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 : learning rate e : errors w : weights of filter m_error : mean squared error p : power of input signal alpha_int :alpha initiate

```
a=1;
b=[1,1.8,0.81];          % impulse response
inputs=randn(1,100);
d=filter(b,a,inputs);
M=length(inputs);
```

part a

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

```
mu max for N=4 and is :
    0.2216
```

part b

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

part c

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

for i=1:k
    [~,cost]=VSLMS(inputs,d,N,alpha_int,M,alpha_max);
    m_error=m_error+cost;
end
m_error=m_error/5;
figure
plot(m_error);
title('Mean squared error, N=4 ');
xlabel('Number of iterations');
```

part d

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

```
excess Mean squared error
   2.4818e-04
```

part e

```
N=4

for i=1:k
    [w,cost]=VSLMS(inputs,d,N,alpha_int,M,alpha_max);
    m_error=m_error+cost;
end
m_error=m_error/5;
disp("weights for N=4  :");
disp(w');
figure
plot(m_error);
title('Mean squared error, N=4');
xlabel('Number of iterations');

% N=4

for i=1:k
    [w,cost]=VSLMS(inputs,d,N,alpha_int,M,alpha_max);
    m_error=m_error+cost;
end
m_error=m_error/5;
disp("weights for N=4  :");
disp(w');
```

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

```
weights for N=4  :
    1.0108
    1.7886
    0.8211
   -0.0069

weights for N=4  :
    1.0108
    1.7886
    0.8211
   -0.0069
```

part e

```
N=2

N=2;

for i=1:k
    [w,cost]=VSLMS(inputs,d,N,alpha_int,M,alpha_max);
    m_error=m_error+cost;
end
m_error=m_error/5;
disp("weights N=2  :");
disp(w');
figure
plot(m_error);
title('Mean squared error, N=2 ');
xlabel('Number of iterations');

%  N=3


N=3;

for i=1:k
    [w,cost]=VSLMS(inputs,d,N,alpha_int,M,alpha_max);
    m_error=m_error+cost;
end
m_error=m_error/5;
disp("weights for N=3 :");
disp(w');
figure
plot(m_error);
title('Mean squared error, N=3');
xlabel('Number of iterations');
disp(" The VSLMS is more quicker than LMS algorithm ")
```

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 :

```
    1.0108
    1.7886
    0.8211
   -0.0069
```

Published with MATLAB® R2022b