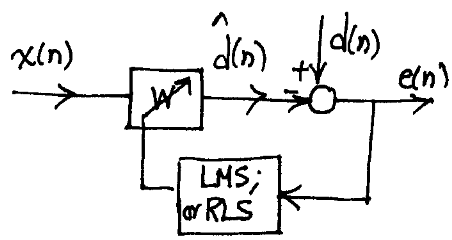
```
% EE254                                                 Prof. Essam Marouf
%
% Comparison of the LMS and RLS Algorithm: Simple Correlation Canceller
% ---------------------------------------------------------------------
clear, close all
%
% Initialize constants
MM=1;                          % MM is the number of taps (M+1 in notes)
Lmax=20;                       % Number of realizations to average
Kmax=500;                      % Length of Data Sequence
Kmin=MM-1;
sigma_sq=0.1;
rho =0.95;                           % Exponential (memory) factor
mu=0.1;                              % LMS step length
w0_rls= zeros(MM,1);                 % weights initialization
w0_lms = zeros(MM,1);
%
%
% Iterate over Different Realizations
for L=1:Lmax,
%
%Initialize R_inv:
eta=1E6 ;                            % Arbitrary large positive constant
R_inv = eta*eye(MM);                 % eye(MM) forms MMxMM unit matrix
%
% Initialize the weigth vectors
W=zeros(MM,1); W_1=zeros(MM,1);
%
% Generate primary and reference signals
x=sqrt(sigma_sq)*randn(1,Kmax);
d= -0.8*x + sqrt(sigma_sq)*randn(1,Kmax);
x=x'; d=d';                          % Column vectors
%
% RLS Iterations
for k=Kmin+1:Kmax,
  X=x(k:-1:k-MM+1);                  %  Form new input vector
  y(k) = W'*X;                       %  A priori output y(k)
  e(k) = d(k) - y(k);                %  A priori error e(k)
            %
  % RLS Iterations
  Z=R_inv*X;                      %  Filtered reference vector
  q=X'*Z;                         %  Normalized error power q
  v= 1/(rho+q);                   %  Gain constant
  Z_t = v*Z;                      %  Normalized filtered reference
vector
  W= W + e(k)*Z_t;                %  Update optimal weight vector
  R_inv = (R_inv - Z_t*Z')/rho;   %  Update R_inv
  w0(k)=W(1);                     %  Save w0 for plotting
```



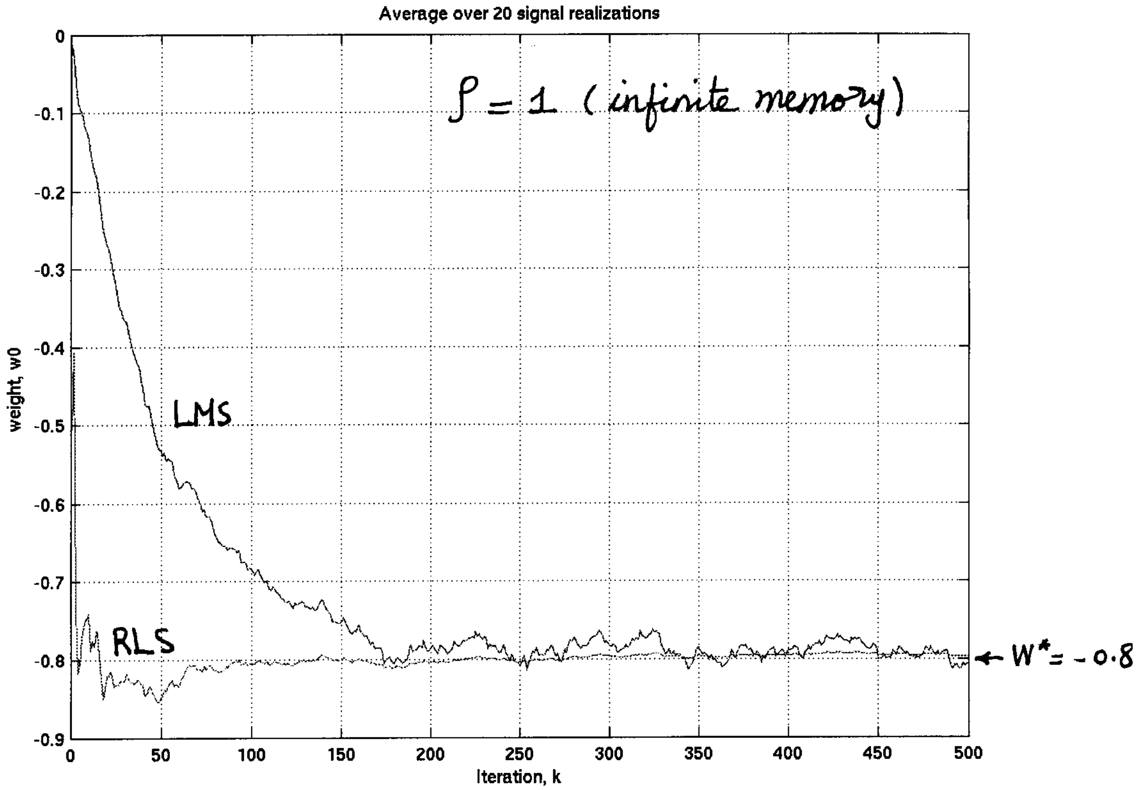
$d(n) = -0.8\,x(n) + v(n)$

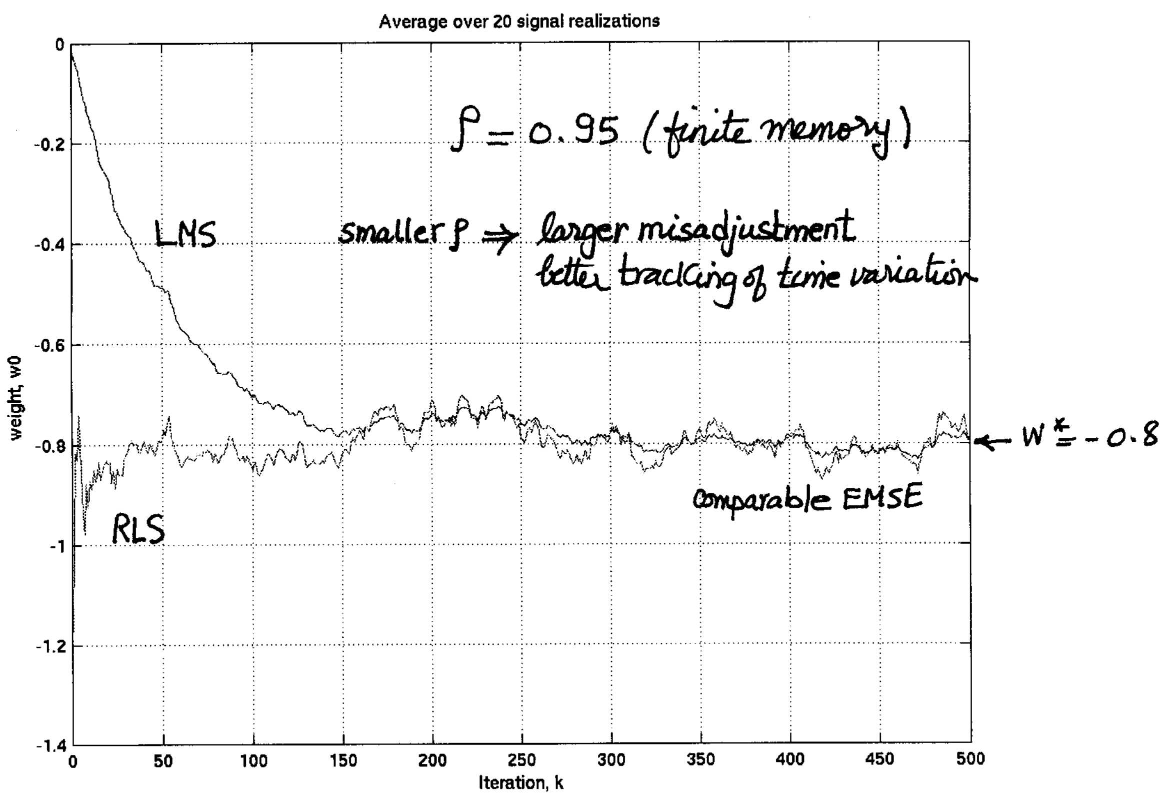
$x(n) : N(0, 0.1)$

$v(n) : N(0, 0.1)$

x, v *uncorrelated*

```
  %
  % LMS Iterations
  y(k) = W_1'*X;                    %  Filter output y(k)
  e(k) = d(k) - y(k);               %  Error signal e(k)
  W_1= W_1 + 2*mu*e(k)*X ;          %  LMS weight update
  w0_1(k)=W_1(1);
end
%
w0_rls = w0_rls + w0;               % Accumulate sum
w0_lms = w0_lms + w0_1;
end
%
w0_rls=w0_rls/Lmax;                 % averaged weights
w0_lms=w0_lms/Lmax;
axis([0 Kmax -1 0]);
plot(Kmin+1:Kmax,w0_lms(Kmin+1:Kmax),Kmin+1:Kmax,w0_rls(Kmin+1:Kmax)),
grid
xlabel('Iteration, k'),ylabel('weight, w0')
title('Average over 20 signal realizations')
axis;
```


Average over 20 signal realizations
$\rho = 1$ (infinite memory)
LMS
RLS
$\leftarrow W^* = -0.8$
weight, w0
Iteration, k


Average over 20 signal realizations
$\rho = 0.95$ (finite memory)
LMS
smaller $\rho \Rightarrow$ larger misadjustment
better tracking of time variation
RLS
comparable EMSE
$\leftarrow W^* = -0.8$
weight, w0
Iteration, k