Kalman Filter HYPED

Trajectory team 2023-2024,
Main Author: Alejandro Lopez

March 13, 2024

Abstract

In this document we the Trajectory team will try to explain:

1. What is a Kalman Filter?
2. What it does?
3. Our implementation, *Input Kalman Filter*

1 What is a Kalman Filter?

Kalman filter is an iterative algorithm that estimates a linear dynamic system (although it can be modified to work on non linear systems) from a series of noisy measurements. The algorithm achieves this by minimizing the mean square error.

Is one of the best algorithms for both **noise cancelling** and **sensor fusion**.

2 Kalman Filter: What is does?

The Kalman filter is an iterative process where we first **predict** with our mathematical model, and then we **update** our estimate using the data from the measurement.

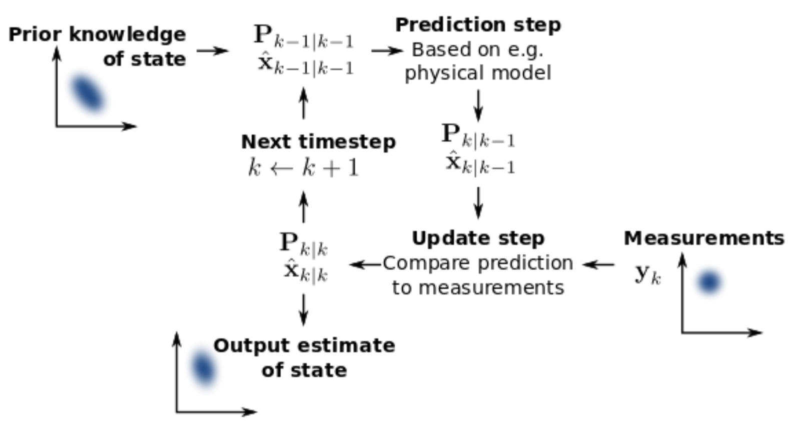

Figure 1: Diagram Kalman Filter, Wikipedia[1]

2.1 State and Measurement

The process of the Kalman Filter involves two main sets of equations: the state equation and the measurement equation, which are defined as follows:

- **State Equation**

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \tag{1}$$

- **Measurement Equation**

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \tag{2}$$

Where:

- $\mathbf{x}_k$ is the state vector at time k.
- $\mathbf{A}$ is the state transition matrix.
- $\mathbf{B}$ is the control input matrix.
- $\mathbf{u}_{k-1}$ is the control vector at time $k-1$.
- $\mathbf{w}_{k-1}$ is the process noise, assumed to be normally distributed with zero mean and covariance matrix $\mathbf{Q}$.
- $\mathbf{z}_k$ is the measurement vector at time k.
- $\mathbf{H}$ is the measurement matrix.
- $\mathbf{v}_k$ is the measurement noise, assumed to be normally distributed with zero mean and covariance matrix $\mathbf{R}$.

2.2 Kalman Filter Equations

The Kalman Filter operates in two phases: predict and update.

Predict Phase

- Predicted State Estimate

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} \tag{3}$$

- Predicted Covariance Estimate

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q} \tag{4}$$

Update Phase

- Kalman Gain

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

- Updated State Estimate

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-) \tag{6}$$

- Updated Covariance Estimate

$$\mathbf{P}_k = (I - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^- \tag{7}$$

The Kalman Filter recursively applies these predict and update phases as new measurements are made, allowing for the estimation to be refined over time.

3 Our implementation, *Input Kalman Filter*

The nature of spatial movement is non-linear, recall from SUVAT

$$x(t) = x_0 + v_0 t + \frac{1}{2} a t^2$$

Although this formula is one of the simplest for forecasting the spatial state (distance), it already shows non-linear behaviour.

We could have implemented a *Extended Kalman Filter* as described in *Bayesian Filtering and Smoothing* [2]. However we decided in favor of a more computational economic option as *Input Kalman Filter*, where we provide the model the acceleration in the matrix **B**.

3.1 Our model

With the same notation as 2.2 we will define the model applied.
<u>Motion Model</u>

$$\hat{x}_k = \begin{bmatrix} x_{k-1} \\ v_{k-1} \end{bmatrix} = \mathbf{A}\hat{x}_{k-1} + \mathbf{B}u_k = \begin{bmatrix} 1 & \triangle t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ v_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\triangle t^2 \\ \triangle t \end{bmatrix} a_{IMU}$$

P is the covariance matrix

$$\mathbf{P} = \begin{bmatrix} \sigma^2_{x,x} & \sigma_{x,v} \\ \sigma_{x,v} & \sigma^2_{v,v} \end{bmatrix} \quad \text{and} \quad \mathbf{P}_0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{initially there is no correlation}$$

Q is the process noise matrix.

$$\mathbf{Q} = \begin{bmatrix} \frac{1}{4}\triangle t^4 & \frac{1}{2}\triangle t^3 \\ \frac{1}{2}\triangle t^3 & \triangle t^2 \end{bmatrix} \sigma^2_{IMU}$$

<u>Measurement Model</u>
z is the measurement from sensors, note that here we have also defined **H**.

$$\mathbf{z} = \begin{bmatrix} x_{\text{ Stripe}} \\ \triangle x_{\text{ Optical}} \end{bmatrix} = \begin{cases} \mathbf{H}\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \triangle t \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} & \text{if only optical flow data available} \\ \mathbf{H}\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \triangle t \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} & \text{if flow data and stripe data are available} \end{cases}$$

With the assumption of strip counter to be accurate, the sensor matrix **R** is:

$$\mathbf{R} = \begin{bmatrix} \sigma^2{}_{\text{Optical}} & 0 \\ 0 & 0 \end{bmatrix}$$

With this specifications we apply the Kalman Filter algorithm as done in 2.2.

References

[1] Wikipedia contributors, "Kalman filter," *Wikipedia, The Free Encyclopedia*, [Online; accessed 13-March-2024]. Available: `https://en.wikipedia.org/wiki/Kalman_filter`.

[2] Simo Särkkä, "Bayesian Filtering and Smoothing," 2013, pp. 51–92.