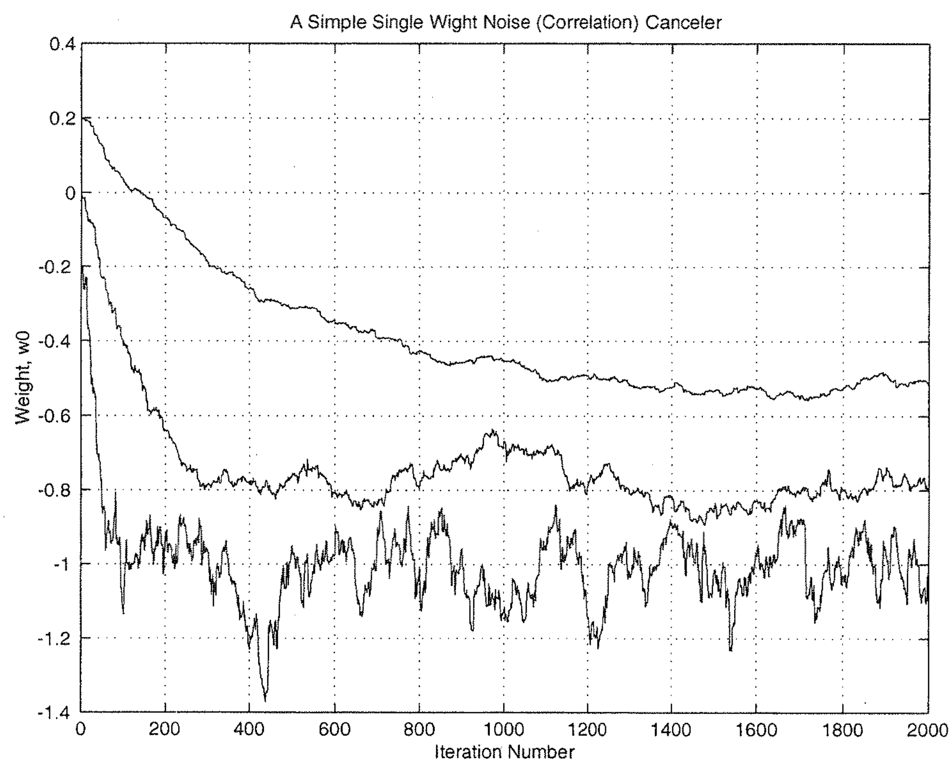
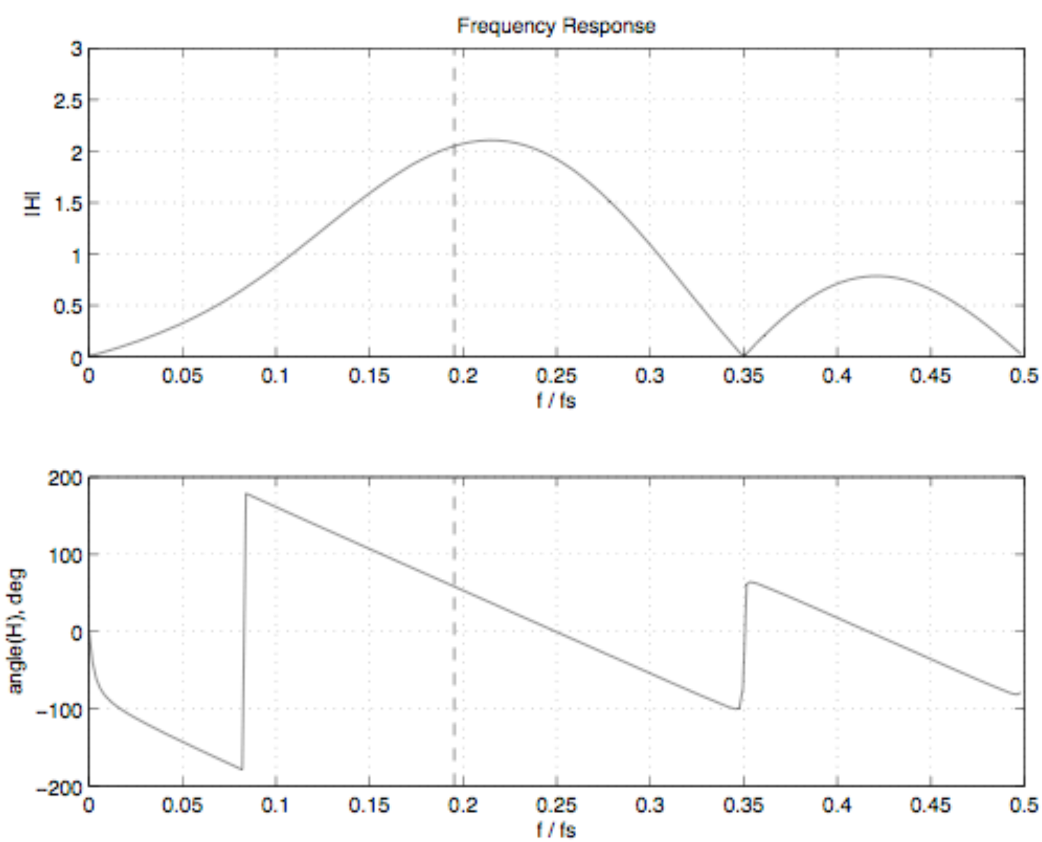
```
% EE 254                                                          Essam Marouf
% The LMS Algorithm: Example Adaptive Filters                     Matlab Handout #5
% ----------------------------------------------------------------------------------

% Example 1. Simple Noise (Correlation) Canceler
% ------------------------------------------------------
%
% the primary (desired) signal d(k)=-0.8*x(k)+v(k), where both x(k)
% and v(k) are samples of white noise of zero mean and variance
% = 0.1. The reference signal is xk.
%
clear, close all
LK=2000;
K=[1:LK];                       % Sample indices
sigma=sqrt(0.1);                % noise standard deviation

shift =[0.2  0.0  -0.2]; % plot shift

indx=0;
for mu=[0.01 0.03 0.1]
  indx=indx+1;

  x=sigma*randn(size(K));   % generates a vector of 200 samples of WGN of zero
                            % mean and sigma^2 variance
  v=sigma*randn(size(K));   % a new call to 'rand' generates a new random vector
  d= -0.8*x+v;
%
                            % The average power of x is its variance =0.1;
                            % thus 0 < mu < 10; here mu << the upper bound so as
                            % to achieve reasonable statistical fluctuations.
  w0(1)=0;                  % Filter has only one weight;
                            % initialized to an arbitrary value = 0.
%
% Iterate
%
  for k=1:1:LK
        y(k)=w0(k)*x(k);                  % Compute the filter output
        e(k)=d(k)-y(k);                   % Compare to the desired signal and
                                          % compute the error
        w0(k+1)=w0(k)+2*mu*e(k)*x(k);     % Update the filter weight
  end
  plot( K,w0(1:LK) + shift(indx) ), grid, hold on
  w0=[];
end,
title('A Simple Single Wight Noise (Correlation) Canceler'),
```

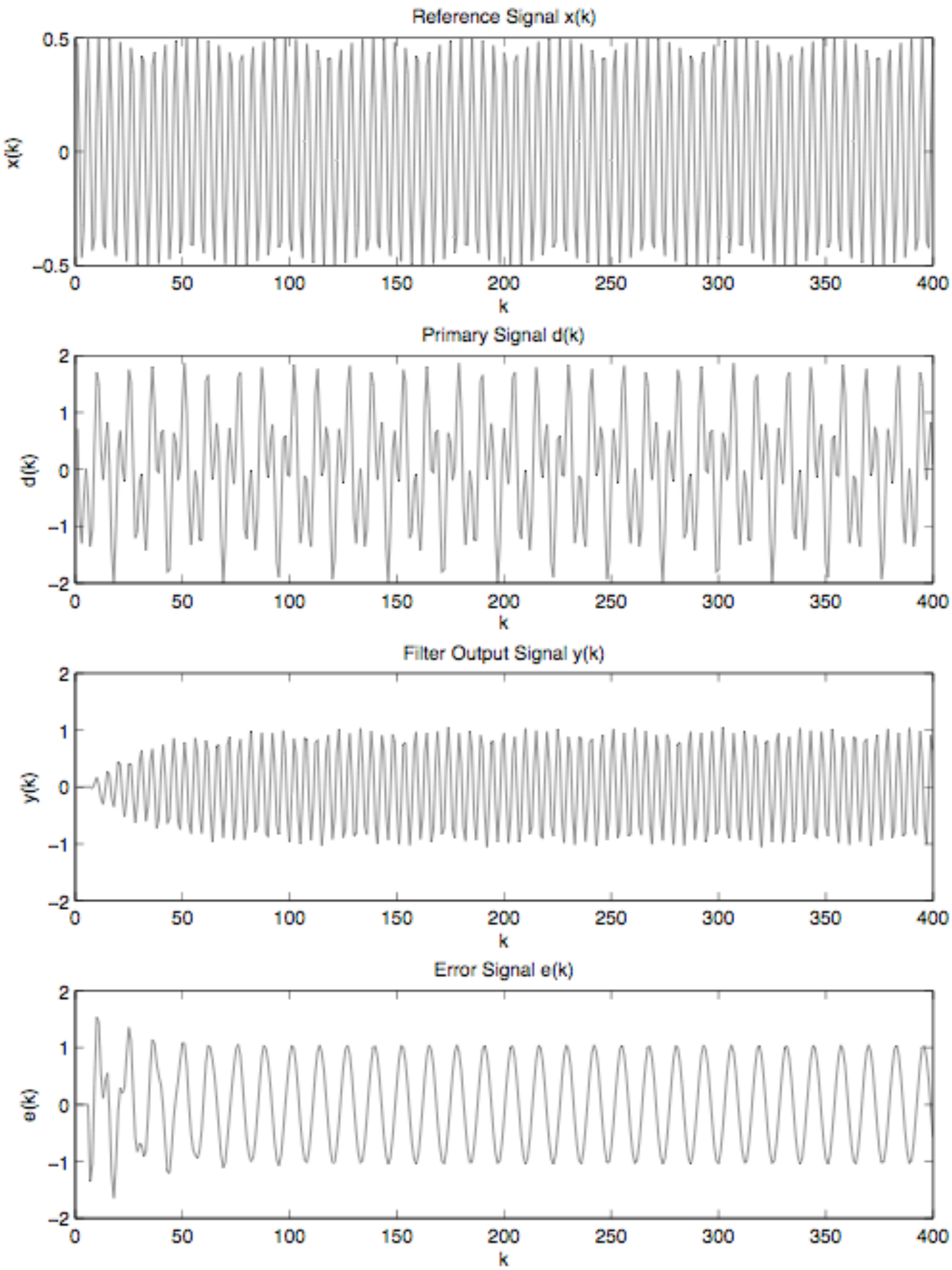
```
xlabel('Iteration Number'), ylabel('Weight, w0'),
hold off
% Exponential convergence towards the theoretical limit w*=-0.8
% is evident.  Also evident is the persistent random fluctuations
% caused by the noisy gradient estimate.  %
% This correlation canceler removes the component of d(k) that is
% correlated with x(k). In principle, this leaves the noise
% component v(k) as the only component of the error signal e(k). To
% check let's plot both v(k) and e(k) for k> 1000
%
% The following conclusions follow:
%
% 1) The larger mu is, the faster the convergenve is. The
% theoretical wight time constants tau=1/(2*mu*lambda) for the three cases
% mu = 0.1, 0.03, and 0.01 are 50, 167, and 500 samples, respectively.
% This is evident in the figure.
%
% 2) The larger mu is, the larger the statistical fluctuations in w0.
% For example, the standard deviation of w0, computed for k > 1000
% (StdZ above) is .0923,.0386 ,and .0283 , for mu=0.1, 0.03, and
% 0.01, repectively. Thus a better statistical behavior is usually
% obtained at the expense of slower adaptation.
%
```



```
% Example 2. Interference Canceler; Deterministic Signals
% -----------------------------------------------------
%
clear , close all
fs=128;          % sampling frequency in Hz
f1=25;           % interference frequency
f2=10;           % frequency of interest
LK=1000;

x= 0.5*cos(2*pi*(f1/fs)*[0:LK-1] + 18*pi/180);
d= cos(2*pi*(f2/fs)*[0:LK-1] + 63*pi/180);
d= d + cos(2*pi*(f1/fs)*[0:LK-1] + 75*pi/180);
%
mu= 0.05;
L=6;                      % 7 weights are assumed (filter of order=6)
W=zeros(1,L+1)';          % Initialize 7 weights to zero;
                          % We now handel W as a vector\
% Iterate
for k=L+1:1:LK
X=x(k:-1:k-L)';           % X is also a vector containing x(k), x(k-1),
                          % ......, X(K-L)
y(k)=W'*X;                % As done in the theoretical formulation
e(k)=d(k)-y(k);           % Remember e is a scalar
W=W+2*mu*e(k)*X ;         % Update weight vector
end

% Plots of relevant results
% ------------------------
%
NP1=1;
NP2=400;

subplot(4,1,1), plot([NP1:NP2],x(NP1:NP2)),
title('Reference Signal x(k)'),xlabel('k'),ylabel('x(k)'),

subplot(4,1,2), plot([NP1:NP2],d(NP1:NP2)),
title('Primary Signal d(k)'),xlabel('k'),ylabel('d(k)'),

subplot(4,1,3), plot([NP1:NP2],y(NP1:NP2)),
title('Filter Output Signal y(k)'),xlabel('k'),ylabel('y(k)'),

subplot(4,1,4), plot([NP1:NP2],e(NP1:NP2)),
title('Error Signal e(k)'),xlabel('k'),ylabel('e(k)'),
subplot,
```

```
% Note that the filter prouces a copy of the the interference
% signal of frequency f1 = 25 Hz . When subtracted from d, the
% output erros signal is a sinusoid of frequency f = 10 Hz, free
% of the interference.
%
b=W';
a=1;
[H,omega]=freqz(b,a,256);
%
tmp1=[f1/fs f1/fs],
tmp3=[0 3];
tmp4=[-200 200];
figure
subplot(2,1,1),
plot(omega/(2*pi),abs(H),tmp1,tmp3,'--'),
grid, title('Frequency Response'),xlabel('f / fs'),ylabel('|H|'),
subplot(2,1,2),
plot(omega/(2*pi),angle(H)*180/pi,tmp1,tmp4,'--'),
grid, xlabel('f / fs'),ylabel('angle(H), deg'),
subplot
```




Reference Signal x(k)
x(k)
0.5
0
−0.5
0 50 100 150 200 250 300 350 400
k
Primary Signal d(k)
d(k)
2
1
0
−1
−2
0 50 100 150 200 250 300 350 400
k
Filter Output Signal y(k)
y(k)
2
1
0
−1
−2
0 50 100 150 200 250 300 350 400
k
Error Signal e(k)
e(k)
2
1
0
−1
−2
0 50 100 150 200 250 300 350 400
k