EE6401 Advanced Digital Signal Processing
Assignment 2
Student Name: FANG MIAO (G1901181B)

Q.1. Consider the autocorrelation matrix of $x[n]$, $\Gamma_M = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$ and the cross-correlation vector between $x[n]$ and $d[n]$, $\gamma_d = \begin{bmatrix} 0.25 \\ -0.25 \end{bmatrix}$.

1. Find $\boldsymbol{h_{opt}} = \begin{bmatrix} h_{opt}[0] \\ h_{opt}[1] \end{bmatrix}$ directly.

2. Let $\mu = 0.1$ and $\boldsymbol{h}_M(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ use the LMS algorithm to find $\boldsymbol{h_{opt}}$.

3. Let $\mu = 0.35$ and $\boldsymbol{h}_M(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ use the LMS algorithm to find $\boldsymbol{h_{opt}}$.

First, according to Eq. (113) in lecture notes, the optimum filter coefficients are given by

$$\boldsymbol{h_{opt}} = \Gamma_M^{-1}\gamma_d = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0.25 \\ -0.25 \end{bmatrix} = \begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix}$$

The LMS algorithm uses an iterative gradient method for solving for $\boldsymbol{h_{opt}}$. In general, algorithm for recursively computing the filter coefficients and therefore searching for the minimum of $\varepsilon_M^h = E[|e[n]|^2]$. Since Γ_M and γ_d are known in this case, so using *steepest descent search* to minimize ε_M^h and calculate $\boldsymbol{h_{opt}}$. And the main equation is

$$h_M(n+1) = h_M(n) - \frac{1}{2}\triangle(n)g(n)$$

Where $g(n)$ in this algorithm is,

$$g(n) = 2[\Gamma_M h_M(n) - \gamma(n)], n = 0,1,2,\ldots$$

Thus,

$$h_M(n+1) = [I - \triangle(n)\Gamma_M]h_M(n) + \triangle(n)\gamma_d, n = 0,1,2,\ldots$$

Where the step size $\triangle(n) = \mu = 0.1$ and $\boldsymbol{h}_M(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

If $\lim_{n\to\infty} g(n) = 0$, $g(n) = \frac{d\varepsilon_M^h(n)}{dh_M(n)} = 2[\Gamma_M h_M(n) - \gamma(n)]$, then $\lim_{n\to\infty} h_M(n) = h_{opt}$.

I select 0.001 as the minimum error, i.e. when the difference between the gradient g and zero less than 0.001 stop the iteration.

With the step size $\mu = 0.1$ and the original filter coefficients $\boldsymbol{h}_M(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, after 124 iterations the $\boldsymbol{h_{opt}} = \begin{bmatrix} 0.4991 \\ -0.4991 \end{bmatrix}$.

With the step size is 0.35, the optimal coefficients obtained after 35 iterations, the $\boldsymbol{h_{opt}} = \begin{bmatrix} 0.4993 \\ -0.4993 \end{bmatrix}$.

As results show before, the filter coefficients obtained by LMS algorithm are same as Wiener-Hopf Equation method. When using the LMS method, the step size can make difference on the speed of coverage.
The entire data of 2 processes with different step size shows below.

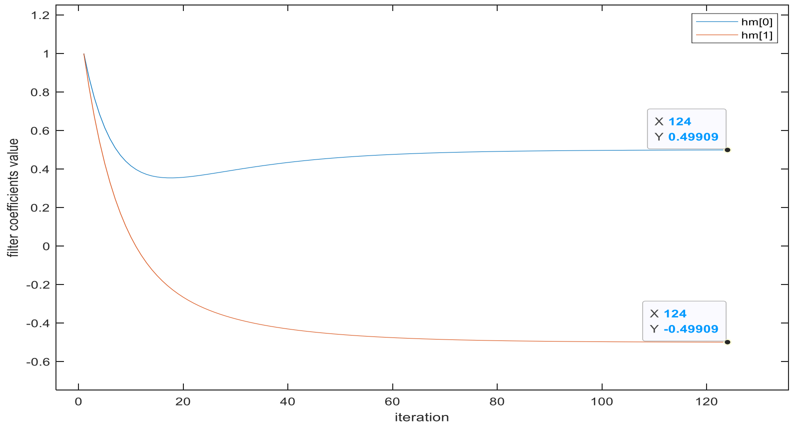


Figure 1 the value of the filter coefficients $h_M[0], h_M[1]$ of each iteration with step size is 0.1

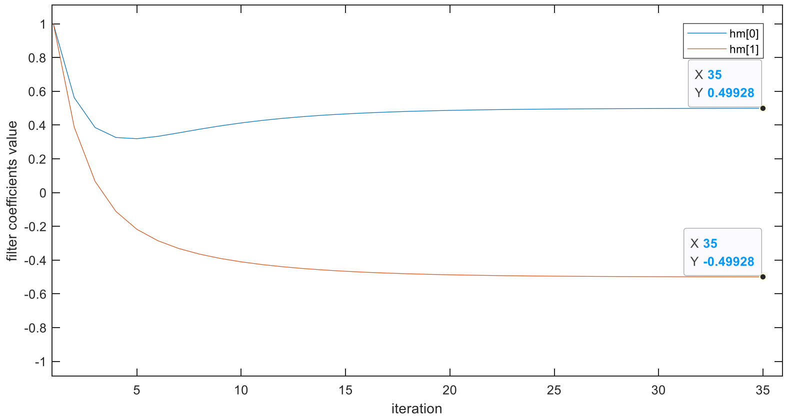


Figure 1 the value of the filter coefficients $h_M[0], h_M[1]$ of each iteration with step size is 0.35
The corresponding MATLAB code shows below.

```
%LSM-Steepest descent search
%autocorrelation matrix
autocorrelation_matrix=[1 0.5; 0.5 1];
%crosscorrelation vector
crosscorrelation_vector=[0.25; -0.25];
%orignal fliter doeffiences
hm_0=[1;1];
%step size
step_size=0.1;
%identify matrix
I=eye(2);
%output matrix
output=zeros(4,1);
output(1,1)=hm_0(1);
output(2,1)=hm_0(2);
%maximum iteration number
max_ite=1000;
```

```
%minimum error
min_err=0.001;

%when loop for max_ite times or the gradient vector error less than min_err, algorithm stop
for i = 2: max_ite
    h_pre = [output(1,i-1); output(2,i-1)];
    g = 2 * (autocorrelation_matrix*h_pre-crosscorrelation_vector);
    h=(I-step_size*autocorrelation_matrix)*h_pre+step_size*crosscorrelation_vector;
    output(1,i)=h(1);
    output(2,i)=h(2);
    output(3,i)=g(1);
    output(4,i)=g(2);
    if(g(1)<min_err && g(2)<min_err)
        break;
    end
end

%plot the evolution graph
x=(1:size(output,2));
y1=output(1,:);
y2=output(2,:);

plot(x,y1);
hold on;
plot(x,y2);
ylim([-1 1]);
```

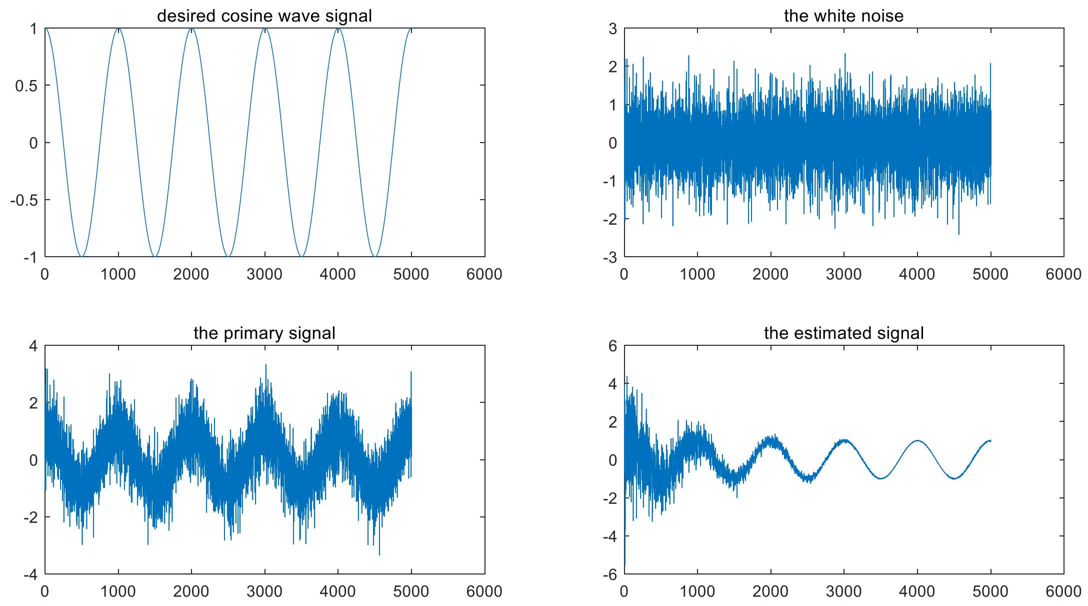
Q.2. Consider the primary signal $d[n] = \cos(2\pi\omega_0 n) + y_0[n]$ where the noise $y_0[n]$ is the output of the $F_0(z) = 0.5 - 0.5z^{-1}$ when excited with a white noise sequence $w[n]$.
Consider the input signal $x[n] = w[n]$.

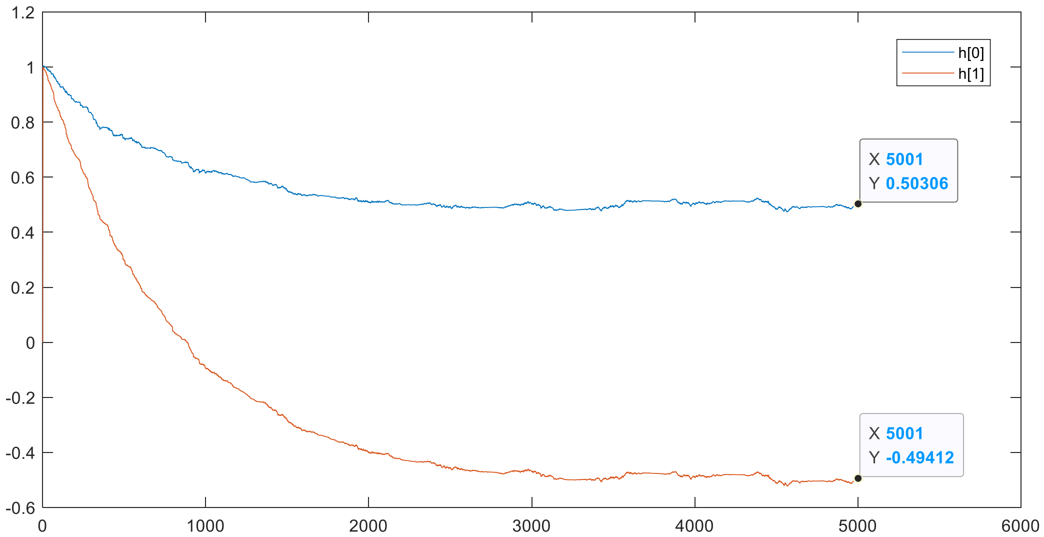
1. Let $\mu = 0.00125$ and $\boldsymbol{h}_M(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ use the LMS algorithm to find $\boldsymbol{h}_{opt}$.
2. Plot the primary signal $d[n]$.
3. Plot $e[n] = d[n] - y[n]$ and the desired sine wave signal.
4. Plot the evolution of the two filter parameters.

Since the autocorrelation matrix and the cross-correlation vector are unknown in this question, LMS with stochastic-gradient-descent algorithm can be used to build the adaptive filter. And the main equation in this algorithm is

$$h_M(n+1) = h_M(n) + \Delta(n)e(n)X_M^*(n), n = 0,1,2 \ldots$$

Where $\Delta(n)$ is the step size (0.00125) and $e(n)$ is error between the primary signal and the output of the input signal through the adaptive filter. $X_M^*(n) = [x[n], x[n-1], x[n-2], \ldots, x[n-M+1]]^T$.
I select 5000 data points of desired sine wave signal and the noise. After iterations, filter coefficients converge to $\boldsymbol{h}_{opt} = \begin{bmatrix} 0.5031 \\ -0.4941 \end{bmatrix}$ (the value in the 5001$^{\text{th}}$ iteration).
The graphs of important signals show below.





The corresponding MATLAB shows below,

```
%obtain cosine wave signal
t=0:pi/500:10*pi;
x=cos(t);
x=x(:);%cosine value
sz=size(x,1);
subplot(3,2,1);
title('the desired sine wave signal');
plot(x);
%white noise
z=tf('z',1);
F0=0.5-0.5*z^-1;
w=randn(size(x));
noise=lsim(F0, w);%y0
subplot(3,2,2);
title('the white nosie');
plot(noise);
% the primary signal
d=x+noise;
```

```
subplot(3,2,3);
title('the primary signal');
plot(d);

%step size
mu=0.00125;
%original coefficients vector
hm_0=[1;1];

[hm,ee]=LMS(mu, hm_0, w, d);
% en = dn - ee
en=d-ee;
subplot(3,2,4);
title('the estimated signal');
plot(en);

title('the filter coefficients');
plot((1:1:5001),hm(1,:));
hold on;
plot((1:1:5001),hm(2,:));

%using stochastic-gradient-descent algorithm
function [hm,ee]=LMS(mu, h0, x, d)
% LMS
% input args:
% mu = the step size
% h0 = original filter coefficients
% x = input signal , whith noise sequence w[n]
% d = desired signal, the primary signal d[n]
% output args:
% hm = filter coefficients in each iterations
% ee = estimated error

% input signal length
N = length(x);
% make sure x and d are column vectos
x=x(:);
d=d(:);
% filter coefficients number
M=size(h0, 1);
hm=zeros(2,1);
ee=zeros(size(x));%adapted output
for n=M:N
    arr=x(n:-1:n-M+1);
    e(n)=d(n)-h0'*arr;
    h0=h0+mu*e(n)*arr;
    hm(1,n)=h0(1);
    hm(2,n)=h0(2);
    ee(n)=h0'*arr;
end

end
```

Q.3.

1. Let $\Gamma_{ss}(z) = \frac{1}{(1-0.6z^{-1})(1-0.6z)}$, $\Gamma_{ww}(z) = 1$, and $\Gamma_{xx}(z) = \Gamma_{ss}(z) + \Gamma_{ww}(z)$. Factorize $\Gamma_{xx}(z)$ as $\Gamma_{xx}(z) = G(z)G(z^{-1})$, where $G(z)$ is the minimum phase causal and stable filter.

Since $\Gamma_{xx}(z) = \Gamma_{ss}(z) + \Gamma_{ww}(z)$, one has

$$\Gamma_{xx}(z) = \Gamma_{ss}(z) + \Gamma_{ww}(z) = \frac{1 + (1-0.6z^{-1})(1-0.6z)}{(1-0.6z^{-1})(1-0.6z)}$$

$$= \frac{2.196(1-0.2732z^{-1})(1-0.2732z)}{(1-0.6z^{-1})(1-0.6z)} = G(z)G(z^{-1})$$

Where the $G(z)$ is

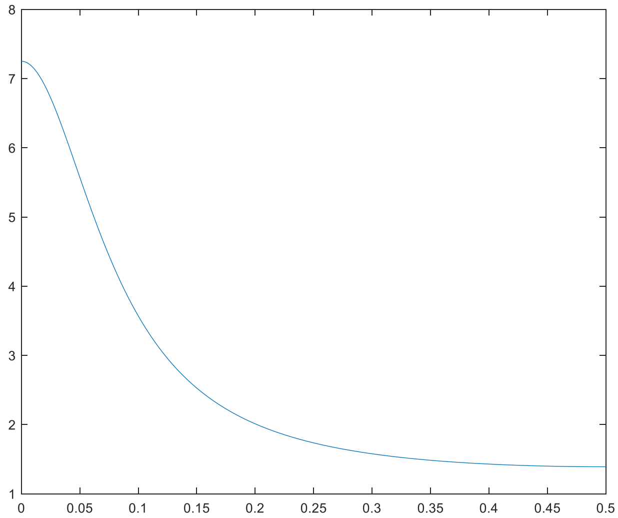
$$G(z) = \frac{\sqrt{2.196}(1-0.2732z^{-1})}{(1-0.6z^{-1})}$$

2. Plot the power spectrum density function $\Gamma_{xx}(f)$.

Evaluate the $\Gamma_{xx}(z)$ equation on the unit circle $z = e^{j2\pi f}$

$$\Gamma_{xx}(f) = 2.196\left|\frac{1-0.2732z}{1-0.6z}\right|^2$$

And plot of the power spectrum density function shows below,



3. Consider an autoregressive AR(2) random process $v[n]$, having auto-correlation function $r_v[m]$, with $r_v[0] = 0.5196, r_v[1] = 0.4171$ *and* $r_v[2] = 0.1654$. Plot the power spectrum density function $\Gamma_{vv}(f)$ using the Yule-Walker method.

Obtained the AR parameters by the equation (187) in lecture,

$$\begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \end{bmatrix} = \begin{bmatrix} r_v[0] & r_v[-1] \\ r_v[1] & r_v[0] \end{bmatrix}^{-1} * \left(-\begin{bmatrix} r_v[1] \\ r_v[2] \end{bmatrix}\right) = \begin{bmatrix} -1.5387 \\ 0.9169 \end{bmatrix}$$

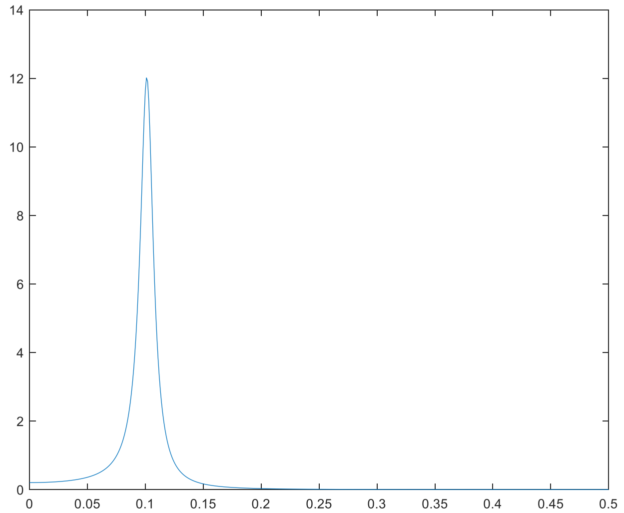
The estimate of σ_w^2 can be obtained from

$$\hat{\sigma}_w^2 = \hat{r}_v(0) + \sum_{k=1}^{2} \hat{a}_k \hat{r}_v(-k) = 0.5196 + (-1.5387 \times 0.4171) + 0.9169 \times 0.1654$$
$$= 0.02946$$

The obtained PSD is

$$\Gamma_{vv}(f) = \frac{0.02946}{|1 + (-1.5387e^{-j2\pi f}) + 0.9169e^{-j4\pi f}|^2}$$

And the plot of it shows below,



4. Compare the plots of $\Gamma_{xx}(f)$ and $\Gamma_{vv}(f)$ and briefly comment on them.

The peak of $\Gamma_{xx}(f)$ is at f = 0 and the peak of $\Gamma_{vv}(f)$ is at f = 0.1.