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;
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
alpha_int = alpha_max*ones(1,N);
[w,~]=VSLMS(inputs,d_t,N,alpha_int,M,alpha_max);
disp("weights for N=4 and l=1  :");
```

```
disp(w');

%N=5 and
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);

alpha_int = alpha_max*ones(1,N);
[w,~]=VSLMS(inputs,d_t,N,alpha_int,M,alpha_max);
disp("weights for N=5 and l=1 :");
disp(w');
disp(" The VSLMS is more quicker than LMS algorithm ")
```

```
mu max for N=4 and is :
    0.1517
```

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
alpha_int = alpha_max*ones(1,N);
[w,~]=VSLMS(inputs,d_t,N,alpha_int,M,alpha_max);
disp("weights for N=4 and l=0.1  :");
disp(w');

%N=5 and  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);

alpha_int = alpha_max*ones(1,N);
[w,~]=VSLMS(inputs,d_t,N,alpha_int,M,alpha_max);
disp("weights N=5 and l=0.1 :");
disp(w');

disp(' 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 :
```

0.1517

VSLMS algorithms

```
function[w,cost,J_min,J_inf]=VSLMS(inputs,d,N,alpha,M,mu_max)
% 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);
    g = ones(1,N);
    g_past = ones(1,N);
    mu_min=1e-6;
    p=5;
    alpha_past=alpha;

    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;

        for j=1:N
            g(j)=e(i-N+1)*u(j);

            if sign(g(j))==sign(g_past(j))
                alpha(j)=p*alpha_past(j);
            else
                alpha(j)=alpha_past(j)/p;

            end

            if alpha(j)>mu_max
                alpha(j)= mu_max;
            end

            if alpha(j)<mu_min
                alpha(j)= mu_min;
            end

            w(j) =  w(j) + alpha(j)*g(j);

        end

        g_past=g;
        alpha_past=alpha;

    end
    cost=e.^2;
    J_min=min(cost);
    J_inf=sum(cost(M-19:M))/20;
```

```
end
```

```
weights for N=4 and l=1  :
    0.8245
    1.9041
    0.7838
    0.0212

mu max for N=5 and is :
    0.1214

weights for N=5 and l=1 :
    0.8207
    1.8908
    0.8377
    0.0690
   -0.1864

 The VSLMS is more quicker than LMS algorithm
```

Published with MATLAB® R2022b