

THE CODES AND EXPLANATIONS

% 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 library 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)");


```

1 % First let's assign some constants we will use
2 %amplitude has chosen to be 1 and since matlab is making it difficult to work with non-integer values for conditional statements let's make T=100
3
4
5 T = 100;
6
7 A = 1;
8
9 t=linspace(0,T-1,T);
10
11 % Now let's define the components of the recieved signal at the input of the filter
12 st1=zeros(0,T);
13 for i = 1 : T/2
14
15     st1(i) = A*sin((2*pi*i)/T);
16
17 endfor
18
19
20
21
22 st2=zeros(0,T);
23 for i = T/2 : T
24
25     st2(i) = A*sin((2*pi*(i-T/2))/T);
26     st1(i) = 0;
27
28 endfor
29
30
31
32 % Now let's calculate the a1 and a2 components of the z
33 a1 = trapz(t,st1.*st1);
34 a2 = trapz(t,-st2.*st2);
35
36 %This communication system tends to reach minimal BER (around  $10^{-9}$ ) around 23 dB
37 SNRdb = linspace(0,23,24);
38

```



```

39 %This next section calculates the energy of the recieved signal
40 Es1=trapz(t,abs(st1.*st1));
41 Es2=trapz(t,abs(st2.*st2));
42
43 Eb=Es1*(1/3) + Es2*(2/3);
44
45 % Impulse response and the energy of the matching filter
46 ht= st1(T-t) - st2(T-t) ;
47
48 Eh=trapz(t,abs(ht.*ht));
49
50 % Now lets calculate Noise power , the desicion treshold and parameters necessary for probabilistic functions
51 N0=Eb.*10.^(-SNRdb/10);
52
53 sigmasquared = (N0/2)*Eh;
54
55 sigma = sigmasquared.^(1/2);
56
57
58 treshold = (sigmasquared/(a1-a2))*(log((2/3)/(1/3))) + (a1+a2)/2 ;
59
60 %Now lets find the BER theoratically.
61 %The matlab is giving me some problems with the communications librariy so i am calculating the Q function with the help of erfc
62
63
64 BERtheory3=zeros(0,length(SNRdb));
65
66
67
68 for i=1 : length(SNRdb)
69
70     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))
71 endfor
72
73
74 A
75 %For the simulation let's create a test input
76

```

```

77 testinput = randi([0 1],10000000,1);
78
79 testoutput = zeros(0,length(testinput));
80
81 % This loop scales the test output for a1 and a2 components
82 for i = 1 : length(testinput)
83     if testinput(i) == 1
84         testoutput(i) = a1 ;
85     elseif testinput(i) == 0
86         testoutput(i) = a2;
87     endif
88
89     endfor
90
91
92
93
94 %Now lets create AWGN
95 n0i = randn(1,10000000);
96 BERSim = zeros(0,length(SNRdb));
97
98 z0=zeros(0,length(testinput));
99 errorcount = 0;
100 %This loop combines proper AWGN and the test output, finds the output of the desicion block and calculates the simulated BER
101 for i = 1 : length(SNRdb)
102
103     errorcount = 0;
104
105     alpha = ((2*N0(i)).^1/2);
106
107     n0= n0i .* alpha ;
108
109     noisyttestoutput1= testoutput + n0 ;
110
111     for c = 1 : length(testoutput)
112         if noisyttestoutput1(c) > treshhold
113             z0(c) = 1;
114         else

```

```

115         z0(c) = 0;
116
117     endif
118
119 endfor
120
121
122
123 for j = 1 : length(testinput)
124     if z0(j) ~= testinput(j)
125         errorcount = errorcount + 1;
126
127     endif
128
129 endfor
130
131 BERSim(i) = errorcount/length(testinput) ;
132
133
134
135 endfor
136
137
138
139
140 %Now let's make the plots
141
142 figure(1)
143 st_1=plot(t,st1)
144 st_2=plot(t,st2)
145
146 legend([st_1 st_2],{'s1(t)' , 's2(t)'});
147
148 figure(2)
149 plot(t,ht)
150 title('h(t)')
151
152

```

```
153 figure(3)
154 semilogy(SNRdb,BERtheory3,'r+-','linewidth',1);
155 title('Calculated BER')
156 xlabel("SNR(dB)");
157 ylabel("BER(Bit Error Rate)");
158
159
160 figure(4)
161 semilogy(SNRdb,BERtheory3);
162 title('Simulated BER')
163 xlabel("SNR(dB)");
164 ylabel("BER(Bit Error Rate)");
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
```

THE GRAPHS







