Table of Contents

```
clear;
clc;
close all;
```

Mohammad Javad Amin 401211193 Problem 1 , exercise 1

definition

d : desired signal N :length of filter M : length of input signal alpha : learning rate e : errors w : weights of filter m_error : mean squared error p : power of input signal

```
a=1;
b=[1,1.8,0.81];         % impulse response
inputs=randn(1,100);
d=filter(b,a,inputs);
M=length(inputs);
```

part a

```
 N=4;
 p= inputs*inputs'/M;
 alpha_max=2/(3*N*p);
 disp('mu max fro N=4 and is :');
 disp(alpha_max);
```

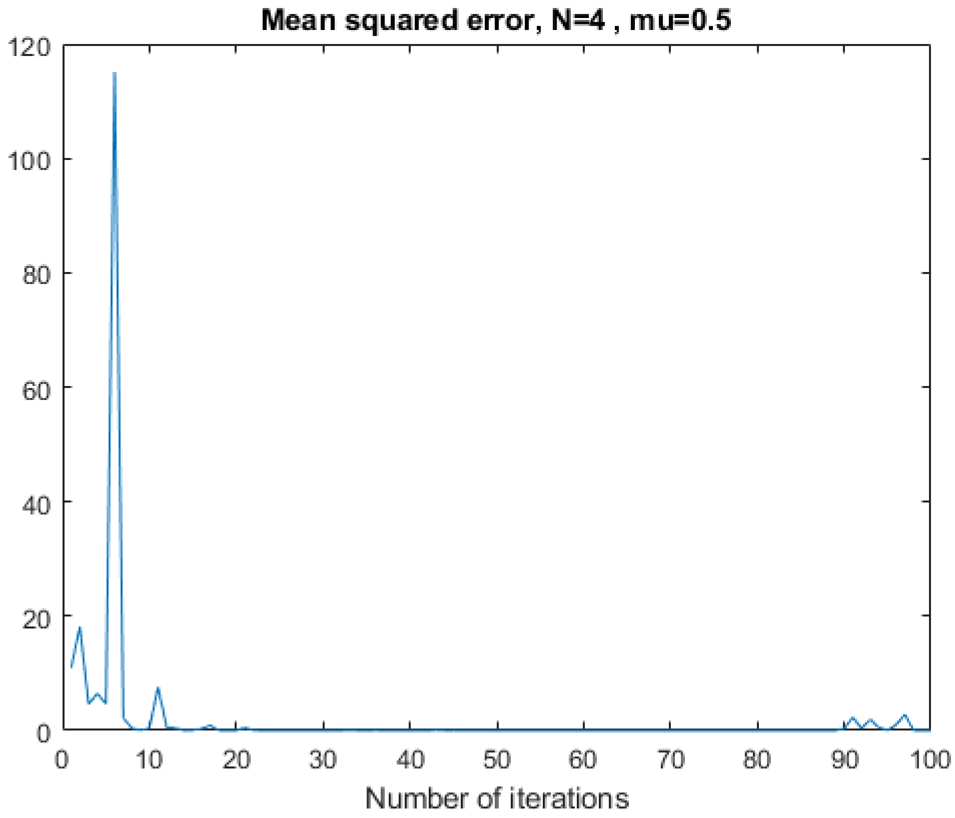
```
mu max fro N=4 and is :
    0.1730
```

part b

```
alpha=0.5;
[w,~]=LMS(inputs,d,N,alpha,M);
disp("weights for mu=0.5 and N=4 :");
disp(w');
disp('because mu is begger than u_max may be LMS algorithm not converged')
```

part c

```
k=5;
m_error=zeros(1,M);

for i=1:k
    [~,cost]=LMS(inputs,d,N,alpha,M);
    m_error=m_error+cost;
end
m_error=m_error/5;
figure
plot(m_error);
title('Mean squared error, N=4 , mu=0.5');
xlabel('Number of iterations');
```



part d

```
[~,~,J_min,J_inf]=LMS(inputs,d,N,alpha,M);
J_ex=J_inf - J_min;
disp("excess Mean squared error")
disp(J_ex)
```

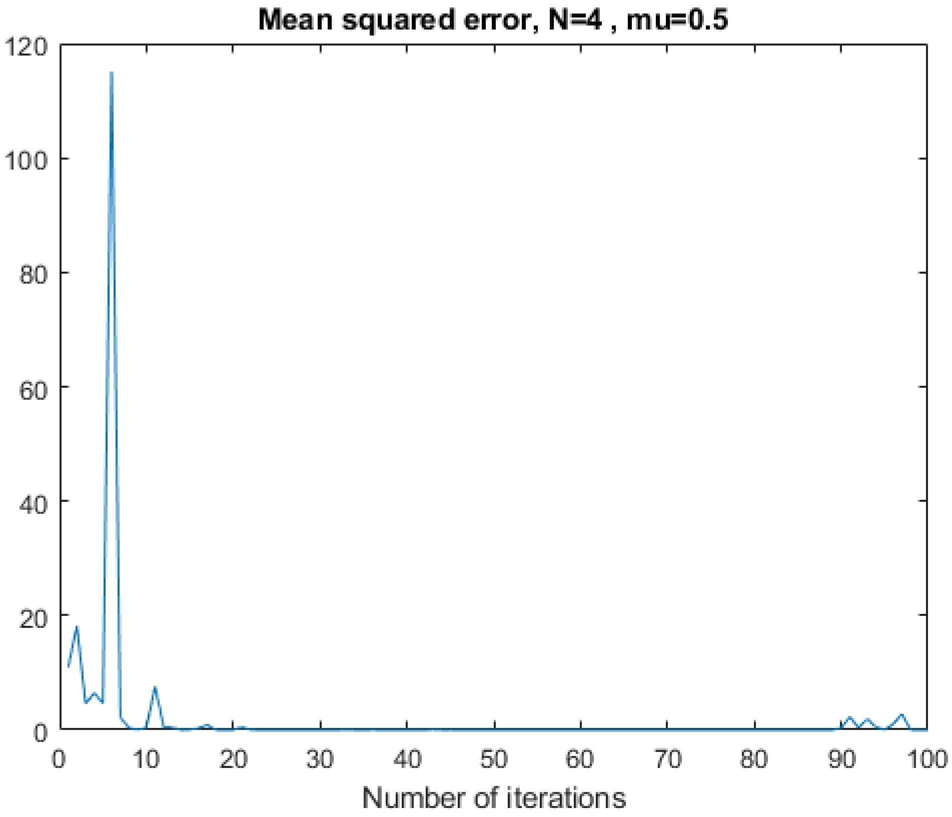
```
excess Mean squared error
    0.4650
```

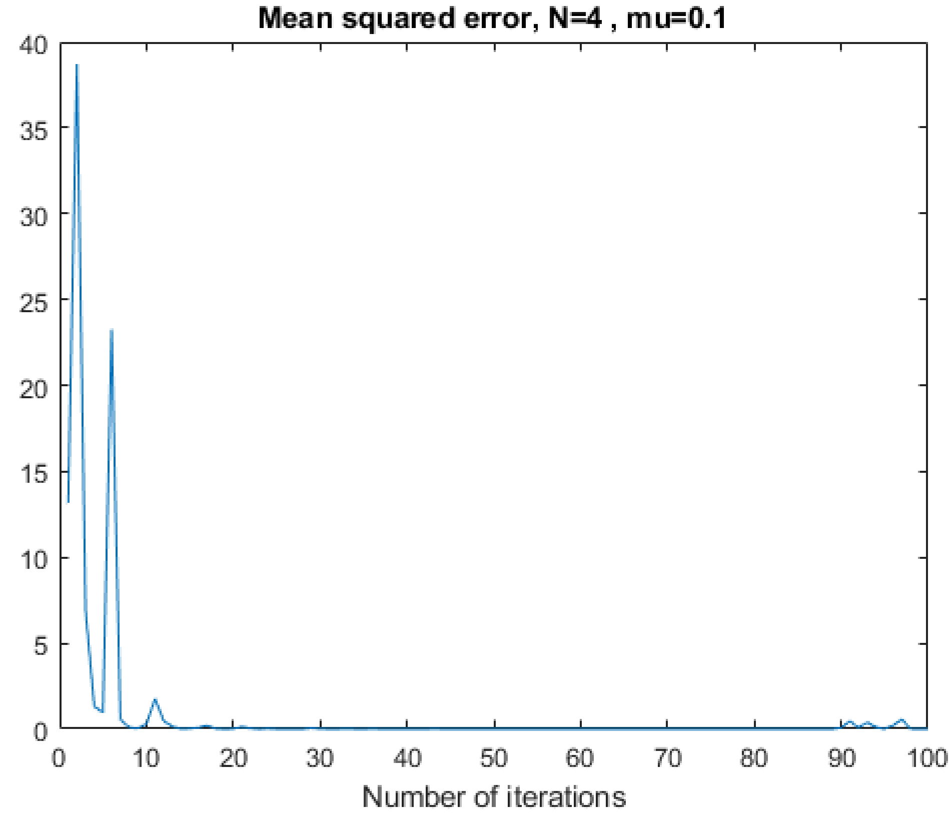
part e

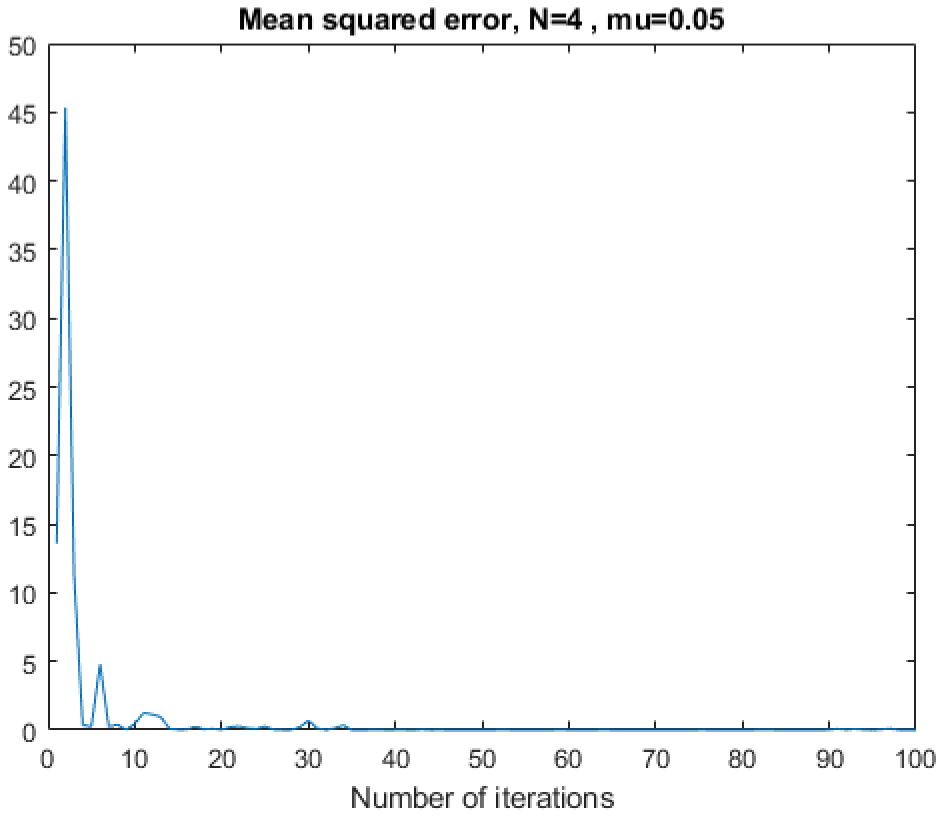
mu = 0.1 and N=4

```
alpha=0.1;

for i=1:k
    [w,cost]=LMS(inputs,d,N,alpha,M);
    m_error=m_error+cost;
end
m_error=m_error/5;
disp("weights for mu=0.1 and N=4  :");
disp(w');
figure
plot(m_error);
title('Mean squared error, N=4 , mu=0.1');
xlabel('Number of iterations');

% mu = 0.05 and N=4
alpha=0.05;

for i=1:k
    [w,cost]=LMS(inputs,d,N,alpha,M);
    m_error=m_error+cost;
end
m_error=m_error/5;
disp("weights for mu=0.05 and N=4  :");
disp(w');
figure
plot(m_error);
title('Mean squared error, N=4 , mu=0.05');
xlabel('Number of iterations');
```

```
weights for mu=0.1 and N=4  :
    1.0000
    1.8000
    0.8100
   -0.0000

weights for mu=0.05 and N=4  :
    1.0016
    1.7968
    0.8096
   -0.0021
```


Mean squared error, N=4 , mu=0.5
120
100
80
60
40
20
0
0
10
20
30
40
50
60
70
80
90
100
Number of iterations


Mean squared error, N=4 , mu=0.1
40
35
30
25
20
15
10
5
0
0
10
20
30
40
50
60
70
80
90
100
Number of iterations



part e

mu = 0.5 and N=2

```
alpha=0.5;
N=2;

for i=1:k
    [w,cost]=LMS(inputs,d,N,alpha,M);
    m_error=m_error+cost;
end
m_error=m_error/5;
disp("weights for mu=0.5 and N=2  :");
disp(w');
figure
plot(m_error);
title('Mean squared error, N=2 , mu=0.5');
xlabel('Number of iterations');

% mu = 0.5 and N=3

alpha=0.5;
N=3;

for i=1:k
    [w,cost]=LMS(inputs,d,N,alpha,M);
    m_error=m_error+cost;
end
m_error=m_error/5;
disp("weights for mu=0.5 and N=3 :");
disp(w');
figure
plot(m_error);
title('Mean squared error, N=3 , mu=0.5');
xlabel('Number of iterations');
```

LMS algorithms

```
function[w,cost,J_min,J_inf]=LMS(inputs,d,N,alpha,M)
% e : error
% u_temp : because LMS run when the first sample arrive, we put M-1 zeros in beging of inputs, if whe don't put this zeros we must wait to m sample arrive
    u_temp=[zeros(1,N-1),inputs];
    e=zeros(1,M);
    w=zeros(1,N);
    for i=N:M
        u=u_temp(i:-1:i-N+1);
        y=dot(w,u);
        e(i-N+1)=d(i-N+1)-y;
        w =  w + alpha*e(i-N+1)*u;
    end
    cost=e.^2;
    J_min=min(cost);
    J_inf=sum(cost(M-19:M))/20;

end
```

```
weights for mu=0.5 and N=4 :
    0.7950
    2.6811
    0.6021
    0.4111

because mu is begger than u_max may be LMS algorithm not converged
```

Published with MATLAB® R2022b