Tracking Systems
Kalman Filter

Mayukh Sattiraju
msattir@clemson.edu

ECE 854
Lab 4 Report

October 12, 2017

1 Introduction

This lab deals with application of the Kalman filter to track the position of a object in both 1D and 2D. The report starts with describing the system, building the Kalman filter equations and discussing the results of the tracking objective.

2 Derivation

2.1 Kalman Filter Equations

The implementation of the Kalman filter has 3 stages,

- Prediction Stage
- Measurement Stage
- Update Stage

The equations for the above stages are:

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

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

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

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

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

2.2 Implementation

The implementations of the Kalman filter to track a 1D and 2D position of a particle is described below.

2.2.1 1D Tracking

The state space equations that describe the position and velocity of a 1D particle is given as

$$\begin{aligned} x_t &= x_{t-1} + T\dot{x}_{t-1} \\ \dot{x}_t &= \dot{x}_{t-1} \end{aligned} \tag{6}$$

Here the states chosen are previous position and previous velocity.

$$X = \begin{bmatrix} x_{t-1} \\ \dot{x}_{t-1} \end{bmatrix}$$

The state transition matrix Φ and observation matrix M is now

$$\Phi = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad (7) \qquad\qquad M = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (8)$$

Constructing the equations for the Kalman filter using the above equations we are able to track the position of the particle. The tracking is below.

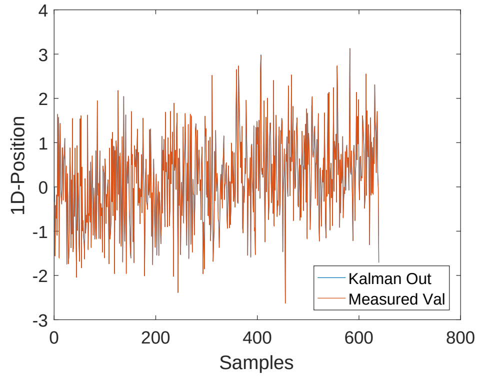

Figure 1: Kalman 1D tracking

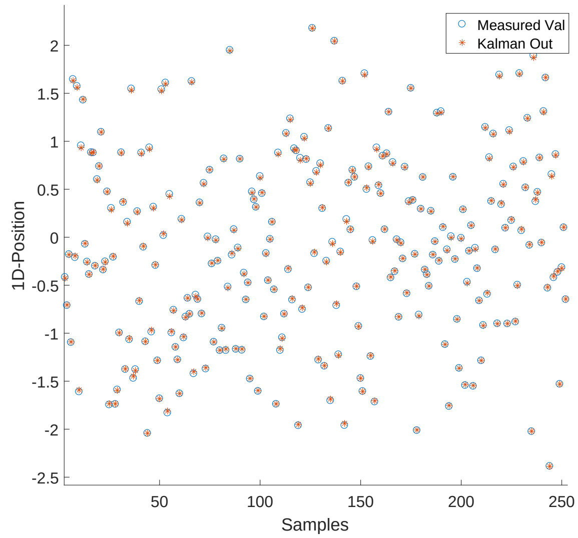

Figure 2: Zoom of Kalman 1D-Tracking

Fig 1 shows that the output of the Kalman filter almost overlaps the measured value y_t. A clearer zoom is shown in Fig 2. These graphs were obtained for $Q_{2,2} = 0.001$ and $R = 0.0001$

2.2.2 2D Tracking

The state space equations that describe the position and velocity of a 2D particle is given as

$$\begin{aligned} x_t &= x_{t-1} + T\dot{x}_{t-1} \\ y_t &= y_{t-1} + T\dot{y}_{t-1} \\ \dot{x}_t &= \dot{x}_{t-1} \\ \dot{y}_t &= \dot{y}_{t-1} \end{aligned} \tag{9}$$

Here the states chosen are x-position, y-position, x-velocity, y-velocity

$$X = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \dot{x}_{t-1} \\ \dot{y}_{t-1} \end{bmatrix} \tag{10}$$

The state transition matrix Φ and observation matrix M is now

$$\Phi = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{12}$$

Constructing the equations for the Kalman filter using the above equations we are able to track the position of the particle. The tracking is shown in the below figures.

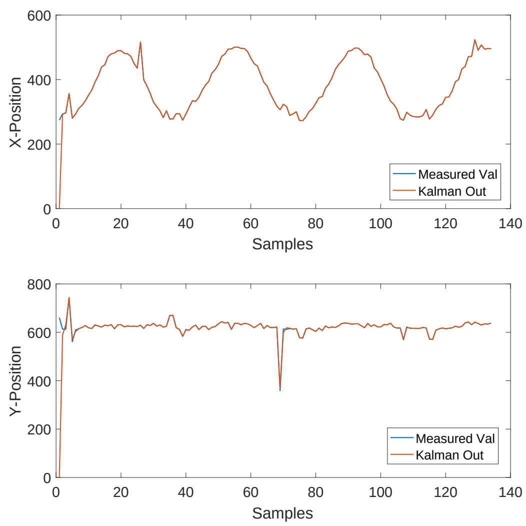

Figure 3: Kalman 2D tracking (X and Y)

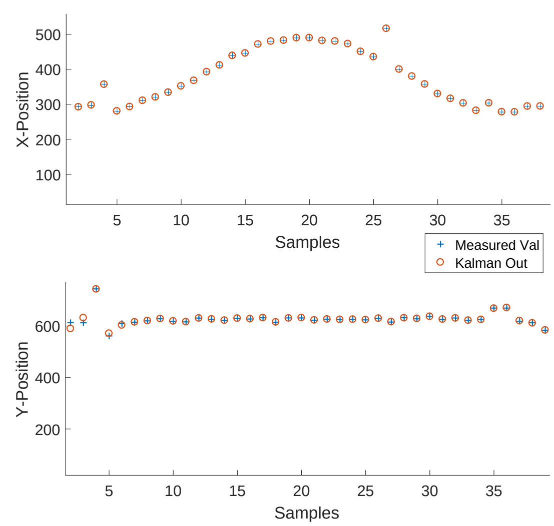

Figure 4: Zoom of Kalman 2D-Tracking

Fig 3 shows that the output of the Kalman filter almost overlaps the measured value y_t. A clearer zoom is shown in Fig 4. These graphs were obtained for $Q_{3,3} = Q_{4,4} = 0.0001; Q_{3,4} = Q_{4,3} = 0.00001$ and $R_{1,1} = 0.0001, R_{2,2} = 0.001, R_{1,2} = R_{2,1} = 0.00001$

3 Results

The Kalman filter was implemented for 1D and 2D position tracking, and the values of Measurement Noise Co-variance, **R** and Dynamic Noise Co-variance, **Q** are varied. This pushes the Kalman output towards the measured value or the predicted output.

3.1 1D tracking

The variation of values of Q and R, for the 1D tracking problem is shown in Fig.

As the value of Q becomes lower, the Kalman output stops tracing the Measured Value and it moves in between the predicted value and the measured Value. The specific Q and R values for this are $Q = 0.0001$ and $R = 0.01$

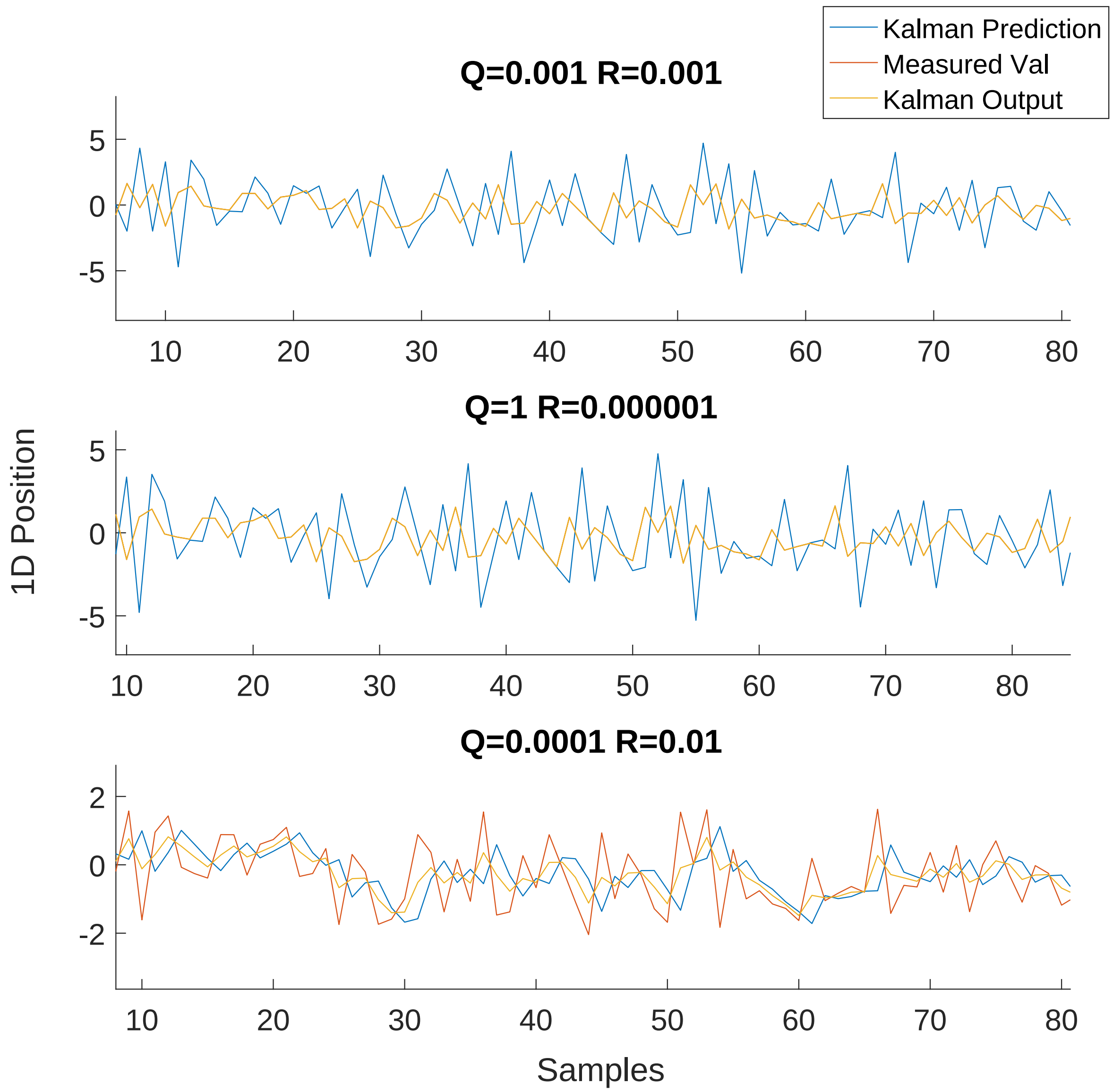

Figure 5: Varying Q and R for 1D Tracking

3.2 2D tracking

The variation of values of Q and R, for the 1D tracking problem is shown in Fig.

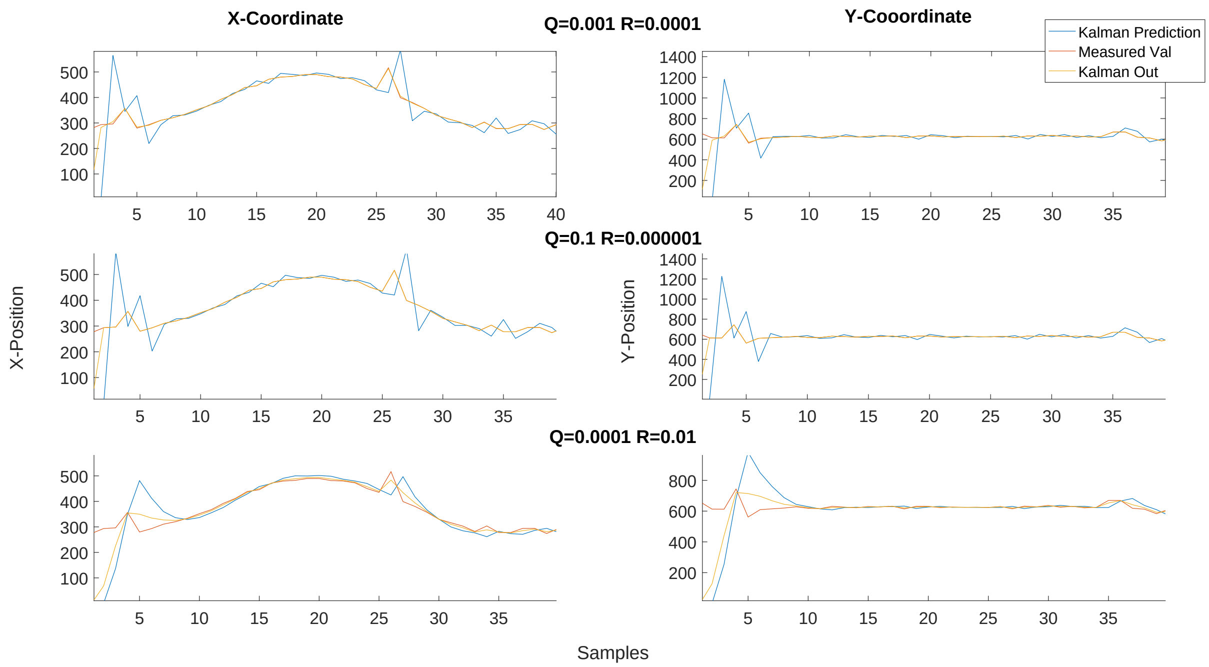

Figure 6: Varying Q and R for 2D Tracking

4 Code

The matlab codes included for the above implementations.

Listing 1: 1D Matlab Implementation

```
%clear all
x1=zeros(2,1);
x=zeros(2,1);
s1=zeros(2,2);
s=zeros(2,2);
phi=zeros(2,2);
q=zeros(2,2);
r=zeros(1,1);
k=zeros(2,1);
i=zeros(2,2);
m=zeros(1,1);

T=5;
i=eye(2);
phi=[1 T; 0 1];
q=[0 0; 0 1];
r=0.000001;
m=[1 0];
```