

1) (see attached)

2)

(a)

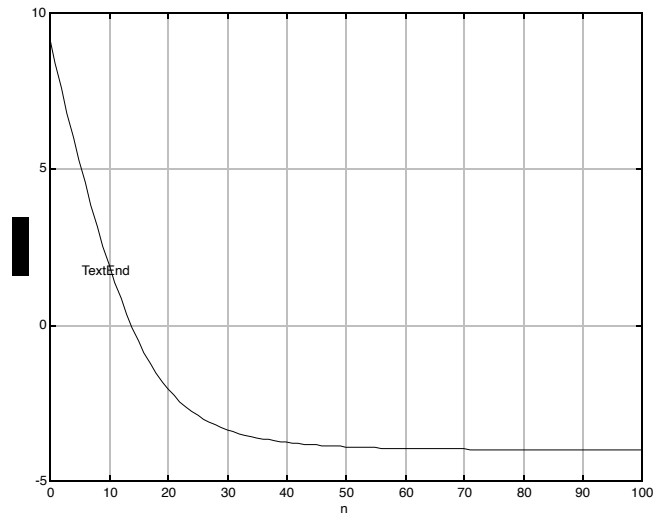
$$J(n) = \sigma_d^2 - \mathbf{w}^H(n)\mathbf{p} - \mathbf{w}(n)\mathbf{p}^H + \mathbf{w}^H(n)\mathbf{R}\mathbf{w}(n)$$
$$= 0.5 + w_1^2 + w_2^2 + 0.8090w_1w_2 - 0.5878w_2$$

(b)

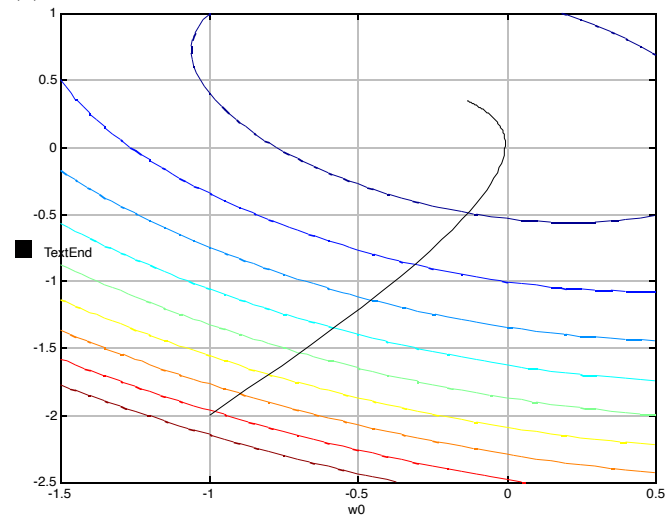
[evecs, evals] =

0.7071	0.7071	0.5955	0
-0.7071	0.7071	0	1.4045

(c)



(d)



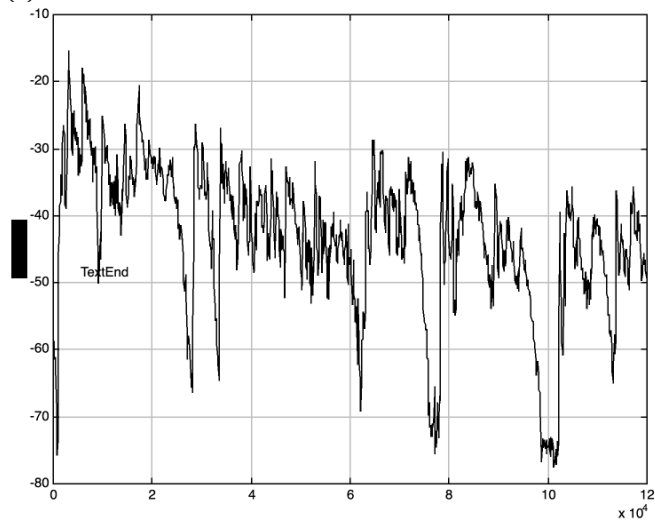
(e) The fast mode associated with the largest eigenvalue moves in a direction aligned with the associated eigenvector. The slow mode associated with the smallest eigenvalue moves in a direction aligned with the associated eigenvector. We see the “turn” in the trajectory as the fast mode dies off and we are left with the slow mode.

3)

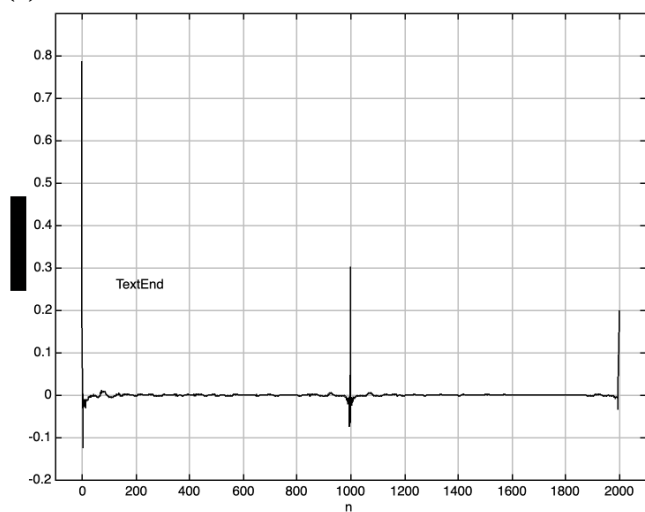
(a) $M = 0.25 * f_s = 2000$

(b) We clearly here the return signal diminishing in volume indicating proper echo cancellation.

(c)



(d)



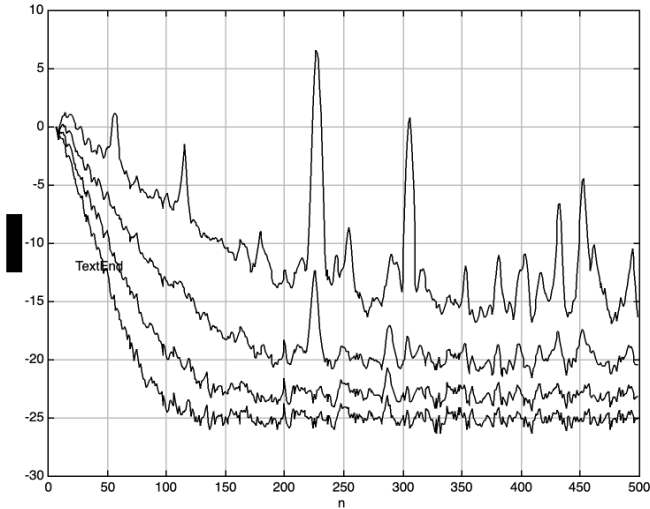
(e)

Listening to $u(n)$ we hear two distinct echoes at $1/8$ second later and $1/4$ second later. This is what we would expect with a room impulse response as in (d).

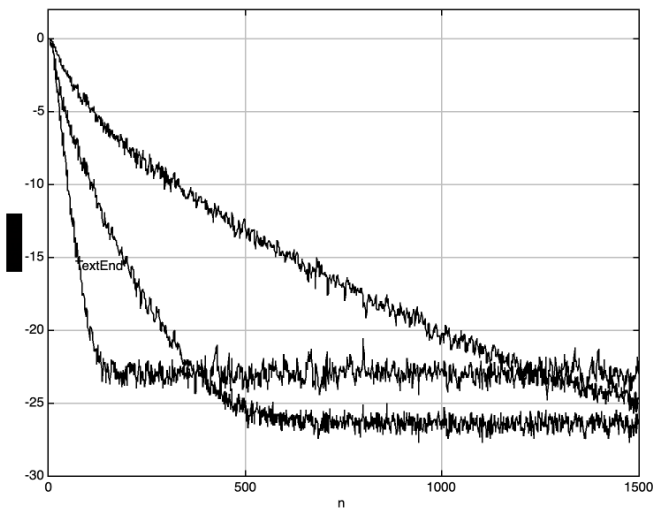
(4)

(a) See attached

(b)



(c)



(d)

δ	Steady-State MSE	δ	Steady-State MSE
0	2.05380268408490	5	0.00519335134051
1	2.05374908942175	6	0.00503746554781
2	0.19613403982023	7	0.00501291413052
3	0.02273353542131	8	0.00504913830660
4	0.00669123478452	9	0.00518379680863

It appears that with $\delta = 7$, we have the smallest steady-state MSE.

```
% Exam #1 Fall 2002
% Haykin, 4th Ed., Prob. 5.22
```

```
%-----
% Prob. 1
%-----
r = [18.1429;-8.5714;8.0440];
R = toeplitz(r);
```

```

p = [-2.1362;10.2858;5.1429];
w_opt = inv(R)*p

% LMS implementation
L = 5000;
M = 3;
sigma_s2 = 10;
sigma_v2 = 1.0;
w_init = zeros(M,1);
mu = 0.001;
randn('state',0);
b1 = [1]';
a1 = [1;-0.5];
b2 = [1]';
a2 = [1;0.8];

s = sqrt(sigma_s2)*randn(L,1);
v = sqrt(sigma_v2)*randn(L,1);
d = filter(b1,a1,s);
x = filter(b2,a2,d);
r = correlation(x,x,4);
p = correlation([0;d(1:L-1)],x,4);
u = x + v;
[e,w] = lms(w_init,u,[0;d(1:L-1)],mu,0);

%-----
% Prob. 2
%-----
%---
% b
%---
r = [1;0.4045];
R = toeplitz(r);
p = [0;0.2939];
[evects,evals] = eig(R);
l_max = max(max(evals));

%---
% c
%---
w_init = [-1;-2];
mu = 0.1/l_max;
sigma_d2 = 0.5;
N = 100;
[J,w] = steepest_descent(w_init,R,p,mu,sigma_d2,N,1);
MSE_plot(J);

%---
% d
%---
contours=[0.1:1:10];
grid_data = [-1.5 0.5 0.05;-2.5 1.0 0.05];
trajectory(sigma_d2,r,p,w,contours,grid_data)

%-----

```

```

% Prob. 3
%-----

% Create data files
fs = 8000;
% load a2_512ll.mat; % from Bell Labs
h = [1;zeros(998,1);0.5;zeros(999,1);0.25]; % fake echo
h = h ./ sqrt(h'*h); % noise gain to one
u=wavread('s7_8k.wav'); % from TIMIT
u = u(1:15*fs); % trim to 15 seconds
u = u ./ max(abs(u));
d = filter(h,1,u);
d = d ./ max(abs(d));
save Exam1.mat u d fs
clear

%---
% a
%---
load Exam1.mat
sound(u, fs);
sound(d, fs);
M = 0.25*fs;

%---
% b
%---
w_init = zeros(M,1);
mu = 0.2 * (2/(3*M*cov(u)));
[e,w] = lms(w_init,u,d,mu,0);
sound(e, fs);

%---
% c
%---
e2 = e.^2;
e2 = filter(0.01*ones(100,1),1,e2);
mse_plot(e2,1, length(e2),1,100);

%---
% d
%---
plot([0:M-1],w, 'k');
ylabel('Estimate of h(n)')
xlabel('n')
grid
axis([-100 2100 -0.2 0.9]);

%-----

% Prob. 4
%-----

%---
% b
%---
% User params

```

```

W = [2.9;3.1;3.3;3.5];
L = 500;
M = 11;
num_runs = 200;
sigma_v2 = 0.001;
delta = 7; % delay
mu = 0.075;
J = zeros(L,4);
for k = 1:4
    k
    rand('state',0);
    randn('state',0);
    sum_e2 = zeros(L,1);
    h = zeros(4,1);
    for n = 1:3
        h(n+1) = 1/2*(1+cos(2*pi/W(k)*(n-2)));
    end
    for run = 1:num_runs
        w_init = zeros(M,1);
        x = sign(rand(L,1)-0.5);
        y = filter(h,1,x);
        v = sqrt(sigma_v2)*randn(L,1);
        u = y + v;
        d = [zeros(delta,1);x(1:L-delta)]; % d = u(n-delta)
        [e,w] = lms(w_init,u,d,mu,0);
        sum_e2 = sum_e2 + e.^2;
    end;
    J(:,k) = sum_e2 / num_runs;
end;
MSE_plot(J(:,1));
hold on
MSE_plot(J(:,2));
MSE_plot(J(:,3));
MSE_plot(J(:,4));
hold off
grid;

%---
% c
%---
% User params
W = 3.1;
L = 1500;
M = 11;
num_runs = 200;
sigma_v2 = 0.001;
delta = 7; % delay
mu = [0.075;0.025;0.0075];
J = zeros(L,3);
for k = 1:3
    k
    rand('state',0);
    randn('state',0);
    sum_e2 = zeros(L,1);
    h = zeros(4,1);
    for n = 1:3

```

```

        h(n+1) = 1/2*(1+cos(2*pi/W*(n-2)));
    end
    for run = 1:num_runs
        w_init = zeros(M,1);
        x = sign(rand(L,1)-0.5);
        y = filter(h,1,x);
        v = sqrt(sigma_v2)*randn(L,1);
        u = y + v;
        d = [zeros(delta,1);x(1:L-delta)]; % d = u(n-delta)
        [e,w] = lms(w_init,u,d,mu(k),0);
        sum_e2 = sum_e2 + e.^2;
    end;
    J(:,k) = sum_e2 / num_runs;
end;
MSE_plot(J(:,1));
hold on
MSE_plot(J(:,2));
MSE_plot(J(:,3));
hold off
grid;

%---
% d
%---
% Evaluate various delays
% User params
W = 3.1;
L = 500;
M = 11;
num_runs = 200;
sigma_v2 = 0.001;
delta = [0:9]'; % delays
mu = 0.075;
MSE_ss = zeros(10,1);
h = zeros(4,1);
for n = 1:3
    h(n+1) = 1/2*(1+cos(2*pi/W*(n-2)));
end
for k = 1:10
    k
    rand('state',0);
    randn('state',0);
    sum_e2 = zeros(L,1);
    for run = 1:num_runs
        w_init = zeros(M,1);
        x = sign(rand(L,1)-0.5);
        y = filter(h,1,x);
        v = sqrt(sigma_v2)*randn(L,1);
        u = y + v;
        d = [zeros(delta(k),1);x(1:L-delta(k))]; % d = u(n-delta)
        [e,w] = lms(w_init,u,d,mu,0);
        sum_e2 = sum_e2 + e.^2;
    end;
    J = sum_e2 / num_runs;
    MSE_ss(k) = sum(J(L-99:L))/100;
end;

```