.

1 Introduction

The objective of Lab 3 is to implement a Kalman Filter. The lab is done in two parts. In Part 1, a Kalman Filter for a constant 1D velocity model is designed for the data "1D-data.txt". In Part 2, the Kalman filter is modified for a 2D model and the data used is provided in "2D-UWB-data.txt".

2 Methods

The Kalman filter for both parts make use of the following equations for each loop.

1. Prediction of next state:

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

2.Prediction of next state covariance:

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

3. Measurement $\mathrm{Y_t}$ is obtained from the data files "1D-data.txt" and "2D-UWB-data.txt" 4. Calculating Kalman gain:

$$K_t = S_{t,t-1} M^T [M S_{t,t-1} M^T + R]^{-1} \tag{3}$$

5. Updating states:

$$X_{t,t} = X_{t,t-1} + K_t (Y_t - M X_{t,t-1}) \tag{4}$$

6.Updating state covariances:

$$S_{t,t} = [I - K_t M] S_{t,t-1} \tag{5}$$

The values of Dynamic noise Q, Measurement noise R, Observation matrix M and covariance of state variables $\mathrm{S_t}$ are different for Part 1 and Part 2.

2.1 Part 1

In Part 1, a 1D velocity model is used. The observation matrix M for this model is,

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

The state variable covariance is given by,

$$\Phi = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

The dyanmic noise Q and measurement noise are varied for three different ratios: high Q and low R, low Q and high R and a balanced pair of Q and R. The values used for each ratio is given below:

1. High Q and Low R

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 0.5 \end{bmatrix} R = \begin{bmatrix} 0.0005 \end{bmatrix}$$

2. Low Q and High R

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 0.0005 \end{bmatrix} R = \begin{bmatrix} 0.5 \end{bmatrix}$$

3. Balanced Q and R

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 0.5 \end{bmatrix} R = \begin{bmatrix} 0.5 \end{bmatrix}$$

In Part 1, X$_($t-1,t-1) and S$_($t-1,t-1) are initialized to [0 0] and

$$S_{t-1,t-1} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

2.2 Part 2

In part 2, the 2D UWB data was applied to a 2D velocity model. The observation matrix used is,

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

The state variable covariance is given by,

$$\Phi = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The dyanmic noise Q and measurement noise are varied for three different ratios: high Q and low R, low Q and high R and a balanced pair of Q and R. The values used for each ratio is given below:

1. High Q and Low R

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.25 \\ 0 & 0 & 0.25 & 0.5 \end{bmatrix} R = \begin{bmatrix} 0.001 & 0.0005 \\ 0.0010.0005 & \end{bmatrix}$$

2. Low Q and High R

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0005 & 0.00025 \\ 0 & 0 & 0.00025 & 0.0005 \end{bmatrix} R = \begin{bmatrix} 1 & 0.5 \\ 10.5 & \end{bmatrix}$$

3. Balanced Q and R

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.25 \\ 0 & 0 & 0.25 & 0.5 \end{bmatrix} R = \begin{bmatrix} 1 & 0.5 \\ 10.5 & \end{bmatrix}$$

In Part 2, X$_($t-1,t-1) and S$_($t-1,t-1) are initialized as shown below,

$$X_{t-1,t-1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} S_{t-1,t-1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0.05 \\ 0 & 0 & 0.05 & 0.1 \end{bmatrix}$$

3 Results

3.1 Part 1

The results obtained by passing the 1D data through the Kalman filter designed in part 1 are as follows:

1. High dynamic noise Q and low measurement noise R

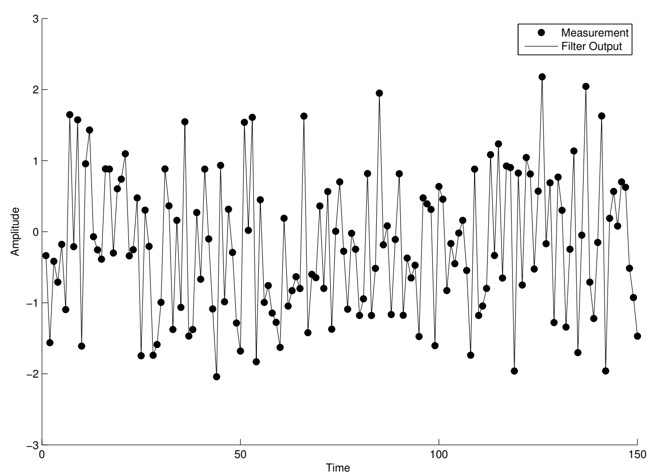

Figure 1: Plot for prediction of x (1D model-high Q, low R ratio)

2. Low dynamic noise Q and high measurement noise R

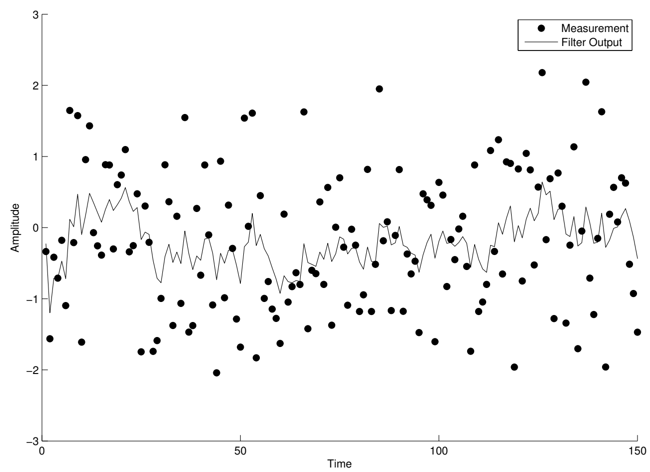

Figure 2: Plot for prediction of x (1D model-low Q, high R ratio)

3. Balanced Q and R

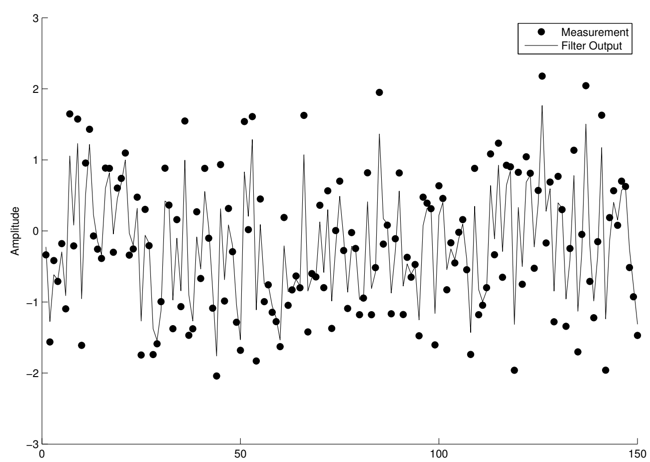

Figure 3: Plot for prediction of x (1D model-Balanced Q/R ratio)

3.2 Part 2

The results obtained by passing the 2D UWB data through the Kalman filter designed in part w are as follows:

1. High dynamic noise Q and low measurement noise R

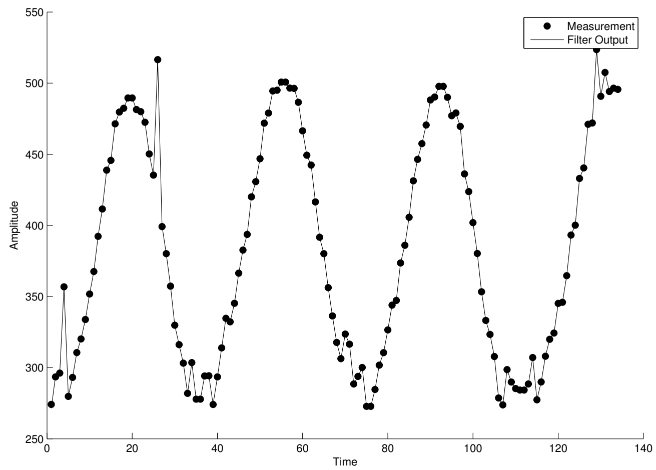

Figure 4: Plot for prediction of x (2D model-high Q, low R ratio)

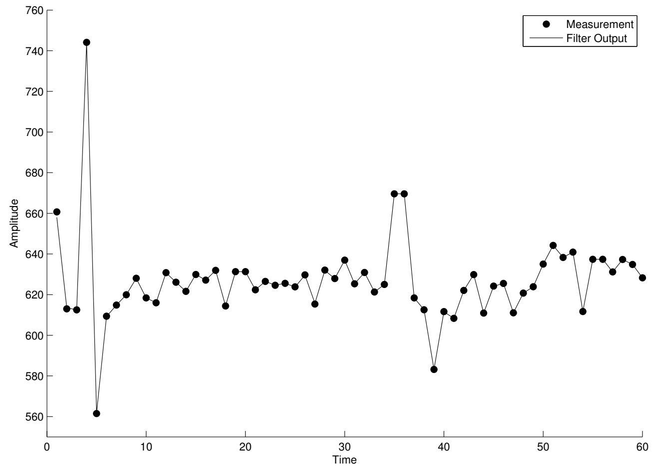

Figure 5: Plot for prediction of y (2D model-high Q, low R ratio)

2. Low dynamic noise Q and high measurement noise R

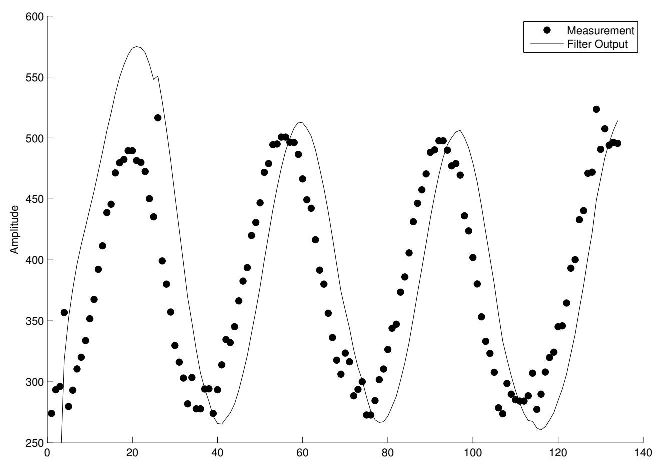

Figure 6: Plot for prediction of x (2D model-low Q, high R ratio)

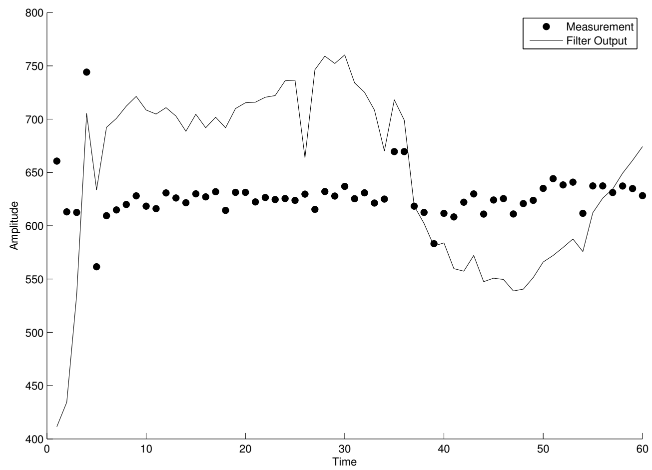

Figure 7: Plot for prediction of y (2D model-low Q, high R ratio)

3. Balanced Q and R

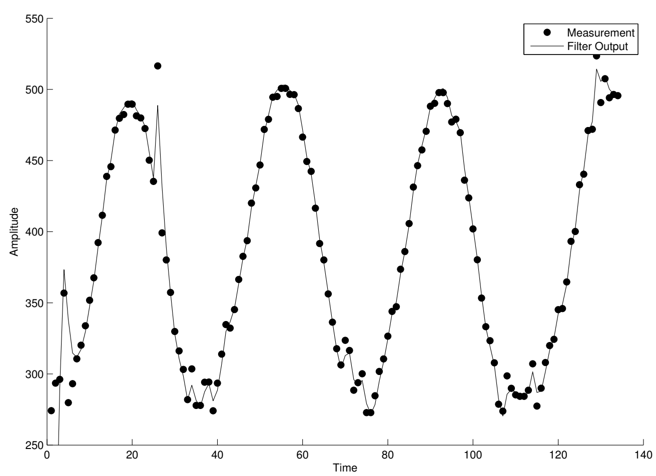

Figure 8: Plot for prediction of x (2D model-Balanced Q/R ratio)

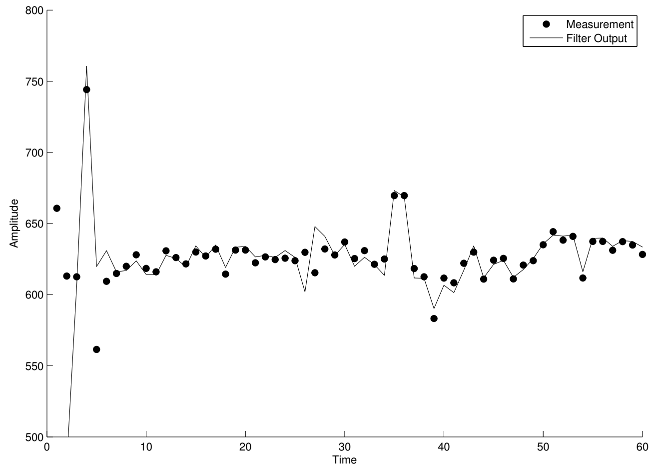

Figure 9: Plot for prediction of y (2D model-Balanced Q/R ratio)

3.3 Inference

From, the above plots it can be observerd that when the dynamic noise Q is set high compared to the measurement noise R, the filter ouput tends to follow the measurement values closely. When Q is set low and R is set to high values, the output doesnot closely follow the measurement data. For a balanced ratio, the output tends to follow the measurements with slight variations depending on the values of Q and R.

4 Conclusions

Thus, the Kalman filter has been applied to 1D and 2D models to predict the next states.

5 Appendix : Matlab Code

5.1 Part 1

```
clear;
Y_t=load('1D-data.txt');

X_t=[0 0]';
S_t=[1 1;0 1];
M=[1 0];
S=[0 0;0 1];
Q=[0 0; 0 0.0005];
R=.0005;

X_predict=S_t*X_t;
S_predict=S_t*S*S_t'+Q;

for i=1:size(Y_t)
    Kt=S_predict*M'*inv(M*S_predict*M'+R);
    X_upd=X_predict+Kt*(Y_t(i)-M*X_predict);
    S_update=(eye(2)-Kt*M)*S_predict;
    X_update(i)=X_upd(1);
    X_predict=S_t*X_update(i);
    S_predict=S_t*S_update*S_t'+Q;

end

scatter(1:1:length(Y_t),Y_t)
hold on
plot(X_update)
hold off
```

5.2 Part 2

```
clear;
Y_t=load('2D-UWB-data.txt');

X_t=[0 0 0 0]';
S_t=[1 0 1 0;0 1 0 1;0 0 1 0;0 0 0 1];
M=[1 0 0 0;0 1 0 0];
S=[0 0 0 0 ;0 0 0 0;0 0 0.1 0.05;0 0  0.05 0.1];
Q=[0 0 0 0 ;0 0 0 0;0 0 0.5 0.25;0 0  0.25 0.5];
R=[1 0.5;1 0.5];

X_predict=S_t*X_t;
```

```
S_predict=S_t*S*S_t'+Q;

for i=1:length(Y_t)
    Kt=S_predict*M'*inv(M*S_predict*M'+R);
    X_upd=X_predict+Kt*(Y_t(i,:)'-M*X_predict);
    S_update=(eye(4)-Kt*M)*S_predict;
    X_update(i)=X_upd(1);
    Y_update(i)=X_upd(2);
    X_predict=S_t*X_upd;
    S_predict=S_t*S_update*S_t'+Q;

end

figure
scatter(1:1:length(Y_t),Y_t(:,1))
hold on
plot(X_update)

figure
scatter(1:1:length(Y_t),Y_t(:,2))
hold on
plot(Y_update)
```