# Table of Contents



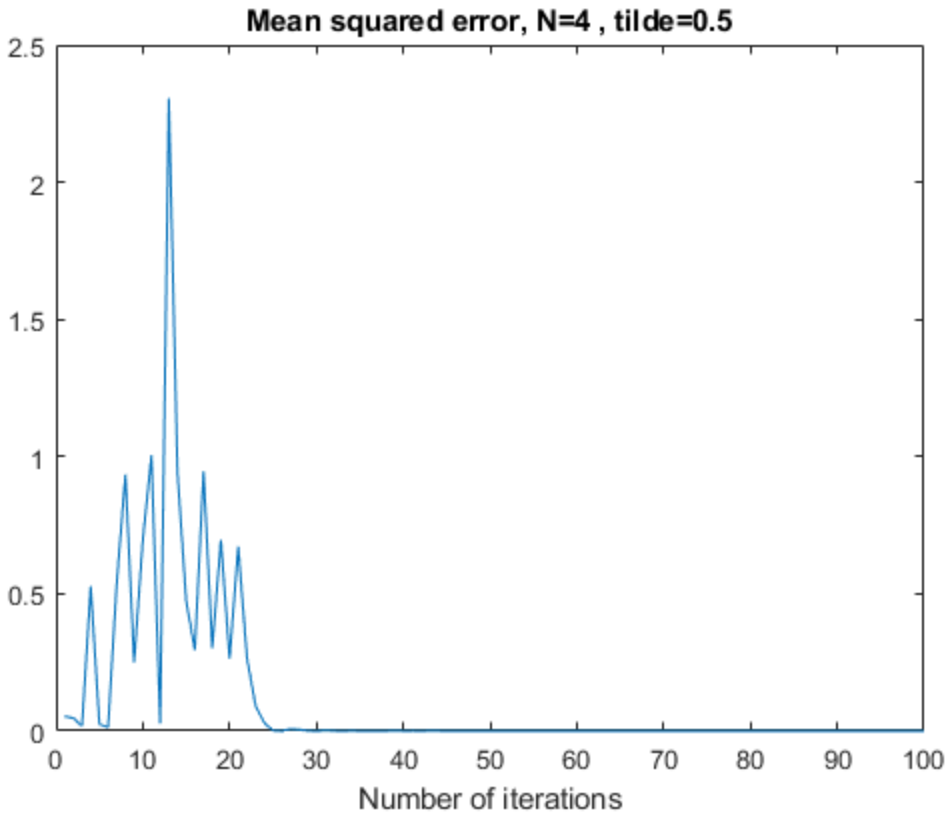
```
clear;
clc;
close all;
```

Mohammad Javad Amin 401211193 Problem 1 , exercise 1

# definition

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

N=4;
alpha=0.5;
[w,~]=NLMS(inputs,d,N,alpha,M);
disp("weights for mu tilde=0.5 and N=4 :");
disp(w');

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

for i=1:k
    [~,cost]=NLMS(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 , tilde=0.5');
xlabel('Number of iterations');
```




# part d

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

```
excess Mean squared error
   5.0502e-10
```

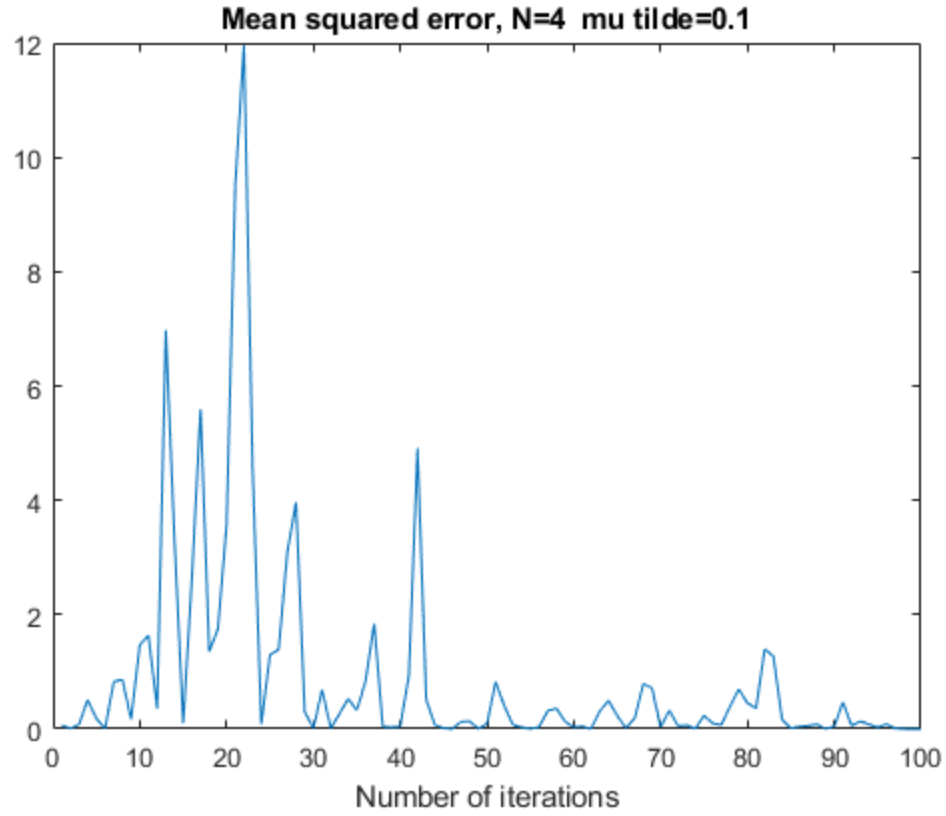
# part e

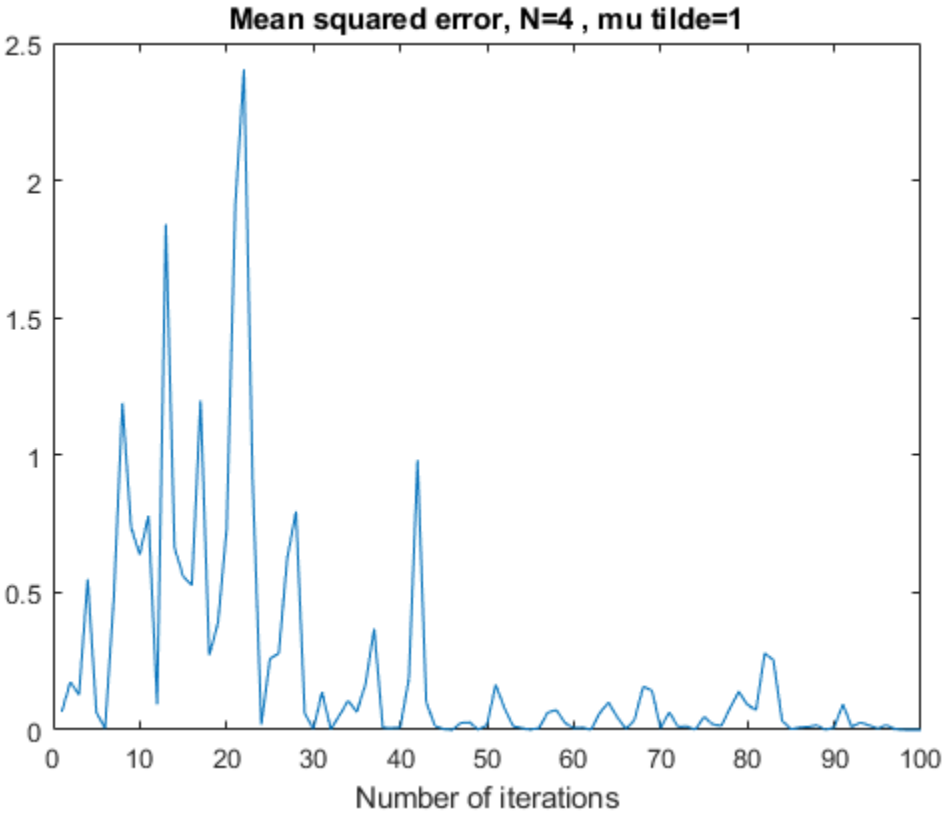
tilde mu = 0.1 and N=4

```
alpha=0.1;

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=0.1 and N=4  :");
disp(w');
figure
```

```
plot(m_error);
title('Mean squared error, N=4  mu tilde=0.1');
xlabel('Number of iterations');

%tilde mu = 1 and N=4
alpha=1;

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=1 and N=4  :");
disp(w');
figure
plot(m_error);
title('Mean squared error, N=4 , mu tilde=1');
xlabel('Number of iterations');
```

```
weights for mu tilde=0.1 and N=4  :
    0.9093
    1.6234
    0.7341
    0.0366

weights for mu tilde=1 and N=4  :
    1.0000
    1.8000
    0.8100
   -0.0000
```


Mean squared error, N=4  mu tilde=0.1
12
10
8
6
4
2
0
0
10
20
30
40
50
60
70
80
90
100
Number of iterations


Mean squared error, N=4 , mu tilde=1
2.5
2
1.5
1
0.5
0
0
10
20
30
40
50
60
70
80
90
100
Number of iterations

# part e

mu tilde = 0.5 and N=2

```
alpha=0.5;
N=2;

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=0.5 and N=2  :");
disp(w');
figure
plot(m_error);
title('Mean squared error, N=2 , mu tilde=0.5');
xlabel('Number of iterations');

% mu tilde = 0.5 and N=3

alpha=0.5;
N=3;

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=0.5 and N=3 :");
disp(w');
figure
plot(m_error);
title('Mean squared error, N=3 , mu tilde=0.5');
xlabel('Number of iterations');
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 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/(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=0.5 and N=4 :
    1.0000
    1.8000
    0.8100
    0.0000
```

*Published with MATLAB® R2022b*