Table of Contents


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

Mohammad Javad Amin 401211193 Problem 1 , exercise 3

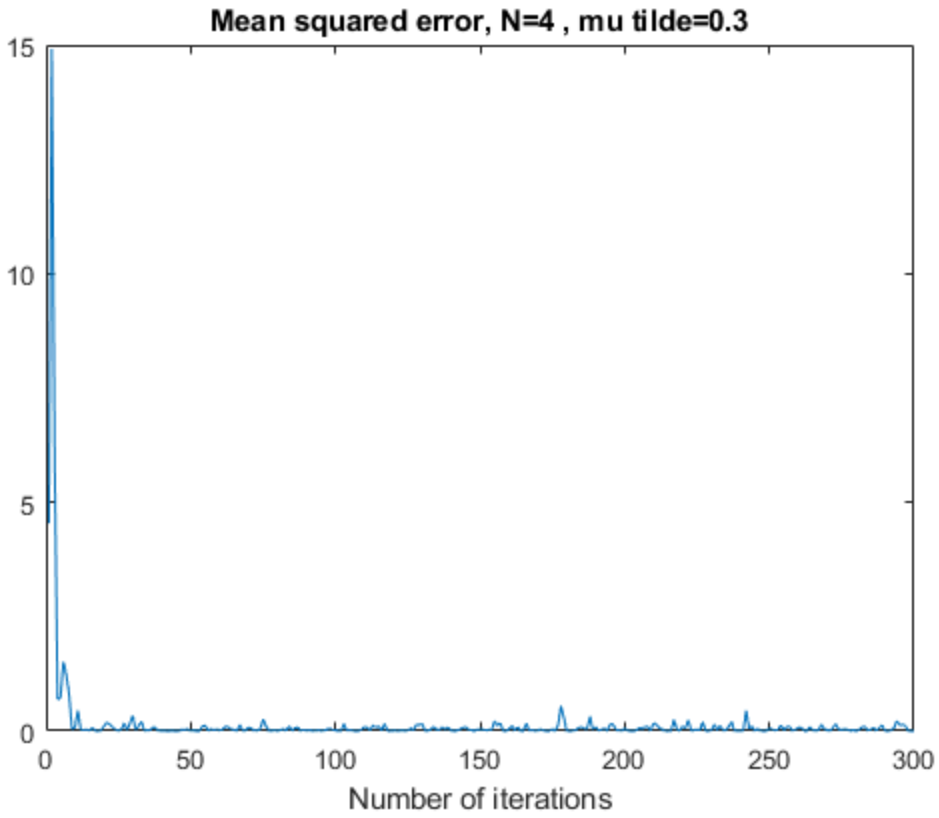
definition

d : desired signal N :length of filter M : length of input signal alpha : mu tilde 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);


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

disp("weights for mu tilde=0.3 and N=4 :");
disp(w');

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

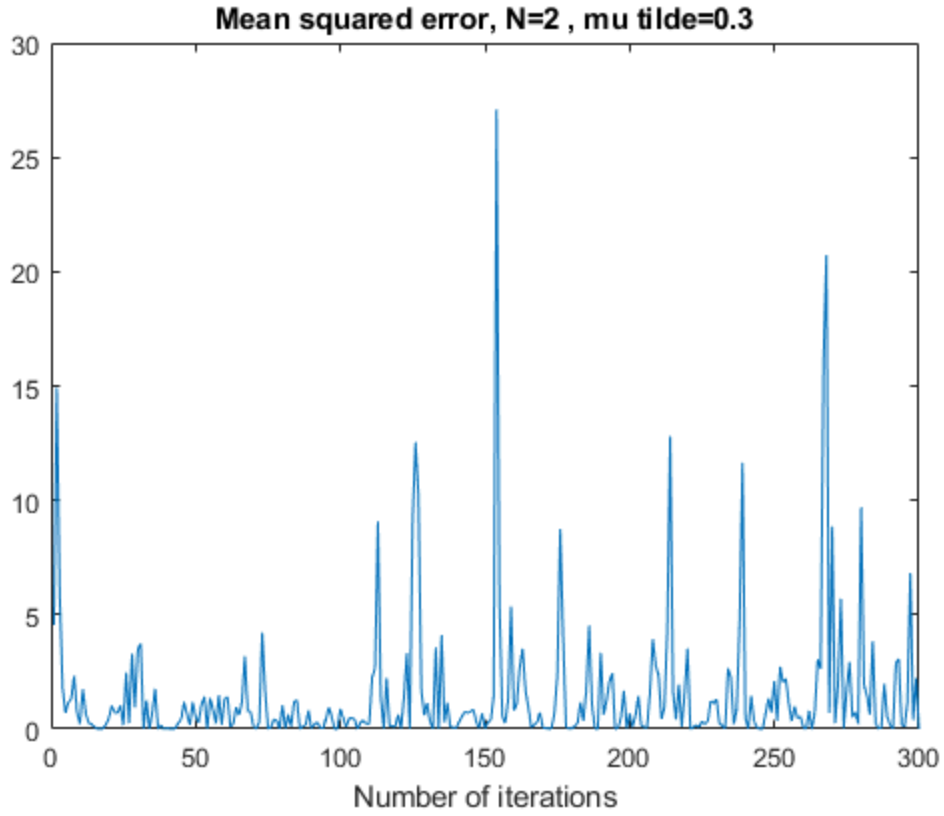


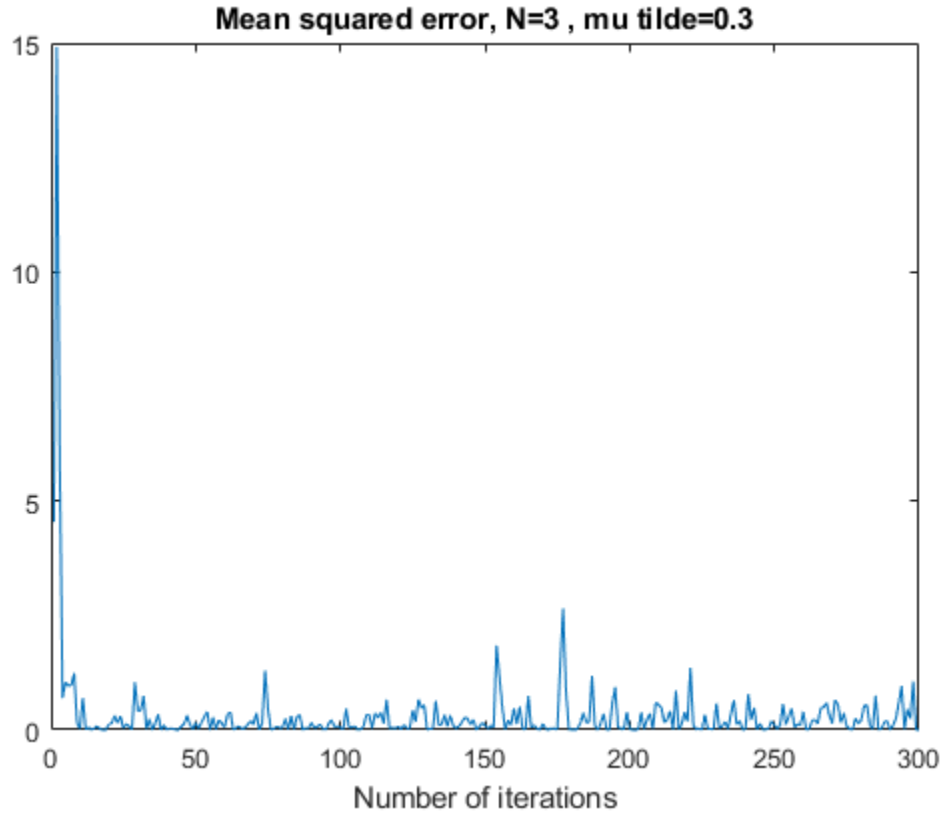
part b

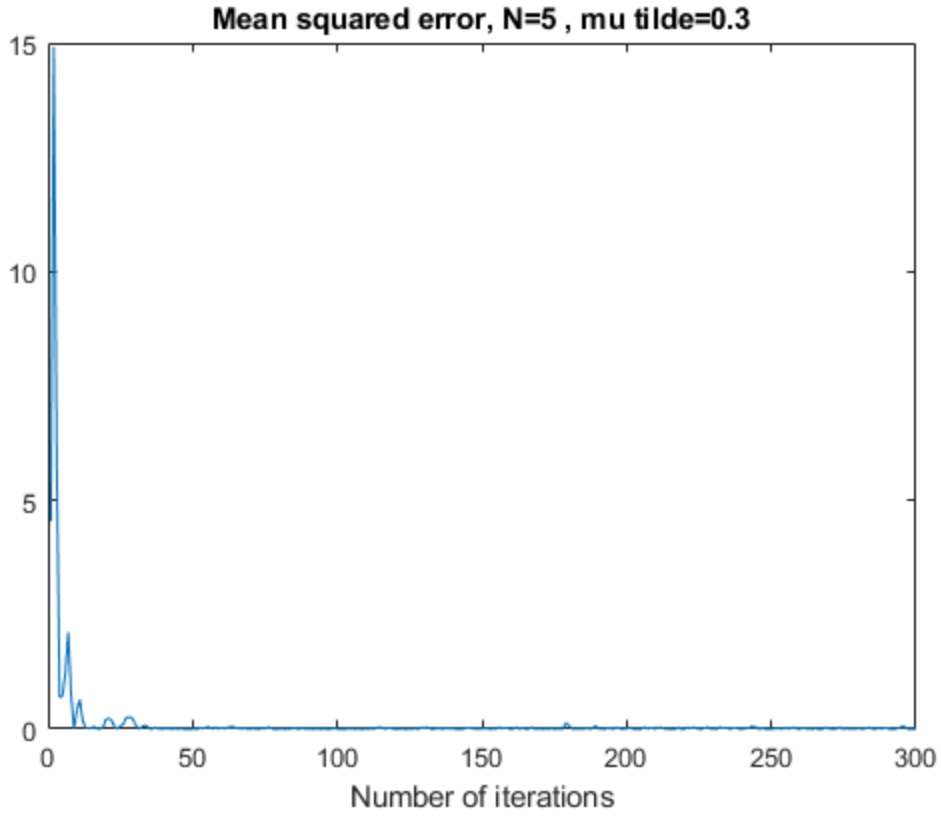
```
N = [2,3,5,7,10];
for i=N
    m_error=zeros(1,M);

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

    disp(['weights for mu tilde=0.3 and N=',num2str(i),':']);
    disp(w');

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

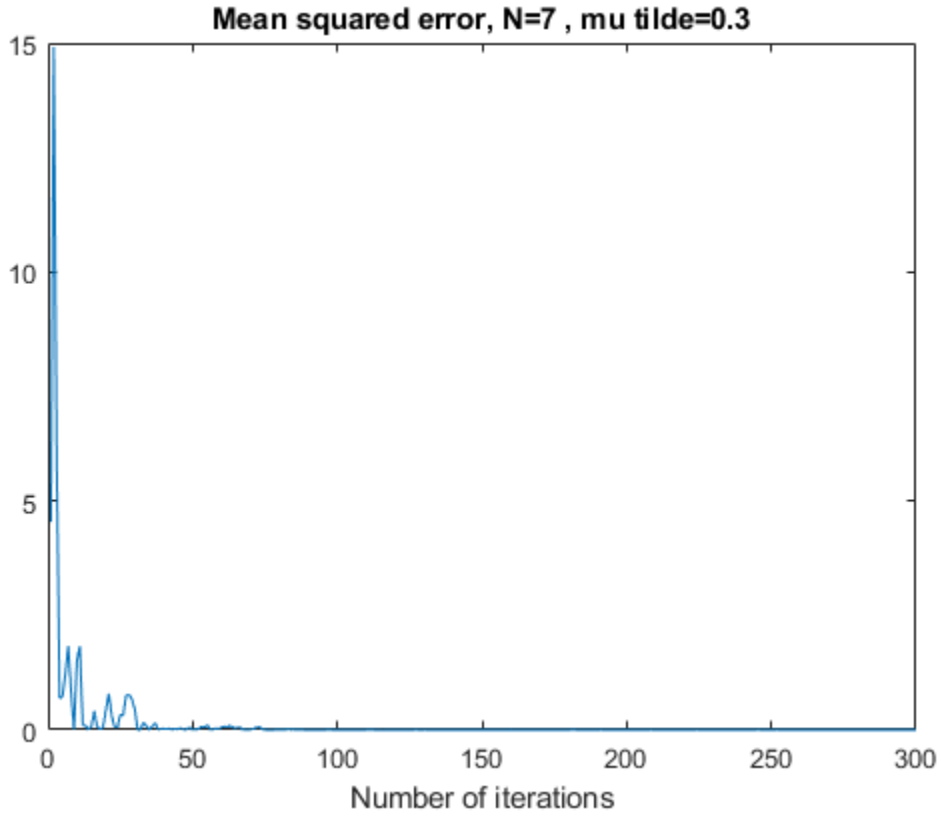
```
weights for mu tilde=0.3 and N=2:
    0.7338
   -1.9832

weights for mu tilde=0.3 and N=3:
```

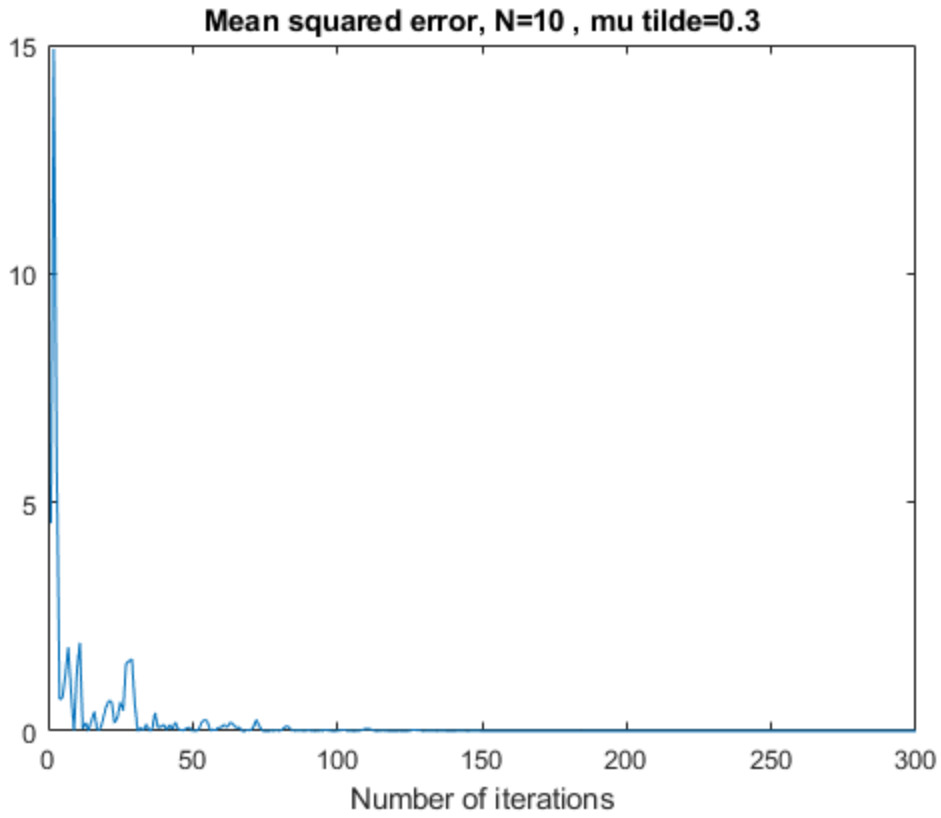
```
    1.3509
   -1.7604
    0.9460

weights for mu tilde=0.3 and N=5:
    0.9992
   -1.4743
    0.7541
   -0.3657
    0.2185

weights for mu tilde=0.3 and N=7:
    1.0049
   -1.4012
    0.7033
   -0.3622
    0.1788
   -0.0837
    0.0405

weights for mu tilde=0.3 and N=10:
    1.0006
   -1.4002
    0.7006
   -0.3504
    0.1753
   -0.0881
    0.0451
   -0.0225
    0.0109
   -0.0049
```


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


Mean squared error, N=3 , mu tilde=0.3
15
10
5
0
0
50
100
150
200
250
300
Number of iterations


Mean squared error, N=5 , mu tilde=0.3
15
10
5
0
0
50
100
150
200
250
300
Number of iterations


Mean squared error, N=7 , mu tilde=0.3
15
10
5
0
0
50
100
150
200
250
300
Number of iterations



part c

```
l = [0.1,0.3,1];
N = 6;
v = randn(1,300);
alpha = 1;
k=5;

for g=l
    m_error=zeros(1,M);
    d_t=d+g*v;


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

    disp(['weights for mu tilde=1 , N=6 and l=',num2str(g), ': ']);
    disp(w')

    figure
    plot(m_error);
    title(['Mean squared error, N=4 , mu tilde=1 and l=', num2str(g),' : ']);
```

```
    xlabel('Number of iterations');
end
disp("if the algorithm not converged must increase the tap of filter ")
```

NLMS algorithms

```
function[w,cost,J_min,J_inf]=NLMS(inputs,d,N,alpha,M)
% e : error
% u_temp : because LMS run when the first sample arrive, we put N-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/(norm(u)^2))*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 tilde=0.3 and N=4 :
    1.1534
   -1.4941
    0.8140
   -0.5037
```

Published with MATLAB® R2022b