Question 7.1

Results Part b:

Using batch BLUE for t = **0:10**

Estimated value of theta : theta_est = [0.134;0.794;-0.366]

i.e. a12 = 0.134, a22 = 0.794, b21= -0.366

so , A = [1, 0.134;
0, 0.794;
];
B = [0;
-0.366
]

Using **batch BLUE** for t = **0:1000**
Estimated value of theta : theta_est = [0.1502;0.9984;0.0303]

i.e. a12 = 0.1502, a22 = 0.9984, b21= 0.0303

so , A = [1, 0.1502;
0, 0.9984;
];
B = [0;
0.0303
]

Results Part d:

Using **Recursive BLUE** for t = **0:1000**
Estimated value of theta : theta_est = [0.1507;0.9983;0.0307]

i.e. a12 = 0.1507, a22 = 0.9983, b21= 0.0304

so , A = [1, 0.1507;
0, 0.9983;
];
B = [0;
0.0304
]

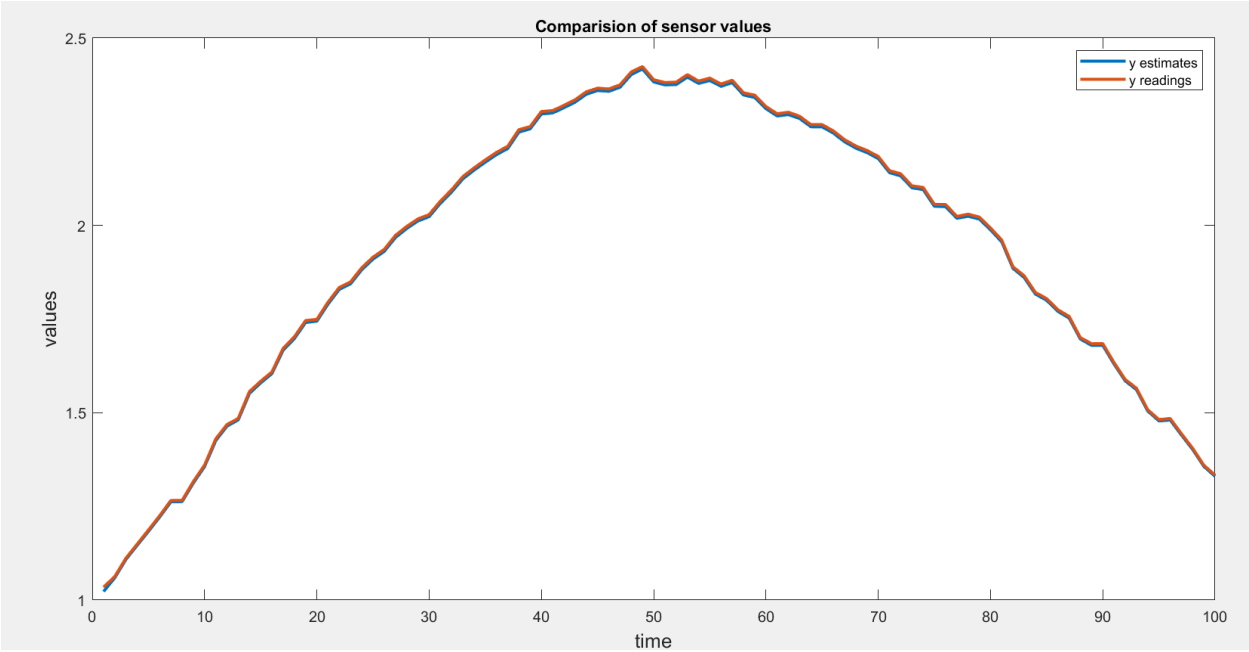
Question 7.2:

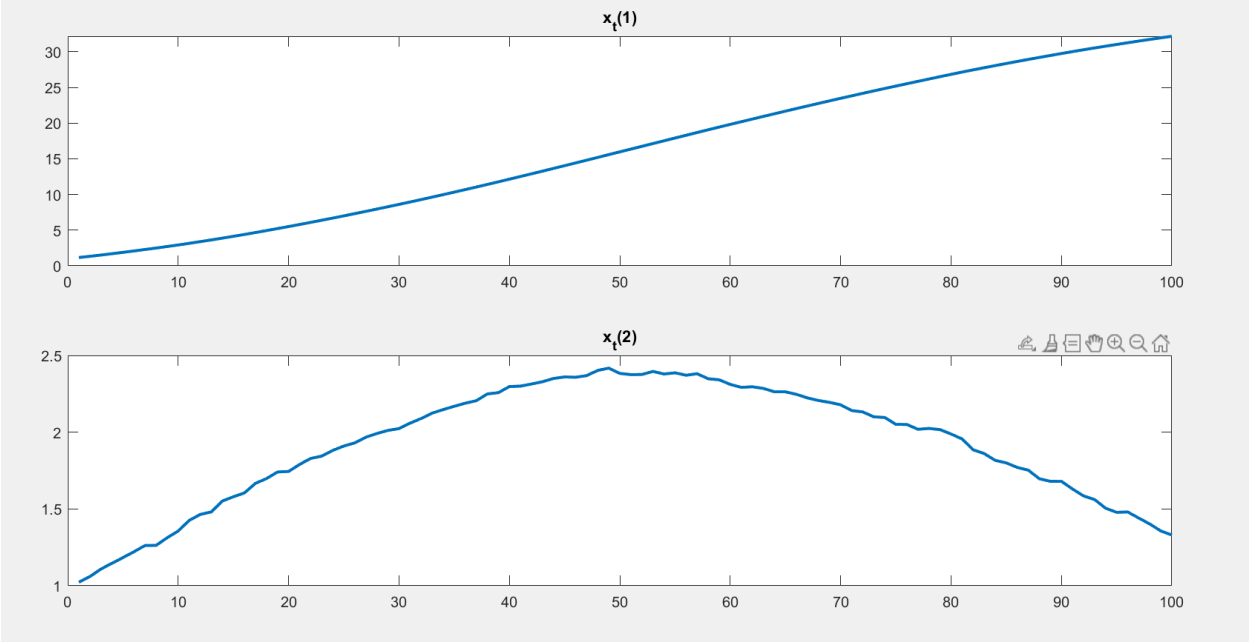
Results Part a:

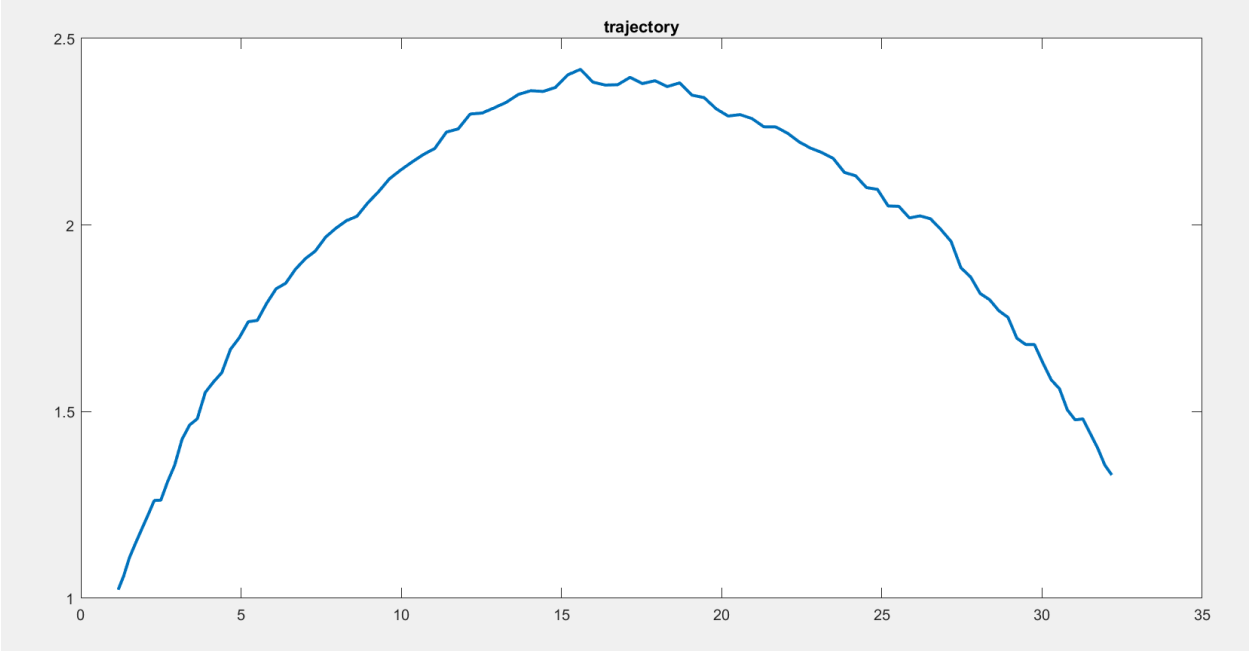
When the **Initial P matrix** is chosen as **Identity matrix**: eye(2)

Kalman gain for step 1: K = [0.1661;
0.9843];

Results Part b:




$x_t(1)$
0
5
10
15
20
25
30
0
10
20
30
40
50
60
70
80
90
100
$x_t(2)$
1
1.5
2
2.5
0
10
20
30
40
50
60
70
80
90
100


trajectory
1
1.5
2
2.5
0
5
10
15
20
25
30
35

Code Question 7.1.b:

```
x = readmatrix("./data_1/x.csv" );
u = readmatrix("./data_1/u.csv" );

H = [];
z = [];
Q = [];

 for i = 1:10

    H_add = [x(i,2),0,0;
     0, x(i,2), u(i)];
    z_add = [x(i+1,1)-x(i,1);x(i+1,2)];
    Q_add = [0.02,0;
    0,0.01;];

    H_new = [H;
          H_add];
    z_new = [z;z_add];
    if(~isempty(Q))
        Q_new = [Q,zeros(length(Q(:,1)),2);
              zeros(2,length(Q(1,:))),Q_add];
    else
        Q_new = Q_add;
    end

    H = H_new;
    Q = Q_new;
    z = z_new;
 end

Q_in = inv(Q);
est_mat = inv((H')*Q_in*H)*(H')*Q_in;
theta_est = est_mat*z;

x_new = validate_dyn(theta_est, x(80,:)', u(80));
```

Code Question 7.1.d:

```
x = readmatrix("./data_1/x.csv" );
u = readmatrix("./data_1/u.csv" );

Q = zeros(3,3);

A_i = [];
for i = 1:999
   H_i = [x(i,2),0,0;
    0, x(i,2), u(i)];
   A_i = [A_i;H_i];
   z_i = [x(i+1,1)-x(i,1);x(i+1,2)];
   s_i = [0.02,0;
          0,0.01];
   if(i>=10)
      C_k_new = H_i;
      S_k_new = inv(s_i);

      Q_new = Q + C_k_new'*S_k_new*C_k_new;
      K_new = inv(Q_new)*C_k_new'*S_k_new;

      x_new = x_old + K_new*(z_i-H_i*x_old);

      x_old = x_new;
      Q = Q_new;
   else
      addthing = H_i'*s_i*H_i;
      x_old = zeros(3,1);
      Q = Q + addthing;

   end
end
```

Code Question 7.2::

```
y = readmatrix("./data_2/y.csv" );
u = readmatrix("./data_2/u.csv" );

x_hat = zeros(100,2);
y_hat = zeros(100,1);

E_k = [];
% theta_est = [0.150;0.998;0.030];
theta_est = [0.134;0.794;-0.366];
x_new = [1;0.3];
P_new = [1,0;
        0, 1];

% [0.134194742312479;0.794913610698950;-0.366540891243778]
% K gail [0.148806565381450;0.990059681671246]
for i = 1:100
    A = [1,theta_est(1);
          0, theta_est(2)];
    C_i = [0,1];
    Q = 0.01;
    y_i = y(i);

    x_old = validate_dyn(theta_est,x_new,u(i));
    P_old = A*P_new*A';

    K = (P_old*C_i')*inv((C_i*P_old*C_i')+Q);

    x_new = x_old + K*(y_i - C_i*x_old);

    y_hat(i) = C_i*x_new;

    x_hat(i,1) = x_new(1);
    x_hat(i,2) = x_new(2);

    if(i==1)
        initial_k_gain = K;
    end

end

fig = figure('visible','on');
plot(y_hat,'LineWidth',2.0);
hold on
plot(y,'LineWidth',2.0);
hold off
legend('y estimates','y readings');
```

```
ylabel('values','FontSize',13)
xlabel('time','FontSize',13)
title("Comparision of sensor values")
saveas(fig, "compare_y_value.png");


fig = figure('visible','on');
plot(x_hat(:,1),x_hat(:,2),'LineWidth',2.0);
title("trajectory")
saveas(fig, "traj.png");


fig = figure('visible','on');
subplot(2,1,1);
plot(x_hat(:,1),'LineWidth',2.0);
subplot(2,1,2);
plot(x_hat(:,2),'LineWidth',2.0);
saveas(fig, "traj_comp.png");
```

Helper function for 7.1 and 7.2

```
function x_new = validate_dyn(theta,x,u)
A = [1, theta(1);
     0, theta(2);];
B = [0;
    theta(3);];
x_new = A*x + B*u;
end
```