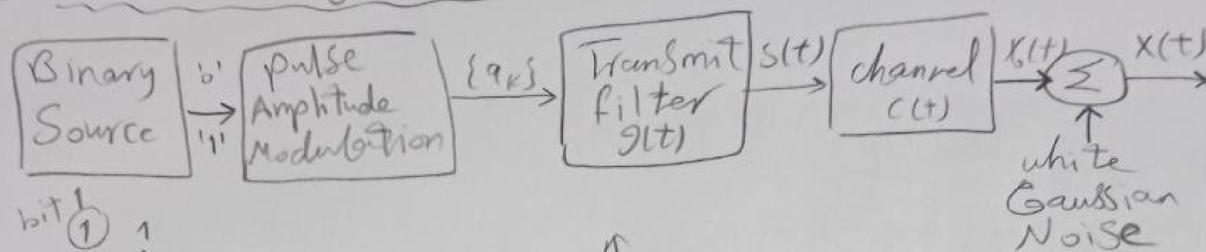
$$y(t)_{\text{'0'}} = \begin{cases} -A^2 t, & 0<t<T \\ -A^2(2T-t), & T<t<2T \\ 0, & \text{otherwise} \end{cases}$$

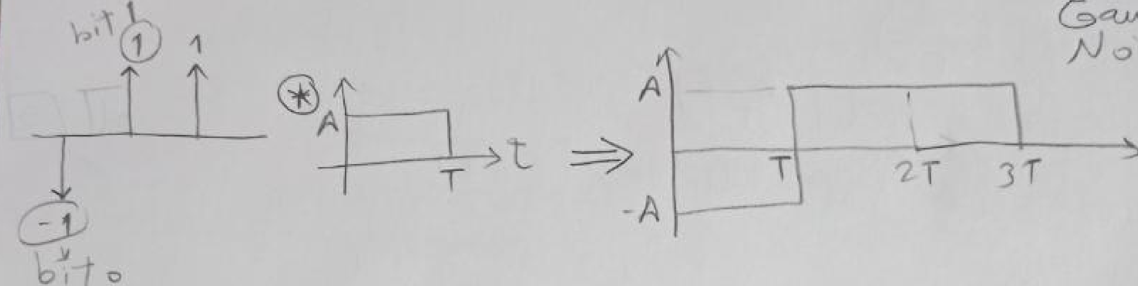
⇒ The o/p of matched filter

(c)

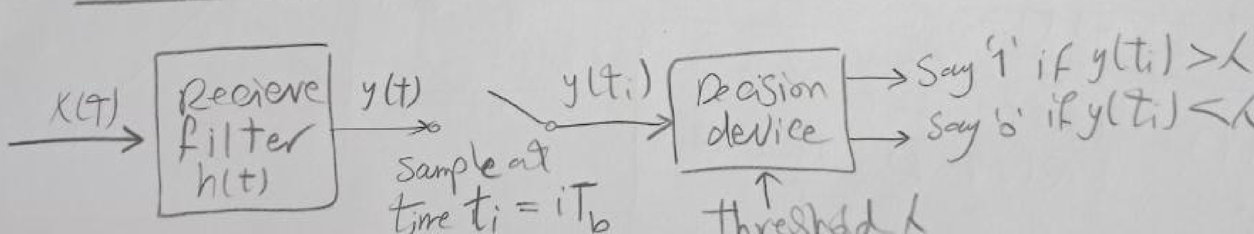


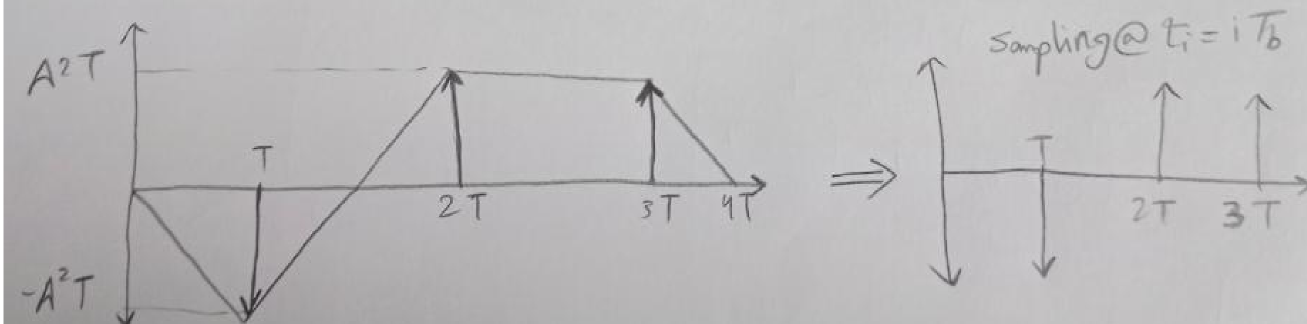

d The Block diagram of the transmitter:




e The block diagram of the reciever:



# Part 2: Probability of Error in the Three cases:

Part II

3

1 Derivation of the probability of error in the three cases:

(a) First case: The receive filter $h(t)$ is a matched filter with unit energy

$$g(t) = \begin{cases} -A, & 0<t<T \\ A, & 0<t<T \end{cases}$$

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

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

$$y(t) = r(t) * h(t) = g(t) * h(t) + w(t) * h(t) = \underline{g_o(t)} + \underline{n(t)}$$

$$g_o(t) = \begin{cases} A^2 t, & 0<t<T \\ A^2(2T-t), & T<t<2T \\ 0, & \text{otherwise} \end{cases} \quad \text{(from previous steps in part I)}$$

$$\therefore y(t) = \begin{cases} A^2T + n(T), & \text{'1'} \\ -A^2T + n(T), & \text{'0'} \end{cases}$$

Calculate $M$, $\delta^2$:

$$M_y = E\{y(T)\} = E\{g_o(T)\} + \underline{E\{n(T)\}} \qquad \boxed{E\{g_o(T)\} = \begin{cases} A^2T, & \text{'1'} \\ -A^2T, & \text{'0'} \end{cases}}$$

$$E\{n(T)\} = E\{\int w(t)\,\underbrace{h(T-\tau)}_{g(\tau)}\,d\tau\} = E\{\int_0^T \pm A\,w(\tau)\,d\tau\} = \boxed{0}$$

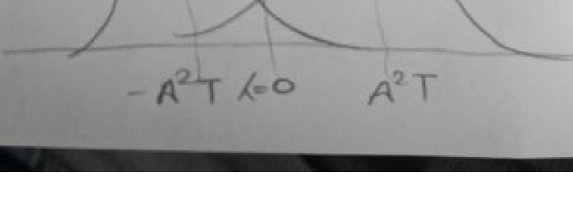
$$\therefore M_y = \begin{cases} A^2T, & \text{'1'} \\ -A^2T, & \text{'0'} \end{cases}$$

$$\delta_y^2 = E\{n^2(T)\} - \underbrace{\cancel{E^2\{n(T)\}}}_{\text{zero}} = E\{n^2(T)\} = \int_{-\infty}^{\infty} S(f)\,df = \frac{N_o}{2}\int_{-\infty}^{\infty} |H(f)|^2\,df$$

$$= \frac{N_o}{2}\int_0^T |h(t)|^2\,dt = \boxed{\frac{N_o}{2}A^2T} \qquad \boxed{p(y) = \frac{1}{\sqrt{2\pi\delta_y^2}}\,e^{-\frac{(x-M_y)^2}{2\delta_y^2}}}$$

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

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



$\boxed{4}$

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

Let $z = \frac{y+A^2T}{\sqrt{N_0 A^2 T}} \longrightarrow dz = \frac{dy}{\sqrt{N_0 A^2 T}}$ $\qquad y=0 \to z = \frac{A^2T}{\sqrt{N_0A^2T}} = \sqrt{\frac{A^2T}{N_0}}$

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

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

$\because p('1') = p('0') = \frac{1}{2}$ , $p(e|'0') = p(e|'1')$

$\therefore p(e) = p(e|'1')p('1') + p(e|'0')p('0') = \frac{1}{2}[p(e|'0') + p(e|'1')]$

$\boxed{p(e) = p(e|'0')}$

$\boxed{\therefore p(e) = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{A^2T}{N_0}}\right)}$

in case of $A=1$, $T=1$: $\boxed{\therefore p(e) = \frac{1}{2} \operatorname{erfc}\left(\frac{1}{\sqrt{N_0}}\right)}$

---

ⓑ The recieve filter $h(t)$ is not exist $(h(t)=\delta(t))$:

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

$y(T) = g_0(T) + n(T) \qquad \therefore y(T) = \begin{cases} A + n(T), & '1' \\ -A + n(T), & '0' \end{cases}$

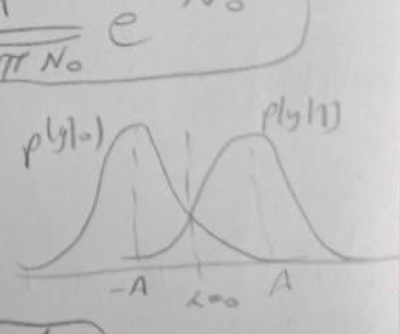
$E\{y(T)\} = E\{g_0(T)\} + \underbrace{\cancel{E\{n(T)\}}}_{0}$

$= \pm A \qquad \boxed{\therefore M_y = \begin{cases} A, & '1' \\ -A, & '0' \end{cases}}$

$\sigma_y^2 = E\{n^2(T)\} = \boxed{\frac{N_0}{2}}$

$\boxed{\therefore p(y|'0') = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y+A)^2}{N_0}}} \qquad \boxed{p(y|'1') = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y-A)^2}{N_0}}}$

$p(e|'0') = \int_0^{\infty} \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y+A)^2}{N_0}} dy \qquad$ Let $\frac{y+A}{\sqrt{N_0}} = z$, $dz = \frac{dy}{\sqrt{N_0}}$




$$= \int_{\frac{A}{\sqrt{N_0}}}^{\infty} \frac{1}{\sqrt{\pi}} e^{-z^2} dz = \boxed{\frac{1}{2} \operatorname{erfc}\left(\frac{A}{\sqrt{N_0}}\right)}$$

if $A=1 \rightarrow \therefore \boxed{p(e|'0') = \frac{1}{2} \operatorname{erfc}\left(\frac{1}{\sqrt{N_0}}\right)}$

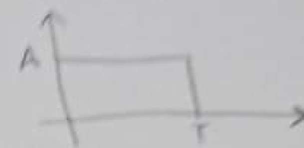
$\underline{p(e)} = \frac{1}{2}[p(e|1)+p(e|0)] = p(e|0) = \boxed{\frac{1}{2}\operatorname{erfc}\left(\frac{1}{\sqrt{N_0}}\right)}$



Q3 The recieve filter $h(t)$ has the following impulse response


$y(t) = g_0(t) + n(t)$

$g_0(t) = g(t) * h(t) = \int_{-\infty}^{\infty} g(\tau)\, h(t-\tau)$




Ⅰ $\underline{t<0}$ : $g_0(t) = 0$

Ⅱ $\underline{0<t\leq T}$ : $g_0(t) = \int_0^t \sqrt{3}A\tau\, d\tau = \sqrt{3}A \left.\frac{\tau^2}{2}\right|_0^t$

$= \frac{\sqrt{3}}{2}At^2$


Ⅲ $\underline{T<t\leq 2T}$ : not need to compute it

$y(T) = \begin{cases} -\frac{\sqrt{3}}{2}AT^2 + n(T), & \underline{'0'} \\ \frac{\sqrt{3}}{2}AT^2 + n(T), & \underline{'1'} \end{cases}$

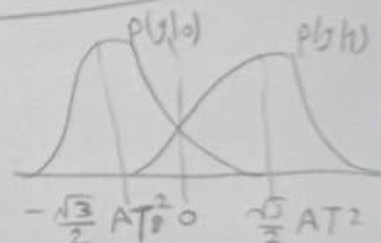
$\mu_y = E\{g_0(T)\} + E\{\cancel{n(T)}\}^{0} = E\{g_0(T)\}$ $\therefore \mu_y = \begin{cases} -\frac{\sqrt{3}}{2}AT^2 & '0' \\ \frac{\sqrt{3}}{2}AT^2 & '1' \end{cases}$

$\sigma_y^2 = E\{n^2(T)\} - E^2\{\cancel{n(T)}\}^{0}$

$= \frac{N_0}{2}\int |h(t)|^2 dt = \frac{N_0}{2}\int_0^T 3t^2\, dt = \frac{N_0}{2}T^3$

$p(y|'0') = \frac{1}{\sqrt{2\pi \frac{N_0}{2}T^3}}\, e^{-\frac{(y+\frac{\sqrt{3}}{2}T^2)^2}{N_0T^3}}$ $\quad \therefore p(y|'0') = \frac{1}{\sqrt{\pi N_0 T^3}}\, e^{-\frac{(y+\frac{\sqrt{3}}{2}AT^2)^2}{N_0 T^3}}$

$p(y|'1') = \frac{1}{\sqrt{\pi N_0 T^3}}\, e^{-\frac{(y-\frac{\sqrt{3}}{2}AT^2)^2}{N_0 T_3}}$




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

$= \int_{\frac{\frac{\sqrt{3}}{2}AT^2}{\sqrt{N_0T^3}}}^{\infty} \frac{1}{\sqrt{\pi}}\, e^{-z^2} dz = \frac{1}{2}\operatorname{erfc}\left(\frac{\frac{\sqrt{3}}{2}AT^2}{\sqrt{N_0 T^3}}\right)$

$p(e) = p(e|0) = \frac{1}{2}\operatorname{erfc}\left(\frac{\frac{\sqrt{3}}{2}AT^2}{\sqrt{N_0T^3}}\right)$ # $\underline{\text{if } A=1,\ T=1:}$

$\therefore p(e) = \frac{1}{2}\operatorname{erfc}\left(\frac{\frac{\sqrt{3}}{2}}{\sqrt{N_0}}\right)$ #

## Part 2:

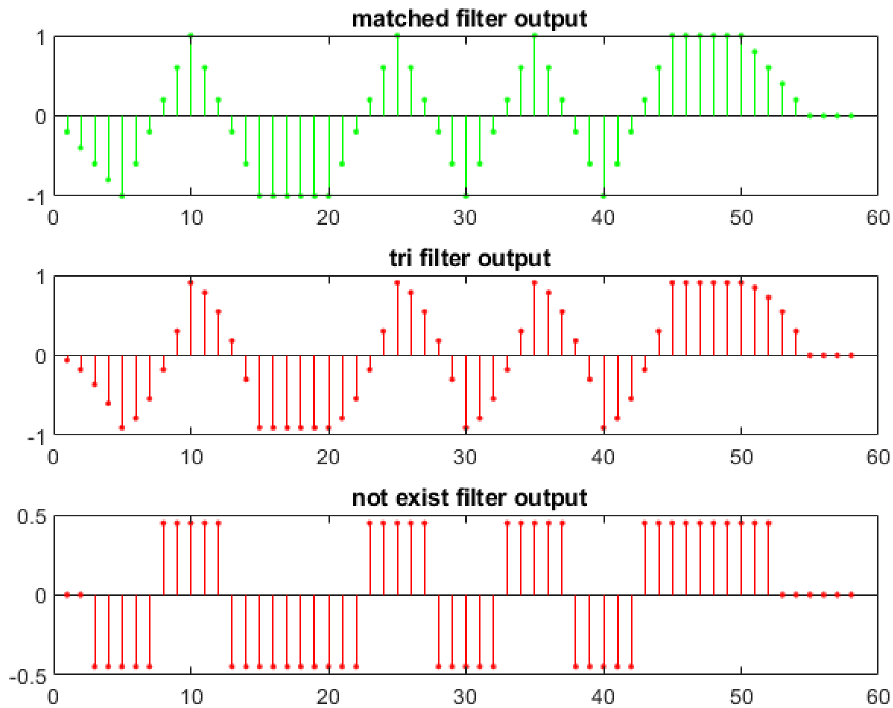



*Fig1. Filters Output in the 3 cases*

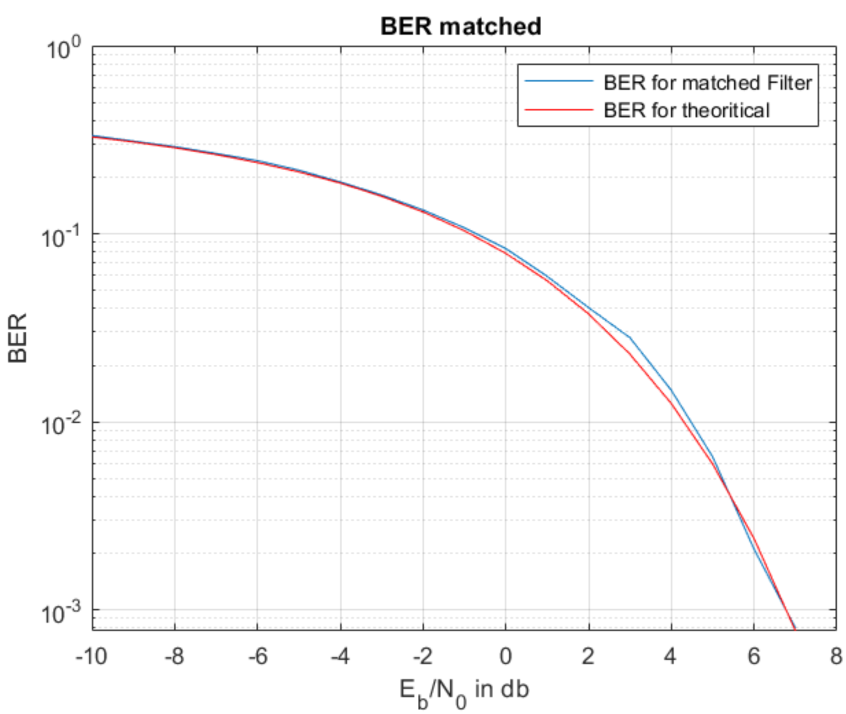



*Fig2. BER Matched Filter*

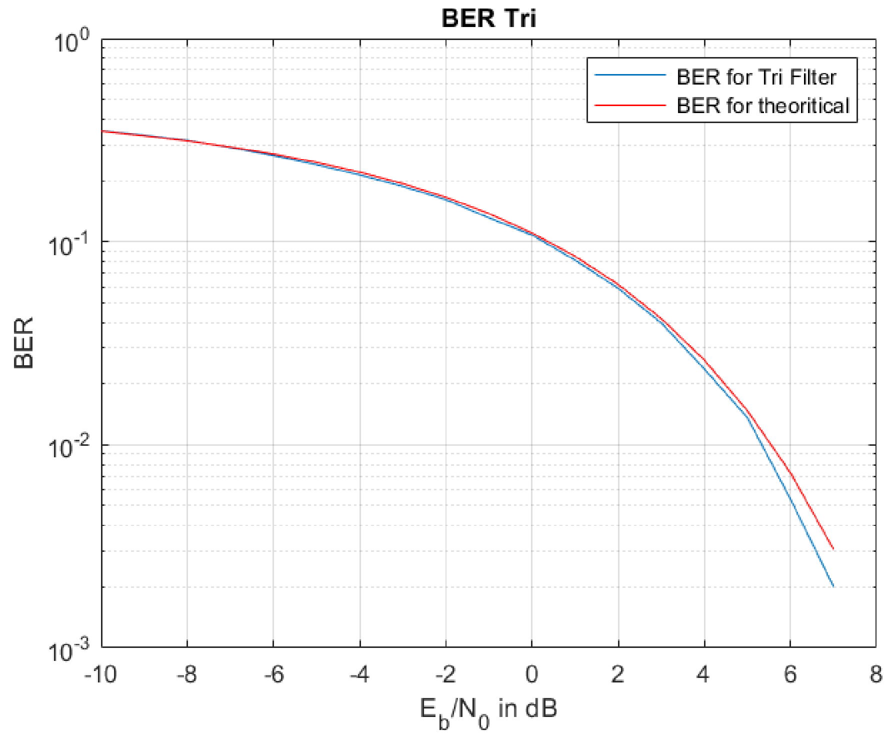



***Fig3. BER Tri Filter***

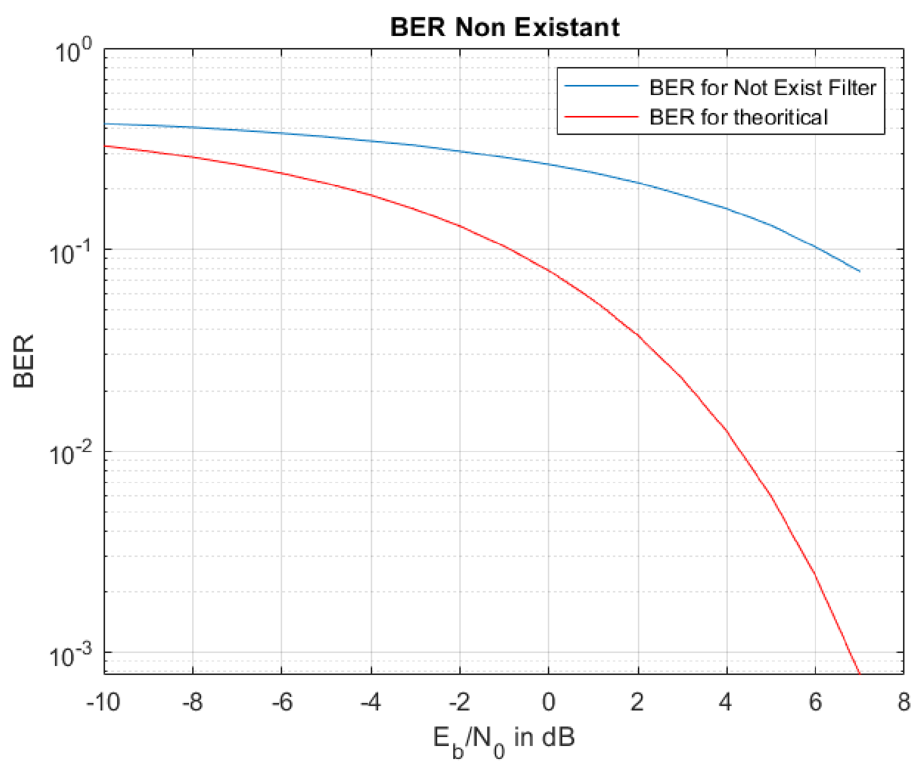



***Fig4. BER Non-Existent Filter***

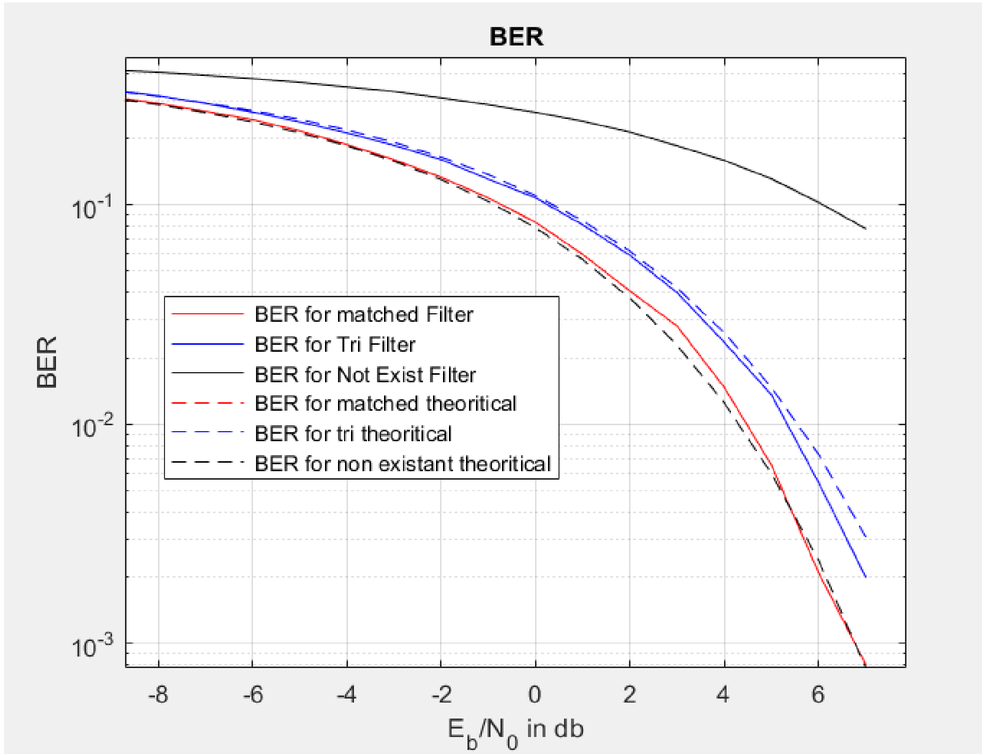



*Fig5. BER of All Together*

## comment:

The theoretical BER of the matched filter and theoretical BER of the non existent filter are completely overlapped so that they are shown as one line in the graph.

## Comment:

These graphs are from -10:7 in Eb/N0 not -10:20 ***to show the scale well.***

***Below is a plot showing the Tri Filter with a range of Eb/N0 = -10:20.***

The BER of the filter is diminished next to the theoretical BER.

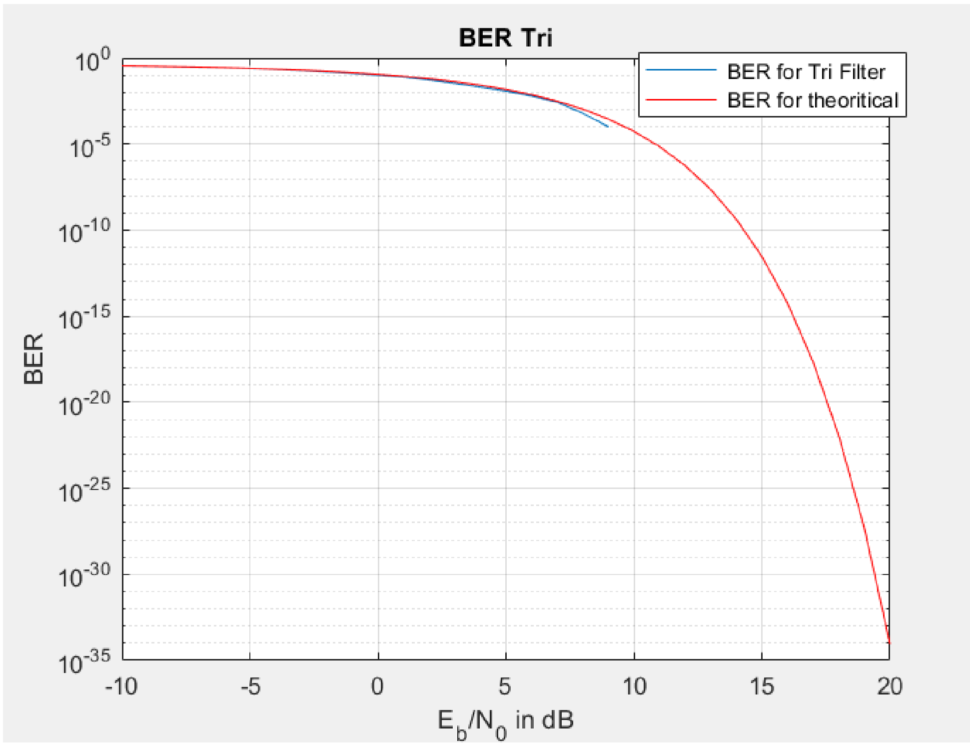



*Fig6. BER Tri Filter -10:20*

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

The Bit Error Rate (BER) is typically a decreasing function of the energy per noise density ($E/N_0$).

This is because $E/N_0$ is a measure of the signal-to-noise ratio (SNR).

SNR = signal power / noise power

As the SNR increases, the received signal becomes stronger relative to the noise, making it easier to detect individual bits accurately.

## 6- Which case has the lowest BER? Why?

The matched filter has the lowest BER as it maximizes the SNR of the received signal which leads to better BER performance.

## Code:

```
clear all; clc; close all;

% need to mul by sqrt(3)
% 3 + 12 + 27 + 48 + 75
tri_filter= sqrt(3) * [1 2 3 4 5 ] / sqrt(165) ;
matched_rect_filter = [1 1 1 1 1 ]/sqrt(5) ;
not_exist_filter = [0 0 1 0 0 ];

figure(1);
subplot(2,1,1);
plot(matched_rect_filter);
subplot(2,1,2);
stem(matched_rect_filter , 'r' , '.')

figure(2);
subplot(2,1,1);
plot(tri_filter);
subplot(2,1,2);
stem(tri_filter , 'r' , '.')

figure(3);
subplot(2,1,1);
plot(not_exist_filter);
subplot(2,1,2);
```

```
stem(not_exist_filter , 'r' , '.')




A=1 ;

number_of_bits1 = 10;


number_of_bits2 = 10000;


number_of_samples =5 ;

[data , data_sampled] = samples_generation(A ,number_of_samples ,
number_of_bits1);


y1 = conv (matched_rect_filter , data_sampled) ;


figure(4)

subplot(2,1,1);

plot(y1);

subplot(2,1,2);

stem(y1 , 'r' , '.')


% filter use the two types of filters and plot the output

[matched_filter_output ,tri_filter_output, not_exist_filter_output ] = filter_block(
matched_rect_filter ,tri_filter ,not_exist_filter, y1) ;

figure(5);

subplot(3,1,1);
```

```
stem(matched_filter_output,'g','.');
title(' matched filter output')
subplot(3,1,2);
stem(tri_filter_output,'r','.')
title(' tri filter output')
subplot(3,1,3);
stem(not_exist_filter_output,'r','.')
title(' not exist filter output')

% Repeat a, b, and c from 1 above but generate 10000 bits instead of 10 bits
[data2 , data_sampled2] = samples_generation(A ,number_of_samples ,
number_of_bits2);
y2 = conv (matched_rect_filter , data_sampled2) ;
n = randn(1 , length(y2));
Eb = 1 ;

BER_matched_cases=ones(1,18);
BER_tri_cases=ones(1,18) ;
BER_not_exist_cases=ones(1,18) ;

BER_theoretical=ones(1,18);
BER_theoretical_tri = ones(1,18);
BER_theoretical_not_existant = ones(1,18);

counter=1;
T = 1;
```

```
for Eb_over_N0 =-10:7

    N0=Eb/(10^(Eb_over_N0/10));
    noise = n*sqrt(N0/2);
    V_n=noise+y2;

    [matched_filter_output ,tri_filter_output, not_exist_filter_output ] = filter_block(matched_rect_filter ,tri_filter ,not_exist_filter, V_n );

    [recived_bits_matched_after_sample,recived_bits_tri_after_sample, recived_bits_not_exist_after_sample ] = sampling_block (matched_filter_output ,tri_filter_output ,not_exist_filter_output, number_of_bits2 ,number_of_samples );

    [matched_after_descision ,tri_after_descision, not_exist_after_descision ]=descision_block(recived_bits_matched_after_sample ,recived_bits_tri_after_sample,recived_bits_not_exist_after_sample  );

    [BER_matched , BER_tri, BER_not_exist ]=bit_error_rate(data2 , matched_after_descision  ,tri_after_descision ,not_exist_after_descision, number_of_bits2 );

    BER_matched_cases(counter)=BER_matched;
    BER_tri_cases(counter)=BER_tri;
    BER_not_exist_cases(counter)=BER_not_exist;

    %BER_theoretical(counter) = (1/3) * erfc(sqrt((3/2) * (Eb/N0)))
    BER_theoretical(counter)=0.5 *erfc(sqrt(Eb/N0));
```

```
   BER_theoretical_tri(counter) = 0.5 * erfc(((sqrt(3)/2)*Eb)/sqrt(N0*T*T*T))
   BER_theoretical_not_existant(counter) = 0.5 * erfc(A/sqrt(N0));


   counter=counter+1 ;
end


semi_log_plot_BER(BER_matched_cases
,BER_tri_cases,BER_not_exist_cases,BER_theoretical, BER_theoretical_tri,
BER_theoretical_not_existant);




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Bit Error Rate
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [BER_matched ,BER_tri, BER_not_exist ] = bit_error_rate( data ,
matched_after_descision , tri_after_descision ,not_exist_after_descision,
number_of_bits)


  Error_num_matched=0;
  Error_num_rect=0;
  Error_num_not_exist=0;


 for i=1:10000


    if(matched_after_descision(i)~=data(i))
      Error_num_matched=Error_num_matched+1;
    end
```

```
     if(tri_after_descision(i)~=data(i))
       Error_num_rect=Error_num_rect+1;
     end

     if(not_exist_after_descision(i)~=data(i))
       Error_num_not_exist=Error_num_not_exist+1;
     end

  end

   BER_tri=Error_num_rect/number_of_bits;
   BER_matched=Error_num_matched/number_of_bits;
   BER_not_exist=Error_num_not_exist/number_of_bits;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Descision Block
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [matched , tri, not_exist] = descision_block (matched , tri, not_exist )

   for i=1:10000

      if(matched(i)>0)
       matched(i)=1;
      else
       matched(i)=0;
```

```
        end


        if(tri(i)>0)
         tri(i)=1;
        else
         tri(i)=0;
        end


        if(not_exist(i)>0)
         not_exist(i)=1;
        else
         not_exist(i)=0;
        end


    end


end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Filter Block
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [matched_filter_output ,tri_filter_output, not_exist_filter_output ] =
filter_block(pulseshape ,tri_filter ,not_exist_filter, y )


matchedfilter = fliplr(pulseshape);

matched_filter_output = conv(matchedfilter , y) ;

tri_filter_output = conv(tri_filter , y);
```

```
not_exist_filter_output = conv(not_exist_filter , y);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Samples Generation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [data , samples_generated] = samples_generation(A,number_of_samples , number_of_bits)

data_size = [1 , number_of_bits] ;
% generate the required array
data = randi([0,1] , data_size);
% mapping logic zero to -1 and logic 1 to 1
data_mapped =(2*data-1)*A;
samples_generated=upsample(data_mapped ,number_of_samples );

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Sampling Block %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [recived_bits_matched_after_sample,recived_bits_tri_after_sample, recived_bits_not_exist_after_sample ] = sampling_block (matched_filter_output ,tri_filter_output ,not_exist_filter_output, number_of_bits , number_of_samples)

   conter=1;
   recived_bits_matched_after_sample = ones (1 , number_of_bits);
   recived_bits_tri_after_sample = ones (1 , number_of_bits);
```

```
    recived_bits_not_exist_after_sample = ones (1 , number_of_bits);


    for i=5:number_of_samples:number_of_bits*number_of_samples
        recived_bits_matched_after_sample(conter)=matched_filter_output(i);
        recived_bits_tri_after_sample(conter)=tri_filter_output(i);
        recived_bits_not_exist_after_sample(conter)=not_exist_filter_output(i);


        conter=conter+1;
    end


end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Drawing in dB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [] = semi_log_plot_BER(BER_matched_cases
,BER_tri_cases,BER_not_exist_cases,BER_theoretical, BER_theoretical_tri,
BER_theoretical_not_existant)
figure(6);
semilogy(-10:7,BER_matched_cases);
hold all
semilogy(-10:7,BER_theoretical,'r');
grid on
title('BER matched ')
ylabel('BER')
xlabel('E_b/N_0 in db')
legend(' BER for matched Filter',' BER for theoretical');
```

```
figure(7);
semilogy(-10:7,BER_tri_cases)
hold all
semilogy(-10:7,BER_theoretical_tri, 'r')
grid on
ylabel('BER')
xlabel('E_b/N_0 in dB')
title('BER Tri')
legend(' BER for Tri Filter',' BER for theoretical');

figure(8);
semilogy(-10:7,BER_not_exist_cases)
hold all
semilogy(-10:7,BER_theoretical_not_existant, 'r')
grid on
ylabel('BER')
xlabel('E_b/N_0 in dB')
title('BER Non Existant')
legend(' BER for Not Exist Filter',' BER for theoretical');



figure(9);
semilogy(-10:7,BER_matched_cases, 'r');
hold all
semilogy(-10:7,BER_tri_cases, 'b');
```

```
semilogy(-10:7,BER_not_exist_cases, 'k');



semilogy(-10:7,BER_theoretical,'r--');


semilogy(-10:7,BER_theoretical_tri, 'b--')


semilogy(-10:7,BER_theoretical_not_existant,'k--')


grid on
title('BER')
ylabel('BER')
xlabel('E_b/N_0 in db')
legend(' BER for matched Filter',' BER for Tri Filter', ' BER for Not Exist Filter',' BER for matched theoretical', ' BER for tri theoretical', ' BER for non existant theoretical');



end
```