Table of Contents


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

Mohammd Javad Amin 401211193 Problem 1 , exercise 2

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
% v : noise
% l : noise amplitude
% d_t : corrupted desired 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

```
l = 1;
alpha = 0.5;
N = 4;

v = randn(1,100);
d_t=d+l*v;

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

% N=4 and mu=0.5
```

```
disp('because mu is begger than u_max may be LMS algorithm not converged')
[w,~]=LMS(inputs,d_t,N,alpha,M);
disp("weights for mu=0.5 , N=4 and l=1  :");
disp(w');

%N=5 and mu = 0.5
N=5;
% calulate mu max for N=5
alpha_max=2/(3*N*p);
 disp('mu max for N=5 and is :');
 disp(alpha_max);
 disp('because mu is begger than u_max may be LMS algorithm not converged')

[w,~]=LMS(inputs,d_t,N,alpha,M);
disp("weights for mu=0.5 , N=5 and l=1 :");
disp(w');
```

```
mu max for N=4 and is :
    0.1530

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

part b

```
l = 0.1;
N = 4;

v = randn(1,100);
d_t=d+l*v;

% calulate mu max for N=4
 alpha_max=2/(3*N*p);
 disp('mu max for N=4 and is :');
 disp(alpha_max);

% M=4 and mu=0.5 and l=0.1
disp('because mu is begger than u_max may be LMS algorithm not converged')
[w,~]=LMS(inputs,d_t,N,alpha,M);
disp("weights for mu=0.5 , N=4 and l=0.1  :");
disp(w');

%N=5 and mu = 0.5 l=0.1
N=5;
% calulate mu max for N=4
alpha_max=2/(3*N*p);
 disp('mu max for N=5 and is :');
 disp(alpha_max);
 disp('because mu is begger than u_max may be LMS algorithm not converged')

[w,~]=LMS(inputs,d_t,N,alpha,M);
disp("weights for mu=0.5 , N=5 and l=0.1 :");
disp(w');
```

```
disp('if mu smaller than mu_max in the best practice noise of desired signal
 not eliminate and if noise amplitude is lower, the output of system is more
 accurate ')
```

```
mu max for N=4 and is :
    0.1530

because 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=0.5 , N=4 and l=1  :
   1.0e+03 *

   -1.8307
    0.2614
    0.2503
   -1.7754

mu max for N=5 and is :
    0.1224

because mu is begger than u_max may be LMS algorithm not converged
weights for mu=0.5 , N=5 and l=1 :
   1.0e+04 *

   -3.5558
    1.3391
    3.7414
    2.3906
   -7.0320
```

Published with MATLAB® R2022b