```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% EE7000-3 Adaptive Filter Theory
% Demonstration of Wiener filter, LMS filter, Steepest-descent algorithm
% Dr. Hsiao-Chun Wu
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear;
clc;

N = 10000;      % the length of the observation sequence
M = 2;          % the filter length

v = randn(1, N);          % white process as the AR excitation
a = poly(sign(randn(1,M)) .* rand(1,M));      % coefficients of AR process

% The input sequence
u = filter(1, a, v);

% The desired response
d = v;

rf = xcorr(u,M,'biased');
rv = rf(M+1:2*M+1);

% The correlation matrix of the input
R = toeplitz(rv);

pf = xcorr(d,u,M,'biased');

% The cross-correlation vector between the input and the desired response
pv = pf(M+1:2*M+1).';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The optimal tap weight vector for Wiener filter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

wopt = inv(R) * pv;

% Selection of a stable step size mu

[V,D]= eig(R);
lambda_max = max(diag(D));
mu = 0.9 * 2/lambda_max;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The steepest descent learning
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

wsd = randn(M+1, 1);                % initial weight vector for steepest descent
total_iteration_number = 100;       % total iteration number

for i = 1: total_iteration_number

    wsd = wsd + mu * (pv - R * wsd);

end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The LMS learning
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

wlms = randn(M+1, 1);               % initial weight vector for LMS
uv = zeros(M+1,1);                  % initial input vector

mu = 0.1 * 2/lambda_max;            % step size mu for LMS

for n = 1: N;

   uv(2:M+1) =uv(1:M);
   uv(1) = u(n);

   y = wlms'*uv;
   e = d(n) - y;
   wlms = wlms + mu * uv * conj(e);

end;
```