Gaussian Signal in Gaussian Noise

Name: Yizhi Lu

Class: ECE254

Date: 12/16/2014

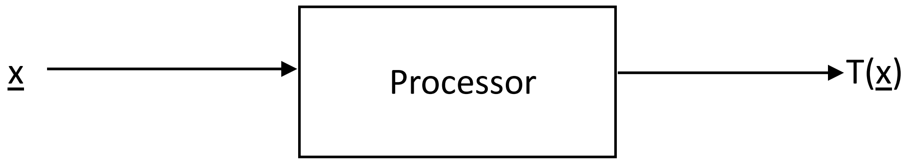
PID:A53075712

Title :

Gaussian Signal in Gaussian noise

Objective (purpose) :

To decide presence or absence of uncorrelated Gaussian signal ~N(0, $\sigma_s{}^2$) buried in uncorrelated Gaussian noise where:
H1:x(n)=s(n)+w(n),n=0,1,…,N-1
H0:x(n)=w(n),n=0,1,…,N-1
w(n) is an uncorrelated, Gaussian noise sequence ~N(0,σ^2).
The processor structure is below:



x(n) equals that s(n) adds noise w(n). And we use matlab to realize all of these tasks and plot what we get.
At first we need to get the functional form of the test statistic T(x) of Gaussian signal in Gaussian Noise problem. Plot PD vs PF with different N and in both ENR=10dB and ENR =15dB on normal probability paper to determine the performance of Energy detector with different N. In this project we also need to plot PD vs N to explore the trade-offs between distributing the signal energy over different numbers of samples for detecting an uncorrelated Gaussian signal ~N (0, $\sigma_s{}^2$) buried in uncorrelated Gaussian noise ~N(0,σ^2).

Background (Introduction):

In the previous project or homework we were able to detect signals in the presence of noise by detecting the change in the mean of a test statistic. This was because the signal was assumed deterministic, and hence its presence altered the mean of the received data. In some case a signal is more appropriately modeled as a random process. An example is speech, for which the waveform of a given sound depends on the identity of the speaker, the context in which the sound is spoken, the health of the speaker, etc. It is therefore, unrealistic to assume that the signal is known. A better approach is to assume that the signal is a random process with a known covariance structure. In this project we assume that signal is an uncorrelated Gaussian signal.(From Kay text Chapter 5)

Approach(Procedure):

1. Express the functional form of test statistic T(x)

Solutions:

s(n)=w_s(n) Uncorrelated Gaussian signal ~ N(0,${\sigma_s}^2$):
Energy detector:
H0: x(n)=w(n) n=0,1,…,N-1
H1: x(n)=s(n)+w(n)
Where: w(n)~ N(0,$\sigma^2 I$)
s(n)~ N(0,${\sigma_s}^2 I$)
thus x(n)~ N(0,$\sigma^2 I$) under H0
x(n)~ N(0,$({\sigma_s}^2+\sigma^2)I$) under H1
A NP detector decides H1 if the likelihood ratio exceeds a threshold or if

$$\Lambda(\mathrm{x})=\frac{p(x;H1)}{p(x;H0)}>\gamma$$

But from our modeling assumption we can get x(n)~ N(0,$\sigma^2 I$) under H0 and thus,x(n)~ x(n)~ N(0,$({\sigma_s}^2+\sigma^2)I$) under H1.As a result, we obtain

$$\Lambda(\mathrm{x})=\frac{p(x|H1)}{p(x|H0)}=\frac{1/(2\pi({\sigma_s}^2+\sigma^2))^{N/2}\exp\{-\frac{1}{2({\sigma_s}^2+\sigma^2)}\sum_{n=0}^{N-1}x^2(n)\}}{1/(2\pi\sigma^2)^{\frac{N}{2}}\exp\{-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}x^2(n)\}}$$

$\ln\Lambda(\mathrm{x})=\frac{N}{2}\ln\left(\frac{\sigma^2}{{\sigma_s}^2+\sigma^2}\right)+\frac{1}{2}\frac{{\sigma_s}^2}{\sigma^2({\sigma_s}^2+\sigma^2)}\sum_{n=0}^{N-1}x^2(n)$
Test statistic : T(x)= $\sum_{n=0}^{N-1}x^2(n)$
The relation between test statistic T(x) and its threshold γ' :
T(x)= $\sum_{n=0}^{N-1}x^2(n)>\gamma'$
The NP detector computes the energy in the received data and computes it to a threshold. Hence, it is known as an energy detector. We can learn that if the signal is present, the energy of the received data increases.

Alternatively T(x)'=$\frac{1}{N}\sum_{n=0}^{N-1}x^2(n)$ is an estimate of the variance and the threshold recognizes that the variance under H0: σ^2 and under H1: ${\sigma_s}^2+\sigma^2$

2. Express the functional form of PD and PF in terms of the threshold, SNR =$\frac{{\sigma_s}^2}{\sigma^2}$, and number of samples N. Indicate how you determine the threshold for a given value of PF when generating plots of PD vs. ENR (e.g. show the iterative formula in Prob. 5.1 and comment on how it is used).

Solutions:
We know that test statistic T(x) are χ_N^2under H0 and H1.
The detection performance can be found according to Chapter2 in Kay text.

$\frac{T(x)}{\sigma^2}\sim\chi_N^2$ under H0;

$\frac{T(x)}{{\sigma_s}^2+\sigma^2}\sim\chi_N^2$ under H1.

PFA=Pr{ T(x)>$\gamma'|H0\}=\Pr\{\frac{\mathrm{T(x)}}{\sigma^2}>\frac{\gamma'}{\sigma^2}|H0\}=Q_{\chi_N^2}(\frac{\gamma'}{\sigma^2})$

$P_D = \Pr\{T(x) > \gamma' | H1\} = \Pr\{\frac{T(x)}{\sigma_s^2+\sigma^2} > \frac{\gamma'}{\sigma_s^2+\sigma^2} | H0\} = Q_{\chi_N^2}(\frac{\gamma'}{\sigma_s^2+\sigma^2})$

SNR=$\frac{\sigma_s^2}{\sigma^2}$ letting $\gamma'' = \frac{\gamma'}{\sigma^2}$

$P_{FA} = Q_{\chi_N^2}(\gamma'')$

$P_D = Q_{\chi_N^2}(\frac{\gamma''}{\frac{\sigma_s^2}{\sigma^2}+1}) = Q_{\chi_N^2}(\frac{\gamma''}{SNR+1})$

Above equations are theoretical and we need some operations to get inverse Q function of Chi-Square distribution for P_{FA}. We can refer to prob.5.1 in Kay text (pp.176-177) for an iterative formula to get an inverse Q function for P_{FA}.
According to the analytical derivation above we easily obtain equation about PFA:

PFA=exp($-\frac{\gamma'}{2\sigma^2}$)$[1 + \sum_{r=1}^{\frac{N-1}{2}} \frac{(\frac{\gamma'}{2\sigma^2})^r}{r!}]$

Let $\gamma'' = \gamma'/2\sigma^2$ and rearranging terms we have

$\gamma'' = -ln\text{PFA} + \ln[\sum_{r=1}^{\frac{N-1}{2}} \frac{(\gamma'')^r}{r!}]$.

This time we only need to solve an equation about γ'' and to solve for γ'' we can use the fixed point iteration

$\gamma''_{k+1} = -ln\text{PFA} + \ln[\sum_{r=1}^{\frac{N-1}{2}} \frac{(\gamma''_k)^r}{r!}]$. (the threshold γ' by iteration with $\gamma''_0 = 1$.)

We need to write a loop for this equation of γ'' and we bring the previous or former γ'' which we write as γ''_k to the equation '$-ln\text{PFA} + \ln[\sum_{r=1}^{\frac{N-1}{2}} \frac{(\gamma''_k)^r}{r!}]$' to update the value of γ'' which we note as γ''_{k+1}. γ''_{k+1} is just the updated value for previous one γ''_k. To get a quite accurate value of γ'' we let $|\gamma''_{k+1} - \gamma''_k| < 10^{-9}$ and this is a threshold for loop. If $|\gamma''_{k+1} - \gamma''_k| < 10^{-9}$, the loop stops and return current value of γ'' i.e. γ''_{k+1}.The specific matlab codes and functions are attached as appendix at the end of this report. (Note: The iterative formula in Problem 5.1 is valid only for N even.)

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. Plot all cases of N for a given ENR on one plot.
(2). PD (linear) vs. ENR (dB) for PF = 10^{-1}, 10^{-2}, and 10^{-3} and ENR from 0 to 20 dB. Plot all cases of N for a given PF on one plot.

Note: ENR = (N)*(SNR) = (N)*($\frac{\sigma_s^2}{\sigma^2}$) is the expected energy-to-noise ratio.

Solutions:

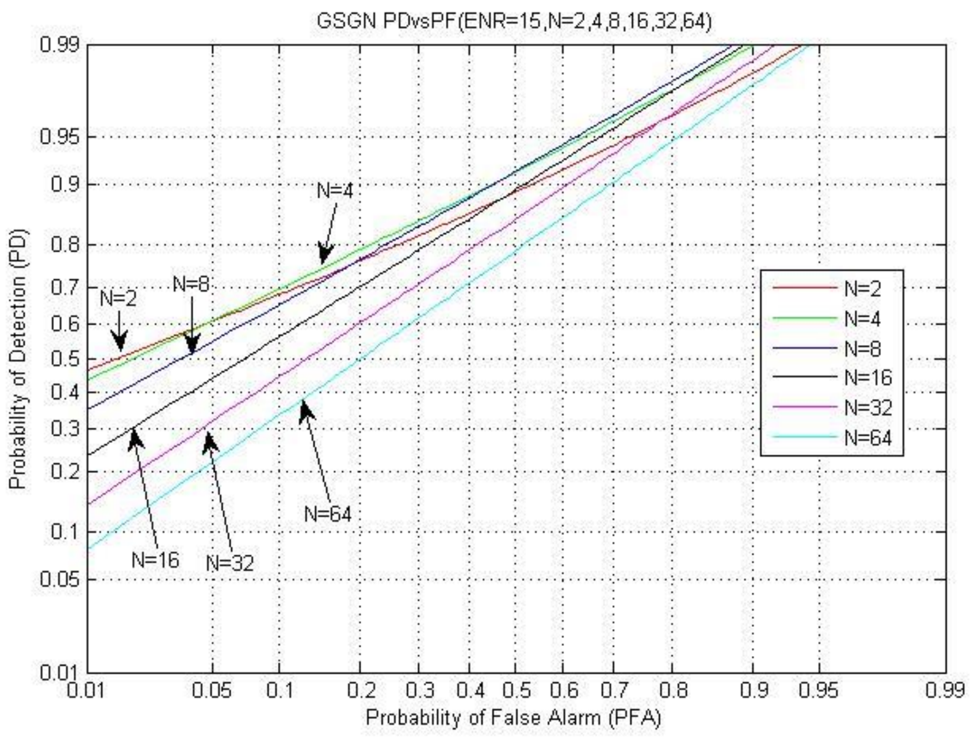
In order to obtain the plot of PD vs PF on normal probability we need to make ENR fixed and ENR1=10^(10/10)=10,ENR2=10^(15/10)$\approx$ 32. P_D=$Q_{\chi_N^2}(\frac{\gamma''}{\frac{\sigma_s{}^2}{\sigma^2}+1})$= $Q_{\chi_N^2}(\frac{\gamma''}{SNR+1})$ and SNR=$\frac{\sigma_s{}^2}{\sigma^2} = ENR/N$,for different N we will get different curves of PD vs PF shown in figure.1.In figure.1 different color represents and illustrates different value of N. Similar to the previous solution, in problem 3.(2) we only make the PF fixed and get the relationship between PD and ENR. Like in figure.1, in figure.2 different color illustrates different value of N. The results are shown from figure.3 to figure.5.

4. Explore the structure of P_D vs. N (linear axes) for fixed ENR and P_F. Investigate for N = {2, 4, 6,8, 10, …, 64}, ENR = {10,15} dB, and P_F = {0.001, 0.01, 0.1}. Plot all cases of P_F for a given ENR on one plot. For a given ENR and P_F, is there an optimum N? Provide a table showing the value of N for maximum P_D for each (ENR, P_F) case.

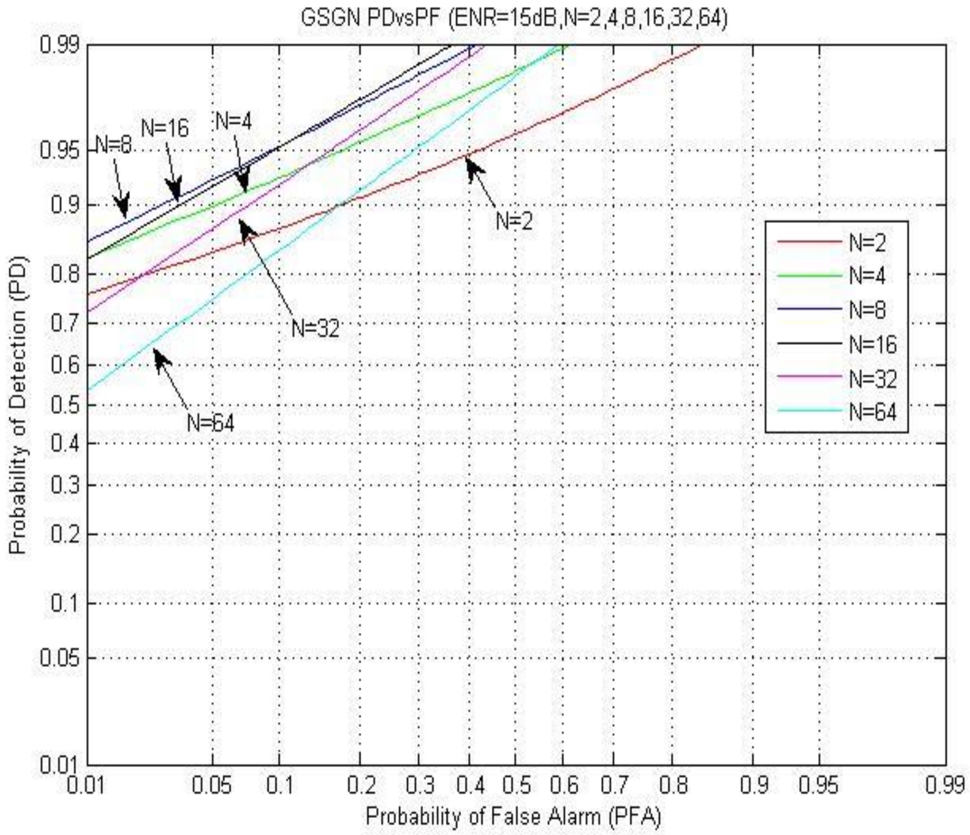
To get the structure of PD vs N we need to set both ENR and PF fixed according to the formula P_{FA}=$Q_{\chi_N^2}(\gamma'')$ and P_D=$Q_{\chi_N^2}(\frac{\gamma''}{\frac{\sigma_s{}^2}{\sigma^2}+1})$= $Q_{\chi_N^2}(\frac{\gamma''}{SNR+1})$ using the equation of iteration above in problem.2. There are 6 kinds of combinations between ENR and PF:

{10dB,0.001},{10dB,0.01},{10dB,0.1},{15dB,0.001},{15dB,0.01} and {15dB,0.1}.

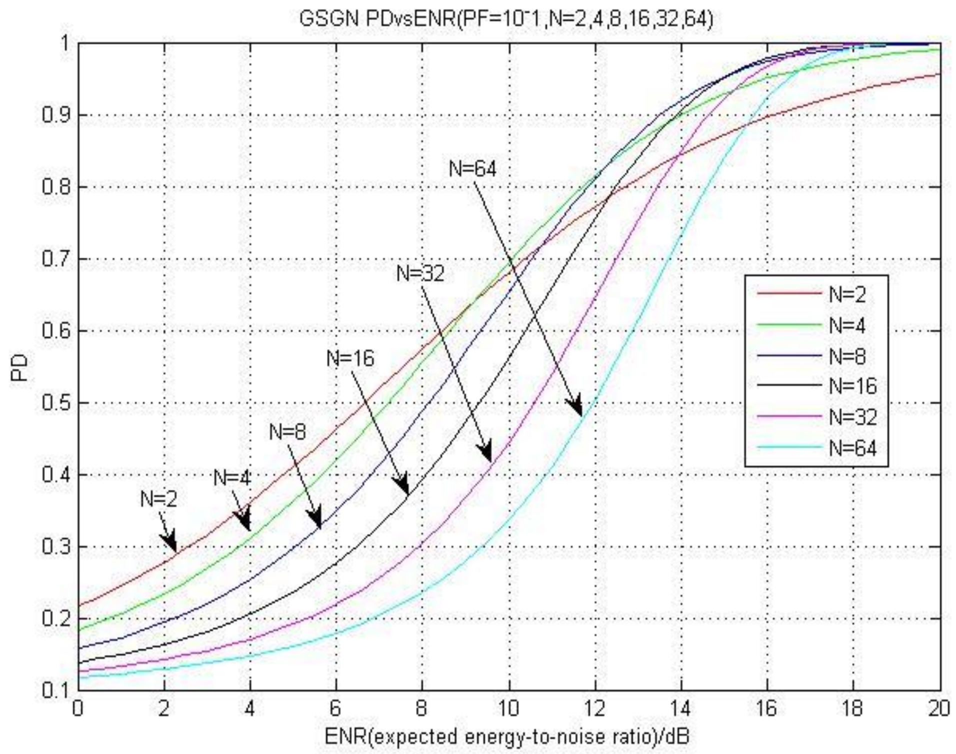
In this case, I put all combinations of ENR and N into one plot shown in figure.6 and the table showing the value of N for maximum PD for each case is shown in table.1 below.



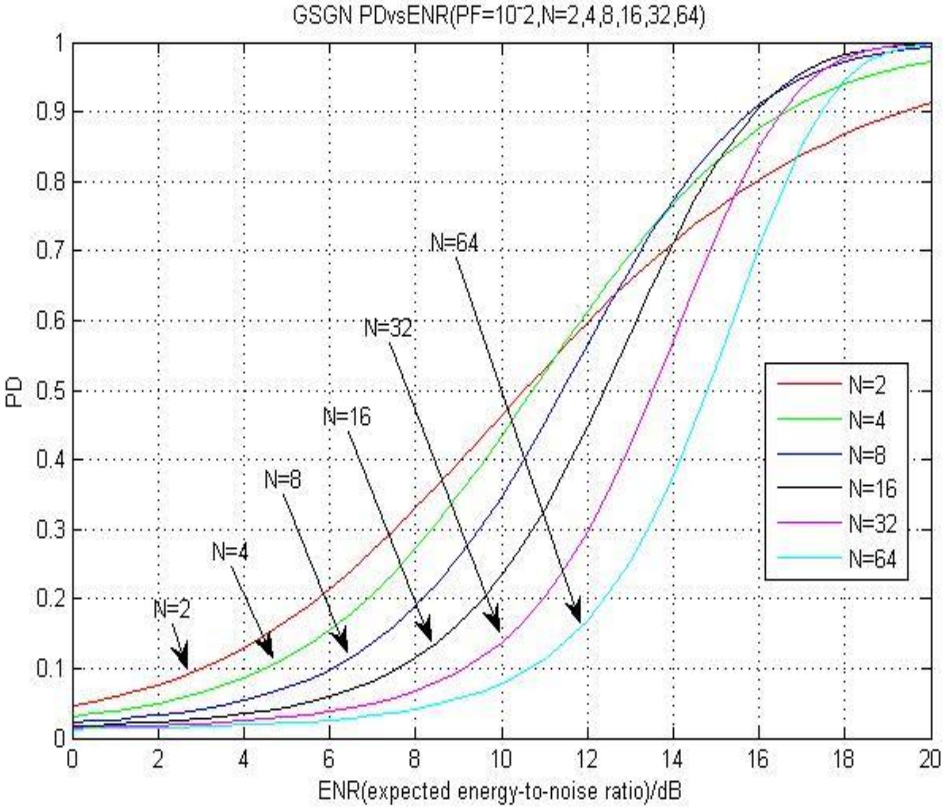
figre.1.GSGN PD vs PF with different value of N (ENR=10dB)



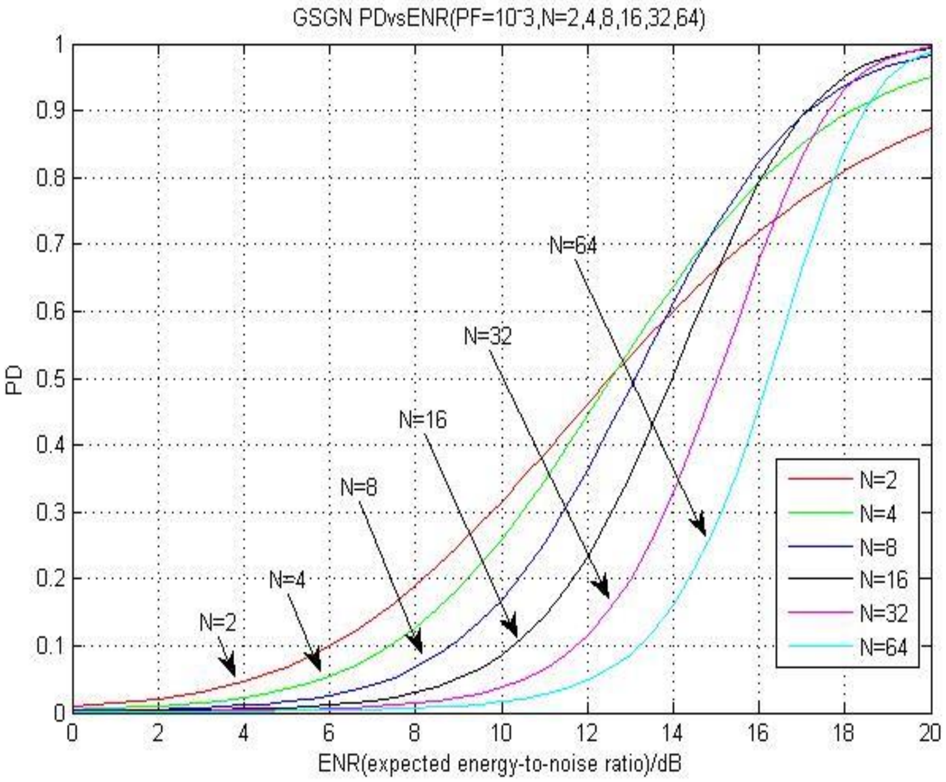
figre.2.GSGN PD vs PF with different value of N (ENR=15dB)



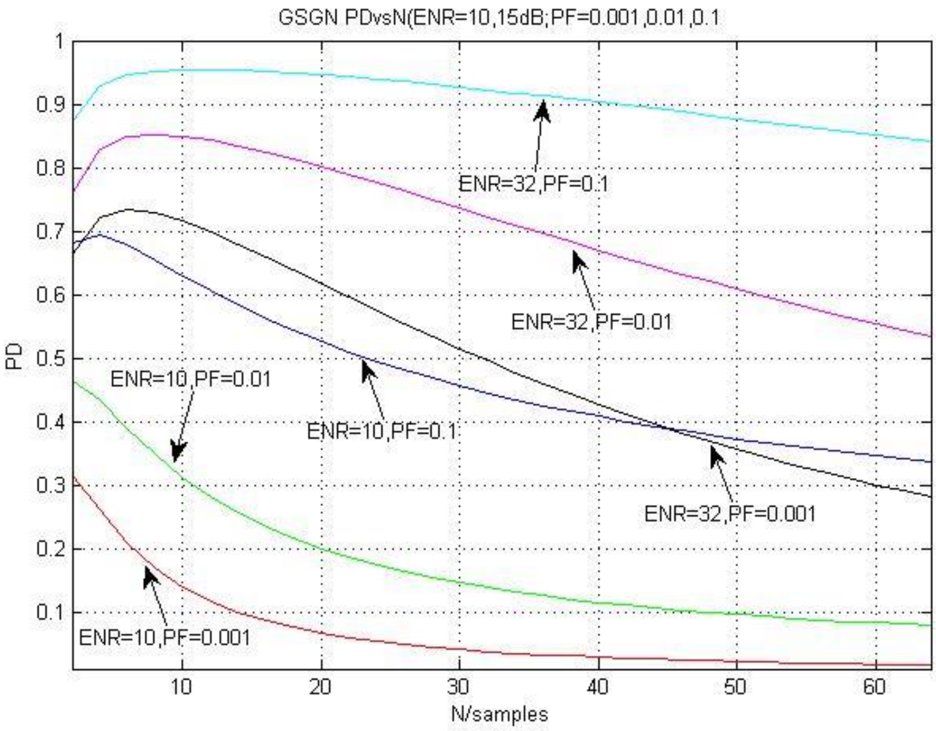
figre.3.GSGN PD vs ENR with different value of N (PF=10^{-1})



figre.4.GSGN PD vs ENR with different value of N (PF=10^{-2})



figre.5.GSGN PD vs ENR with different value of N (PF=10^{-3})



figre.6.GSGN PD vs N with different kind of combinations of ENR and PF

Case (ENR,PF)	(10,0.001)	(10,0.01)	(10,0.1)	(32,0.001)	(32,0.01)	(32,0.1)
Samples N	2	2	4	6	8	12
PD	0.316228	0.464159	0.694876	0.733095	0.852009	0.954462

table.1.table with value of N showing maximum PD for each case

case(ENR=10,PF=0.001):

N	2	4	6	8	10	12	14	16
PD	0.316228	0.26011	0.20881	0.169428	0.139749	0.117198	0.099804	0.086167
N	18	20	22	24	26	28	30	32
PD	0.075305	0.066524	0.059329	0.053361	0.048354	0.044111	0.040483	0.037354
N	34	36	38	40	42	44	46	48
PD	0.034635	0.032255	0.03016	0.028304	0.026651	0.025171	0.023841	0.022639
N	50	52	54	56	58	60	62	64
PD	0.02155	0.020559	0.019653	0.018824	0.018062	0.017359	0.01671	0.016109

case(ENR=10,PF=0.01):

N	2	4	6	8	10	12	14	16
PD	0.464159	0.434696	0.389963	0.348322	0.312388	0.281953	0.256228	0.234402
N	18	20	22	24	26	28	30	32
PD	0.215768	0.199747	0.185875	0.173779	0.163162	0.153785	0.145455	0.138016
N	34	36	38	40	42	44	46	48
PD	0.131338	0.125316	0.119862	0.114902	0.110375	0.106227	0.102416	0.098902
N	50	52	54	56	58	60	62	64
PD	0.095654	0.092643	0.089845	0.087239	0.084805	0.082528	0.080394	0.078389

case(ENR=10,PF=0.1):

N	2	4	6	8	10	12	14	16
PD	0.681292	0.694876	0.677794	0.654124	0.629464	0.605625	0.583233	0.562462
N	18	20	22	24	26	28	30	32
PD	0.543296	0.525644	0.509385	0.494394	0.480548	0.467735	0.455852	0.444808
N	34	36	38	40	42	44	46	48
PD	0.434521	0.424917	0.415934	0.407514	0.399605	0.392164	0.385151	0.378528
N	50	52	54	56	58	60	62	64
PD	0.372265	0.366332	0.360705	0.355359	0.350273	0.34543	0.340811	0.336401

case(ENR=32,PF=0.001):

N	2	4	6	8	10	12	14	16
PD	0.663055	0.722224	0.733095	0.727859	0.715147	0.698403	0.679351	0.658983
N	18	20	22	24	26	28	30	32
PD	0.637932	0.616624	0.595358	0.574347	0.553743	0.533657	0.514162	0.49531
N	34	36	38	40	42	44	46	48
PD	0.477132	0.459644	0.442851	0.42675	0.411329	0.396574	0.382467	0.368986
N	50	52	54	56	58	60	62	64
PD	0.356109	0.343811	0.332069	0.320858	0.310155	0.299935	0.290175	0.280854

case(ENR=32,PF=0.01):

N	2	4	6	8	10	12	14	16
PD	0.760384	0.828268	0.847672	0.852009	0.849533	0.843298	0.834722	0.82458
N	18	20	22	24	26	28	30	32
PD	0.813343	0.801327	0.788759	0.775808	0.762605	0.749256	0.735844	0.722438
N	34	36	38	40	42	44	46	48
PD	0.709094	0.695859	0.682769	0.669854	0.65714	0.644645	0.632384	0.620369
N	50	52	54	56	58	60	62	64
PD	0.608608	0.597107	0.585869	0.574897	0.564191	0.553749	0.54357	0.53365

case(ENR=32,PF=0.1):

N	2	4	6	8	10	12	14	16
PD	0.872	0.928327	0.945308	0.951871	0.954236	0.954462	0.953414	0.951517
N	18	20	22	24	26	28	30	32
PD	0.949007	0.946027	0.942671	0.939005	0.935078	0.93093	0.926591	0.922089
N	34	36	38	40	42	44	46	48
PD	0.917445	0.912679	0.90781	0.902853	0.897821	0.892728	0.887585	0.882403
N	50	52	54	56	58	60	62	64
PD	0.877191	0.871957	0.86671	0.861456	0.856203	0.850955	0.845719	0.840498

(The value marked with yellow is the maximum PD for optimum N)

Result and discussion:

1. In figure.1 we can see that in case ENR=10 when PF is lower than approximately 0.04 ,PD(N=2)> PD(N=4)> PD(N=8)> PD(N=16)> PD(N=32)> PD(N=64).In figure.2 we can know in case ENR=32,the rank of PD related to N is totally different from in case ENR=32.Easily we can learn that in very low PF Energy detector with N=8 has best performance.
2. Comparing figure.1 and figure.2 we can learn that when ENR becomes larger (from 10 to 32),the whole group of Energy detectors with different N performs better. For a fixed N which means a specific line in figure.1 or figure.2 and ENR=N*SNR, P_D=$Q_{\chi_N^2}(\frac{\gamma''}{\frac{\sigma_s{}^2}{\sigma^2}+1})$= $Q_{\chi_N^2}(\frac{\gamma''}{SNR+1})$,when ENR becomes larger or higher ,the SNR becomes larger. So the curves on the whole in figure.2 are above those in figure.1 which means Energy detector with higher signal to noise ratio performs better.
3. From figure.3 to figure.5, in a case PF is fixed when ENR is lower than a certain value, Energy detector N=2 has the best performance ,N=4 second …, and the N=64 worst. However, when ENR is higher than a certain value, the rank of performance changes as well. For example, in figure.3 we can see Energy detector with N=2 performs better than that with N=4 when ENR is smaller than about 11dB but when ENR is larger than 11dB the situation is opposite and Energy detector with N =4 performs better than N=2.
4. From figure.3. to figure.5. we can get that on a condition that PF is fixed, the larger the ENR is the larger the PD is or the better the performance is.
5. From figure.3 to figure.5. we can learn that when PF decrease from 10^{-1} to 10^{-3} the curves of PD vs ENR of Energy detector with N=2,4…,64 are shifted to the right. When PF decrease, the slope of curve of PD vs ENR becomes steeper or the value of slope becomes larger.
6. In figure.6 when ENR=10 ,PF=0.001 or 0.01 the curve of PD vs N is just a monotonic and its peak is at N=2 .When PF is larger ,PF=0.01 its peak vary from N=2 to N=4.In the case ENR=32,when PF becomes larger the peak is shifted to right.(from N=6 to 12).On a fixed ENR we can learn that the curve with high PF is above that with low PF, due to the equation of PD and PF. Similarly, on a fixed PF the curve with high value of ENR is above the curve with low value of ENR.

7. Draw the plot of PD vs N in the case of fixed PF and ENR we can get the optimum Energy detector with best performance by exploring where the peak is or which value of N samples is.

Summary (Conclusions):

1. In figure.1 we can see that in case ENR=10 when PF is lower than approximately 0.04 ,PD(N=2)> PD(N=4)> PD(N=8)> PD(N=16)> PD(N=32)> PD(N=64).In figure.2 we can know in case ENR=32,the rank of PD related to N is totally different from in case ENR=32.Easily we can learn that in very low PF Energy detector with N=8 has best performance.
2. When ENR becomes larger (from 10 to 32),the whole group of Energy detectors with different N performs better. For a fixed N which means a specific line in figure.1 or figure.2 and ENR=N*SNR, P_D=$Q_{\chi_N^2}(\frac{\gamma''}{\frac{\sigma_s{}^2}{\sigma^2}+1})$= $Q_{\chi_N^2}(\frac{\gamma''}{SNR+1})$,when ENR becomes larger or higher ,the SNR becomes larger. So the curves on the whole in figure.2 are above those in figure.1 which means Energy detector with higher signal to noise ratio performs better.
3. We can learn that when PF decrease from 10^{-1} to 10^{-3} the curves of PD vs ENR of Energy detector with N=2,4…,64 are shifted to the right. When PF decrease, the slope of curve of PD vs ENR becomes steeper or the value of slope becomes larger.
4. When ENR=10 ,PF=0.001 or 0.01 the curve of PD vs N is just a monotonic and its peak is at N=2 .When PF is larger ,PF=0.01 its peak vary from N=2 to N=4.In the case ENR=32,when PF becomes larger the peak is shifted to right.(from N=6 to 12).On a fixed ENR we can learn that the curve with high PF is above that with low PF, due to the equation of PD and PF. Similarly, on a fixed PF the curve with high value of ENR is above the curve with low value of ENR.

Appendix:

PD vs PF:

```
%GSGN PD vs PF
ENR1=10.^(10/10);
ENR2=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=getgama(PFA(i),N(1));
R1=2*r1;
gama1=R1/(ENR1/N(1)+1);
PD1(i)=Qchipr2(N(1),0,gama1,1e-5);
end
figure(1)
probpaper(PFA,PD1)
%N=4,ENR=10dB
for i=1:100
r2=getgama(PFA(i),N(2));
R2=2*r2;
gama2=R2/(ENR1/N(2)+1);
PD2(i)=Qchipr2(N(2),0,gama2,1e-5);
end
figure(1)
hold on
probpaper(PFA,PD2)
%N=8,ENR=10dB
for i=1:100
r3=getgama(PFA(i),N(3));
R3=2*r3;
gama3=R3/(ENR1/N(3)+1);
PD3(i)=Qchipr2(N(3),0,gama3,1e-5);
end
figure(1)
probpaper(PFA,PD3)
%N=16,ENR=10dB
for i=1:100
```

```
r4=getgama(PFA(i),N(4));
R4=2*r4;
gama4=R4/(ENR1/N(4)+1);
PD4(i)=Qchipr2(N(4),0,gama4,1e-5);
end
figure(1)
probpaper(PFA,PD4)
%N=32,ENR=10dB
for i=1:100
r5=getgama(PFA(i),N(5));
R5=2*r5;
gama5=R5/(ENR1/N(5)+1);
PD5(i)=Qchipr2(N(5),0,gama5,1e-5);
end
figure(1)
probpaper(PFA,PD5)
%N=64,ENR=10dB
for i=1:100
r6=getgama(PFA(i),N(6));
R6=2*r6;
gama6=R6/(ENR1/N(6)+1);
PD6(i)=Qchipr2(N(6),0,gama6,1e-5);
end
figure(1)
probpaper(PFA,PD6)
grid;
%--------------------------------------------------------------------
%N=2,ENR=15dB
for i=1:100
r1=getgama(PFA(i),N(1));
R1=2*r1;
gama1=R1/(ENR2/N(1)+1);
PD1(i)=Qchipr2(N(1),0,gama1,1e-5);
end
figure(2)
probpaper(PFA,PD1)
%N=4,ENR=15dB
for i=1:100
r2=getgama(PFA(i),N(2));
R2=2*r2;
gama2=R2/(ENR2/N(2)+1);
PD2(i)=Qchipr2(N(2),0,gama2,1e-5);
end
figure(2)
```

```
hold on
probpaper(PFA,PD2)
%N=8,ENR=15dB
for i=1:100
r3=getgama(PFA(i),N(3));
R3=2*r3;
gama3=R3/(ENR2/N(3)+1);
PD3(i)=Qchipr2(N(3),0,gama3,1e-5);
end
figure(2)
probpaper(PFA,PD3)
%N=16,ENR=15dB
for i=1:100
r4=getgama(PFA(i),N(4));
R4=2*r4;
gama4=R4/(ENR2/N(4)+1);
PD4(i)=Qchipr2(N(4),0,gama4,1e-5);
end
figure(2)
probpaper(PFA,PD4)
%N=32,ENR=15dB
for i=1:100
r5=getgama(PFA(i),N(5));
R5=2*r5;
gama5=R5/(ENR2/N(5)+1);
PD5(i)=Qchipr2(N(5),0,gama5,1e-5);
end
figure(2)
probpaper(PFA,PD5)
%N=64,ENR=15dB
for i=1:100
r6=getgama(PFA(i),N(6));
R6=2*r6;
gama6=R6/(ENR2/N(6)+1);
PD6(i)=Qchipr2(N(6),0,gama6,1e-5);
end
figure(2)
probpaper(PFA,PD6)
grid;
```

PD vs ENR:

```
%GSGN PD vs ENR
PFA1=10^-1;
PFA2=10^-2;
PFA3=10^-3;
```

```
ENR=[0:0.5:20];
x=10.^(ENR/10);
N=[2,4,8,16,32,64];
PD1=zeros(1,41);
PD2=zeros(1,41);
PD3=zeros(1,41);
PD4=zeros(1,41);
PD5=zeros(1,41);
PD6=zeros(1,41);
%PFA1=10^-1
r1=getgama(PFA1,N(1));
R1=2*r1;
r2=getgama(PFA1,N(2));
R2=2*r2;
r3=getgama(PFA1,N(3));
R3=2*r3;
r4=getgama(PFA1,N(4));
R4=2*r4;
r5=getgama(PFA1,N(5));
R5=2*r5;
r6=getgama(PFA1,N(6));
R6=2*r6;
for i=1:41
gama1=R1/(x(i)/N(1)+1);
PD1(i)=Qchipr2(N(1),0,gama1,1e-5);
gama2=R2/(x(i)/N(2)+1);
PD2(i)=Qchipr2(N(2),0,gama2,1e-5);
gama3=R3/(x(i)/N(3)+1);
PD3(i)=Qchipr2(N(3),0,gama3,1e-5);
gama4=R4/(x(i)/N(4)+1);
PD4(i)=Qchipr2(N(4),0,gama4,1e-5);
gama5=R5/(x(i)/N(5)+1);
PD5(i)=Qchipr2(N(5),0,gama5,1e-5);
gama6=R6/(x(i)/N(6)+1);
PD6(i)=Qchipr2(N(6),0,gama6,1e-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,'m') %pink
plot(ENR,PD6,'c') %cyan
```

```
grid;
%PFA1=10^-2
r1=getgama(PFA2,N(1));
R1=2*r1;
r2=getgama(PFA2,N(2));
R2=2*r2;
r3=getgama(PFA2,N(3));
R3=2*r3;
r4=getgama(PFA2,N(4));
R4=2*r4;
r5=getgama(PFA2,N(5));
R5=2*r5;
r6=getgama(PFA2,N(6));
R6=2*r6;
for i=1:41
gama1=R1/(x(i)/N(1)+1);
PD1(i)=Qchipr2(N(1),0,gama1,1e-5);
gama2=R2/(x(i)/N(2)+1);
PD2(i)=Qchipr2(N(2),0,gama2,1e-5);
gama3=R3/(x(i)/N(3)+1);
PD3(i)=Qchipr2(N(3),0,gama3,1e-5);
gama4=R4/(x(i)/N(4)+1);
PD4(i)=Qchipr2(N(4),0,gama4,1e-5);
gama5=R5/(x(i)/N(5)+1);
PD5(i)=Qchipr2(N(5),0,gama5,1e-5);
gama6=R6/(x(i)/N(6)+1);
PD6(i)=Qchipr2(N(6),0,gama6,1e-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,'m') %pink
plot(ENR,PD6,'c') %cyan
grid;
%PFA1=10^-3
r1=getgama(PFA3,N(1));
R1=2*r1;
r2=getgama(PFA3,N(2));
R2=2*r2;
r3=getgama(PFA3,N(3));
R3=2*r3;
```

```
r4=getgama(PFA3,N(4));
R4=2*r4;
r5=getgama(PFA3,N(5));
R5=2*r5;
r6=getgama(PFA3,N(6));
R6=2*r6;
for i=1:41
gama1=R1/(x(i)/N(1)+1);
PD1(i)=Qchipr2(N(1),0,gama1,1e-5);
gama2=R2/(x(i)/N(2)+1);
PD2(i)=Qchipr2(N(2),0,gama2,1e-5);
gama3=R3/(x(i)/N(3)+1);
PD3(i)=Qchipr2(N(3),0,gama3,1e-5);
gama4=R4/(x(i)/N(4)+1);
PD4(i)=Qchipr2(N(4),0,gama4,1e-5);
gama5=R5/(x(i)/N(5)+1);
PD5(i)=Qchipr2(N(5),0,gama5,1e-5);
gama6=R6/(x(i)/N(6)+1);
PD6(i)=Qchipr2(N(6),0,gama6,1e-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,'m') %pink
plot(ENR,PD6,'c') %cyan
grid;
```

PD vs N:

```
%GSGN PD vs N
N=[2:2:64];
ENR=[10,15];
x=10.^(ENR/10);
PFA=[0.001,0.01,0.1];
PD1=zeros(1,32);
PD2=zeros(1,32);
PD3=zeros(1,32);
PD4=zeros(1,32);
PD5=zeros(1,32);
PD6=zeros(1,32);
%ENR=10dB,PFA=0.001
for i=1:32
 r1=getgama(PFA(1),N(i));
```

```
 R1=2*r1;
 gama1=R1/(x(1)/N(i)+1);
 PD1(i)=Qchipr2(N(i),0,gama1,1e-5);
end
figure(6)
plot(N,PD1,'r')
%ENR=10dB,PFA=0.01
for i=1:32
 r2=getgama(PFA(2),N(i));
 R2=2*r2;
 gama2=R2/(x(1)/N(i)+1);
 PD2(i)=Qchipr2(N(i),0,gama2,1e-5);
end
figure(6)
hold on
plot(N,PD2,'g')
%ENR=10dB,PFA=0.1
for i=1:32
 r3=getgama(PFA(3),N(i));
 R3=2*r3;
 gama3=R3/(x(1)/N(i)+1);
 PD3(i)=Qchipr2(N(i),0,gama3,1e-5);
end
figure(6)
plot(N,PD3,'b')
%ENR=15dB,PFA=0.001
for i=1:32
 r4=getgama(PFA(1),N(i));
 R4=2*r4;
 gama4=R4/(x(2)/N(i)+1);
 PD4(i)=Qchipr2(N(i),0,gama4,1e-5);
end
figure(6)
plot(N,PD4,'k')
%ENR=15dB,PFA=0.01
for i=1:32
 r5=getgama(PFA(2),N(i));
 R5=2*r5;
 gama5=R5/(x(2)/N(i)+1);
 PD5(i)=Qchipr2(N(i),0,gama5,1e-5);
end
figure(6)
plot(N,PD5,'m')
%ENR=15dB,PFA=0.1
```

```
for i=1:32
 r6=getgama(PFA(3),N(i));
 R6=2*r6;
 gama6=R6/(x(2)/N(i)+1);
 PD6(i)=Qchipr2(N(i),0,gama6,1e-5);
end
figure(6)
plot(N,PD6,'c')
axis([2,64,0.01,1])
grid;
```

Iteration function:

```
function y=iteration(PFA,r,N)
s=0;
for i=1:(N/2)-1
    s=s+(r.^i)/factorial(i);
end
y=-log(PFA)+log(1+s);
```

Get threshold function:

```
function y=getgama(PFA,N)
x1=1;
x2=0;
while(abs(x1-x2)>1e-9)
    x2=x1;
    x1=iteration(PFA,x1,N);
end
y=x1;
```