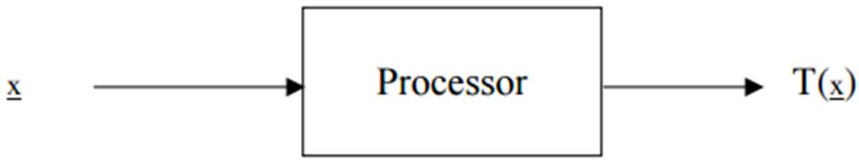
ECE 254 Homework 3

Correlated Noise

Name: Mingxuan Wang

Date: 2015/10/18

- Title: Correlated Noise
- Objective:



Our goal is to decide presence or absence of a sinusoid buried in correlated Gaussian noise

where: $x(n) = s(n) + n(n)$, $n = 0,1,\ldots, N-1$ and $N = 16$

$s(n) = A \sin(2\pi f_c n)$, $f_c = 1/16$, $A = (0.5)^{1/2}$

$n(n) = h(n) * w(n)$ is a correlated, Gaussian noise sequence

$w(n)$ in an uncorrelated, Gaussian noise sequence $\sim N(0,1)$

1. Let h(n) take on the following structure:
   a) h(n) = {1}
   b) h(n) = $(1/1.81)^{1/2}\{1, 0.9\}$
2. Compute and plot:
   a) $|S(f)|^2$(dB)
   b) $|H(f)|^2$(dB)
3. Determine the ROC performance of the following process
   a) SKE with h(n) as in 1a above. What is the processor input SNR?
      What is d^2?
   b) SKE with h(n) as in 1b above. What is the processor input SNR?
      What is d^2
   c) Mismatched matched filter with the actual data as in 1b above but the processor assuming the data was from 1a above. Thus, the processor is the conventional matched filter. What is the processor input SNR? What is is d^2

- Approach:

1. This one is only the definition of two signals.
   a) h(n) = {1} means that h(0) = 1, which is actually an impulse signal.
   b) h(n) = $(1/1.81)^{1/2}\{1, 0.9\}$ means that h(0) = $(1/1.81)^{1/2}$ and h(0) = $(1/1.81)^{1/2} * 0.9$. Actually there are two impulses at n = 0 and n = 1.

2. Compute these two values is not hard to do. I do Fourier Transform on s(n) and h(n), calculate square of them and then transfer them to dB

$$10 * log10(|S(f)|^2)$$

   In order to plot in the range(-pi, pi), we have to manipulate the result to transfer the second half part to the first half position

3. This problem contains 3 subproblems
   a) First, we can easily compute with

$$SNR1 = \frac{A^2}{2\sigma^2} = \frac{1}{4}$$

   From the lecture, we know that:

$$d^2 = \frac{\left(E(T|H0) - E(T|H1)\right)^2}{Var(T|H0)}$$

   For T above:

$$T(x) = x^T C^{-1} s$$

   In this case, C is diagonal matrix with all 1s. So that $T(x) = x^T s$.

   We can obtain:

$$E(T|H0) = 0$$

$$E(T|H1) = \frac{NA^2}{2}$$

$$Var(T|H0) = \frac{NA^2\sigma^2}{2}$$

   So that d1 = 2

   b) In case, C is not diagonal matrix with all 1s.

$$n(n) = \left(\frac{1}{1.81}^{0.5}\right) w(n) + \left(\frac{1}{1.81}^{0.5}\right) * 0.9$$

$$* w(n-1), w(n)\ is\ white\ Gaussian\ noise$$

C can be computed by:

$$C_{mn} = E\big(n(m)n(n)\big)$$

$$= \frac{1}{1.81}\big(1.81r_{ww}(m-n) + 0.9r_{ww}(m-n-1)$$

$$+ 0.9r_{ww}(m-n+1)\big),$$

$$where\ r_{ww}(n) = 1\ when\ n = 0;\ r_{ww}(n) = 0\ otherwise$$

So, Cmn = 1 when m =n, Cmn = 0.9/1.81 when |m-n| = 1, Cmn = 0 otherwise.

Then we can obtain:

$$E(T|H0) = 0$$

$$E(T|H1) = E(s^T C^{-1} s + n^T C^{-1} s) = s^T C^{-1} s$$

$$Var(T|H0) = E\big((x^T C^{-1} s - 0)^T (x^T C^{-1} s - 0)\big)$$

$$= E(s^T C^{-1} n n^T C^{-1} s) = s^T C^{-1} s$$

Then d = 1.4469.

We know that for correlated noise, its power is diagonal elements of C.

So that SNR2 = 1/4.

c) The procedure of computation is the combination of a) and b), we compute C like b), and for the rest part we treat it like a), which means the covariance of noise is just white Gaussian:

$$E(T|H0) = 0$$

$$E(T|H1) = E(s^T C^{-1} s + n^T C^{-1} s) = s^T s$$

$$Var(T|H0) = s^T C s$$

So d = 1.4438 and SNR3 = 1/4.

- Results(including plots):

1. In figure 1, we can see three plots for Problem 2. The first one is for $|S(f)|^2$(dB). We can conclude that the signal energy is mainly in a range less that (-1, 1)

   The second one is for $|H(f)|^2$(dB) of 1a. We can conclude that in this case h(n) is not a specific filter because it does nothing filtering signal and it is always 1 for the signal.

   The third one is for $|H(f)|^2$(dB) of 1b. It is a simple low pass filter that signal is mainly in low w area.
2. In figure 2, we can find that the ROC curve of b) and c) are nearly the same because according to our computation result, d of b) is 1.4469 and d of c) is 1.4438. The error of wrong filter we choose can be ignored.(conventional matched filter for color noise)
3. In figure 2, we can conclude that the larger the d is, the better ROC performance is.

Plots:

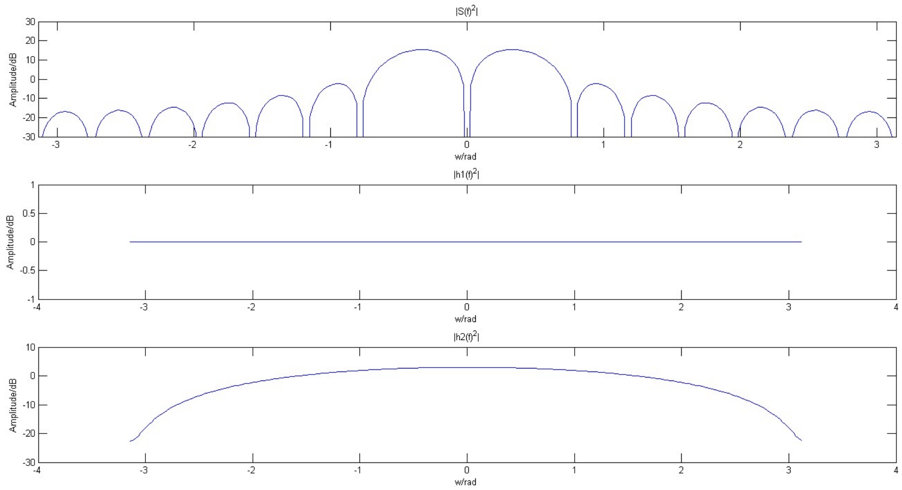


Figure 1 Spectrum of |S(f)|^2 and |H(f)|^2

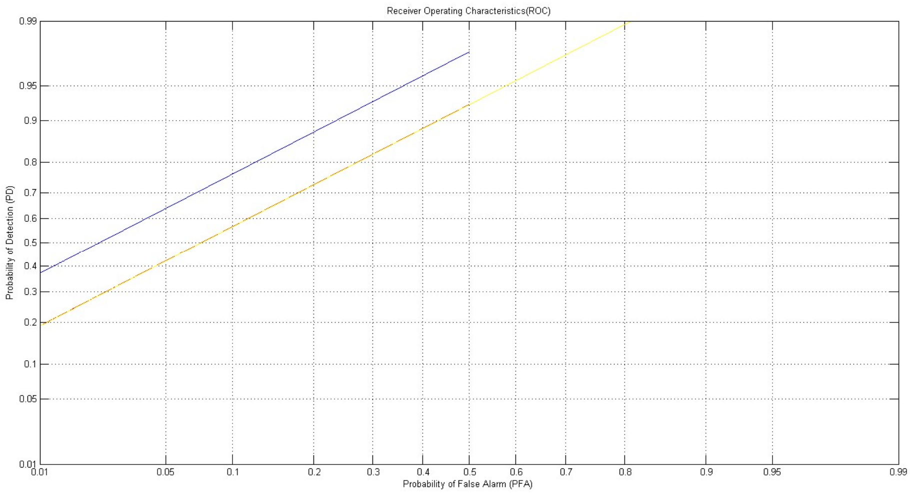


Figure 2 ROC performance of three different processors

- Appendix

```
%% define constance

n = 0:15;

N= 256;

fc = 1/16;

A = (1/2)^(1/2);

sn = A*sin(2*pi*fc*n);

hn1 =[1];

hn2 = (1/1.81)^(1/2)*[1, 0.9];

omega = -pi:2*pi/256:255*pi/256;


%% p2

sf = 10*log10((abs(fft(sn,N))).^2);

s = sf(129:256);

sf = [s, sf(1:128)];

hf1 = 10*log10((abs(fft(hn1, N))).^2);

hf2 = 10*log10((abs(fft(hn2, N))).^2);

h = hf2(129:256);

hf2 = [h, hf2(1:128)];

figure(1)

subplot(3,1,1)

plot(omega,sf)
```

```
title('|S(f)^2|')

xlabel('w/rad')

ylabel('Amplitude/dB')

axis([-pi pi -30 30])

subplot(3,1,2)

plot(omega,hf1)

title('|h1(f)^2|')

xlabel('w/rad')

ylabel('Amplitude/dB')

subplot(3,1,3)

plot(omega,hf2)

title('|h2(f)^2|')

xlabel('w/rad')

ylabel('Amplitude/dB')



%% p3

%1st

d1=4;

PF1=0:0.0001:0.5;

a1=Qinv(PF1);

PD1=Q(a1-d1^(1/2));
```

```
%2nd
C=eye(16);
for m=1:16
    for n=1:16
        if abs(m-n)==1
            C(m,n)=0.9/1.81;
            C(n,m)=0.9/1.81;
        end
    end
end
d2=sn/C*sn';
PF2=0:0.0001:0.5;
a2=Qinv(PF2);
PD2=Q(a2-d2^(1/2));

% 3rd
ETH1=sn*sn';
varTH0=sn*C*sn';
d3=ETH1^2/varTH0;
PF3=0:0.0001:0.95;
a3=Qinv(PF3);
```

```
PD3=Q(a3-d3^(1/2));

figure(2)
probpaper(PF1,PD1, 'b')
hold on
probpaper(PF2,PD2, 'r')
probpaper(PF3,PD3, 'y')
```