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
% e : errors
% w : weights of filter
% 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;

% N=4 and

[w,~]=RLS(inputs,d_t,N,M);
disp("weights for N=4 and l=1  :");
disp(w');

%N=5 and
N=5;

[w,~]=RLS(inputs,d_t,N,M);
disp("weights for N=5 and l=1 :");
```

```
disp(w');
disp(" The LMS is more quicker than RLS algorithm but the error in RLS is much
 better than LMS ");
```

part b

```
l = 0.1;
N = 4;

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


% M=4 and l=0.1

[w,~]=RLS(inputs,d_t,N,M);
disp("weights for N=4 and l=0.1  :");
disp(w');

%N=5 and  l=0.1
N=5;

[w,~]=RLS(inputs,d_t,N,M);
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 ')
```

RLS algorithms

```
function[w,cost,J_min,J_inf]=RLS(inputs,d,N,M)
% z : error
% N :length of filter
% M : length of input signal

    z=zeros(1,M-N+1);
    w=zeros(1,N);
    lambda=0.6;
    delta= 1e-10;

    p=delta*eye(N);

    for i=N:M-1
        u=inputs(i:-1:i-N+1);
        y=dot(w,u);
        z(i-N+1)=d(i)-y;
        k=(p*u')/(lambda+u*p*u');
        w=w+k'*conj(z(i-N+1));
        p=(p -k*conj(u)*p)/lambda;

    end
    cost=z.^2;
```

```
    J_min=min(z);
    J_inf=sum(z(M-N-19:M-N))/20;

end
```

```
weights for N=4 and l=1  :
    0.6941
    2.0778
   -0.3311
   -0.1451

weights for N=5 and l=1 :
    0.5566
    1.9743
   -0.8008
   -0.6804
   -0.7823
```

The LMS is more quicker than RLS algorithm but the error in RLS is much better than LMS

Published with MATLAB® R2022b