Table of Contents

..... 1
definition 1
part a 1
part b 2
part c 7
LMS algorithms 8


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

Mohammad Javad Amin 401211193 Problem 1 , exercise 3

definition

d : desired signal N :length of filter M : length of input signal alpha : learning rate e : errors w : weights of filter p : power of input signal l : noise amplitude d_t : corrupted desired signal

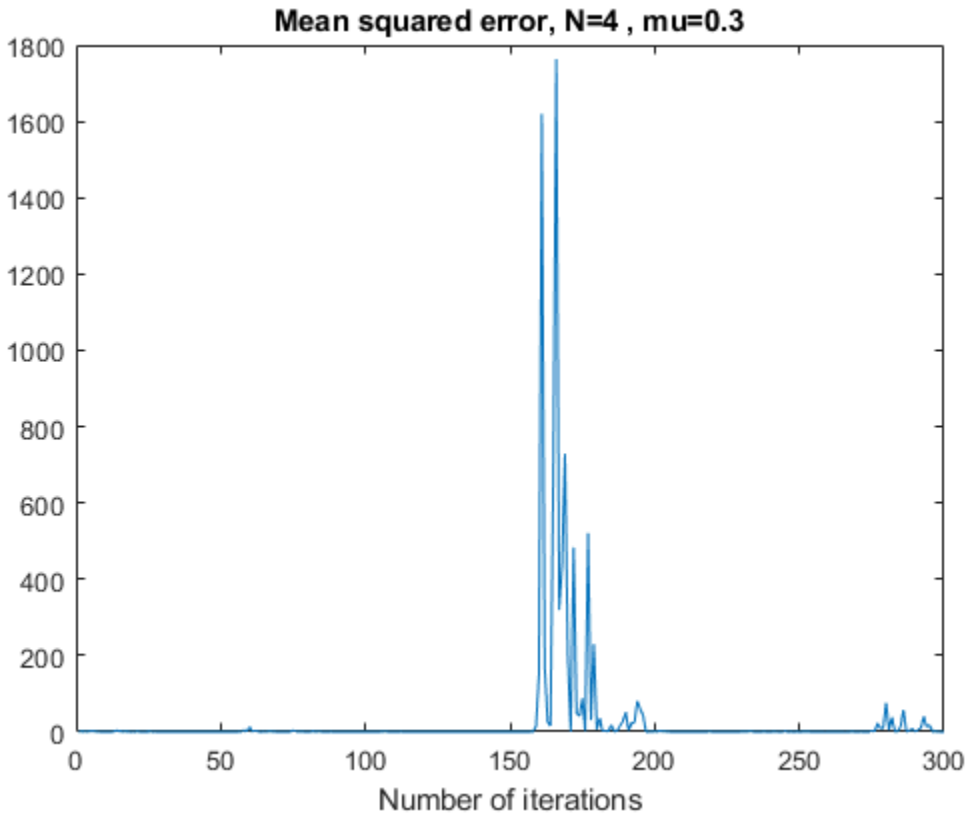
```
a=[1,0.5];
b=[1,-0.9];          % impulse response
inputs=randn(1,300);
d=filter(b,a,inputs);
M=length(inputs);
```

part a

```
alpha = 0.3;
N = 4;
k=5;
m_error=zeros(1,M);

% calulate mu max for N=4
p= inputs*inputs'/M;
 alpha_max=2/(2*N*p);
 disp('mu max for N=4 and is :');
 disp(alpha_max);

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=0.3 and N=4 :");
disp(w');
disp('if mu is begger than u_max may be LMS algorithm not converged')

figure
plot(m_error);
```

```
title('Mean squared error, N=4 , mu=0.3');
xlabel('Number of iterations');
```

```
mu max for N=4 and is :
    0.2217
```



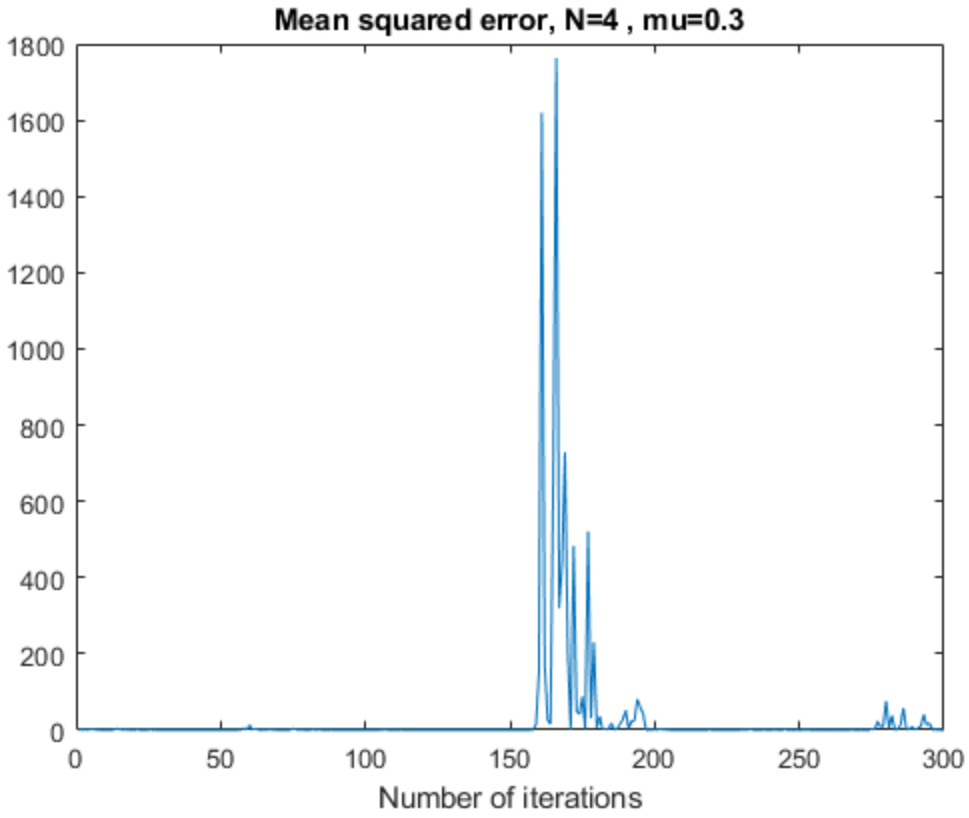
part b

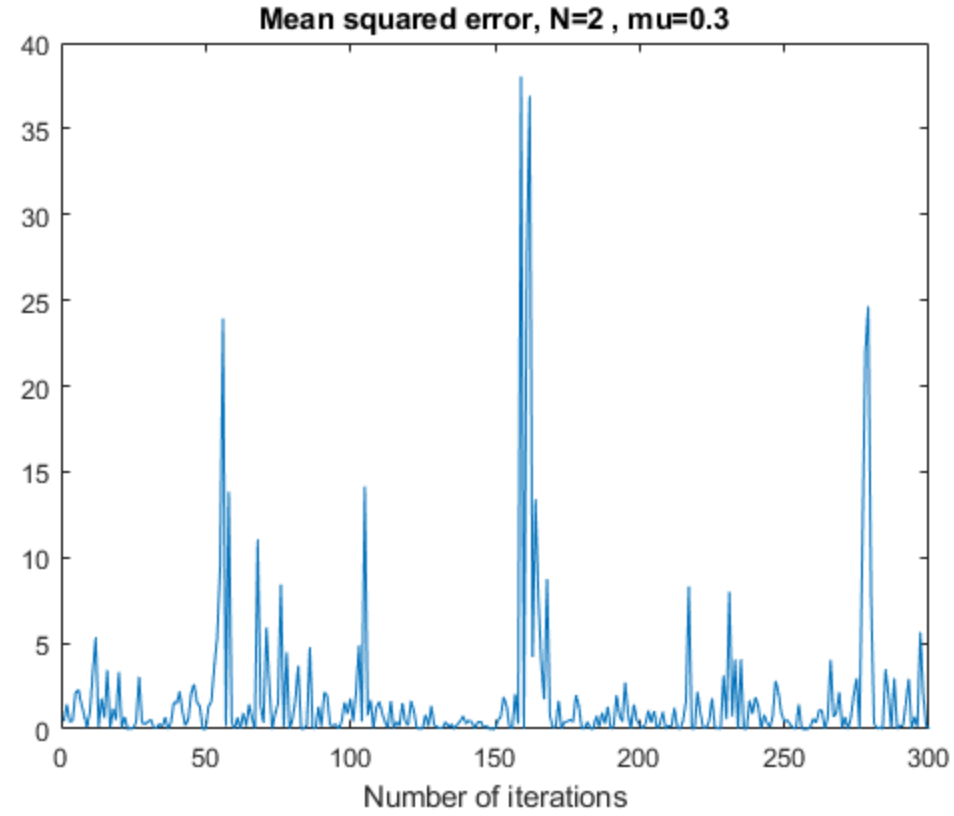
```
N = [2,3,5,7,10];
for i=N
    alpha_max=2/(3*i*p);
    disp(['mu max for N=',num2str(i),' and mu= 0.3 is :']);
    disp(alpha_max);

    m_error=zeros(1,M);

    for g=1:k
        [w,cost]=LMS(inputs,d,i,alpha,M);
        m_error=m_error+cost;
    end
    m_error=m_error/5;

    disp(['weights for mu=0.3 and N=',num2str(i),':']);
    disp(w');
    disp('if mu is begger than u_max may be LMS algorithm not converged')
```

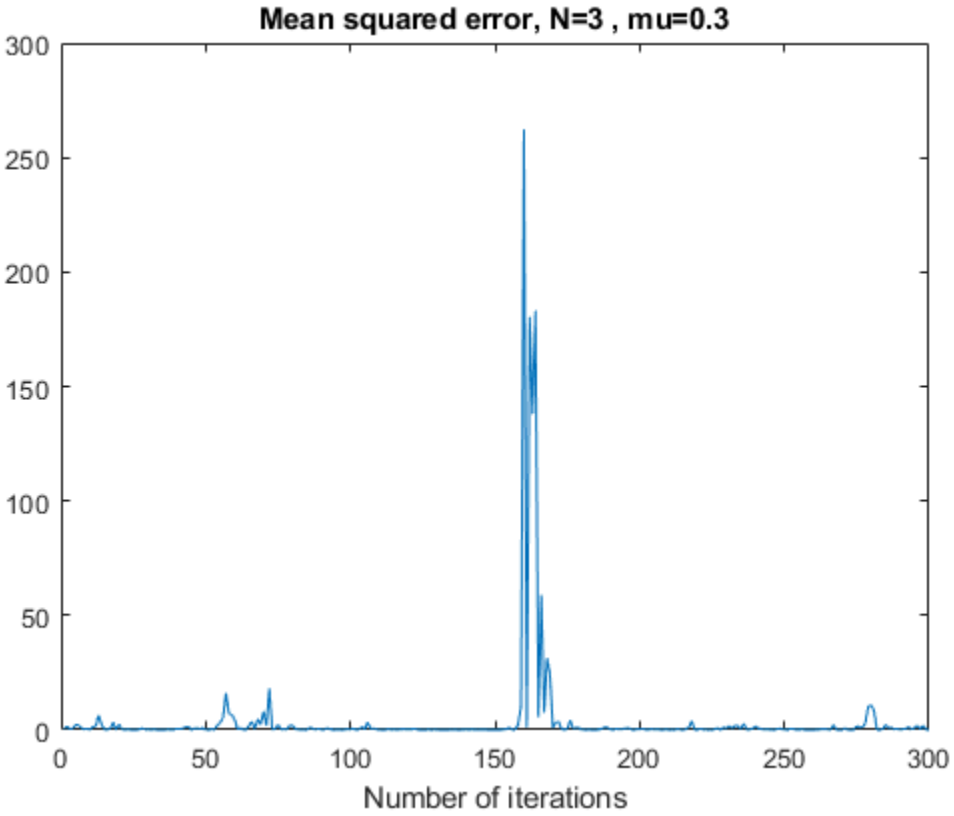
```
    figure
    plot(m_error);
    title(['Mean squared error, N=',num2str(i),' , mu=0.3']);
    xlabel('Number of iterations');
end

mu max for N=2 and mu= 0.3 is :
    0.2957

weights for mu=0.3 and N=2:
    1.5540
   -1.7501

if mu is begger than u_max may be LMS algorithm not converged
mu max for N=3 and mu= 0.3 is :
    0.1971

weights for mu=0.3 and N=3:
    1.0896
   -2.0639
    1.0974

if mu is begger than u_max may be LMS algorithm not converged
mu max for N=5 and mu= 0.3 is :
    0.1183

weights for mu=0.3 and N=5:
   -1.4362
    1.3253
   -2.7794
   -1.2081
    0.4730

if mu is begger than u_max may be LMS algorithm not converged
mu max for N=7 and mu= 0.3 is :
    0.0845

weights for mu=0.3 and N=7:
   1.0e+10 *

    0.2396
    0.5208
    0.2405
    0.4486
   -5.4098
    3.3696
   -1.7381

if mu is begger than u_max may be LMS algorithm not converged
mu max for N=10 and mu= 0.3 is :
    0.0591

weights for mu=0.3 and N=10:
```

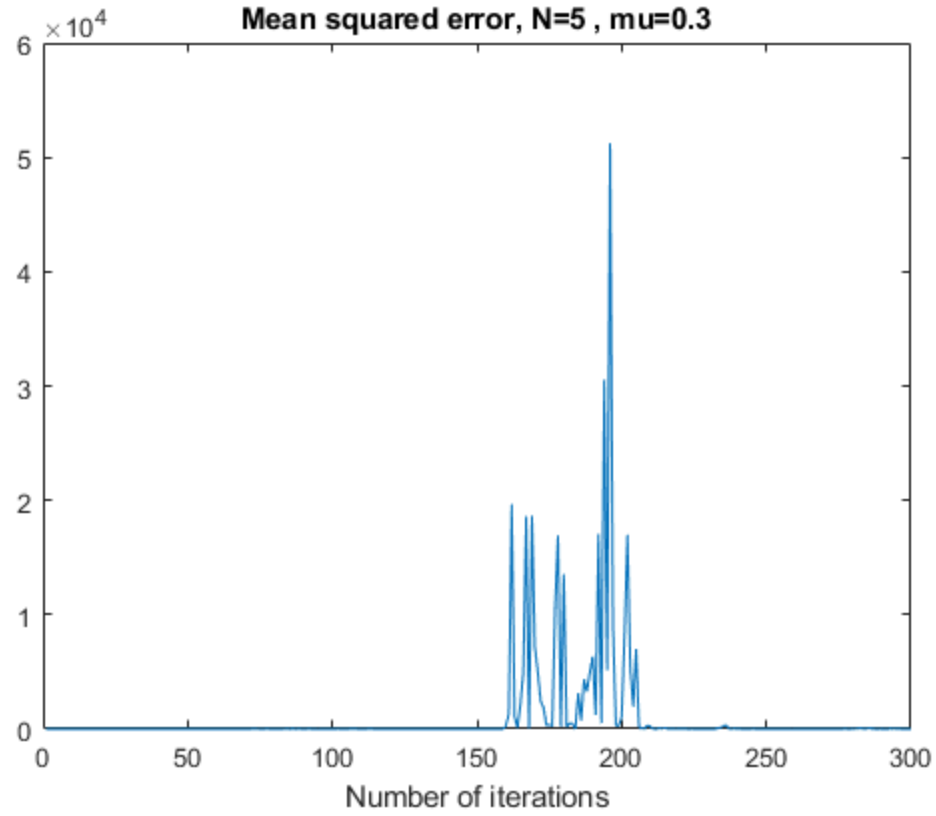
```
   1.0e+21 *

    0.1353
    0.9931
    0.8193
   -0.4980
   -0.3445
    2.4894
   -1.9225
   -0.4096
    1.3771
    0.9470
```

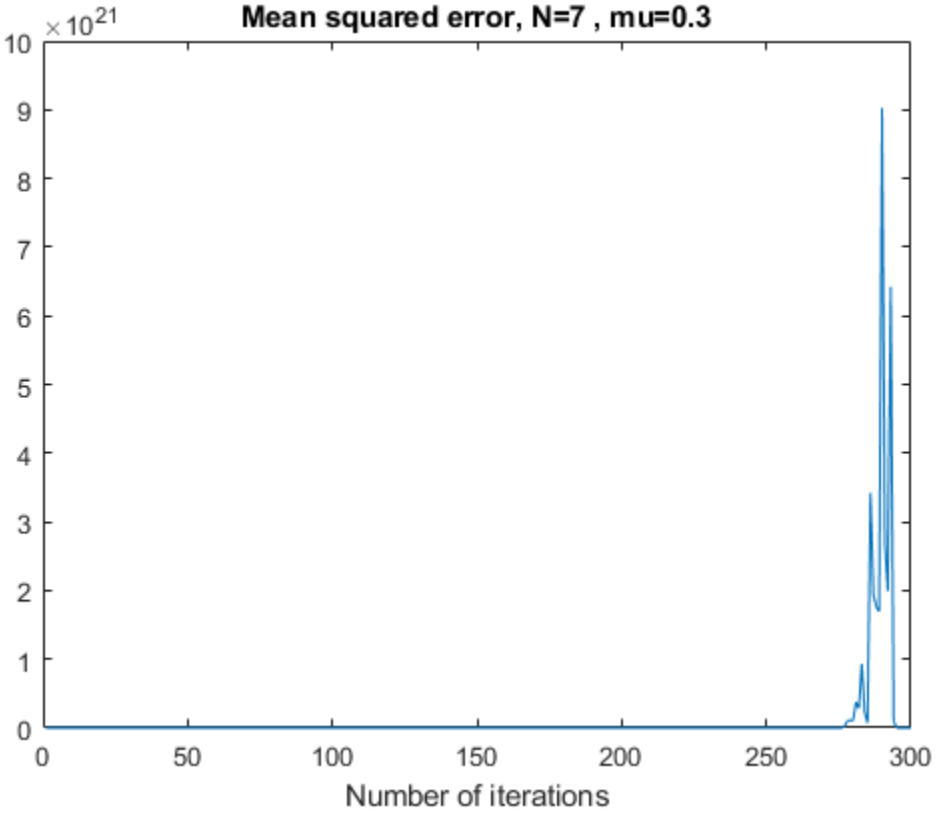
if mu is begger than u_max may be LMS algorithm not converged

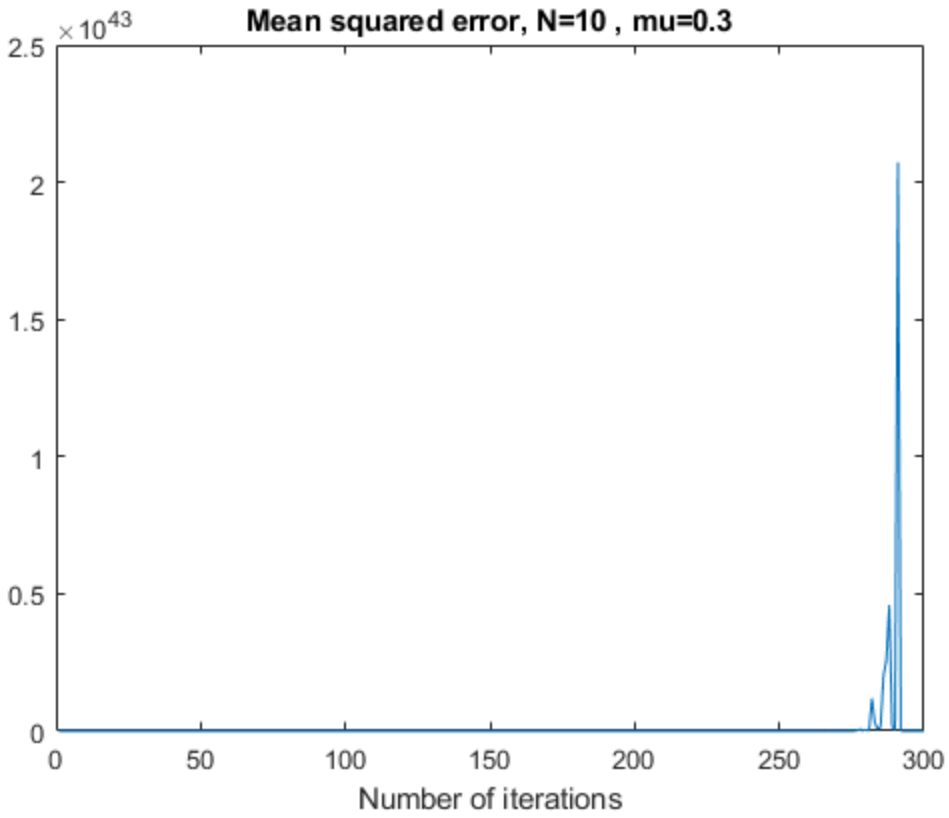



Mean squared error, N=2 , mu=0.3
40
35
30
25
20
15
10
5
0
0
50
100
150
200
250
300
Number of iterations


Mean squared error, N=3 , mu=0.3
300
250
200
150
100
50
0
0
50
100
150
200
250
300
Number of iterations


Mean squared error, N=5 , mu=0.3
×10^4
6
5
4
3
2
1
0
0
50
100
150
200
250
300
Number of iterations


Mean squared error, N=7 , mu=0.3
×10^21
10
9
8
7
6
5
4
3
2
1
0
0
50
100
150
200
250
300
Number of iterations



part c

```
l = [0.1,0.3,1];
N = 4;
v = randn(1,300);
alpha = 0.3;
k=5;

% calulate mu max for N=4
p= inputs*inputs'/M;
alpha_max=2/(2*N*p);
disp('mu max for N=4 and is :');
disp(alpha_max);
disp('if mu is begger than u_max may be LMS algorithm not converged')
for g=l
    m_error=zeros(1,M);
    d_t=d+g*v;


    for i=1:k
        [w,cost]=LMS(inputs,d_t,N,alpha,M);
        m_error=m_error+cost;
    end
    m_error=m_error/5;
```

```
    disp(['weights for mu=0.3 , N=4 and l=',num2str(g), ': ']);
    disp(w')

    figure
    plot(m_error);
    title(['Mean squared error, N=4 , mu=0.3 and l=', num2str(g),' : ']);
    xlabel('Number of iterations');
end
```

```
mu max for N=4 and is :
    0.2217
```

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

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=0.3 and N=4 :
    3.7184
   -1.5416
    1.4777
    1.6576
```

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

Published with MATLAB® R2022b