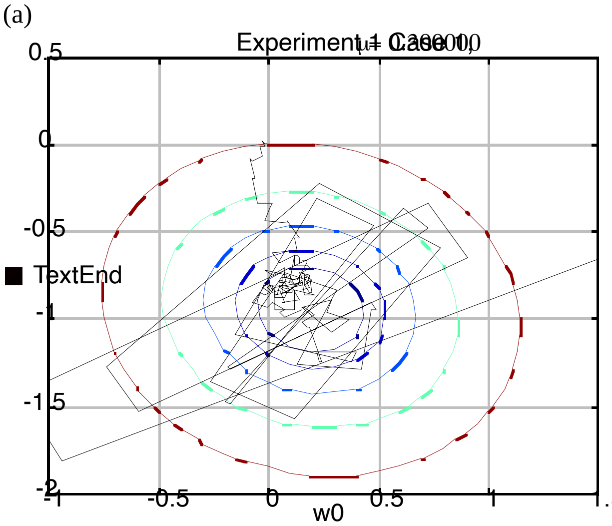
Solution #4 LMS Adaptive Filters

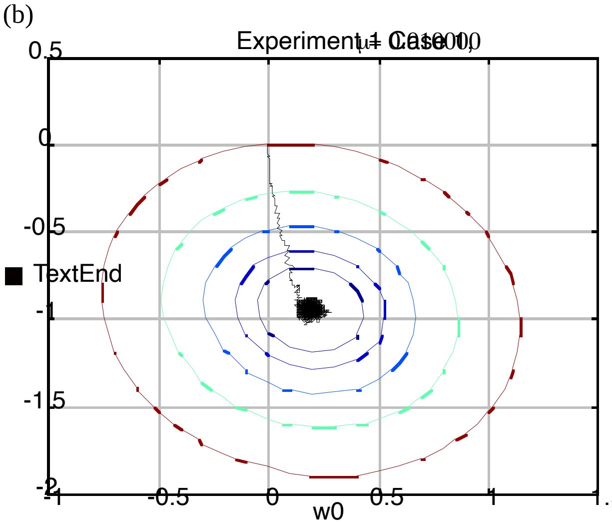
1)

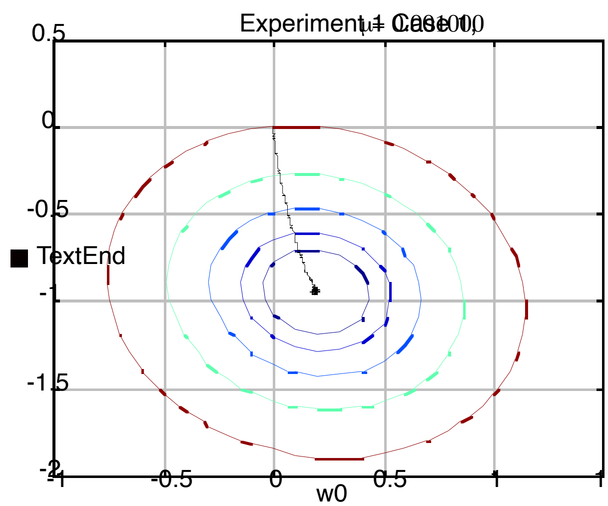
(a)

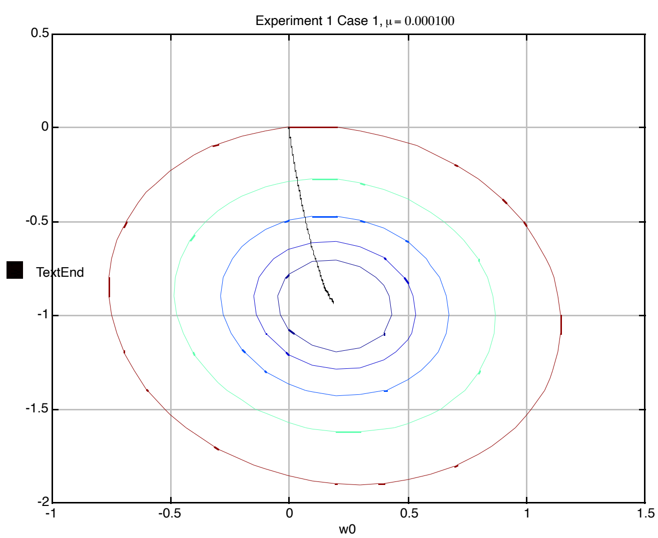


Clearly our weight vector is off in the weeds even though $\mu < \dfrac{2}{3tr(R)} = \dfrac{2}{3(1.1+0.9)}$. Our bounds were based on many assumptions obviously not entirely true in this experiment.

(b)





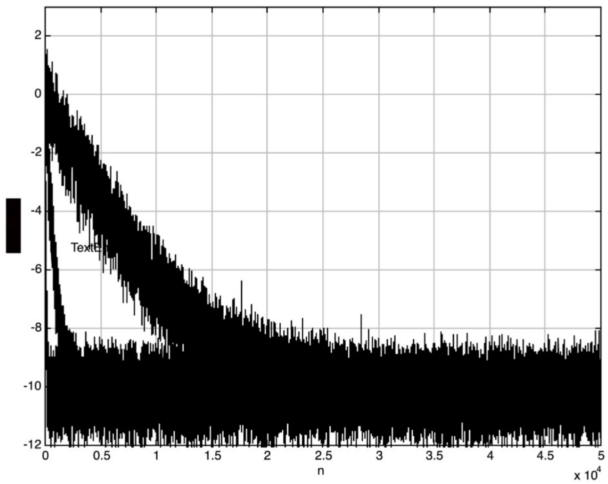


Wiener filter	μ = 0.01	μ = 0.001	μ = 0.0001

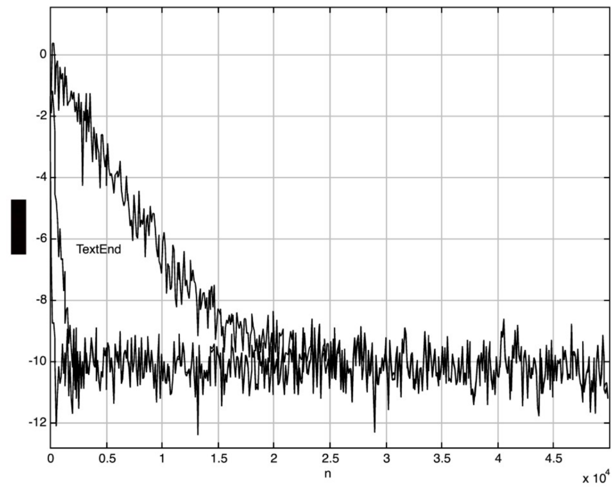
`wo =`
`0.1950`
`-0.9500`
	`w(:,50001) =`
	`0.1453`
	`-0.9416`
		`w(:,50001) =`
		`0.1913`
		`-0.9549`
			`w(:,50001) =`
			`0.1872`
			`-0.9445`

- As μ decreases we are “closer” to the Wiener filter.
- As μ decreases the amount we “wander at the bottom of the error surface” or J_{ex} decreases.

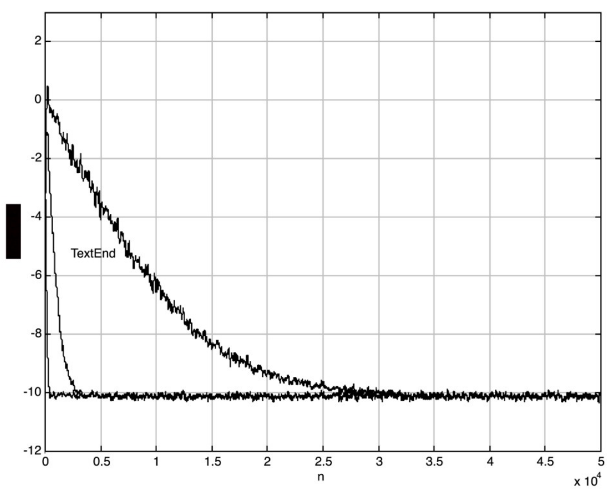
(c)



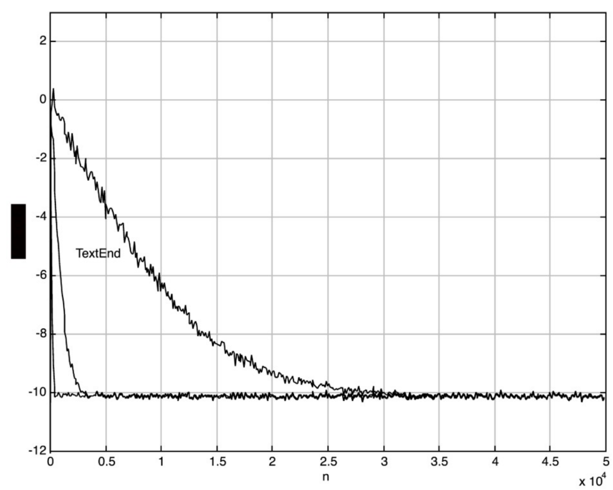
Raw data



Downsample data



100-pt MA filtered data

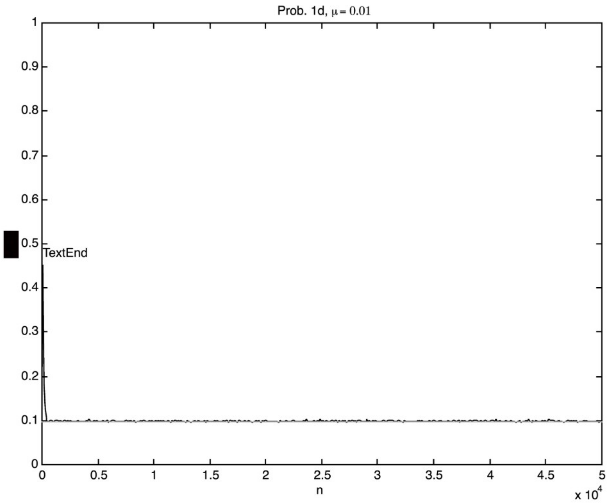


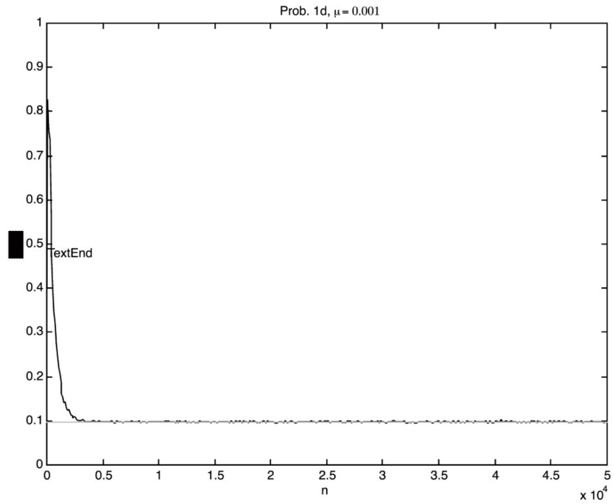
100-pt MA filtered, downsampled data

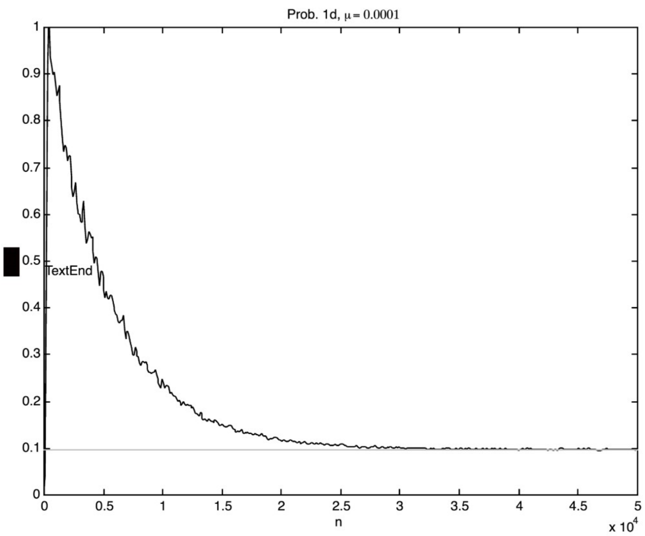
There are several ways to make the MSE plots more readable: downsampling the MSE data (fewer data points) and MA filtering or smoothing (length 100 filter is typical). These methods are illustrated above.

100 runs used in the simulations. Clearly as we increase the step size our convergence rates increase, however, the next part will show the consequences of this.

(d) All data has been MA filtered and downsampled for readability





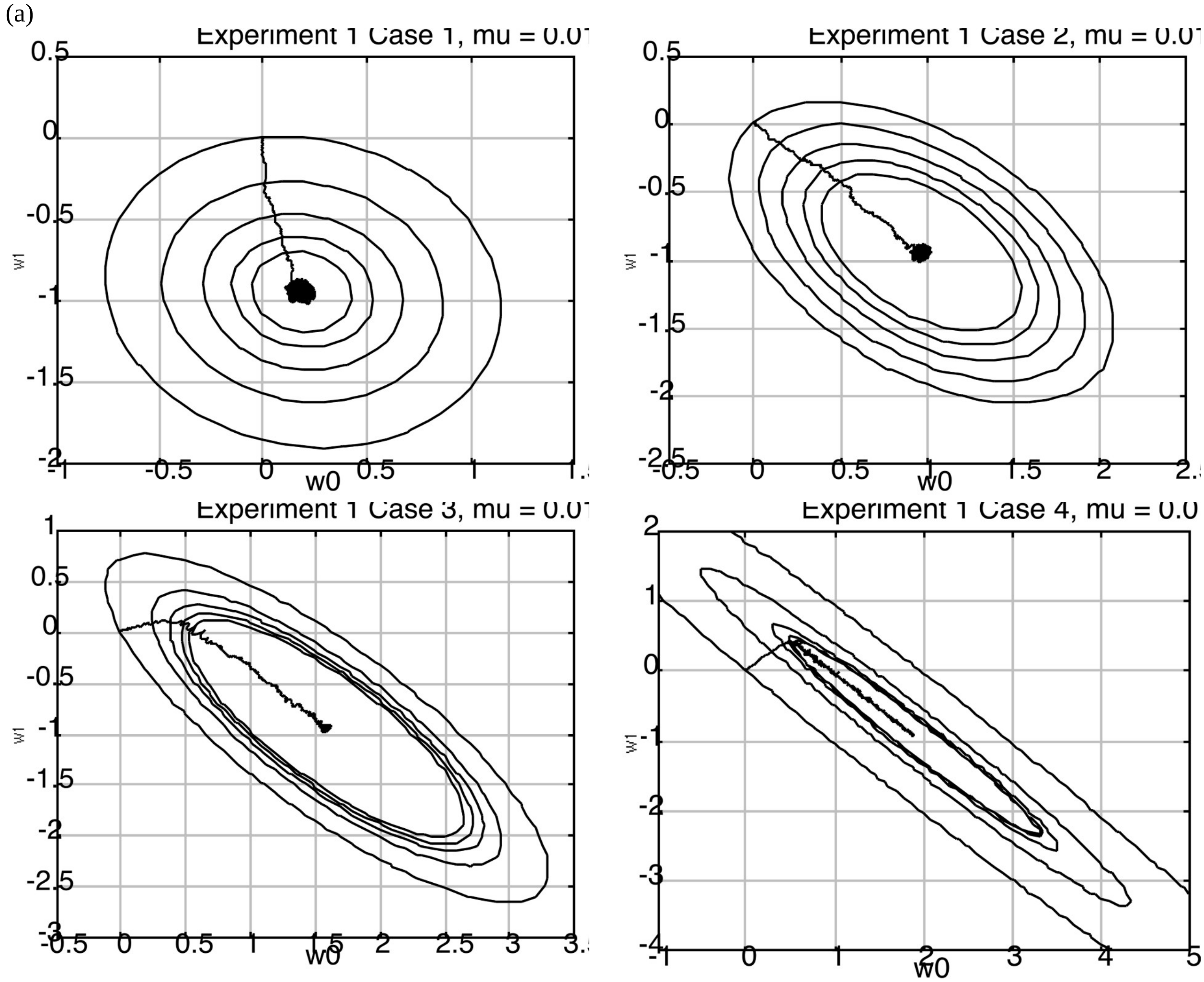


$\mu = 0.01$
J_{ex} = 9.75e-04 (theory)
J_{ex} = 10.266e-04 (experiment)
M = 0.0106

$\mu = 0.001$
J_{ex} = 9.66e-04 (theory)
J_{ex} = 11.484e-04 (experiment)
M = 0.0012

$\mu = 0.0001$
J_{ex} = 9.65e-06 (theory)
J_{ex} = 1.277e-04 (not enough samples?)
M = 0.0013

2)
(a)

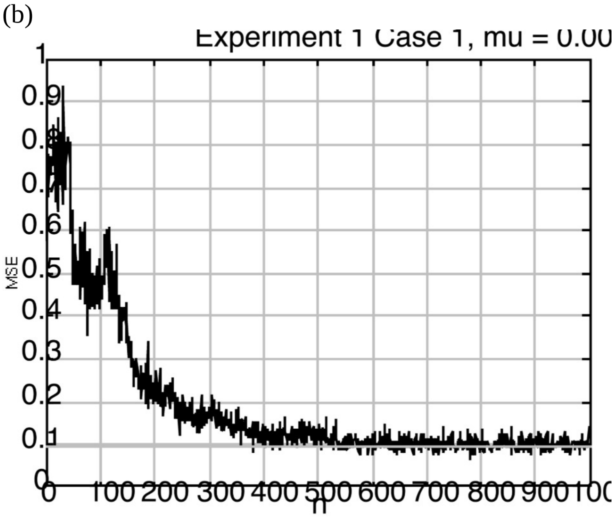


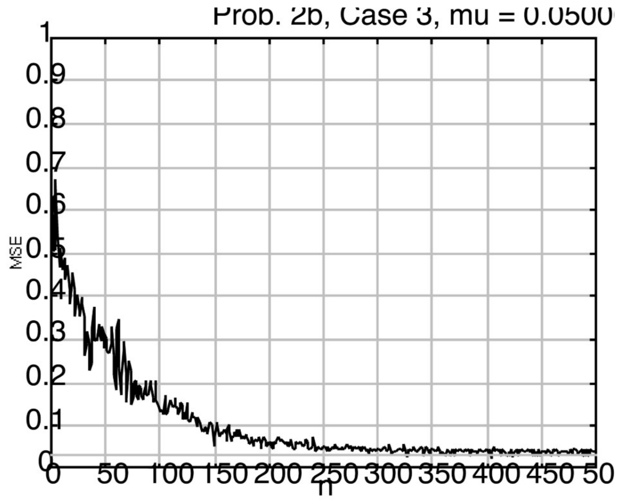
Case 1: χ = 1.22	Case 2: χ = 3	Case 3: χ = 10	Case 4: χ = 100

`wo =`
`0.1950`
`-0.9500`
`w(:,25001) =`
`0.2176`
`-0.9379`
	`wo =`
	`0.9750`
	`-0.9500`
	`w(:,25001) =`
	`0.9753`
	`-0.9674`
		`wo =`
		`1.5955`
		`-0.9500`
		`w(:,25001) =`
		`1.5871`
		`-0.9392`
			`wo =`
			`1.9114`
			`-0.9500`
			`w(:,25001) =`
			`1.9041`
			`-0.9431`

We see “noisy” versions of the trajectories from steepest-descent which is due to LMS’s noisy estimate of the gradient.

(b)



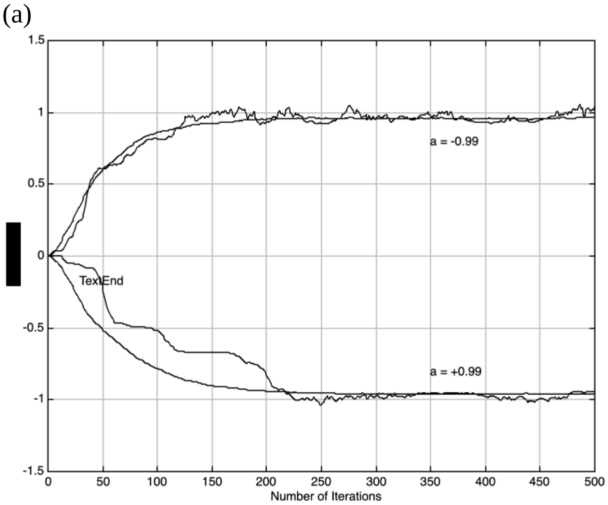


Case 1: $\chi = 1.22$ ($\mu = 0.007$)	Case 3: $\chi = 10$ ($\mu = 0.05$)
$J_{ex} = 0.0029$ (experiment) $J_{ex} = 4.8495\text{e-}04$ (theory) $J_{min} = 0.0965$ (theory) Misadjustment = 0.0305 (experiment) Misadjustment = 0.005 (theory)	$J_{ex} = 0.0033$ (experiment) $J_{ex} = 0.0053$ (theory) $J_{min} = 0.0322$ (theory) Misadjustment = 0.1034 (experiment) Misadjustment = 0.0546 (theory)

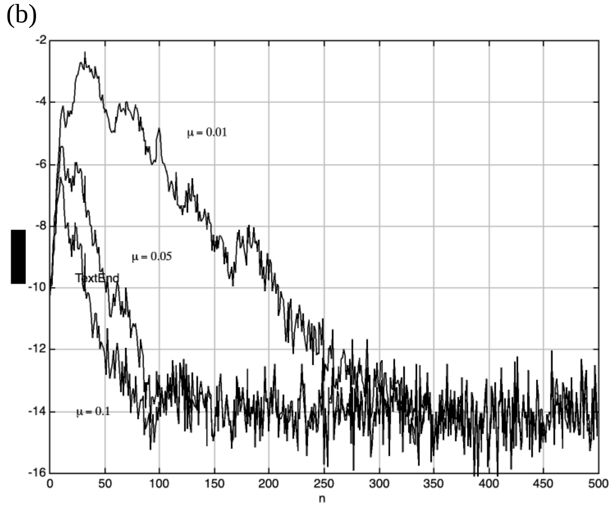
Clearly our experiments have J_{ex} which exceeds that found in our experiments--reasonable perhaps when we consider the assumptions used in determining a theoretical J_{ex}.

3)

(a)



(b)




```
% EE594 - Fall 2002 - Homework #4

%---
% 1
%---
% a
sigma_v = 0.096525;
a = [1 -0.1950 0.95]';
sigma_u = (1+a(3))/(1-a(3))*(sigma_v/((1+a(3))^2-a(2)^2));
```

```
r = [sigma_u;-a(2)/(1+a(3))*sigma_u;(-a(3) + (a(2)^2)/(1+a(3)))*sigma_u];
R = toeplitz(r(1:2));
p = [r(2) r(3)]';
J = [0.9997 0.5516 0.3273 0.2142 0.1568]';  % values from SD experiment

M = 2;
mu = 0.3;
w_init = zeros(M,1);
L = 200;
ts = 20;
randn('state',0);
u = AR_synthesizer(a,L,sigma_v);
[e,w] = lms2(w_init,u(ts:L-1),u(ts+1:L),mu,1);
grid_data = [-1.0 1.5 0.1;-2.0 0.5 0.1];
trajectory(sigma_u,[r(1) r(2)]',p,w,J,grid_data)
title(['Experiment 1 Case 1, \mu = ',sprintf('%f', mu)]);

% b
sigma_v = 0.096525;
a = [1 -0.1950 0.95]';
sigma_u = (1+a(3))/(1-a(3))*(sigma_v/((1+a(3))^2-a(2)^2));
r = [sigma_u;-a(2)/(1+a(3))*sigma_u;(-a(3) + (a(2)^2)/(1+a(3)))*sigma_u];
R = toeplitz(r(1:2));
p = [r(2) r(3)]';
J = [0.9997 0.5516 0.3273 0.2142 0.1568]';  % values from SD experiment
M = 2;
L = 50000;
ts = 20;
grid_data = [-1.0 1.5 0.1;-2.0 0.5 0.1];

% mu = 0.01
mu = 0.01;
w_init = zeros(M,1);
randn('state',0);
u = AR_synthesizer(a,L,sigma_v);
[e,w] = lms2(w_init,u(ts:L-1),u(ts+1:L),mu,1);
trajectory(sigma_u,[r(1) r(2)]',p,w,J,grid_data)
title(['Experiment 1 Case 1, \mu = ',sprintf('%f', mu)]);

% mu = 0.001
mu = 0.001;
w_init = zeros(M,1);
randn('state',0);
u = AR_synthesizer(a,L,sigma_v);
[e,w] = lms2(w_init,u(ts:L-1),u(ts+1:L),mu,1);
trajectory(sigma_u,[r(1) r(2)]',p,w,J,grid_data)
title(['Experiment 1 Case 1, \mu = ',sprintf('%f', mu)]);

% mu = 0.0001
mu = 0.0001;
w_init = zeros(M,1);
randn('state',0);
u = AR_synthesizer(a,L,sigma_v);
[e,w] = lms2(w_init,u(ts:L-1),u(ts+1:L),mu,1);
trajectory(sigma_u,[r(1) r(2)]',p,w,J,grid_data)
title(['Experiment 1 Case 1, \mu = ',sprintf('%f', mu)]);

% c
a = [1 -0.1950 0.95]';
```

```
sigma_v = 0.0965;
Jmin = 0.0965;
M = 2;
mu = 0.01;  % 0.01, 0.001, 0.0001
w_init = zeros(M,1);
L = 50020;
ts = 20;
acc_sqrd_e = zeros(L-ts,1);
num_runs = 100;
randn('seed',0);
for runs = 1:num_runs
   runs
   u = AR_synthesizer(a,L,sigma_v);
   % u = u ./ sqrt(cov(u(ts+1:L)));
   [e,w] = lms2(w_init,u(ts:L-1),u(ts+1:L),mu,0);
   acc_sqrd_e = acc_sqrd_e + e.^2;
end;
MSE = acc_sqrd_e / num_runs;
MSE = filter(ones(100,1)./100,1,MSE);  % Smooth this out a bit
MSE_plot(MSE,0,L-ts-1,1,100);  % optional downsampling

% d
L = 50020;
ts = 20;
Jmin = 0.0965;
Jex = mean(MSE(L-ts-999:L-ts)) - Jmin
Misadjustment = Jex / Jmin
MSE = filter(ones(100,1)./100,1,MSE);  % Smooth this out a bit
MSE_plot(MSE,0,L-ts-1,0,100);  % optional downsampling
grid;
hold on
plot([0 L-ts],[Jmin Jmin],'k:')
hold off
title('Prob. 1d, \mu = 0.01');
axis([0 L 0 1])

%---
% 2
%---
% a
%a = [1 -0.1950 0.95]';  % X = 1.22
%sigma_v = 0.0965;
%J = [0.9997 0.5516 0.3273 0.2142 0.1568]';  % values from SD experiment

%a = [1 -0.9750 0.95]';  % X = 3
%sigma_v = 0.0731;
%J = [0.9997 0.7424 0.5567 0.4225 0.3256]';  % values from SD experiment

%a = [1 -1.5955 0.95]';  % X = 10
%sigma_v = 0.0322;
%J = [0.9991 0.6366 0.5187 0.4560 0.4088]';  % values from SD experiment

%a = [1 -1.9114 0.95]';  % X = 100
%sigma_v = 0.0038;
%J = [0.9943 0.2359 0.1080 0.0858 0.0814]';  % values from SD experiment

sigma_u = (1+a(3))/(1-a(3))*(sigma_v/((1+a(3))^2-a(2)^2));
r = [sigma_u;-a(2)/(1+a(3))*sigma_u;(-a(3) + (a(2)^2)/(1+a(3)))*sigma_u];
R = toeplitz(r(1:2));
```

```
p = [r(2) r(3)]';

M = 2;
mu = 0.01;
w_init = zeros(M,1);
L = 25020;
ts = 20;
randn('seed',0);
u = AR_synthesizer(a,L,sigma_v);
[e,w] = lms2(w_init,u(ts:L-1),u(ts+1:L),mu,1);
grid_data = [-1 5 0.1;-4 2 0.1];
trajectory(sigma_u,[r(1) r(2)]',p,w,J,grid_data)
title(['Experiment 1 Case 4, mu = ',sprintf('%f', mu)]);
[inv(R)*p w(:,L-ts+1)]

% b
a = [1 -0.1950 0.95]';  % X = 1.22
sigma_v = 0.0965;
J = [0.9997 0.5516 0.3273 0.2142 0.1568]';  % values from SD experiment
Jmin = 0.0965;

%a = [1 -1.5955 0.95]';  % X = 10
%sigma_v = 0.0322;
%J = [0.9991 0.6366 0.5187 0.4560 0.4088]';  % values from SD experiment
%Jmin = 0.0322;

sigma_u = (1+a(3))/(1-a(3))*(sigma_v/((1+a(3))^2-a(2)^2));
r = [sigma_u;-a(2)/(1+a(3))*sigma_u;(-a(3) + (a(2)^2)/(1+a(3)))*sigma_u];
R = toeplitz(r(1:2));
p = [r(2) r(3)]';

M = 2;
mu = 0.005;% Case 1 mu=0.005 Jex=0.0029. Case 3 mu=0.05 Jex=0.0033
w_init = zeros(M,1);
L = 1020;
ts = 20;
acc_sqrd_e = zeros(L-ts,1);
num_runs = 100;
randn('seed',0);

for runs = 1:num_runs
   runs
   u = AR_synthesizer(a,L,sigma_v);
   [e,w] = lms2(w_init,u(ts:L-1),u(ts+1:L),mu,0);
   acc_sqrd_e = acc_sqrd_e + e.^2;
end;

MSE = acc_sqrd_e / num_runs;
MSE_plot(MSE,0,L-ts-1,0);
w_o = inv(R)*p
Jex = mean(MSE(L-ts-99:L-ts))-Jmin
Misadjustment = Jex/Jmin

title(['Prob. 2b, Case 3, mu = ',sprintf('%f', mu)]);
hold on
plot([0 L-ts-1],[Jmin Jmin],':')
axis([0  L-ts-1 0 10]); % for normal unit plots
hold off
axis([0 L-ts 0 1])
```

```
%---
% 3
%---
% Case 1 - Single Realization
randn('state',0);
a = [1;-0.99]; % Case 1
N = 501;
u = AR_synthesizer(a,N,1);
sigma_u2 = 0.93627; % experiment condition
u = u .* sqrt(sigma_u2) ./ sqrt(cov(u));
w_init = [0];
mu = 0.05;
[e,w] = lms2(w_init,u(1:N-1),u(2:N),mu,1);
plot([0:N-1],w,'k');

% Case 1 - Ensemble-Averaged Result
randn('state',0);
a = [1;-0.99]; % Case 1
N = 501;
sigma_u2 = 0.93627; % experiment condition
w_init = [0];
mu = 0.05;
num_runs = 100;
acc_w = zeros(1,N);
for runs = 1:num_runs
   runs
   u = AR_synthesizer(a,N,1);
   u = u .* sqrt(sigma_u2) ./ sqrt(cov(u));
   [e,w] = lms2(w_init,u(1:N-1),u(2:N),mu,1);
   acc_w = acc_w + w;
end;
mean_w = acc_w ./ num_runs;
hold on
plot([0:N-1],mean_w,'k');

% Case 2 - Single Realization
randn('state',0);
a = [1;0.99]; % Case 2
N = 501;
u = AR_synthesizer(a,N,1);
sigma_u2 = 0.93627; % experiment condition
u = u .* sqrt(sigma_u2) ./ sqrt(cov(u));
w_init = [0];
mu = 0.05;
[e,w] = lms2(w_init,u(1:N-1),u(2:N),mu,1);
plot([0:N-1],w,'k');

% Case 2 - Ensemble-Averaged Result
randn('state',0);
a = [1;0.99]; % Case 2
N = 501;
sigma_u2 = 0.93627; % experiment condition
w_init = [0];
mu = 0.05;
num_runs = 100;
acc_w = zeros(1,N);
for runs = 1:num_runs
   runs
```

```
   u = AR_synthesizer(a,N,1);
   u = u .* sqrt(sigma_u2) ./ sqrt(cov(u));
   [e,w] = lms2(w_init,u(1:N-1),u(2:N),mu,1);
   acc_w = acc_w + w;
end;
mean_w = acc_w ./ num_runs;
plot([0:N-1],mean_w,'k');
hold off
ylabel('Tap Weight');
xlabel('Number of Iterations');
grid;
axis([0 500 -1.5 1.5]);
text(350,0.8,'a = -0.99');
text(350,-0.8,'a = +0.99');

% Case 1 - MSE vs. Step-Size Results
randn('state',0);
a = [1;-0.99]; % Case 1
N = 501;
sigma_u2 = 0.93627; % experiment condition
w_init = [0];
mu = 0.1; % 0.01, 0.05 0.1
num_runs = 100;
acc_sqrd_e = zeros(N-1,1);
for runs = 1:num_runs
   runs
   u = AR_synthesizer(a,N,1);
   u = u .* sqrt(sigma_u2) ./ sqrt(cov(u));
   [e,w] = lms2(w_init,u(1:N-1),u(2:N),mu,0);
   acc_sqrd_e = acc_sqrd_e + e.^2;
end;
MSE = acc_sqrd_e / num_runs;
MSE_plot(MSE);
```