Kalman Filter

Ugenteraan Manogaran

E-mail : m.ugenteraan15@gmail.com

29 December 2019

Contents

1.0 Problem Statement

Time series data from sensors and other measuring instruments almost certainly contain inaccuracies and statistical noise. In some cases, the readings may even have relatively long delays in between the readings. Therefore, the need to optimally predict and estimate the data points from the uncertain measurements are needed.

Assuming that the internal *state of a system*,[1] $\mathbf{x}_k$ at time k can be linearly modelled, the internal state can be represented as such:

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

where $\mathbf{A}$ is a state transition matrix, $\hat{\mathbf{x}}_{k+1}$ is the estimated state on the previous time step and $\mathbf{w}_k$ is the associated white noise which is assumed to be normally distributed. However, the measurement/observation from the measuring instruments or sensors may not contain every information that are in our state vectors, $\mathbf{x}_k$ or $\hat{\mathbf{x}_k}$. For example, a sensor might only provide the reading of the velocity of a moving car. Since displacement of the car can be derived from the velocity, we could use the state vectors to predict the displacement and the velocity. Therefore, the measurement vector, $\mathbf{z}_k$ at time k can be modelled as such :

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

where $\mathbf{H}$ is the noiseless connection between the state vector and the measurement vector and $\mathbf{v}_k$ is the associated measurement error which is also assumed to be normally distributed. In order to find the estimates of the measurements, some sort of averaging or weighting process is required. For demonstration purpose, let's assume that we want to simply average our measurements to provide an estimate, $\hat{\mathbf{x}}_k$ at time k. Therefore,

$$\hat{\mathbf{x}}_k = \frac{1}{k}\sum_{j=1}^{k}\mathbf{z}_j \tag{3}$$

By rearranging the above equation we obtain the following:

[1] A set of measurable properties such as velocity, displacement, acceleration, etc.

$$
\begin{aligned}
\hat{\mathbf{x}}_k &= \frac{1}{k}\sum_{j=1}^{k}\mathbf{z}_j = \frac{1}{k}(\sum_{j=1}^{k-1}\mathbf{z}_j + \mathbf{z}_k) = \frac{1}{k}\sum_{j=1}^{k-1}\mathbf{z}_j + \frac{1}{k}\mathbf{z}_k \\
&= (\frac{1}{k})(\frac{k-1}{k-1})\sum_{j=1}^{k-1}\mathbf{z}_j + \frac{1}{k}\mathbf{z}_k \\
&= (\frac{k-1}{k})(\frac{1}{k-1})\sum_{j=1}^{k-1}\mathbf{z}_j + \frac{1}{k}\mathbf{z}_k \\
&= (\frac{k-1}{k})\hat{\mathbf{x}}_{k-1} + \frac{1}{k}\mathbf{z}_k \\
&= \frac{k(\hat{\mathbf{x}}_{k-1}) - \hat{\mathbf{x}}_{k-1}}{k} + \frac{1}{k}\mathbf{z}_k \\
&= \hat{\mathbf{x}}_{k-1} + \frac{1}{k}(\mathbf{z}_k - \hat{\mathbf{x}}_{k-1})
\end{aligned}
$$

The final equation :

$$
\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \frac{1}{k}(\mathbf{z}_k - \hat{\mathbf{x}}_{k-1}) \tag{4}
$$

The final equation implies that to get the estimate of the state at time k, we can use the previous estimate added with the average of the difference between the current measurement and the previous estimate. The equation $\mathbf{z}_k - \hat{\mathbf{x}}_{k-1}$ is known as *innovation*.

In order to get an optimal estimation, we should strive to minimise the difference between our prediction and estimation (*error*). In short, the predicted state (without considering the measurement) should be as close as possible to the estimate (considering the measurement). As a result, the prediction will get better over time. Therefore, instead of averaging the data naively, we need a weight that considers the reliability of our measurements and prediction and at the same time reduce the error.

2.0 Kalman Filter

Kalman Filter (**a.k.a.** Linear Quadratic Estimation **(LQE)**) is an algorithm that estimates the state of a system from *indirect*[2] and/or uncertain measurements such that the *estimation error*[3], e is minimised.

[2] A measurement value that can be used to derive other properties such as acceleration from velocity.

[3] The difference between the predicted vector and the estimated vector

2.1 Estimation Error

Our error function should be positive and monotonically increasing. Mean squared error function exhibit the said properties. Therefore, our estimation error function, $f(e)$ would be

$$f(e) = (\hat{\mathbf{x}}_k - \mathbf{x}_k)^2 \tag{5}$$

Since both the prediction and the estimation has the same mean,

$$\mathbf{E}[(\hat{\mathbf{x}}_k - \mathbf{x}_k)] = 0 \tag{6}$$

Note that:

$$\mathbf{Q} = cov(\mathbf{w}_k, \mathbf{w}_k) = \mathbf{E}[\mathbf{w}_k \mathbf{w}_k^{\mathbf{T}}]$$
$$\mathbf{R} = cov(\mathbf{v}_k, \mathbf{v}_k) = \mathbf{E}[\mathbf{v}_k \mathbf{v}_k^{\mathbf{T}}]$$

and since the mean of the error term is zero [6],

$$\mathbf{P} = cov(\hat{\mathbf{x}}_k - \mathbf{x}_k, \hat{\mathbf{x}}_k - \mathbf{x}_k) = \mathbf{E}[(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^{\mathbf{T}}]$$

Borrowing the idea from **Equation** 4, we can derive an estimate for time k with

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \mathbf{K}(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k-1}) \tag{7}$$

where $\hat{\mathbf{x}}_k$ is the estimate at time k, $\hat{\mathbf{x}}_{k-1}$ is the prediction at time *k-1*, $\mathbf{K}$ is the weight that controls how much contribution does the prediction and the measurement each makes towards the estimation, and $\mathbf{z}_k$ is the measurement at time k. **A more rigorous proof for the derivation of the estimate can be found in [1].**

2.2 Error Covariance Matrix, P

Consider the the vectors, $\hat{\mathbf{x}}_k$, $\mathbf{x}_k$, and $\hat{\mathbf{x}}_k - \mathbf{x}_k$ as below:

$$\hat{\mathbf{x}}_k = \begin{bmatrix} \hat{x}_k^{(1)} \\ \hat{x}_k^{(2)} \\ . \\ . \\ . \\ \hat{x}_k^{(n)} \end{bmatrix}, \quad \mathbf{x}_k = \begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \\ . \\ . \\ . \\ x_k^{(n)} \end{bmatrix}, \quad \hat{\mathbf{x}}_k - \mathbf{x}_k = \begin{bmatrix} \hat{x}_k^{(1)} - x_k^{(1)} \\ \hat{x}_k^{(2)} - x_k^{(2)} \\ . \\ . \\ . \\ \hat{x}_k^{(n)} - x_k^{(n)} \end{bmatrix}$$

The error covariance matrix $\mathbf{P}$, would be:

$$\mathbf{P} = \begin{bmatrix} \hat{x}_k^{(1)} - x_k^{(1)} \\ \hat{x}_k^{(2)} - x_k^{(2)} \\ \cdot \\ \cdot \\ \cdot \\ \hat{x}_k^{(n)} - x_k^{(n)} \end{bmatrix} \begin{bmatrix} \hat{x}_k^{(1)} - x_k^{(1)} & \hat{x}_k^{(2)} - x_k^{(2)} & \cdot & \cdot & \cdot & \hat{x}_k^{(n)} - x_k^{(n)} \end{bmatrix}$$

$$= \begin{bmatrix} (\hat{x}_k^{(1)} - x_k^{(1)})^2 & (\hat{x}_k^{(1)} - x_k^{(1)})(\hat{x}_k^{(2)} - x_k^{(2)}) & \cdot\ \cdot\ \cdot & (\hat{x}_k^{(1)} - x_k^{(1)})(\hat{x}_k^{(n)} - x_k^{(n)}) \\ (\hat{x}_k^{(1)} - x_k^{(1)})(\hat{x}_k^{(2)} - x_k^{(2)}) & (\hat{x}_k^{(2)} - x_k^{(2)})^2 & \cdot\ \cdot\ \cdot & (\hat{x}_k^{(2)} - x_k^{(2)})(\hat{x}_k^{(n)} - x_k^{(n)}) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ (\hat{x}_k^{(n)} - x_k^{(n)})(\hat{x}_k^{(1)} - x_k^{(1)}) & (\hat{x}_k^{(n)} - x_k^{(n)})(\hat{x}_k^{(2)} - x_k^{(2)}) & \cdot\ \cdot\ \cdot & (\hat{x}_k^{(n)} - x_k^{(n)})^2 \end{bmatrix}$$

Notice that the sum of the diagonal entries of the matrix are the mean squared error, $f(e)$. Therefore, the trace of the matrix $\mathbf{P}$, $\mathbf{Tr(P)}$ is

$$\mathbf{Tr(P)} = f(e) \tag{8}$$

2.3 Minimizing Estimation Error

Let's start off by expanding the error term $(\hat{\mathbf{x}}_k - \mathbf{x}_k)$.

$$\begin{aligned}
(\hat{\mathbf{x}}_k - \mathbf{x}_k) &= (\hat{\mathbf{x}}_{k-1} + \mathbf{K}(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k-1}) - \mathbf{x}_k) \\
&= (\hat{\mathbf{x}}_{k-1} + \mathbf{K}((\mathbf{H}\mathbf{x}_k + \mathbf{v}_k) - \mathbf{H}\hat{\mathbf{x}}_{k-1}) - \mathbf{x}_k) \\
&= (\hat{\mathbf{x}}_{k-1} + \mathbf{K}\mathbf{H}\mathbf{x}_k + \mathbf{K}\mathbf{v}_k - \mathbf{K}\mathbf{H}\hat{\mathbf{x}}_{k-1} - \mathbf{x}_k) \\
&= (\hat{\mathbf{x}}_{k-1} - \mathbf{x}_k + \mathbf{K}\mathbf{H}\mathbf{x}_k - \mathbf{K}\mathbf{H}\hat{\mathbf{x}}_{k-1} + \mathbf{K}\mathbf{v}_k) \\
&= ((\mathbf{I} - \mathbf{K}\mathbf{H})(\hat{\mathbf{x}}_{k-1} - \mathbf{x}_k) + \mathbf{K}\mathbf{v}_k)
\end{aligned}$$

Plugging the expanded term into $\mathbf{P}$ and simplifying it would produce:

$$\begin{aligned}
\mathbf{P} &= \mathbf{E}[((\mathbf{I} - \mathbf{K}\mathbf{H})(\hat{\mathbf{x}}_{k-1} - \mathbf{x}_k) + \mathbf{K}\mathbf{v}_k)((\mathbf{I} - \mathbf{K}\mathbf{H})(\hat{\mathbf{x}}_{k-1} - \mathbf{x}_k) + \mathbf{K}\mathbf{v}_k)^{\mathbf{T}}] \\
&= (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{E}[(\hat{\mathbf{x}}_{k-1} - \mathbf{x}_k)(\hat{\mathbf{x}}_{k-1} - \mathbf{x}_k)^{\mathbf{T}}](\mathbf{I} - \mathbf{K}\mathbf{H})^{\mathbf{T}} + \mathbf{K}\mathbf{E}[\mathbf{v}_k\mathbf{v}_k^{\mathbf{T}}]\mathbf{K}^{\mathbf{T}}
\end{aligned}$$

Notice that $\mathbf{E}[(\hat{\mathbf{x}}_{k-1} - \mathbf{x}_k)(\hat{\mathbf{x}}_{k-1} - \mathbf{x}_k)^{\mathbf{T}}]$ is actually the prior estimate error, $\mathbf{P}_{k-1}$ and $\mathbf{E}[\mathbf{v}_k\mathbf{v}_k^{\mathbf{T}}] = \mathbf{R}$. Therefore,

$$\begin{aligned}\mathbf{P} &= (\mathbf{I}-\mathbf{KH})\mathbf{P}_{k-1}(\mathbf{I}-\mathbf{KH})^{\mathbf{T}} + \mathbf{KRK}^{\mathbf{T}} \\ &= (\mathbf{P}_{k-1} - \mathbf{K}_k\mathbf{HP}_{k-1})(\mathbf{I}-\mathbf{K}_k\mathbf{H})^{\mathbf{T}} + \mathbf{K}_k\mathbf{RK}_k^{\mathbf{T}} \\ &= \mathbf{P}_{k-1} - \mathbf{K}_k\mathbf{HP}_{k-1} - \mathbf{P}_{k-1}\mathbf{H}^{\mathbf{T}}\mathbf{K}_k^{\mathbf{T}} + \mathbf{K}_k\mathbf{HP}_{k-1}\mathbf{H}^{\mathbf{T}}\mathbf{K}_k^{\mathbf{T}} + \mathbf{K}_k\mathbf{RK}_k^{\mathbf{T}} \\ &= \mathbf{P}_{k-1} - \mathbf{K}_k\mathbf{HP}_{k-1} - \mathbf{P}_{k-1}\mathbf{H}^{\mathbf{T}}\mathbf{K}_k^{\mathbf{T}} + \mathbf{K}_k(\mathbf{HP}_{k-1}\mathbf{H}^{\mathbf{T}} + \mathbf{R})\mathbf{K}_k^{\mathbf{T}}\end{aligned}$$

Since minimizing the estimation error is equivalent to minimizing $\mathbf{Tr(P)}$, we want to find $\mathbf{K}$ such that $\frac{d}{d\mathbf{K}_k}\mathbf{Tr(P)} = 0$. Noting that the trace of a matrix equals to the trace of its transpose,

$$\begin{aligned}\frac{d}{d\mathbf{K}_k}\mathbf{Tr(P)} &= \mathbf{Tr}(\mathbf{P}_{k-1}) - 2\mathbf{Tr}(\mathbf{K}_k\mathbf{HP}_{k-1}) + \mathbf{Tr}(\mathbf{K}_k(\mathbf{HP}_{k-1}\mathbf{H}^{\mathbf{T}} + \mathbf{R})\mathbf{K}^{\mathbf{T}}) \\ &= -2(\mathbf{HP}_{k-1})^{\mathbf{T}} + \mathbf{K}_k(\mathbf{HP}_{k-1}\mathbf{H}^{\mathbf{T}} + \mathbf{R})^{\mathbf{T}} + \mathbf{K}_k(\mathbf{HP}_{k-1}\mathbf{H}^{\mathbf{T}} + \mathbf{R})\end{aligned}$$

Since $\mathbf{R}^{\mathbf{T}} = \mathbf{R}$,

$$\frac{d}{d\mathbf{K}_k}\mathbf{Tr(P)} = -2(\mathbf{HP}_{k-1})^{\mathbf{T}} + 2\mathbf{K}_k(\mathbf{HP}_{k-1}\mathbf{H}^{\mathbf{T}} + \mathbf{R})$$

By setting $\frac{d}{d\mathbf{K}_k}\mathbf{Tr(P)} = 0$,

$$\begin{aligned}-2(\mathbf{HP}_{k-1})^{\mathbf{T}} + 2\mathbf{K}_k(\mathbf{HP}_{k-1}\mathbf{H}^{\mathbf{T}} + \mathbf{R}) &= 0 \\ \mathbf{K}_k(\mathbf{HP}_{k-1}\mathbf{H}^{\mathbf{T}} + \mathbf{R}) &= (\mathbf{HP}_{k-1})^{\mathbf{T}} \\ \mathbf{K} &= \mathbf{P}_{k-1}\mathbf{H}^{\mathbf{T}}(\mathbf{HP}_{k-1}\mathbf{H}^{\mathbf{T}} + \mathbf{R})^{-1}\end{aligned}$$

$$\mathbf{K} = \mathbf{P}_{k-1}\mathbf{H}^{\mathbf{T}}(\mathbf{HP}_{k-1}\mathbf{H}^{\mathbf{T}} + \mathbf{R})^{-1} \tag{9}$$

$\mathbf{K}$ is the weight that will be used for our estimation calculation. Note that the measurement error, $\mathbf{R}$ is an addition to the inverse matrix. This means that as $\mathbf{R}$ gets bigger, $\mathbf{K}$ will approach 0. Since $\mathbf{K}$ is applied to the *innovation* term, the measurement will not be contributing much or at all to the estimation calculation. $\mathbf{K}$ is known as **Kalman Gain**.

For simplicity sake, let

$$\mathbf{S} = \mathbf{HP}_{k-1}\mathbf{H}^{\mathbf{T}} + \mathbf{R} \tag{10}$$

Therefore,

$$\mathbf{K}_k = \mathbf{P}_{k-1}\mathbf{H}^{\mathbf{T}}(\mathbf{S})^{-1}$$

It is also worth noting that

$$\mathbf{S^T} = \mathbf{S}$$

Substituting $\mathbf{K}$ into $\mathbf{P}$ yields :

$$\begin{aligned}
\mathbf{P} &= \mathbf{P}_{k-1} - \mathbf{K}_k\mathbf{HP}_{k-1} - \mathbf{P}_{k-1}\mathbf{H^T}\mathbf{K}_k^{\mathbf{T}} + \mathbf{K}_k(\mathbf{HP}_{k-1}\mathbf{H^T} + \mathbf{R})\mathbf{K}_k^{\mathbf{T}} \\
&= \mathbf{P}_{k-1} - (\mathbf{P}_{k-1}\mathbf{H^T}\mathbf{S}^{-1})\mathbf{HP}_{k-1} - \mathbf{P}_{k-1}\mathbf{H^T}(\mathbf{P}_{k-1}\mathbf{H^T}\mathbf{S}^{-1})^{\mathbf{T}} + \\
&\quad (\mathbf{P}_{k-1}\mathbf{H^T}\mathbf{S}^{-1})(\mathbf{S})(\mathbf{P}_{k-1}\mathbf{H^T}\mathbf{S}^{-1})^{\mathbf{T}} \\
&= \mathbf{P}_{k-1} - (\mathbf{P}_{k-1}\mathbf{H^T}\mathbf{S}^{-1})\mathbf{HP}_{k-1} - \mathbf{P}_{k-1}\mathbf{H^T}(\mathbf{S}^{-1}\mathbf{HP}_{k-1}^{\mathbf{T}}) + \\
&\quad (\mathbf{P}_{k-1}\mathbf{H^T}\mathbf{S}^{-1})(\mathbf{S})(\mathbf{S}^{-1}\mathbf{HP}_{k-1}^{\mathbf{T}}) \\
&= \mathbf{P}_{k-1} - (\mathbf{P}_{k-1}\mathbf{H^T}\mathbf{S}^{-1})\mathbf{HP}_{k-1} - \mathbf{P}_{k-1}\mathbf{H^T}(\mathbf{S}^{-1}\mathbf{HP}_{k-1}^{\mathbf{T}}) + \\
&\quad (\mathbf{P}_{k-1}\mathbf{H^T}\mathbf{S}^{-1})(\mathbf{HP}_{k-1}^{\mathbf{T}}) \\
&= \mathbf{P}_{k-1} - (\mathbf{P}_{k-1}\mathbf{H^T}\mathbf{S}^{-1})\mathbf{HP}_{k-1} \\
&= \mathbf{P}_{k-1} - \mathbf{K}_k\mathbf{HP}_{k-1} \\
&= (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_{k-1}
\end{aligned}$$

2.4 State and Error Projection

The estimate at time k, $\hat{\mathbf{x}}_k$ can be projected into time *k+1* using

$$\hat{\mathbf{x}}_{k+1} = \mathbf{A}\hat{\mathbf{x}}_k \tag{11}$$

The prediction at time step *k-1*, $\mathbf{x}_{k+1}$ is calculated as such :

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{w}_k \tag{12}$$

Therefore, the error at time *k+1*, e_{k+1} is

$$\begin{aligned}
e_{k+1} &= \hat{\mathbf{x}}_{k+1} - \mathbf{x}_{k+1} \\
&= \mathbf{A}\hat{\mathbf{x}}_k - (\mathbf{A}\mathbf{x}_k + \mathbf{w}_k) \\
&= \mathbf{A}(\hat{\mathbf{x}}_k - \mathbf{x}_k) + \mathbf{w}_k \\
&= \mathbf{A}e_k + \mathbf{w}_k
\end{aligned}$$

Using the error term at time *k+1*, $\mathbf{P}_{k+1}$ can be derived as such :

$$\begin{aligned}
\mathbf{P}_{k+1} &= \mathbf{E}[e_{k+1}e_{k+1}^{\mathbf{T}}] \\
&= \mathbf{E}[(\mathbf{A}e_k + \mathbf{w}_k)(\mathbf{A}e_k + \mathbf{w}_k)^{\mathbf{T}}] \\
&= cov((\mathbf{A}e_k + \mathbf{w}_k), (\mathbf{A}e_k + \mathbf{w}_k)) \\
&= cov(\mathbf{A}e_k, \mathbf{A}e_k) + cov(\mathbf{w}_k, \mathbf{w}_k) \\
&= \mathbf{AP}_k\mathbf{A^T} + \mathbf{Q}
\end{aligned}$$

3.0 Kalman Filter's Recursion Process

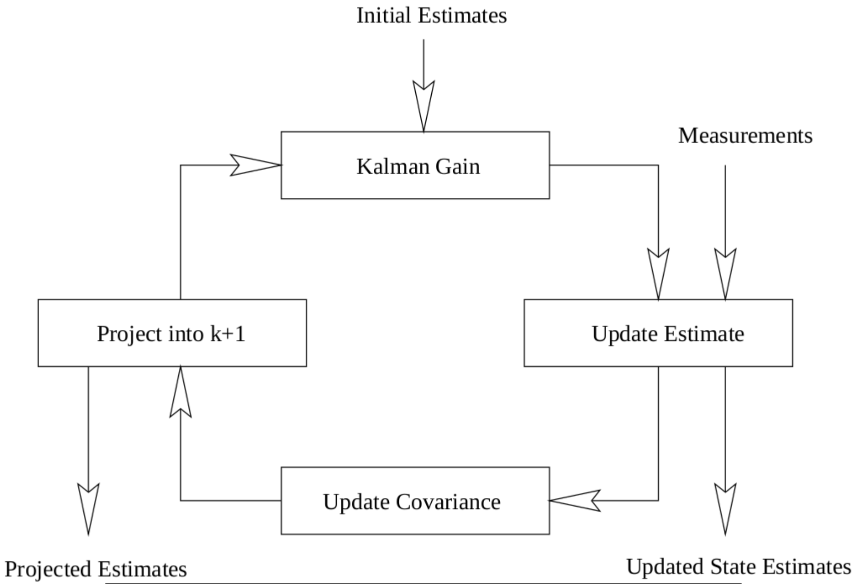

Figure 1 : Kalman Filter's Recursion Process [2]

Description	Equation
Kalman Gain	$\mathbf{K} = \mathbf{P}_{k-1}\mathbf{H}^{\mathbf{T}}(\mathbf{H}\mathbf{P}_{k-1}\mathbf{H}^{\mathbf{T}} + \mathbf{R})^{(-1)}$
Estimate Update	$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \mathbf{K}(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k-1})$
Covariance Update	$\mathbf{P} = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_{k-1}$
Estimate Projection	$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{w}_k$
Estimate Error Covariance	$\mathbf{P}_{k+1} = \mathbf{A}\mathbf{P}_k\mathbf{A}^{\mathbf{T}} + \mathbf{Q}$

Table 1: Kalman Filter Processes

4.0 Summary

In order for Kalman Filter process to work, one must provide initial estimates as this is a recursive algorithm. Choosing the initial values may prove to be excruciating at times. However, note that even if the initial estimates are off by a large margin, Kalman Filter will be able to converge to the true values given the time. During implementation, the prediction vectors and the estimation vectors are the same. In short, $\hat{\mathbf{x}_k} = \mathbf{x}_k$. I assume that in the literature, it is necessary to treat them as different vectors in order to derive the equations. Since Kalman Filter originates from estimation theory, many assumptions and derivations were taken as definitions.

References

[1] *Masnadi-Shirazi, H., Masnadi-Shirazi, A., Dastgheib, M. A. (2019). A Step by Step Mathematical Derivation and Tutorial on Kalman Filters. arXiv preprint arXiv:1910.03558.*

[2] *Thacker, N. A., Lacey, A. J. (1998). Tutorial: The kalman filter. Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester, 61.*

[3] *cwlinghk, V. (2019). Derivation of Kalman Filter's Formulas. Retrieved 28 December 2019, from https://missingueverymoment.wordpress.com/2019/12/02/derivation-of-kalman-filter/*

[4] *How a Kalman filter works, in pictures — Bzarg. (2019). Retrieved 28 December 2019, from https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/*

[5] *Becker, A. (2019). Online Kalman Filter Tutorial. Retrieved 28 December 2019, from https://www.kalmanfilter.net/default.aspx*