1 Introduction

In this example we introduce the classical Kalman Filter (KF) which is used in estimation theory for linear models and Gaussian random variables. We begin by introducing the problem setup for batch discrete time estimation and follow by definitions and implementations of the KF in a 2D localization problem.

1.1 Linear Gaussian Batch Discrete Time Estimation

We define the following motion and observation models:

$$\text{motion model: } \mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_k + \mathbf{w}_k, \quad k = 1 \ldots K \tag{1}$$

$$\text{observation model: } \mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{n}_k, \quad k = 0 \ldots K \tag{2}$$

Where we have:

$$\begin{aligned}
\text{system state } &: \mathbf{x}_k \in R^N \\
\text{initial state } &: \mathrm{x}_0 \in R^N \sim \mathcal{N}\left(\check{\mathrm{x}}_0, \check{\mathrm{P}}_0\right) \\
\text{input } &: \mathbf{v}_k \in R^N \\
\text{process noise } &: \mathbf{w}_k \in R^N \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}_k\right) \\
\text{measurement } &: \mathbf{y}_k \in R^M \\
\text{measurement noise } &: \mathbf{n}_k \in R^M \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}_k\right)
\end{aligned}$$

These are all random variables except for $\mathbf{v}_k$. The matrix $\mathbf{A}_k$ is known as the transition matrix. The matrix $\mathbf{C}_k$ is known as the observation matrix. We base our estimation on the state given the initial state knowledge, the inputs, and the measurements. In general our goal is to come up with an estimate of the true state given the initial estimate, a sequence of measurements, a sequence of inputs, as well as the motion and observation models. In [1], it was conveyed that the Bayesian interference and Maximum a posteriori (MAP) methods produce the same answer with regards to the Linear-Gaussian (LG) estimation problem. This is because in Gaussian distribution the mean and mode of the posterior are the same. This property does not hold true for non- Gaussian systems such as the famous Extended Kalman Filter.

1.2 Kalman Filter

Batch solutions face a particular drawback in that they cannot be used online due to required information from future states. To be used online, we need a solution which can use the data up until a current timestep. The Kalman filter is a classical solution to this problem. The Kalman filter has two steps: the prediction step, where the next state of the system is predicted from the last measurement that took place, and an update step, where the current state of the system is estimated by incorporating a new measurement.

Prediction Steps:

$$\check{\mathbf{P}}_k = \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_k \tag{3}$$

$$\check{\mathbf{x}}_k = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{v}_k \tag{4}$$

Kalman Gain:

$$\mathbf{K}_k = \mathbf{P}_k\mathbf{C}_k^T\left(\mathbf{C}_k\check{\mathbf{P}}_k\mathbf{C}_k^T + \mathbf{R}_k\right)^{-1} \tag{5}$$

Update Steps:

$$\mathbf{P}_k = \left(\mathbf{1} - \mathbf{K}_k\mathbf{C}_k\right)\mathbf{P}_k \tag{6}$$

$$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}_k\left(\mathbf{y}_k - \mathbf{C}_k\check{\mathbf{x}}_k\right) \tag{7}$$

The Kalman gain is to be interpreted as properly weighting the difference between the actual and expected measurements contribution to the estimate. The difference between the actual and expected measurements is also known as the innovation.

2 Implementation and Results

The problem at hand contained a velocity model of the robot given by:

$$\dot{x} = \frac{r}{2}\left(u_r + u_l\right) + w_x \quad , \quad \dot{y} = \frac{r}{2}\left(u_r + u_i\right) + w_y \tag{8}$$

where r = 0.1 m is the radius of the wheel, $\mathbf{u}_l$, and $\mathbf{u}_r$, are the control signals applied to the left and right wheels. Each wheel was assumed to have a fixed speed of 0.1 m/s. The process noise was given as $\mathbf{w}_x = N(0, 0.1)$, and $\mathbf{w}_y = N(0, 0.15)$ accordingly. The velocity model can be re-written in the form:

$$\dot{X} = \begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \begin{bmatrix}\frac{r}{2}(u_r + u_l) + w_x\\ \frac{r}{2}(u_r + u_l) + w_y\end{bmatrix} \tag{9}$$

The initial values were given as:

$$x_0 = 0, y_0 = 0, P_0 = \begin{bmatrix} 0 & 0\\ 0 & 0\end{bmatrix} \tag{10}$$

The motion model is computed 8 times a second. Lastly the measurement model was given as:

$$z_k = \begin{bmatrix}1 & 0\\ 0 & 2\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix} + \begin{bmatrix}r_x & 0\\ 0 & r_y\end{bmatrix} \tag{11}$$

where their corresponding noise terms were given as: $r_x \sim N(0, 0.05)$ and $r_y \sim N(0, 0.075)$. By noting that $\dot{X} = \frac{X_k - X_{k-1}}{T}$, we can re write the velocity model as:

$$\check{X}_k = \begin{bmatrix} \check{x} \\ \check{y} \end{bmatrix} = X_{k-1} + T\dot{X}_k \tag{12}$$

Where T = 1/8 seconds. By observation we can now compare our derived terms in this specific problem setup and make correspondences to the KF equations 3-7. We see that our transition matrix A_k takes the form of the identity matrix. Our z_k can be replaced by y_k. Lastly our covariance matrix of the process noise, and covariance matrix of our measurement noise take the form:

$$Q_k = \begin{bmatrix} w_x & 0 \\ 0 & w_y \end{bmatrix} \tag{13}$$

$$R_k = \begin{bmatrix} r_x & 0 \\ 0 & r_y \end{bmatrix} \tag{14}$$

The system was simulated for 10 seconds. This consisted of 8 motion predictions steps (Equations 3,4 and 12) applied per second following an update incorporating the measurement at that instant time. We can easily notice that the covariance of the state decreases after each measurement update. The animations were conducted on pygame, and scales were applied to the x and y positions, as well as the covariance of the state parameters accordingly, to allow for a much observable animation. The dark blue line represents the estimated trajectory, and the red dotted line represents the trajectory of all the predicted positions.

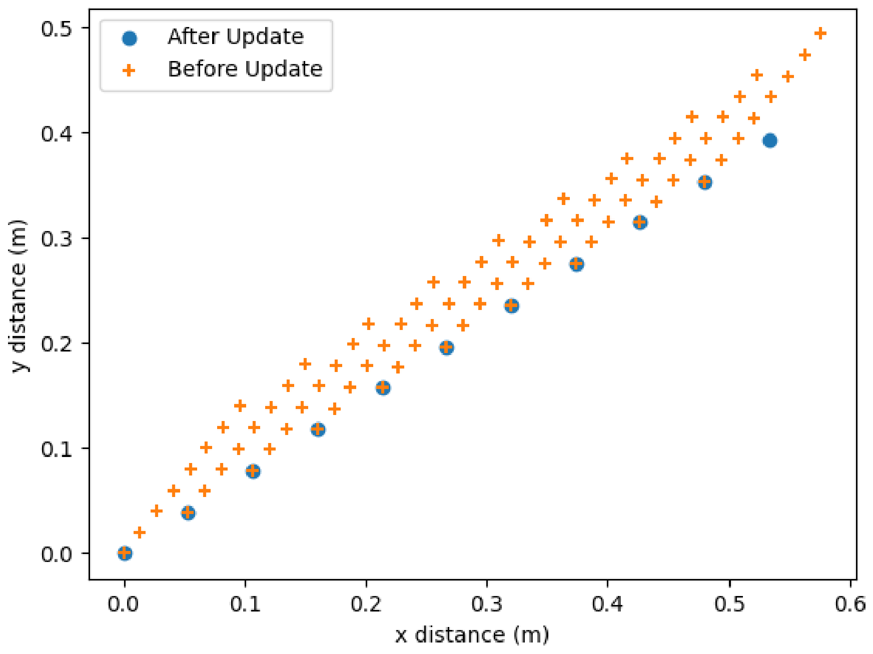

Figure 1: Kalman Filter Graphical Trajectory.

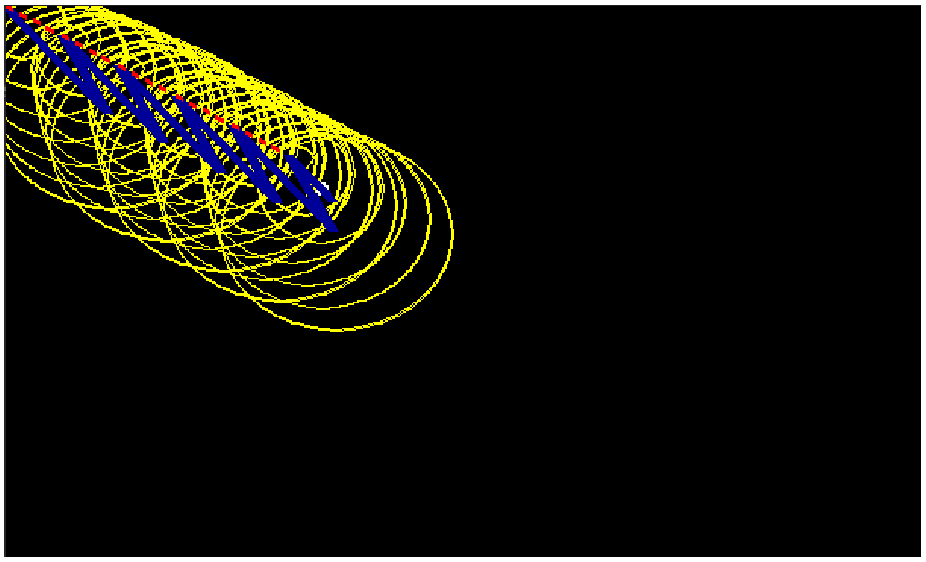

Figure 2: Kalman Filter Animation In Progress.

References

[1] T. D. Barfoot, *State Estimation for Robotics.* Cambridge University Press, 2020.