**ASSIGNMENT 4**

**QUESTION 1:**

% To plot graph for true values

N = 1000; % total number of time steps

true\_pop = zeros(N+1,1);

true\_food = zeros(N+1,1);

% entering the initial states

dt = 1;

time = [1:dt:11];

true\_pop(1) = 650;

true\_food(1) = 250;

for i=2:N+1

true\_pop(i,1) = (true\_pop(i-1,1)/2)+(2\*true\_food(i-1,1));

true\_food(i,1) = true\_food(i-1,1);

end

% to calculate the estimated value= zeros(N+1,1);

plus\_state = zeros(2,N+1);

minus\_state = zeros(2,N+1);

gain= zeros(2,N+1);

%error= zeros(2,N);

F = [1 0; 2 0.5];

L = [1 0 ;0 0];

Q = [10 0;0 0];

Q\_tilde = L'\*Q\*L;

R = [10];

H = [0 1];

P\_plus = [200 0;0 500];

P\_minus = [0 0;0 0];

I = [1 0;0 1];

plus\_state(1,1) = 200;

plus\_state(2,1) = 600;

std\_1 = zeros(N+1,1);

std\_2 = zeros(N+1,1);

for j= 2:N+1

P\_minus = F\*P\_plus\*F'+Q\_tilde;

value = (inv(H\*P\_minus\*H'+R));

gain(:,j) = P\_minus\*H'\*value;

minus\_state(:,j) = F\*plus\_state(:,j-1);

plus\_state(:,j) = minus\_state(:,j) + gain(:,j)\*(true\_pop(j,:)-H\*minus\_state(:,j));

P\_plus= (I-gain(:,j)\*H)\*P\_minus;

std\_1(j,1)= sqrt(P\_plus(1,1));

std\_2(j,1)= sqrt(P\_plus(2,2));

end

P\_plus

% To plot population comparison

set(gcf,'color','w');

plot(time,true\_pop(:,1));

hold on;

plot(time,plus\_state(2,:));

hold off;

% to plot food supply comparison

% plot(time,true\_food(:,1));

% hold on;

% plot(time,plus\_state(1,:));

% hold off;

% To plot the standard deviations

% plot(time,std\_1(:,1));

%plot(time,std\_2(:,1));

% % to plot gain as a function of time

% plot(time,gain(1,:)) % food

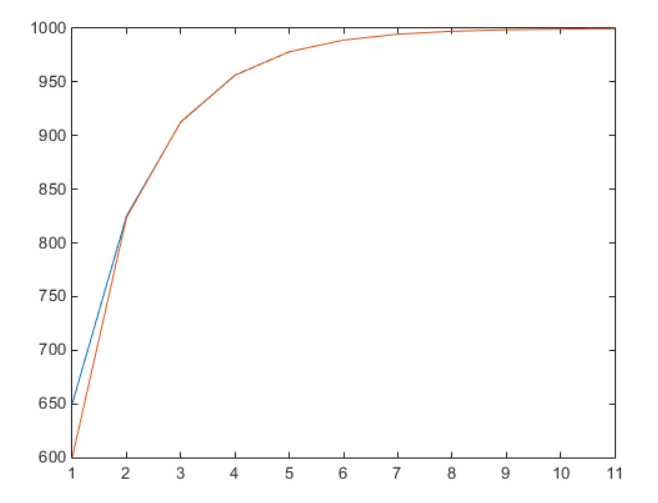
% plot(time,gain(2,:)) % population

% to calculate std deviation as a function of time

GRAPH 1:

True population and expected population:

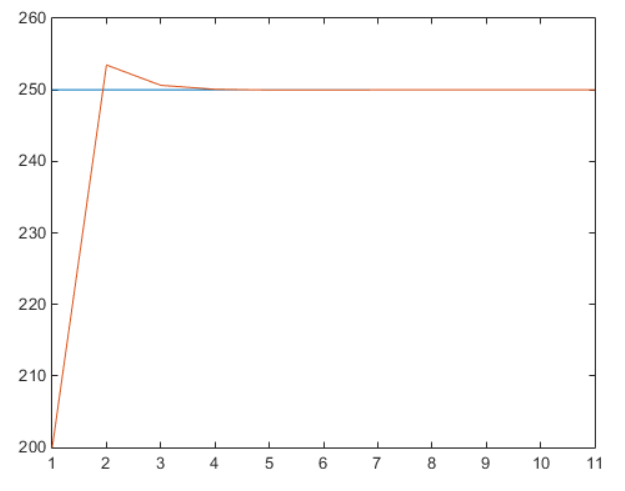
X axis : Time step Y axis : Population



GRAPH 2:

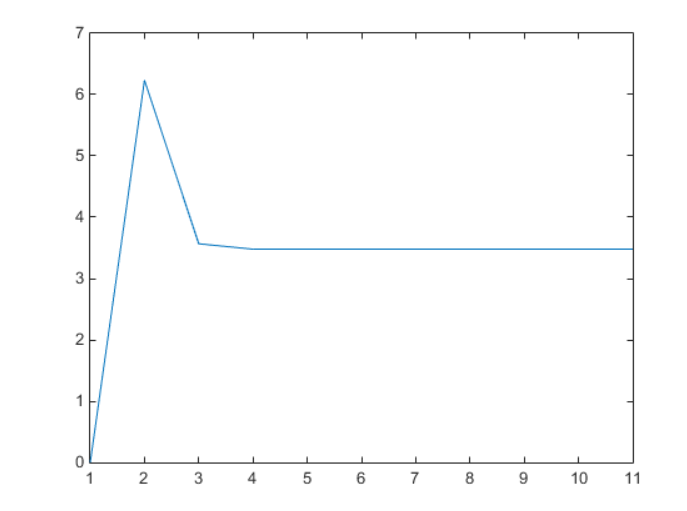
TRUE FOOD SUPPLY AND ESTIMATED FOOD SUPPLY:

X AXIS : TIME STEP Y AXIS : FOOD SUPPLY



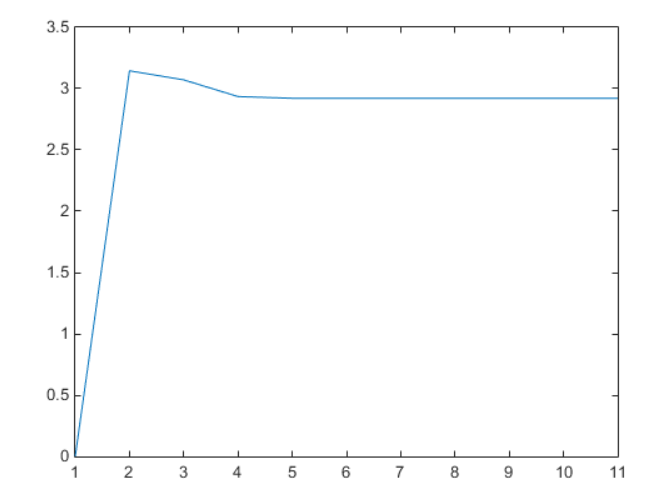
STANDARD DEVIATION OF FOOD ESTIMATION ERROR

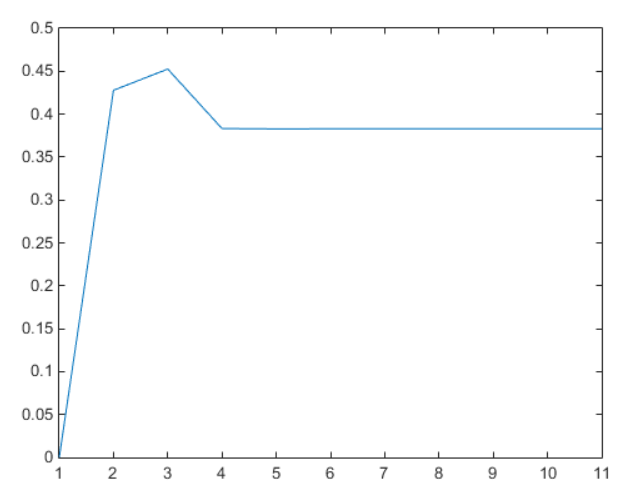
X : TIME SERIES Y= STD OF ERROR IN FOOD ESTIMATION



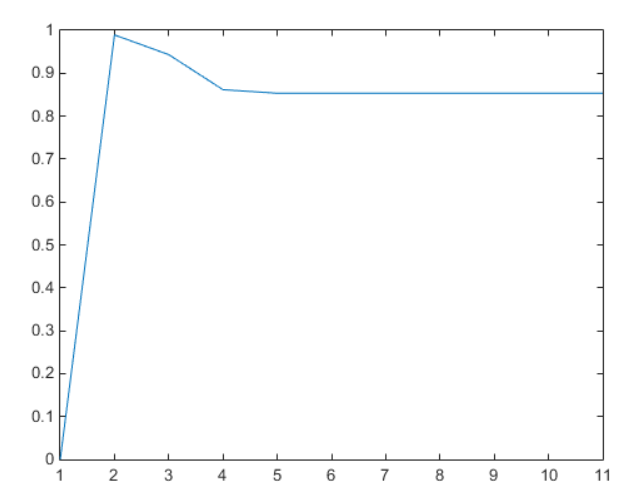
STANDARD DEVIATION OF POPULATION ESTIMATION ERROR

X : TIME SERIES Y= STD OF ERROR IN POPULATION ESTIMATION



GAIN FOR FOOD:

GAIN FOR POPULATION



QUESTION 3:

A = [0 2; -1 -3];

H = [1 0];

dt = 1/10;

time = [0:dt:2];

N = 20;

R =1/10;

Q = [1 0; 0 1];

L = [0 1];

plus\_state = zeros(2,N);

minus\_state = zeros(2,N);

gain= zeros(2,N);

P\_plus = [0 0;0 0];

F = [1 0.2; -0.1 0.7];

Q\_c = L\*Q\*L';

var\_1 = zeros(N+1,1);

var\_2 = zeros(N+1,1);

for j= 2:N

Q\_tilde = [0.2\*time(j)^2 0.2\*time(j)-0.6\*(time(j)^2);0.2\*time(j)-0.6\*(time(j)^2) 0.1+0.9\*time(j)^2-0.2\*time(j)];

P\_minus = F\*P\_plus\*F'+Q\_tilde;

value = (inv(H\*P\_minus\*H'+R));

gain(:,j) = P\_minus\*H'\*value;

%minus\_state(:,j) = F\*plus\_state(:,j-1);

%error(1,j-1) = true\_pop(j,:)-H\*minus\_state(:,j);

%plus\_state(2,j) = minus\_state(2,j) + gain(2,j)\*(0-H\*minus\_state(1,j));

P\_plus= (I-gain(:,j)\*H)\*P\_minus;

var\_1(j,1)= (P\_plus(1,1));

var\_2(j,1)= (P\_plus(2,2));

var\_1(j,2)= (P\_minus(1,1));

var\_2(j,2)= (P\_minus(2,2));

end

set(gcf,'color','w');

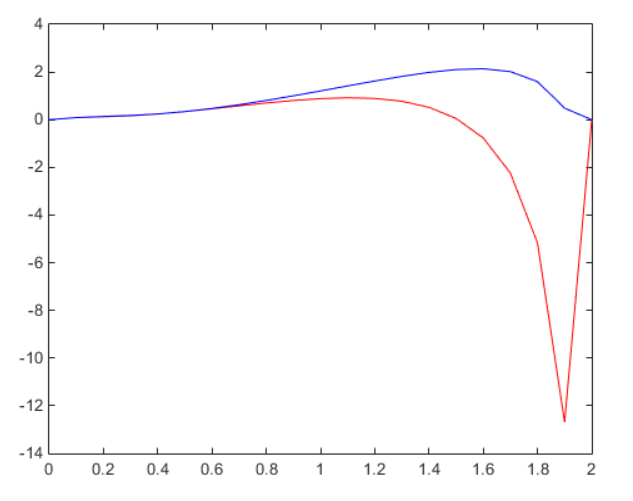
plot(time,var\_2(:,1),'r');

hold on;

plot(time,var\_2(:,2),'b');

hold off;

GRAPH:



QUESTION 5:

F = [1 0;2 0.5];

L = [1 0;0 0];

H = [0 1];

Q = [10 0;0 0];

%F = A;

% (4) Initial Conditions and Noise Matrices

x\_0 = [200;600];

P\_0 = [200 0;0 500];

R = [10];

LQL = L'\*Q\*L;

% (5) Forward Propagation of Covariance

cov\_f\_p = zeros(drl,4); % Forward + covariance history

cov\_f\_m = zeros(drl,4); % Forward - covariance history

Pf = P\_0;

cov\_f\_p(1,:) = [Pf(1,1) Pf(1,2) Pf(2,1) Pf(2,2)];

cov\_f\_m(1,:) = [Pf(1,1) Pf(1,2) Pf(2,1) Pf(2,2)];

for k=2:drl

Pf = F\*Pf\*F' + LQL;

cov\_f\_m(k,:) = [Pf(1,1) Pf(1,2) Pf(2,1) Pf(2,2)];

K = Pf\*H'\*inv(H\*Pf\*H' + R);

Pf = (eye(2) - K\*H)\*Pf;

cov\_f\_p(k,:) = [Pf(1,1) Pf(1,2) Pf(2,1) Pf(2,2)];

end

%% (6) Backward Propagation of Covariance

F\_b = inv(F);

LQL\_b = F\_b\*LQL\*F\_b';

cov\_b\_p = zeros(drl,4); % Backward covariance history

Pb = 10000\*P\_0; % Not quite infinity but large enough

I = [0 0;0 0]; % For Homework fix this so that you are

% using an information matrix

cov\_b\_p(end,:) = [Pb(1,1) Pb(1,2) Pb(2,1) Pb(2,2)];

eps = 0.0001;

for k=drl-1:-1:1

I = I + H'\*pinv(R)\*H;

Pb = pinv(I);

inter\_1 = F\_b\*pinv(I)\*F\_b';

inter\_2 = F\_b\*LQL\*F\_b';

I = pinv(inter\_1+inter\_2);

% Pb = pinv(I);

% Pb = F\_b\*Pb\*F\_b' + LQL\_b;

% K = Pb\*H'\*inv(H\*Pb\*H' + R);

% Pb = (eye(2) - K\*H)\*Pb;

cov\_b\_p(k,:) = [Pb(1,1) Pb(1,2) Pb(2,1) Pb(2,2)];

end

%% (7) RTS Backward Propagation of Covariance

cov\_s = zeros(drl,4); % Smoother covariance history

Ps = Pf;

cov\_s(end,:) = [Ps(1,1) Ps(1,2) Ps(2,1) Ps(2,2)];

for k=(drl-1):-1:1

Pf\_kp = [cov\_f\_p(k,1) cov\_f\_p(k,2);cov\_f\_p(k,3) cov\_f\_p(k,4)];

Pf\_k1m = [cov\_f\_m(k+1,1) cov\_f\_m(k+1,2);cov\_f\_m(k+1,3) cov\_f\_m(k+1,4)];

K = Pf\_kp\*F'\*inv(Pf\_k1m);

Ps = Pf\_kp - K\*(Pf\_k1m - Ps)\*K';

cov\_s(k,:) = [Ps(1,1) Ps(1,2) Ps(2,1) Ps(2,2)];

end

figure(gcf)

subplot(121)

h1f = semilogy(t,cov\_f\_p(:,1),'b-');hold on;grid on;

h1b = semilogy(t,cov\_b\_p(:,1),'r-');

h1s = semilogy(t,cov\_s(:,1),'g\*');

legend('P\_f','P\_b','P\_{RTS}')

set(h1f,'LineWidth',2);set(h1b,'LineWidth',2);set(h1s,'LineWidth',2);

xlabel('Time (sec)');ylabel('P\_{11}');

subplot(122)

h2f = semilogy(t,cov\_f\_p(:,4),'b-');hold on;grid on;

h2b = semilogy(t,cov\_b\_p(:,4),'r-');

h2s = semilogy(t,cov\_s(:,4),'g\*');

set(h2f,'LineWidth',2);set(h2b,'LineWidth',2);set(h2s,'LineWidth',2);

xlabel('Time (sec)');ylabel('P\_{22}');

GRAPH:

