THE CODES AND EXPLINATIONS

```
% First let's assign some constants we will use

%amplitude has choosen to be 1 and since matlab is making it difficult to work with non-integer values for conditional statements let's  make T=100

T = 100;

A = 1;

t=linspace(0,T-1,T);

% Now let's define the components of the recieved signal at the input of the filter

st1=zeros(0,T);

for i = 1 : T/2

   st1(i) = A*sin((2*pi*i)/T);

  endfor

st2=zeros(0,T);

for i = T/2 : T

   st2(i) = A*sin((2*pi*(i-T/2))/T);
```

```
st1(i) = 0;

endfor


% Now let's calculate the a1 and a2 components of the z
a1 = trapz(t,st1.*st1);
a2 = trapz(t,-st2.*st2);

%This communication system tends to reach minimal BER (around 10^-9) around 23 dB
SNRdb = linspace(0,23,24);

%This next section calculates the energy of the recieved signal
Es1=trapz(t,abs(st1.*st1));
Es2=trapz(t,abs(st2.*st2));

Eb=Es1*(1/3) + Es2*(2/3);

% Impulse response and the energy of the matching filter
ht= st1(T-t) - st2(T-t) ;

Eh=trapz(t,abs(ht.*ht));

% Now lets calculate Noise power , the desicion treshold and parameters necessary for probabilistic functions
N0=Eb.*10.^(-SNRdb/10);

sigmasquared = (N0/2)*Eh;
```

```
sigma = sigmasquared.^(1/2);

treshold = (sigmasquared/(a1-a2))*(log((2/3)/(1/3))) + (a1+a2)/2 ;

%Now lets find the BER theoratically.
%The matlab is giving me some problems with the communications librariy so i am calculating the Q function with the help of erfc

BERtheory3=zeros(0,length(SNRdb));

for i=1 : length(SNRdb)

  BERtheory3(i)=      (1-((1/2)*erfc(((treshold(i)-a1)/(sigma(i)))/2.^1/2)))*(1/3)       + ((1/2)*erfc(((treshold(i)-a2)/(sigma(i)))/2.^1/2))*(2/3);
endfor

A
%For the simulation let's create a test input

testinput = randi([0 1],10000000,1);

testoutput = zeros(0,length(testinput));
```

% This loop scales the test output for a1 and a2 components

```
for i = 1 : length(testinput)
  if testinput(i) == 1
  testoutput(i) = a1 ;
elseif testinput(i) == 0
  testoutput(i) = a2;
  endif

  endfor
```

%Now lets create AWGN

```
n0i = randn(1,10000000);
BERSim = zeros(0,length(SNRdb));

z0=zeros(0,length(testinput));
errorcount = 0;
```

%This loop combines proper AWGN and the test output, finds the output of the desicion block and calculates the simulated BER

```
for i = 1 : length(SNRdb)

errorcount = 0;

alpha =((2*N0(i)).^1/2);

n0= n0i .* alpha ;
```

```
noisytestoutput1= testoutput  + n0 ;


  for c = 1 : length(testoutput)
    if noisytestoutput1(c) > treshold
        z0(c) = 1;
    else
        z0(c) = 0;


  endif


  endfor




  for j = 1 : length(testinput)
    if z0(j) ~= testinput(j)
    errorcount = errorcount + 1;


      endif


endfor


BERSim(i) = errorcount/length(testinput) ;




endfor
```

```
%Now let's make the plots

figure(1)
st_1=plot(t,st1)
st_2=plot(t,st2)

legend([st_1 st_2],{'s1(t)' , 's2(t)'});

figure(2)
plot(t,ht)
title('h(t)')


figure(3)
semilogy(SNRdb,BERtheory3,'r+-','linewidth',1);
title('Calculated BER')
xlabel("SNR(dB)");
ylabel("BER(Bit Error Rate)");


figure(4)
semilogy(SNRdb,BERtheory3);
title('Simulated BER')
xlabel("SNR(dB)");
ylabel("BER(Bit Error Rate)");
```

```
% First let's assign some constants we will use
%amplitude has choosen to be 1 and since matlab is making it difficult to work with non-integer values for conditional statements let's  make T=100



T = 100;

A = 1;

t=linspace(0,T-1,T);

% Now let's define the components of the recieved signal at the input of the filter
st1=zeros(0,T);
for i = 1 : T/2

    st1(i) = A*sin((2*pi*i)/T);

  endfor




st2=zeros(0,T);
for i = T/2 : T

    st2(i) = A*sin((2*pi*(i-T/2))/T);
    st1(i) = 0;

  endfor



% Now let's calculate the a1 and a2 components of the z
a1 = trapz(t,st1.*st1);
a2 = trapz(t,-st2.*st2);

%This communication system tends to reach minimal BER (around 10^-9) around 23 dB
SNRdb = linspace(0,23,24);

```

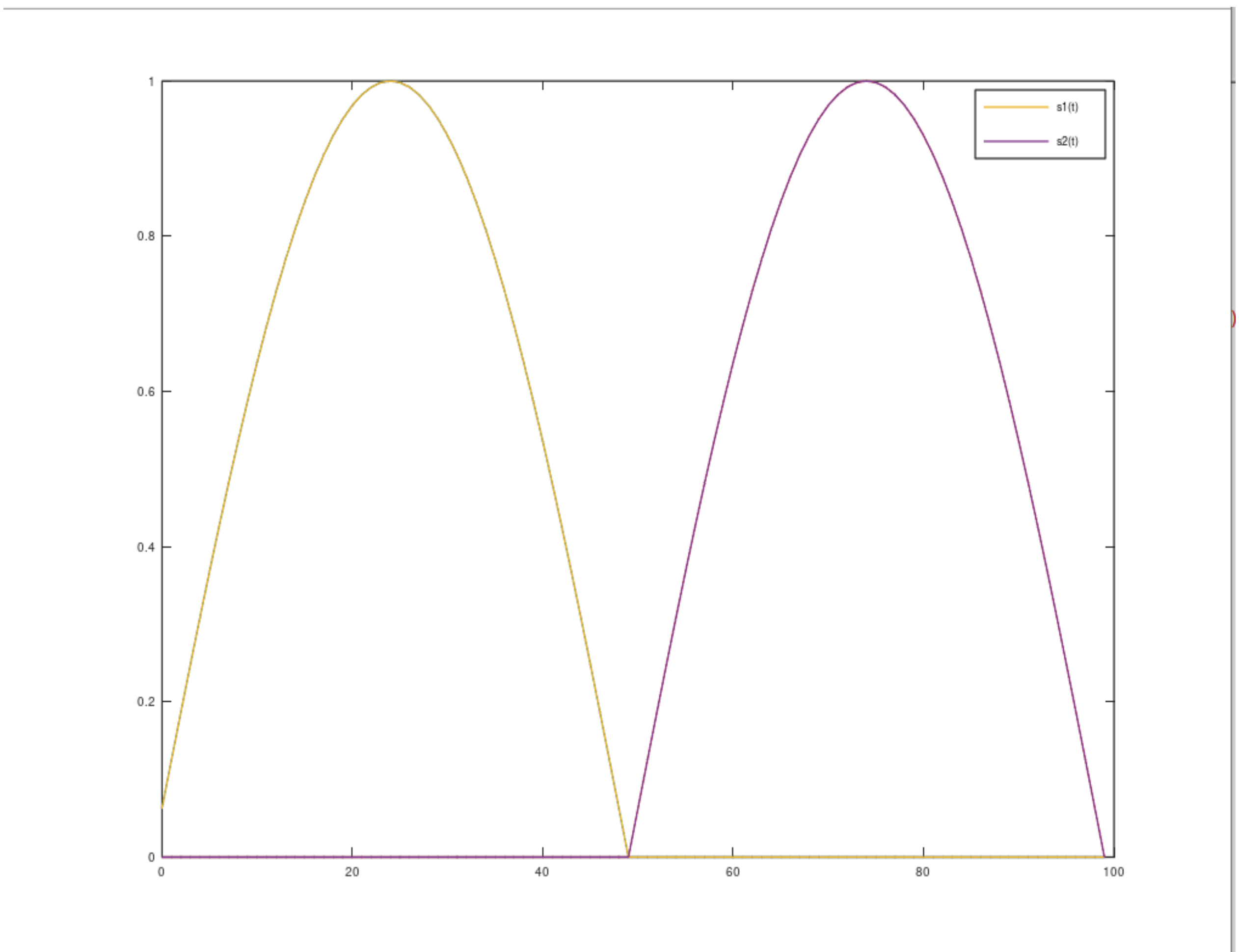
```
%This next section calculates the energy of the recieved signal
Es1=trapz(t,abs(st1.*st1));
Es2=trapz(t,abs(st2.*st2));

Eb=Es1*(1/3) + Es2*(2/3);

% Impulse response and the energy of the matching filter
ht= st1(T-t) - st2(T-t) ;

Eh=trapz(t,abs(ht.*ht));

% Now lets calculate Noise power , the desicion treshold and parameters necessary for probabilistic functions
N0=Eb.*10.^(-SNRdb/10);

sigmasquared = (N0/2)*Eh;

sigma = sigmasquared.^(1/2);


treshold = (sigmasquared/(a1-a2))*(log((2/3)/(1/3))) + (a1+a2)/2 ;

%Now lets find the BER theoratically.
%The matlab is giving me some problems with the communications librariy so i am calculating the Q function with the help of erfc


BERtheory3=zeros(0,length(SNRdb));



for i=1 : length(SNRdb)

  BERtheory3(i)= (1-((1/2)*erfc(((treshold(i)-a1)/(sigma(i)))/2.^1/2)))*(1/3)   +   ((1/2)*erfc(((treshold(i)-a2)/(sigma(i)))/2.^1/2)
endfor


A
%For the simulation let's create a test input

```

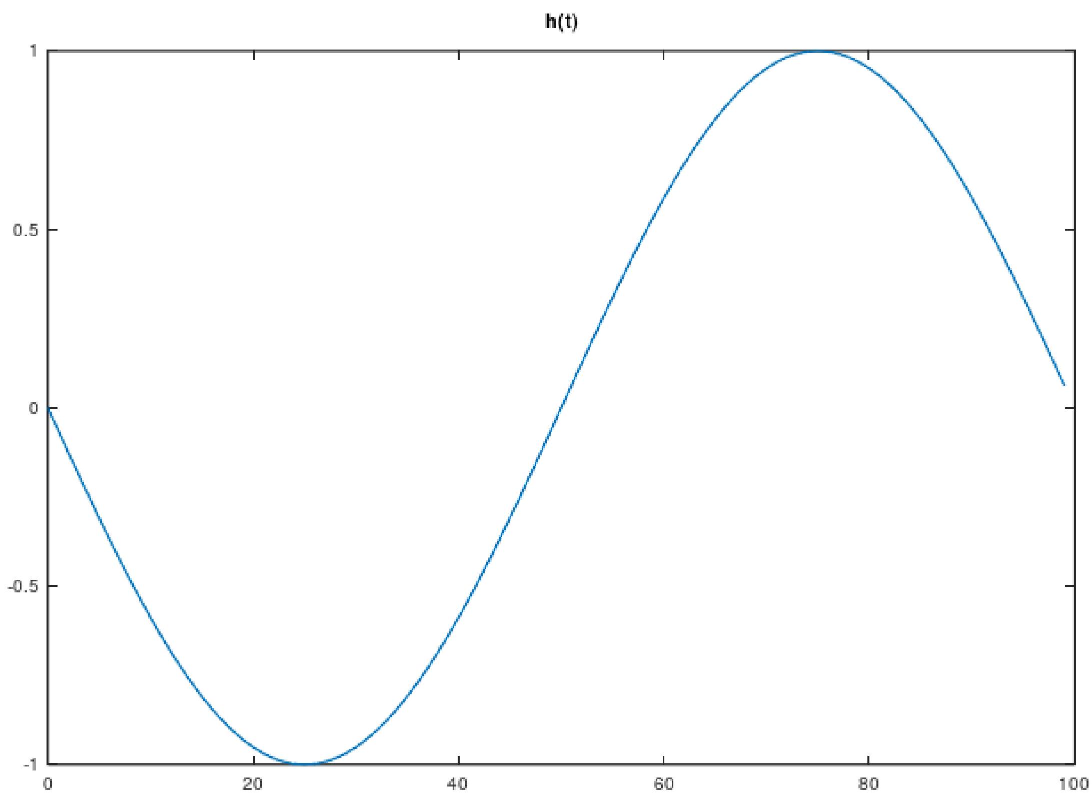
```
testinput = randi([0 1],10000000,1);

testoutput = zeros(0,length(testinput));

% This loop scales the test output for a1 and a2 components
for i = 1 : length(testinput)
  if testinput(i) == 1
  testoutput(i) = a1 ;
elseif testinput(i) == 0
  testoutput(i) = a2;
  endif

  endfor




%Now lets create AWGN
n0i = randn(1,10000000);
BERSim = zeros(0,length(SNRdb));

z0=zeros(0,length(testinput));
errorcount = 0;
%This loop combines proper AWGN and the test output, finds the output of the desicion block and calculates the simulated BER
for i = 1 : length(SNRdb)

errorcount = 0;

alpha =((2*N0(i)).^1/2);

n0= n0i .* alpha ;

noisytestoutput1= testoutput  + n0 ;

  for c = 1 : length(testoutput)
    if noisytestoutput1(c) > treshold
        z0(c) = 1;
    else
```

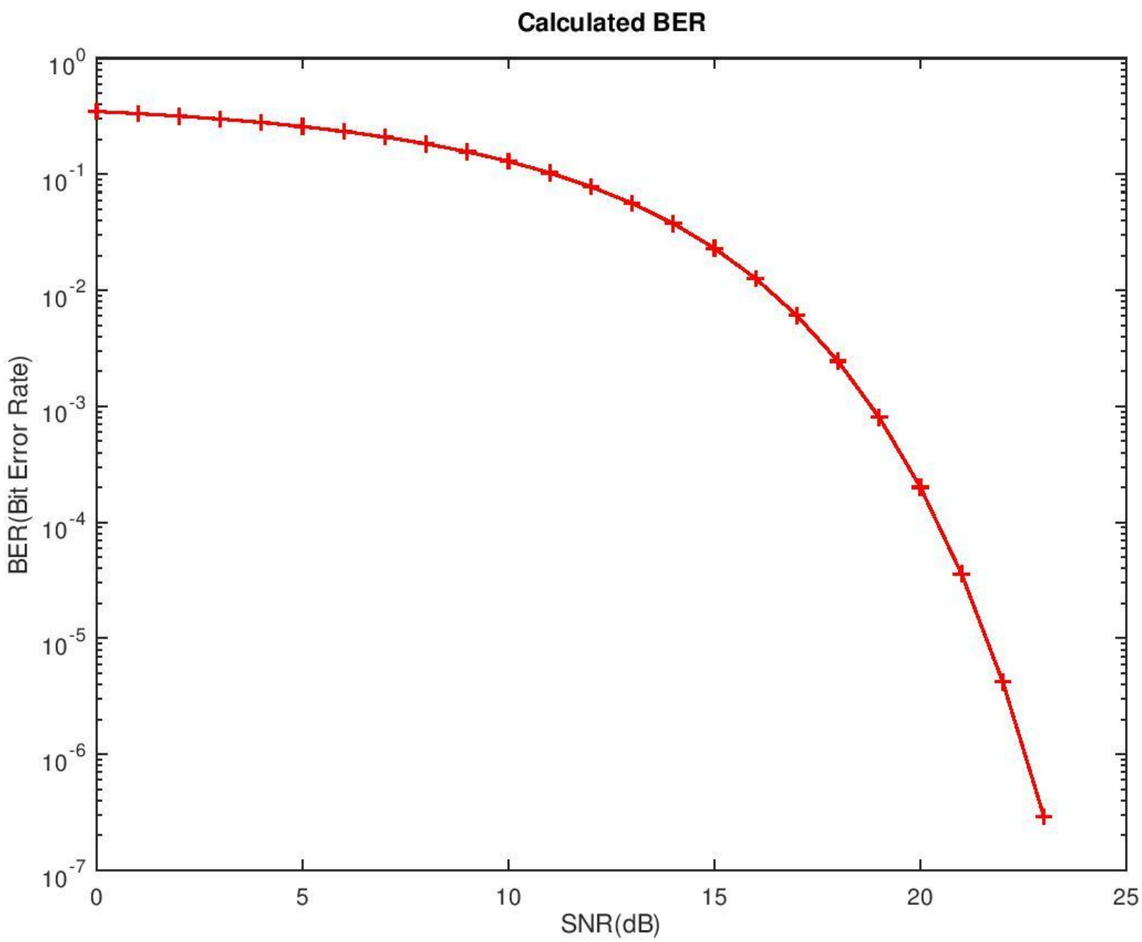
```
        z0(c) = 0;

  endif

  endfor



  for j = 1 : length(testinput)
    if z0(j) ~= testinput(j)
    errorcount = errorcount + 1;

      endif

endfor

BERSim(i) = errorcount/length(testinput) ;



endfor




%Now let's make the plots

figure(1)
st_1=plot(t,st1)
st_2=plot(t,st2)

legend([st_1 st_2],{'s1(t)' , 's2(t)'});

figure(2)
plot(t,ht)
title('h(t)')


```

```
figure(3)
semilogy(SNRdb,BERtheory3,'r+-','linewidth',1);
title('Calculated BER')
xlabel("SNR(dB)");
ylabel("BER(Bit Error Rate)");


figure(4)
semilogy(SNRdb,BERtheory3);
title('Simulated BER')
xlabel("SNR(dB)");
ylabel("BER(Bit Error Rate)");
```

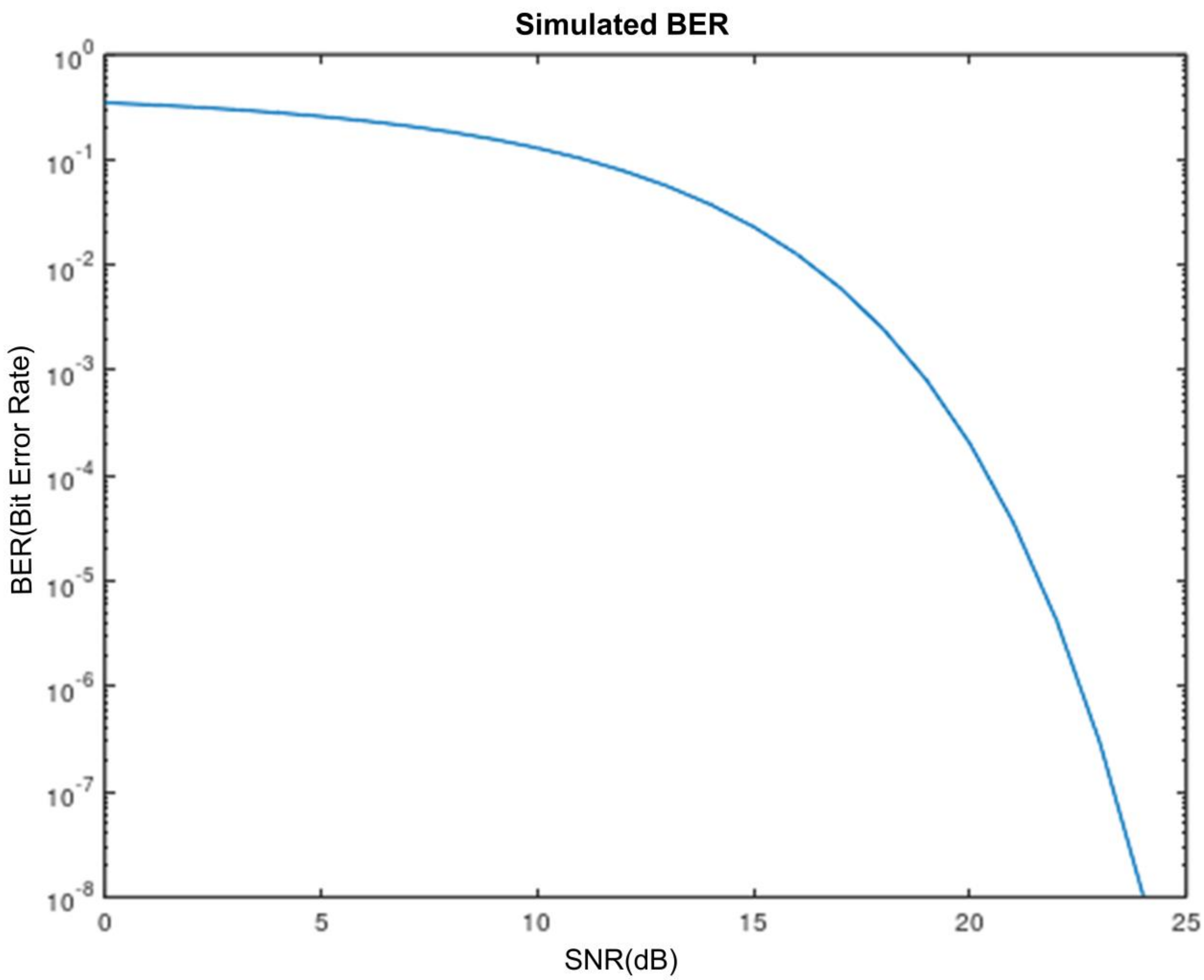
THE GRAPHS




h(t)
1
0.5
0
-0.5
-1
0
20
40
60
80
100



Calculated BER
BER(Bit Error Rate)
SNR(dB)
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
0
5
10
15
20
25


Simulated BER
BER(Bit Error Rate)
SNR(dB)
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
10^-8
0
5
10
15
20
25