**ECE 254 End-term**

**Gaussian Signal in Gaussian Noise**

**Name: Mingxuan Wang**

**Date: 2015/12/5**

* Title: Gaussian Signal in Gaussian Noise
* Objective:

We are going to explore the trade-off between distributing the signal energy over different numbers of samples for detecting an uncorrelated Gaussian signal ~N(0, σs2) buried in uncorrelated Gaussian noise ~N(0, σ2).

Below are the details of our goals:

1. Get the functional form of the test statistic T(x).
2. Get the functional form of PD and PF in terms of the threshold, SNR = σs2/σ2, and number of samples N.
3. Compute and plot the performance of the detection receiver for N = 2, 4, 8, 16, 32, and 64:
   1. PD vs. PF on normal probability paper for 10 log (ENR) = 10 dB and 15 dB.
   2. PD (linear) vs. ENR (dB) for PF = 10-1, 10-2, and 10-3 and ENR from 0 to 20 dB.
4. Plot PD vs. N (linear axes) for fixed ENR and PF with N = {2, 4, 6, 8, 10,…, 64}, ENR = {10,15} dB, and PF = {0.001, 0.01, 0.1}.

* Approach:

See hand-writing part. (Page 1 - 3)

* Results(including plots):

Plots:



**Figure 1 PF vs PF ROC curve with 10 log (ENR) = 10 dB**



**Figure 2 PF vs PF ROC curve with 10 log (ENR) = 15 dB**



**Figure 3 PD vs ENR with PF = 0.1**



**Figure 4 PD vs ENR with PF = 0.01**



**Figure 5 PD vs ENR with PF = 0.001**



**Figure 6 PD vs N with PF = 0.1, 0.01 0.001 and ENR = 10dB, 15dB**

**Table 1 Optimum N for each case in Plot 6**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ENR  PF | 15,  0.1 | 15,  0.01 | 15， 0.001 | 10，  0.1 | 10，  0.01 | 10，  0.001 |
| Optimum N | 12 | 8 | 6 | 4 | 2 | 2 |
| Corresponding PD | 0.954462 | 0.852009 | 0.733095 | 0.694876 | 0.464159 | 0.316228 |

Discussion:

See hand-writing part. (Page 4 - )

* Appendix:

Main.m:

clear;clc;

%% PD vs PF

x1=10.^(10/10);

x2=10.^(15/10);

N=[2,4,8,16,32,64];

PFA=0.01:0.01:1;

PD1=zeros(1,100);

PD2=zeros(1,100);

PD3=zeros(1,100);

PD4=zeros(1,100);

PD5=zeros(1,100);

PD6=zeros(1,100);

%N=2,ENR=10dB

for i=1:100

r1=getthres(PFA(i),N(1));

PD1(i)=Qchipr2(N(1),0,2\*r1/(x1/N(1)+1),1e-5);

end

figure(1);probpaper(PFA,PD1, 'r');

% N=4,ENR=10dB

for i=1:100

r2=getthres(PFA(i),N(2));

PD2(i)=Qchipr2(N(2),0,2\*r2/(x1/N(2)+1),1e-5);

end

figure(1);hold on;probpaper(PFA,PD2, 'g') ;

% N=8,ENR=10dB

for i=1:100

r3=getthres(PFA(i),N(3));

PD3(i)=Qchipr2(N(3),0,2\*r3/(x1/N(3)+1),1e-5);

end

figure(1);hold on;probpaper(PFA,PD3, 'b');

%N=16,ENR=10dB

for i=1:100

r4=getthres(PFA(i),N(4));

PD4(i)=Qchipr2(N(4),0,2\*r4/(x1/N(4)+1),1e-5);

end

figure(1);probpaper(PFA,PD4, 'k');

%N=32,ENR=10dB

for i=1:100

r5=getthres(PFA(i),N(5));

PD5(i)=Qchipr2(N(5),0,2\*r5/(x1/N(5)+1),1e-5);

end

figure(1);probpaper(PFA,PD5, 'c') ;

%N=64,ENR=10dB

for i=1:100

r6=getthres(PFA(i),N(6));

PD6(i)=Qchipr2(N(6),0,2\*r6/(x1/N(6)+1),1e-5);

end

figure(1);probpaper(PFA,PD6, 'm');grid;

legend('N=2','N=4','N=8','N=16','N=32','N=64');hold off;title('PF VS PD ROC paper ENR=10dB')

%N=2,ENR=15dB

for i=1:100

r1=getthres(PFA(i),N(1));

PD1(i)=Qchipr2(N(1),0,2\*r1/(x2/N(1)+1),1e-5);

end

figure(2);probpaper(PFA,PD1, 'r');

% N=4,ENR=15dB

for i=1:100

r2=getthres(PFA(i),N(2));

PD2(i)=Qchipr2(N(2),0,2\*r2/(x2/N(2)+1),1e-5);

end

figure(2);hold on;probpaper(PFA,PD2, 'g') ;

% N=8,ENR=15dB

for i=1:100

r3=getthres(PFA(i),N(3));

PD3(i)=Qchipr2(N(3),0,2\*r3/(x2/N(3)+1),1e-5);

end

figure(2);hold on;probpaper(PFA,PD3, 'b');

%N=16,ENR=15dB

for i=1:100

r4=getthres(PFA(i),N(4));

PD4(i)=Qchipr2(N(4),0,2\*r4/(x2/N(4)+1),1e-5);

end

figure(2);probpaper(PFA,PD4, 'k');

%N=32,ENR=15dB

for i=1:100

r5=getthres(PFA(i),N(5));

PD5(i)=Qchipr2(N(5),0,2\*r5/(x2/N(5)+1),1e-5);

end

figure(2);probpaper(PFA,PD5, 'c') ;

%N=64,ENR=15dB

for i=1:100

r6=getthres(PFA(i),N(6));

PD6(i)=Qchipr2(N(6),0,2\*r6/(x2/N(6)+1),1e-5);

end

figure(2);probpaper(PFA,PD6, 'm');grid;

legend('N=2','N=4','N=8','N=16','N=32','N=64');hold off;title('PF VS PD ROC paper ENR=15dB');

%% PD vs ENR

PFA =[10^-1;10^-2;10^-3];

N=[2,4,8,16,32,64];

ENR=0:0.5:20;

x=10.^(ENR/10);

%PFA1=10^-1

PD1=zeros(1,41);

r1 = getthres(PFA(1), N(1));

for i = 1: 41

PD1(i) = Qchipr2(N(1),0,2\*r1/(x(i)/N(1)+1),1e-5);

end

PD2=zeros(1,41);

r2 = getthres(PFA(1), N(2));

for i = 1: 41

PD2(i) = Qchipr2(N(2),0,2\*r2/(x(i)/N(2)+1),1e-5);

end

PD3=zeros(1,41);

r3 = getthres(PFA(1), N(3));

for i = 1: 41

PD3(i) = Qchipr2(N(3),0,2\*r3/(x(i)/N(3)+1),1e-5);

end

PD4=zeros(1,41);

r4 = getthres(PFA(1), N(4));

for i = 1: 41

PD4(i) = Qchipr2(N(4),0,2\*r4/(x(i)/N(4)+1),1e-5);

end

PD5=zeros(1,41);

r5 = getthres(PFA(1), N(5));

for i = 1: 41

PD5(i) = Qchipr2(N(5),0,2\*r5/(x(i)/N(5)+1),1e-5);

end

PD6=zeros(1,41);

r6 = getthres(PFA(1), N(6));

for i = 1: 41

PD6(i) = Qchipr2(N(6),0,2\*r6/(x(i)/N(6)+1),1e-5);

end

figure(3)

plot(ENR,PD1,'r')

hold on

plot(ENR,PD2,'g')

plot(ENR,PD3,'b')

plot(ENR,PD4,'k')

plot(ENR,PD5,'c')

plot(ENR,PD6,'m')

grid;

legend('N=2','N=4','N=8','N=16','N=32','N=64');title('PD vs ENR PF=10^-1');xlabel('ENR in dB');ylabel('PD');

%PFA1=10^-2

PD1=zeros(1,41);

r1 = getthres(PFA(2), N(1));

for i = 1: 41

PD1(i) = Qchipr2(N(1),0,2\*r1/(x(i)/N(1)+1),1e-5);

end

PD2=zeros(1,41);

r2 = getthres(PFA(2), N(2));

for i = 1: 41

PD2(i) = Qchipr2(N(2),0,2\*r2/(x(i)/N(2)+1),1e-5);

end

PD3=zeros(1,41);

r3 = getthres(PFA(2), N(3));

for i = 1: 41

PD3(i) = Qchipr2(N(3),0,2\*r3/(x(i)/N(3)+1),1e-5);

end

PD4=zeros(1,41);

r4 = getthres(PFA(2), N(4));

for i = 1: 41

PD4(i) = Qchipr2(N(4),0,2\*r4/(x(i)/N(4)+1),1e-5);

end

PD5=zeros(1,41);

r5 = getthres(PFA(2), N(5));

for i = 1: 41

PD5(i) = Qchipr2(N(5),0,2\*r5/(x(i)/N(5)+1),1e-5);

end

PD6=zeros(1,41);

r6 = getthres(PFA(2), N(6));

for i = 1: 41

PD6(i) = Qchipr2(N(6),0,2\*r6/(x(i)/N(6)+1),1e-5);

end

figure(4)

plot(ENR,PD1,'r')

hold on

plot(ENR,PD2,'g')

plot(ENR,PD3,'b')

plot(ENR,PD4,'k')

plot(ENR,PD5,'c')

plot(ENR,PD6,'m')

grid;

legend('N=2','N=4','N=8','N=16','N=32','N=64');title('PD vs ENR PF=10^-2');xlabel('ENR in dB');ylabel('PD');

%PFA1=10^-3

PD1=zeros(1,41);

r1 = getthres(PFA(3), N(1));

for i = 1: 41

PD1(i) = Qchipr2(N(1),0,2\*r1/(x(i)/N(1)+1),1e-5);

end

PD2=zeros(1,41);

r2 = getthres(PFA(3), N(2));

for i = 1: 41

PD2(i) = Qchipr2(N(2),0,2\*r2/(x(i)/N(2)+1),1e-5);

end

PD3=zeros(1,41);

r3 = getthres(PFA(3), N(3));

for i = 1: 41

PD3(i) = Qchipr2(N(3),0,2\*r3/(x(i)/N(3)+1),1e-5);

end

PD4=zeros(1,41);

r4 = getthres(PFA(3), N(4));

for i = 1: 41

PD4(i) = Qchipr2(N(4),0,2\*r4/(x(i)/N(4)+1),1e-5);

end

PD5=zeros(1,41);

r5 = getthres(PFA(3), N(5));

for i = 1: 41

PD5(i) = Qchipr2(N(5),0,2\*r5/(x(i)/N(5)+1),1e-5);

end

PD6=zeros(1,41);

r6 = getthres(PFA(3), N(6));

for i = 1: 41

PD6(i) = Qchipr2(N(6),0,2\*r6/(x(i)/N(6)+1),1e-5);

end

figure(5)

plot(ENR,PD1,'r')

hold on

plot(ENR,PD2,'g')

plot(ENR,PD3,'b')

plot(ENR,PD4,'k')

plot(ENR,PD5,'c')

plot(ENR,PD6,'m')

grid;

legend('N=2','N=4','N=8','N=16','N=32','N=64');title('PD vs ENR PF=10^-3');xlabel('ENR in dB');ylabel('PD');

%% PD vs N

N=2:2:64;

ENR=[10,15];

x=10.^(ENR/10);

PFA=[0.001,0.01,0.1];

%ENR = 10dB, PFA = 0.001

PD1=zeros(1,32);

for i=1:32

r1=getthres(PFA(1),N(i));

PD1(i)=Qchipr2(N(i),0,2\*r1/(x(1)/N(i)+1),1e-5);

end

%ENR = 10dB, PFA = 0.01

PD2=zeros(1,32);

for i=1:32

r2=getthres(PFA(2),N(i));

PD2(i)=Qchipr2(N(i),0,2\*r2/(x(1)/N(i)+1),1e-5);

end

%ENR = 10dB, PFA = 0.1

PD3=zeros(1,32);

for i=1:32

r3=getthres(PFA(3),N(i));

PD3(i)=Qchipr2(N(i),0,2\*r3/(x(1)/N(i)+1),1e-5);

end

%ENR = 15dB, PFA = 0.001

PD4=zeros(1,32);

for i=1:32

r4=getthres(PFA(1),N(i));

PD4(i)=Qchipr2(N(i),0,2\*r4/(x(2)/N(i)+1),1e-5);

end

%ENR = 15dB, PFA = 0.01

PD5=zeros(1,32);

for i=1:32

r5=getthres(PFA(2),N(i));

PD5(i)=Qchipr2(N(i),0,2\*r5/(x(2)/N(i)+1),1e-5);

end

%ENR = 15dB, PFA = 0.1

PD6=zeros(1,32);

for i=1:32

r6=getthres(PFA(3),N(i));

PD6(i)=Qchipr2(N(i),0,2\*r6/(x(2)/N(i)+1),1e-5);

end

figure(6)

plot(N,PD1,'r')

hold on

plot(N,PD2,'g')

plot(N,PD3,'b')

plot(N,PD4,'k')

plot(N,PD5,'c')

plot(N,PD6,'m')

axis([2,64,0.01,1])

xlabel('N(even numbers)');ylabel('PD');grid on;hold off

legend('PF=10^-3, ENR = 10dB','PF=10^-2, ENR = 10dB','PF=10^-1, ENR = 10dB', 'PF=10^-3, ENR = 15dB', 'PF=10^-2, ENR = 15dB', 'PF=10^-1, ENR = 15dB');title('PD vs N PF=10^-1 10^-2 10^-3');

getthres.m:

function ret=getthres(PFA,N)

gamma1=1;

gamma2=0;

epis = 1e-5;

while(abs(gamma1-gamma2)>epis)

gamma2=gamma1;

sigma=0;

for r=1:(N/2)-1

sigma=sigma+(gamma2.^r)/factorial(r);

end

gamma1=-log(PFA)+log(1+sigma);

end

ret=gamma1;

end