Table of Contents


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

Mohammad Javad Amin 401211193 Problem 2

Read audio file

```
[y,Fs] = audioread('s1.wav');
s=y(:,1);                           % Extrac 1 channel form audio
s = s - mean(s);                    % Normalize s
s = s/std(s);
```

Noise

```
n0=randn(size(s));
b=rand(1,5);
a=1;
n=filter(b,a,n0);
disp("please wait until all sounds play")
```

```
please wait until all sounds play
```

Noisy signal

```
x=s+n;
sound(x,Fs)
```

FIR Filter with unit variance noise

N :length of filter M : length of input signal alpha : learning rate e : errors w : weights of filter

```
M = length(s);
N = 10;
alpha_max = 1e-3;

alpha_int = alpha_max*ones(1,N);
```

```
[~,e]=VSLMS(n0,x,N,alpha_int,M,alpha_max);
pause(14)
sound(e,Fs)
```

FIR Filter with a noise of variance 10

```
n0 = sqrt(10).*randn(size(s));
n=filter(b,a,n0);

pause(14)
x=s+n;
sound(x,Fs)

[~,e]=VSLMS(n0,x,N,alpha_int,M,alpha_max);
pause(14)
sound(e,Fs)
```

IIR filter

```
a= [1,0.5];
b=[1,-0.9];

n=filter(b,a,n0);


x=s+n;
[w,e]=VSLMS(n0,x,N,alpha_int,M,alpha_max);
pause(14)
sound(e,Fs)

figure(1)
plot(x)
title('noisy signal')
figure(2)
plot(e)
title('out signal');
disp("The diffrence between LMS and VSLMS algorithm is the VSLMS is more
 qucikly converge than LMS but the  ouput quality of LMS I think better than
 VSLMS ")
```

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=7;
    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
```

Published with MATLAB® R2022b