ECE 8540 Analysis of Tracking Systems

Lab 4- Kalman Filtering

Mayuresh Bhosale
mbhosal@clemson.edu

Department of Automotive Engineering,
Clemson University

November 7, 2022

1 Introduction

The lab report addresses problem of applying Kalman Filter for multiple measurements. Measurements are set of points over time with error value to be filtered. Error or noise is deviation from true value. True value is not always the same as the measurements. Kalman Filtering is performed to obtain the best estimate of true value from measurements. Kalman filtering can be applied in real time as well as to recorded measurements. Previous methods such as model fitting techniques are used to get the best estimate over set of measurements. The model fitting method cannot be applied in real time. Kalman filtering is applied to accurately predict the next value from the previous measurement. Weights or gains are applied in Kalman filtering to predictions and measurements to obtain true values.

2 Methodology

Kalman filtering is done with set of equations that continuously predict and update the value. Firstly, the next state and the state covariance are predicted. After the prediction, the sensor recorded measurement is obtained and Kalman gain is calculated. The Kalman gain consists of weights that are applied to the prediction and the measured value. Based on this Kalman gain, the state and state covariance is updated. The updated state provides final output for that particular state and this cycle is repeated over time. The following equations define Kalman filtering:

1. Predict next state

$$X_{t,t-1} = \Phi X_{t-1,t-1} \tag{1}$$

2. Predict next state covariance

$$S_{t,t-1} = \Phi S_{t-1,t-1} \Phi^T + Q \tag{2}$$

3. Obtain measurement(s)

$$Y_t \tag{3}$$

4. Calculate the Kalman gain (weights)

$$K_t = S_{t,t-1} M^T [M S_{t,t-1} M^T + R]^{-1} \tag{4}$$

5. Update state

$$X_{t,t} = X_{t,t-1} + K_t (Y_t - M X_{t,t-1}) \tag{5}$$

6. Update state covariance

$$S_{t,t} = [I - K_t M] S_{t,t-1} \tag{6}$$

7. Loop (t becomes t+1)

Here, the initial state, state covariance and measurement values are to be defined.Moreover depending on the state variables the all the matrices should be checked before executing the loop. The dynamic noise covariance Q and measurement noise covariance R are adjusted and defined by the user. The update state provides final output value at that time step.

2.1 1-D Kalman Filtering

Following equations define the matrices to solve the 1D Kalman filtering to estimate position with a constant velocity model.

1. State variables

$$\mathbf{X_t} = \begin{bmatrix} x_t \\ \dot{x}_t \end{bmatrix} \tag{7}$$

2. State transition equations

$$x_{t+1} = x_t + T\dot{x}_t \tag{8}$$

$$\dot{x}_{t+1} = \dot{x}_t \tag{9}$$

State transition matrix based on the state variables and state transition equations on can be written as matrix in equation 10. Here the time step T is written as 1 for 1 second interval.

$$\mathbf{\Phi} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \tag{10}$$

3. Observation variables are defined as:

$$\mathbf{Y_t} = \begin{bmatrix} \tilde{x_t} \end{bmatrix} \tag{11}$$

here, $\tilde{x_t}$ is 1-D measurement at consecutive time steps t.

4. Observation equation is written as:

$$\tilde{x_t} = x_t \tag{12}$$

5. State covariance matrix S_t that defines the covariance of the state variables is written as:

$$\mathbf{S_t} = \begin{bmatrix} \sigma^2_{x_t} & \sigma_{x_t,\dot{x}_t} \\ \sigma_{x_t,\dot{x}_t} & \dot{\sigma}^2_{x_t} \end{bmatrix} \tag{13}$$

here, as there are two variables in the state, the matrix size is 2x2

6. The dynamic noise covariance matrix Q and measurement noise covariance matrix R can be written as:

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 \\ 0 & \sigma^2_a \end{bmatrix} \tag{14}$$

$$\mathbf{R} = \begin{bmatrix} \sigma^2_n \end{bmatrix} \tag{15}$$

7. The observation matrix M is 1x2 matrix as defined below:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{16}$$

The final step after defining matrices is to check the size of matrices in the Kalman filtering equations.

2.2 2-D Kalman Filtering

Following equations define the matrices to solve the 2D Kalman filtering to estimate position with a constant velocity model.

1. State variables

$$\mathbf{X_t} = \begin{bmatrix} x_t \\ t_t \\ \dot{x}_t \\ \dot{y}_t \end{bmatrix} \tag{17}$$

2. State transition equations

$$x_{t+1} = x_t + T\dot{x}_t \tag{18}$$

$$y_{t+1} = y_t + T\dot{y}_t \tag{19}$$

$$\dot{x}_{t+1} = \dot{x}_t \tag{20}$$

$$\dot{y}_{t+1} = \dot{y}_t \tag{21}$$

State transition matrix based on the state variables and state transition equations on can be written as matrix in equation 22. Here the time step T is written as 1 for 1 second interval.

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{22}$$

3. Observation variables are defined as:

$$\mathbf{Y_t} = \begin{bmatrix} \tilde{x_t} \\ \tilde{y_t} \end{bmatrix} \tag{23}$$

here, $\tilde{x_t}$ is 1-D measurement at consecutive time steps t.

4. Observation equation is written as:

$$\tilde{x_t} = x_t \tag{24}$$

$$\tilde{y_t} = y_t \tag{25}$$

5. State covariance matrix S_t that defines the covariance of the state variables is written as:

$$\mathbf{S_t} = \begin{bmatrix} \sigma_{x_t}^2 & \sigma_{x_t,y_t} & \sigma_{x_t,\dot{x}_t} & \sigma_{x_t,\dot{y}_t} \\ \sigma_{x_t,y_t} & \sigma_{y_t}^2 & \sigma_{y_t,\dot{x}_t} & \sigma_{y_t,\dot{y}_t} \\ \sigma_{x_t,\dot{x}_t} & \sigma_{y_t,\dot{x}_t} & \sigma_{\dot{x}_t}^2 & \sigma_{\dot{y}_t,\dot{y}_t} \\ \sigma_{x_t,\dot{y}_t} & \sigma_{y_t,\dot{y}_t} & \sigma_{\dot{y}_t,\dot{y}_t} & \sigma_{\dot{y}_t}^2 \end{bmatrix} \tag{26}$$

here, as there are two variables in the state, the matrix size is 2x2

6. The dynamic noise covariance matrix Q and measurement noise covariance matrix R can be written as:

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{a_1}^2 & \sigma_{a_1,a_2} \\ 0 & 0 & \sigma_{a_1,a_2} & \sigma_{a_2}^2 \end{bmatrix} \tag{27}$$

$$\mathbf{R} = \begin{bmatrix} \sigma_{n1}^2 & \sigma_{n_1,n_2} \\ \sigma_{n_1,n_2} & \sigma_{n2}^2 \end{bmatrix} \tag{28}$$

7. The observation matrix M is 1x2 matrix as defined below:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{29}$$

The final step after defining matrices is to check the size of matrices in the Kalman filtering equations.

2.3 MATLAB Programming and Implementation

The Kalman filtering are further solved by using MATLAB Scripts. R2021a version of MATLAB was used on a Windows 11 Operating System. All the matrices were constructed in MATLAB and the update state $X_{t,t}$ is calculated. To visualize the Kalman filtering, graphs were plotted depicting Kalman filtered output to the measured data.

3 Results

3.1 1-D Kalman Filtering

The Kalman filtered output of 1-D measured data is calculated and plotted. Different iterations are performed for multiple measurement noise covariance R and dynamic noise covariance Q. The Table 1 shows 3 iterations made for Q and R values to obtain the results against 1-D measured data.

3.1.1 Trial 1

Figure 1 shows results from Kalman filtering for R = 1 and Q = 10. In this trial, the measurement noise covariance in the Kalman filtering equations is weighted lower than the dynamic noise covariance. Here it is considered that there is huge

Table 1: Trials of dynamic noise covariance matrix Q and measurement Noise covariance matrix R for 1-D Kalman filtering

Trial No	R (σ_n^2)	Q (σ_a^2)
1	1	10
2	10^4	10^{-4}
3	10^{10}	10^{-10}

impact of the dynamic noise on the filtered output than the measurement noise. As observed in Figure 1, the filtered output largely matches the measurement readings. The top plot consists Kalman filtered output with respect to measured data with reduced data points and bottom plot is the similar to top plot but with all the data points included.

3.1.2 Trial 2

Figure 2 shows results from Kalman filtering for $R = 10^4$ and $Q = 10^{-4}$. In this trial, the measurement noise covariance in the Kalman filtering equations is weighted higher than the dynamic noise covariance. Here it is considered that there is more impact of the measurement noise on the filtered output than the dynamic noise. As observed in Figure 2, the filtered output largely matches the approximate average of measurement readings. The top plot consists Kalman filtered output with respect to measured data with reduced data points and bottom plot is the similar to top plot but with all the data points included. If we observe carefully, at initial stages the kalman filtered output takes time till 70 data points to match the average of measured data and achieves a stable state after 70 readings.

3.1.3 Trial 3

Figure 3 shows results from Kalman filtering for $R = 10^4$ and $Q = 10^{-4}$. In this trial, the measurement noise covariance in the Kalman filtering equations is weighted higher than the dynamic noise covariance. Here it is considered that there is significant impact of the measurement noise on the filtered output than the dynamic noise. As observed in Figure 2, the filtered output does not deviate from abnormalities in the measurement readings. The top plot consists Kalman filtered output with respect to measured data with reduced data points and bottom plot is the similar to top plot but with all the data points included. Since there is no impact of measurements on the Kalman filtered output, the straight line plotted corresponds to the initial reading of 0 with almost zero dynamic noise.

3.2 2-D Kalman Filtering

The Kalman filtered output of 2-D measured data is calculated and plotted. The X-position and Y-position measured and filtered data are plotted separately

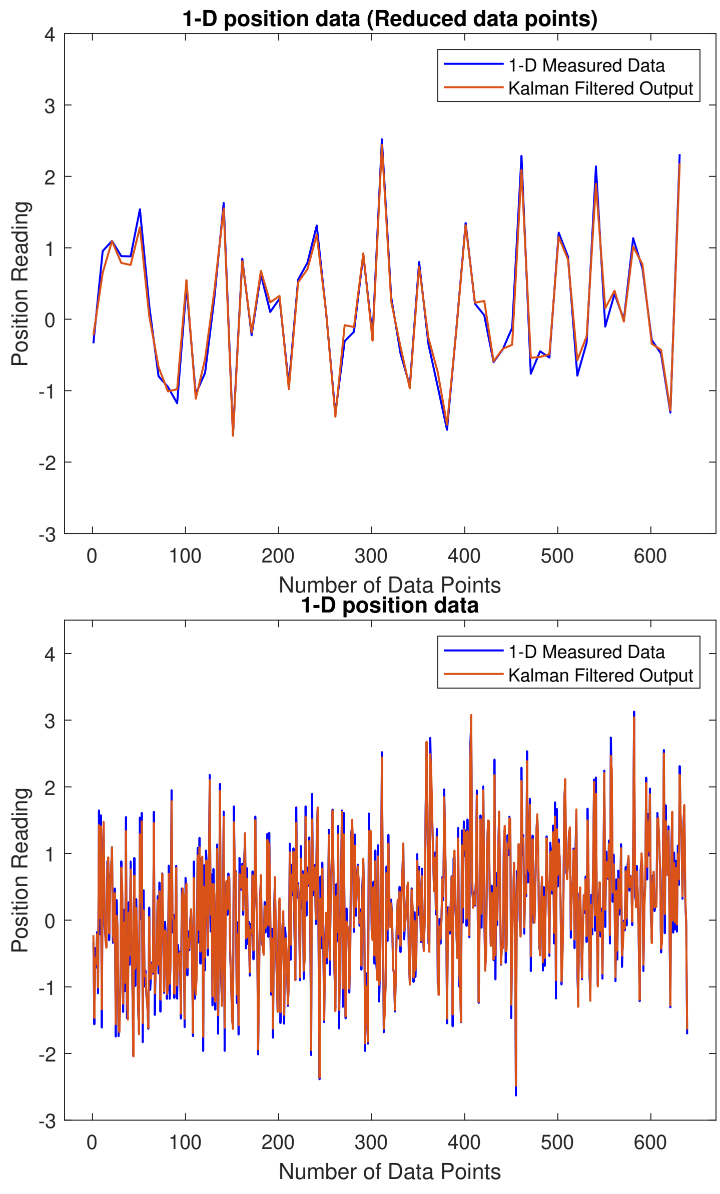

Figure 1: Trial 1 Results for Kalman filtering for R(σ_n^2) = 1 and Q(σ_a^2) = 10

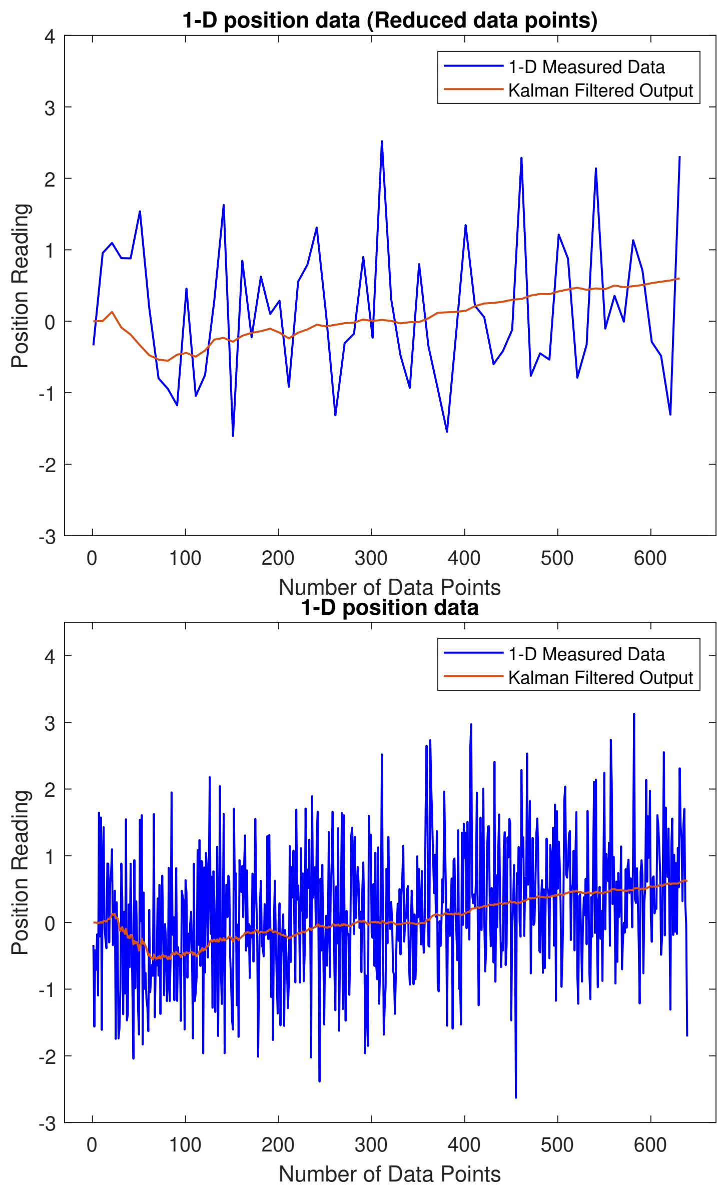

Figure 2: Trial 2 Results for Kalman filtering for $R(\sigma_n^2) = 10^4$ and $Q(\sigma_a^2) = 10^{-4}$

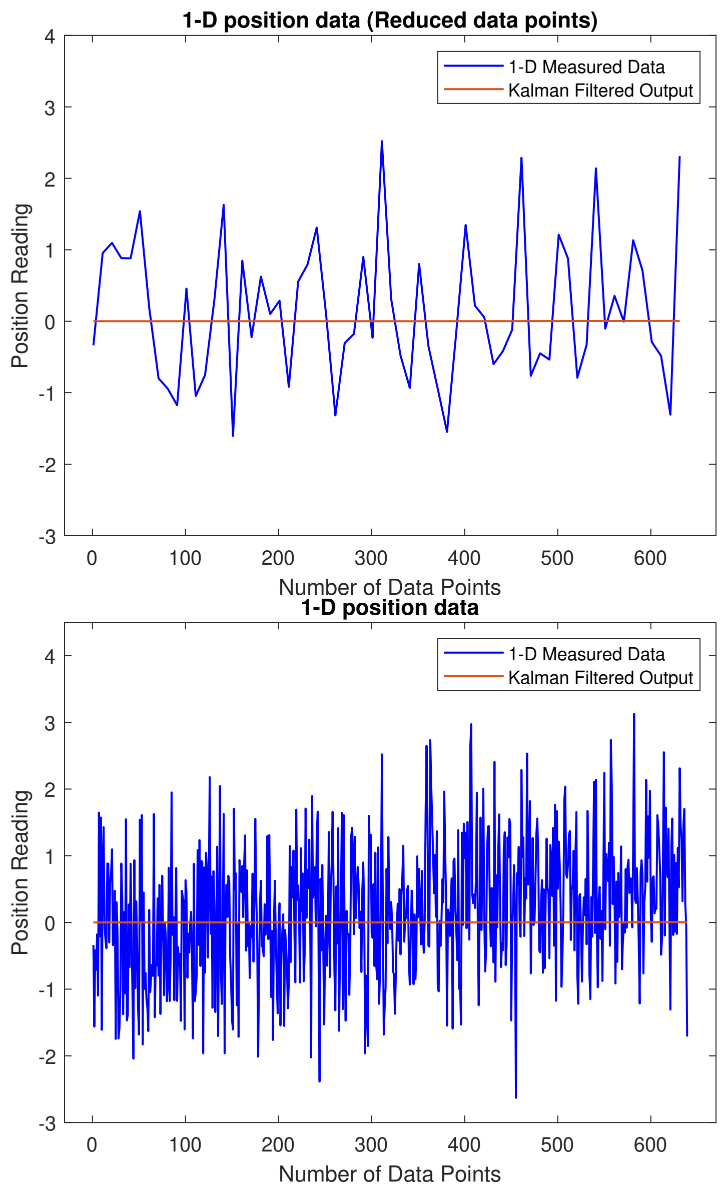

Figure 3: Trial 3 Results for Kalman filtering for $R(\sigma_n^2) = 10^{10}$ and $Q(\sigma_a^2) = 10^{-10}$

over number of data points. Different iterations are performed for multiple measurement noise covariance R and dynamic noise covariance Q. The Table 2 and 3 shows 3 iterations made for Q and R values to obtain the results against 2-D measured data.

Table 2: Trials of measurement noise covariance matrix R for 2-D Kalman filtering

Trial No	$\sigma^2_{n_1}$	σ_{n_1,n_2}	σ_{n_1,n_2}	$\sigma^2_{n_2}$
1	1	0.1	0.1	1
2	10^4	10^{-1}	10^{-1}	10^4
3	25	10^{-1}	10^{-1}	25

Table 3: Trials of dynamic noise covariance matrix Q for 2-D Kalman filtering

Trial No	$\sigma {a_1}^2$	σ_{a_1,a_2}	σ_{a_1,a_2}	$\sigma^2_{a_2}$
1	10	0.1	0.1	10
2	10^{-4}	10^{-3}	10^{-3}	10^{-4}
3	1	10^{-1}	10^{-1}	1

3.2.1 Trial 1

Figure 4 shows results from Kalman filtering for trial 1, R, and Q matrices values mentioned in Tables 2 and 3. X-axis represents a number of data points, and Y-axis represents the respective position data. In this trial, the measurement noise covariance in the Kalman filtering equations is weighted lower than the dynamic noise covariance. Here it is considered that there is more impact of the dynamic noise on the filtered output than the measurement noise. As observed in Figure 4, the filtered output largely matches the measurement readings. The top plot consists of Kalman-filtered output with respect to X-positions, and the bottom plot consists of Kalman-filtered output with respect to Y-positions.

3.2.2 Trial 2

Figure 5 shows results from Kalman filtering for trial 2, R, and Q matrices values mentioned in Tables 2 and 3. X-axis represents a number of data points, and Y-axis represents the respective position data. In this trial, the measurement noise covariance in the Kalman filtering equations is weighted higher than the dynamic noise covariance. Here it is considered that there is high impact of the measurement noise on the filtered output than the dynamic noise. As observed in Figure 5, the filtered output is largely deviating from measurement readings. The top plot consists of Kalman filtered output with respect to X-positions and the bottom plot consists Kalman filtered output with respect to Y-positions.

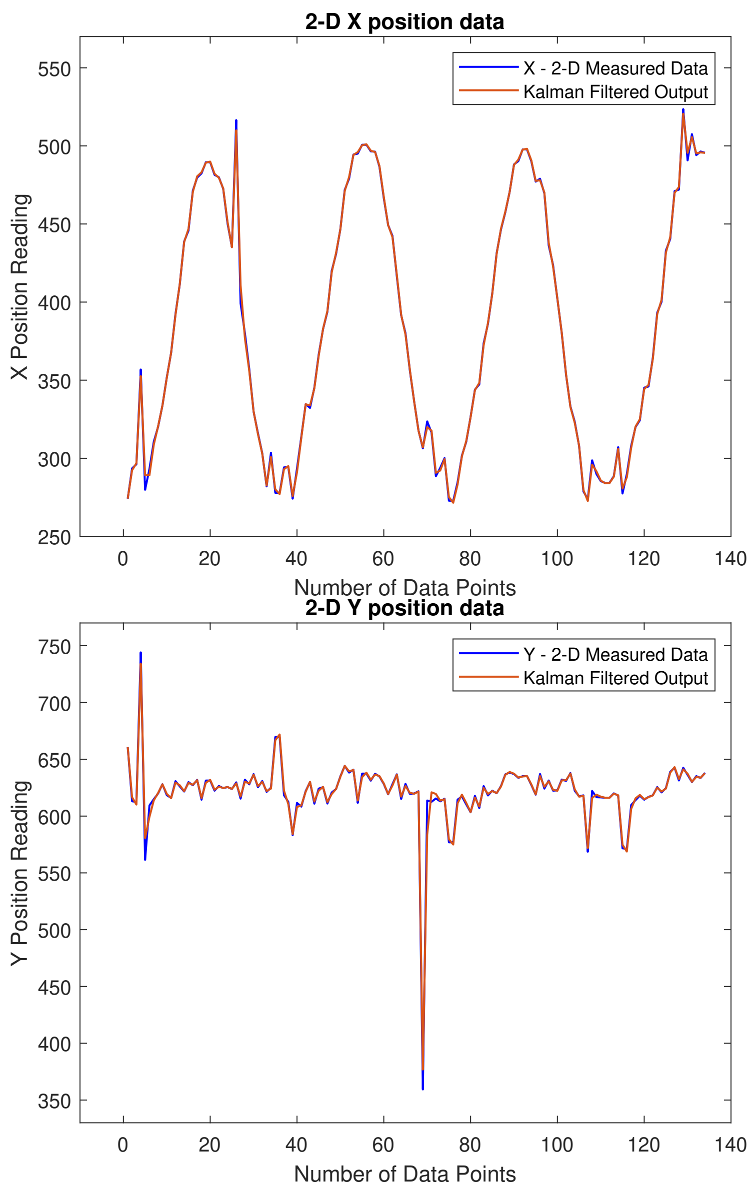

Figure 4: Trial 1 Results for 2-D Kalman filtering (Top - X-postion, Bottom - Y-position)

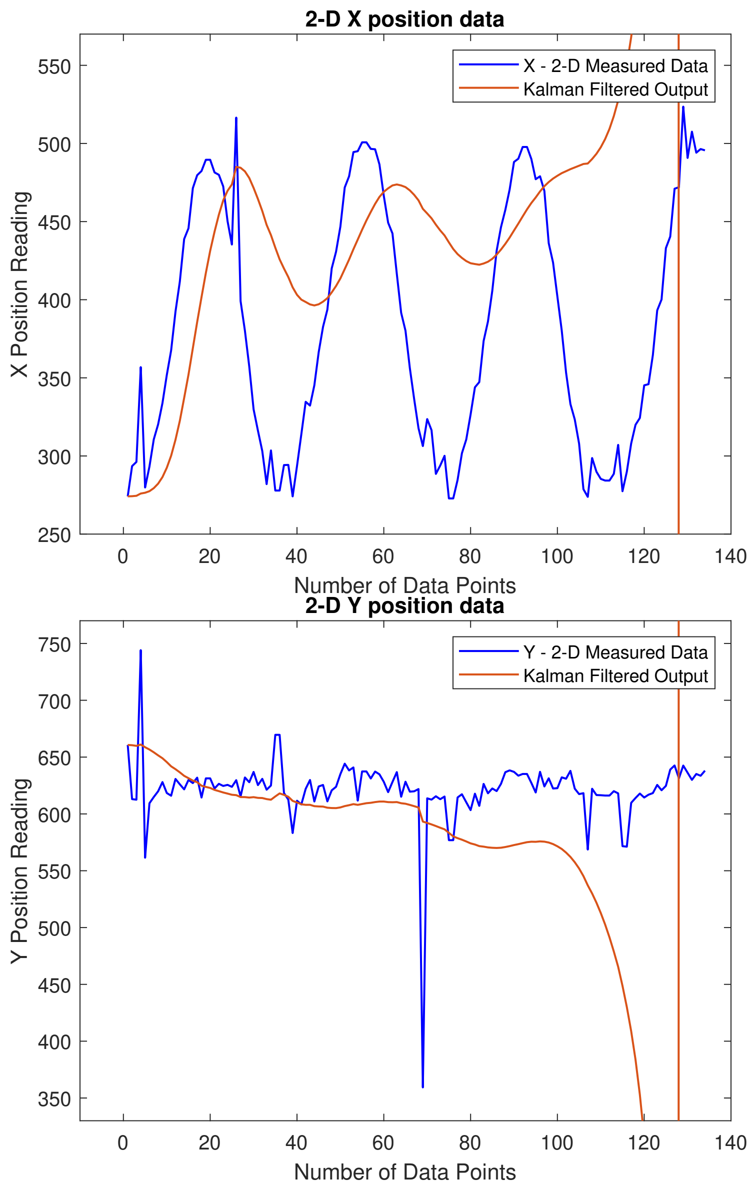

Figure 5: Trial 2 Results for 2-D Kalman filtering (Top - X-postion, Bottom - Y-position)

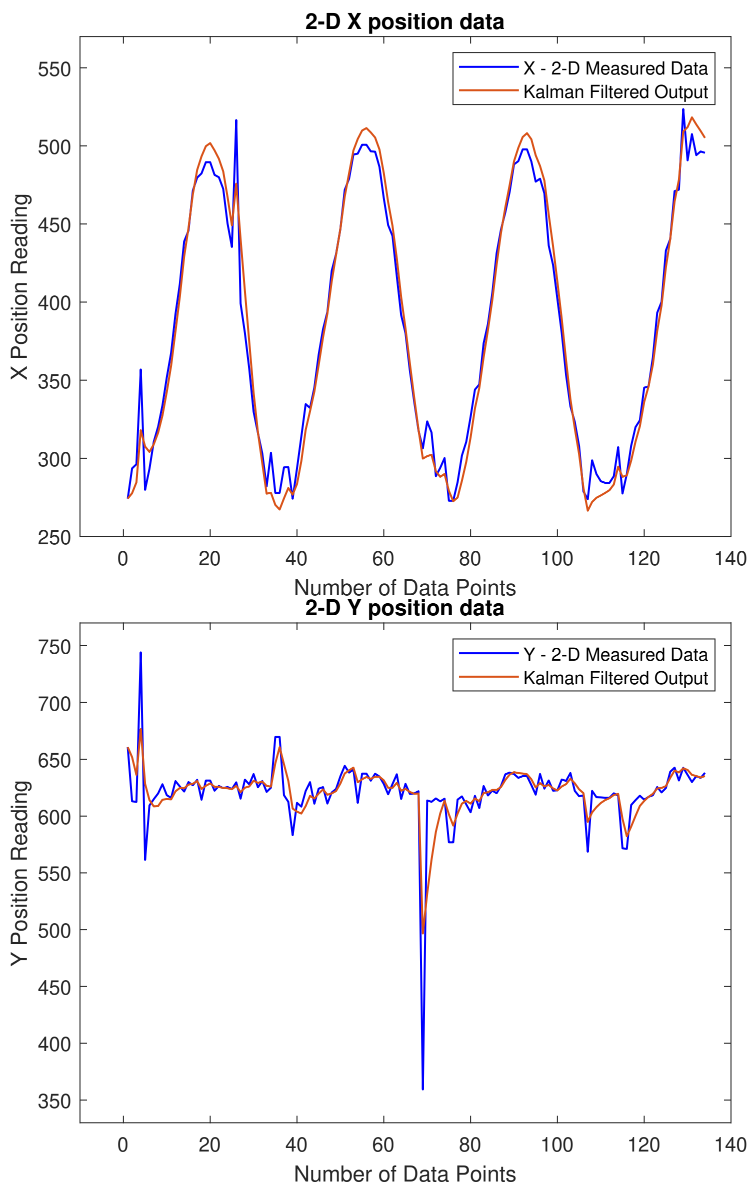

Figure 6: Trial 3 Results for 2-D Kalman filtering (Top - X-postion, Bottom - Y-position)

3.2.3 Trial 3

Figure 6 shows results from Kalman filtering for trial 3, R and Q matrices values mentioned in Tables 2 and 3. X-axis represents a number of data points, and Y-axis represents the respective position data. In this trial, the measurement noise covariance in the Kalman filtering equations is weighted significantly higher than the dynamic noise covariance. Here it is considered that there is more of the measurement noise on the filtered output than the dynamic noise, with some presence of dynamic noise. As observed in Figure 6, the filtered output does not deviate due to changes in measurement data largely. The Kalman filtered output is a balanced output between measurement noise and dynamic noise. The top plot consists of Kalman filtered output with respect to X-positions and the bottom plot consists of Kalman filtered output with respect to Y-positions.

4 Conclusion

Following are they key points that can be concluded from Kalman Filtering technique applied in t,his lab:

- Dynamic noise covariance and measurement noise covariance is to be weighed by the user to generate the accurate Kalman filtered output.
- Sensor properties providing accuracy are useful tools to determine the measurement noise covariance matrix. Such knowledge assists to get accurate Kalman filtered output.
- As depicted in Figure 2 and Trial 2 of 1-D Kalman filtering, Kalman filtered output generates accurate values after certain time has passed. In a dynamic system where Kalman filter is applied, it is always better to have a delay before getting accurate estimate from Kalman Filter.
- Accuracy of noise covariances can only be determined if we compare Kalman filtered output to the ground truth data.
- Kalman filtering is an excellent technique that is used broadly in various applications to obtain the best estimate of true value.

5 Appendix

The code for this lab can be found below:

```
% This Program is written by Mayuresh Bhosale that depicts Kalman Filter
% applied to 1-D and 2-D data

clc
clear all
close all
%1-D Data & Plot
Adata = importdata('1D-data.txt');
xaxis = 1:height(Adata);


%Initializing
x_t = [0; 0]; %Position and Velocity Initialization
Q_t = [0 1]; %Dynamic Noise
Y_t = 0; %Measurement
% R_t = 1; %Measurement Noise
S = [1 0; 0 1]; % State Covariance Matrix
Q_var = [0 0; 0 10^(-10)]; %Dynamic noise Covariance Matrix
R_var = 10^10; %Measurement Covariance Matrix
```

```
Phi = [1 1; 0 1]; %State transition Matrix
M = [1 0]; %Observation Matrix
I = eye(2); %Identity matrix

%Equations for Kalman Filtering

for i = 1:height(Adata)
    %Predict next state
    x_tt1 = Phi*x_t;
    %Predict Next State Covariance
    S_tt1 = Phi*S*Phi'+Q_var;
    %Obtain Measurements
    Y_t = Adata(i);
    %Calculating Kalman Gain
    K_t = S_tt1*M'* inv(M*S_tt1*M' + R_var);
    %Update State
    x_t = x_tt1 + K_t*(Y_t-M*x_tt1);
    X_t(i) = x_t(1);
    %Update State Covariance
    S = (I - K_t*M)*S_tt1;
end
t1 = tiledlayout(1,1);
nexttile
%Plotting Reduced Data
plot(xaxis(1:10:639),Adata(1:10:639),'b',"LineWidth",1);
xlabel('Number of Data Points');
ylabel('Position Reading');
title('1-D position data (Reduced data points)');
hold on
plot(xaxis(1:10:639),X_t(1:10:639),"LineWidth",1);
legend('1-D Measured Data','Kalman Filtered Output')
xlim([-30,670])
ylim([-3,4])
hold off
exportgraphics(t1,'Trial_4_Graph_1.eps')

t2 = tiledlayout(1,1);
nexttile
%Plotting Reduced Data
plot(xaxis,Adata,'b',"LineWidth",1);
xlabel('Number of Data Points');
ylabel('Position Reading');
title('1-D position data');
hold on
plot(xaxis,X_t,"LineWidth",1);
legend('1-D Measured Data','Kalman Filtered Output')
```

```
xlim([-30,670])
ylim([-3,4.5])
hold off
exportgraphics(t2,'Trial_4_Graph_2.eps')

% This Program is written by Mayuresh Bhosale that
    depicts Kalman Filter
% applied to 1-D and 2-D data

clc
clear all
close all
%1-D Data & Plot
Adata = importdata('2D-UWB-data.txt');
xaxis = 1:height(Adata);

%Initializing
X_t = [Adata(1,1); Adata(1,2); 0; 0]; %Position and
    Velocity Initialization
Q_t = [0 1]; %Dynamic Noise
Y_t = [0 0]; %Measurement
% R_t = [0.01 0.0001;
% %        0.001 0.01]; %Measurement Noise
% R_t = [1 0.1;
%        0.1 1]; %Measurement Noise

S = [1 0 0 0;
     0 1 0 0;
     0 0 1 0;
     0 0 0 1]; % State Covariance Matrix
Q_var = [0 0 0 0;
         0 0 0 0;
         0 0 1 0.1;
         0 0 0.1 1]; %Dynamic noise Covariance Matrix
R_var = [25 0.1;
         0.1 25]; %Measurement Covariance Matrix
Phi = [1 0 1 0;
       0 1 0 1;
       0 0 1 0;
       0 0 0 1]; %State transition Matrix
M = [1 0 0 0;
     0 1 0 0]; %Observation Matrix
I = eye(4); %Identity matrix

%Equations for Kalman Filtering

```

```
for i = 1:height(Adata)
    %Predict next state
    x_tt1 = Phi*X_t;
    %Predict Next State Covariance
    S_tt1 = Phi*S*Phi'+Q_var;
    %Obtain Measurements
    Y_t = [Adata(i,1); Adata(i,2)];
    %Calculating Kalman Gain
    K_t = S_tt1*M'* inv(M*S_tt1*M' + R_var);
    %Update State
    X_t = x_tt1 + K_t*(Y_t-M*x_tt1);
    X1(i) = X_t(1,1);
    X2(i) = X_t(2,1);
    %Update State Covariance
    S = (I - K_t*M)*S_tt1;
end

t1 = tiledlayout(1,1);
nexttile
%Plotting Reduced Data
plot(xaxis,Adata(:,1),'b',"LineWidth",1);
xlabel('Number of Data Points');
ylabel('X Position Reading');
title('2-D X position data');
hold on
plot(xaxis,X1,"LineWidth",1);
legend('X - 2-D Measured Data','Kalman Filtered Output')
xlim([-10,140])
ylim([250,570])
hold off
exportgraphics(t1,'2D_Trial_4_Graph_X.eps')

t2 = tiledlayout(1,1);
nexttile
%Plotting Reduced Data
plot(xaxis,Adata(:,2),'b',"LineWidth",1);
xlabel('Number of Data Points');
ylabel('Y Position Reading');
title('2-D Y position data');
hold on
plot(xaxis,X2,"LineWidth",1);
legend('Y - 2-D Measured Data','Kalman Filtered Output')
xlim([-10,140])
ylim([330,770])
hold off
exportgraphics(t2,'2D_Trial_4_Graph_Y.eps')
```