

AM205 Workshop: The Kalman Filter

Notes Written by Yuexia Luna Lin (Fall 2020)

Presented by Michael S. Emanuel
23-Sep-2021

In the noisy world that we live in, the Kalman filter provides a way to combine our model of a system and the measurements we make from that system, to estimate the state of the system. Filtering describes the process of separating noise from signal. In the case of Kalman filter, we not only filter out the noise, but also make a prediction on the future state of the system, as well as the probability distributions of the state variables. We can view the Kalman Filter as finding an estimate of the state that has minimal variance, given measurements.

Named after the Hungarian-American electrical engineer and mathematician, Rudolf Kalman, Kalman filters have been widely applied since its development in the 1960s. During the Apollo 11 moon landing mission, Kalman filter was used to estimate the spacecraft's coordinates, given Newtonian models of orbital dynamics and the onboard sensor measurements. Kalman filter's use in navigation has been extended to individual use, in GPS systems in our phones and cars. For another instance, Kalman filter is also indispensable for manufacturing, when engineers need to know the positions of motors and other mechanical parts but cannot directly or frequently measure them. Although we will not cover it here, Kalman filter has been extended and modified to work with large scale, nonlinear dynamics, like in weather forecasting.

In this tutorial, we will learn how to derive the basic algorithm of Kalman filter in a discrete time system, given two assumptions:

1. The dynamical system is linear.
2. The noise in the system is Gaussian distributed white noise, categorized as disturbances to the input, \mathbf{w} , and measurement noise \mathbf{v} .

Our aim is to create an *iterative* algorithm that estimates the state $\hat{\mathbf{x}}_{k+1}$ and the covariance matrix \mathbf{P}_{k+1} at the next time step $t = k + 1$, based on our current estimate $\hat{\mathbf{x}}_k$ and \mathbf{P}_k , and measurement \mathbf{z}_k . A key quantity in this calculation is the *Kalman gain* at time step k , \mathbf{K}_k , which give the optimal combination of model predictions and data measurements based on the strength of noises.

The 1D case

Consider a discrete time 1D dynamical system with state variable x . We make a measurement z . There are noises in the input as well as in the measurement,

$$\begin{aligned}x_{k+1} &= \phi x_k + w \\ z_{k+1} &= x_{k+1} + v\end{aligned}\tag{1}$$

where $w \sim \mathcal{N}(0, \tau^2)$ and $v \sim \mathcal{N}(0, \sigma^2)$.

Given initial conditions x_0, P_0 , where P_0 is the variance in the initial condition x_0 , we will get to an estimate of x_1 that maximizes the probability of the state being x_1 given measurement z_1 , i.e. $p(x_1|z_1)$. Using Bayes' rule, we have,

$$p(x_1|z_1) \propto p(z_1|x_1)p(x_1)\tag{2}$$

We use \propto because we haven't included a normalization factor. For our purpose here, it's not super important.

We know that the measurement noise probability density is $\mathcal{N}(0, \sigma^2)$ around the mean x_1 . So $p(z_1|x_1) \sim \mathcal{N}(x_1, \sigma^2)$. However, we have to compute $p(x_1)$ before data becomes available, using the law of total probability,

$$\begin{aligned}p(x_1) &= \int p(x_1|x_0)p(x_0)dx_0 \\ &= \int \mathcal{N}(\phi x_0, \tau^2) \times \mathcal{N}(x_0, P_0)dx_0 \\ &= \mathcal{N}(\phi x_0, \phi^2 P_0 + \tau^2) \\ &= \mathcal{N}(\hat{x}_1, \hat{P}_1)\end{aligned}\tag{3}$$

where we defined a predictor estimate of state $\hat{x}_1 = \phi x_0$ and variance $\hat{P}_1 = \phi^2 P_0 + \tau^2$. The intuition behind the second to the third line is that the initial variance P_0 gets amplified by the linear dynamics by a multiplicative factor (ϕ^2) as well as a Gaussian white noise by an additive factor (τ^2).

So now we can revisit Eq. 2,

$$\begin{aligned}p(x_1|z_1) &\propto p(z_1|x_1)p(x_1) \\ &= \mathcal{N}(x_1, \sigma^2) \times \mathcal{N}(\hat{x}_1, \hat{P}_1) \\ &\propto \exp\left(-\frac{(z_1 - x_1)^2}{2\sigma^2}\right) \exp\left(-\frac{(x_1 - \hat{x}_1)^2}{2\hat{P}_1}\right)\end{aligned}\tag{4}$$

We maximize the log of the posterior,

$$\begin{aligned}
\frac{\partial \log p(x_1|z_1)}{\partial x_1} &= -\frac{(x_1 - z_1)}{\sigma^2} - \frac{(x_1 - \hat{x}_1)}{\hat{P}_1} = 0 \\
\rightarrow x_1 &= \left(\frac{z_1}{\sigma^2} + \frac{\hat{x}_1}{\hat{P}_1} \right) \bigg/ \left(\frac{1}{\sigma^2} + \frac{1}{\hat{P}_1} \right) \\
&= \frac{\hat{P}_1 z_1 + \sigma^2 \hat{x}_1}{\hat{P}_1 + \sigma^2} \\
&= \hat{x}_1 + K_1(z_1 - \hat{x}_1); \quad K_1 \equiv \frac{\hat{P}_1}{\hat{P}_1 + \sigma^2}
\end{aligned} \tag{5}$$

By trying to maximize the conditional probability $p(x_1|z_1)$, we not only have derived the optimal estimate x_1 , we have also derived a key quantity, *Kalman gain*, K_1 . We can also derive the variance of the conditional probability distribution,

$$P_1 = (1 - K_1)\hat{P}_1 \tag{6}$$

Now we have completed one entire step. Notice at each step, we only need x_k and P_k from the previous step, and to wait for data z_k to become available. Next, let's see how Kalman Filter is extended to multi-dimensional systems.

The general case

Consider a system or a process that can be modeled by a linear dynamical system with N state variables,

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{w} \tag{7}$$

where \mathbf{A} is $N \times N$ the transition matrix, $\mathbf{x}_k, \mathbf{x}_{k+1}$ are column vectors of state variables at time k and $k + 1$, respectively, and \mathbf{w} is Gaussian white noise disturbance in the input. The term $\mathbf{B}\mathbf{u}_k$ can be used to represent control or explicitly time dependent quantities, if applicable.

The measurement vector, \mathbf{z}_k , has M elements, is also subject to noise,

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v} \tag{8}$$

where \mathbf{H} is the $M \times N$ connection matrix between the state vector and the measurement vector, and \mathbf{v} is Gaussian white noise in the measurements.

Denote an estimate of the state as $\hat{\mathbf{x}}_k$, and the estimation error $\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k$. The cost function to be minimized as the covariance matrix of the errors,

$$\mathbf{P}_k = E(\mathbf{e}_k \mathbf{e}_k^T) = E((\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T) \tag{9}$$

The use of the covariance matrices enables us to not only track the state of the dynamics, but also get a sense of the probability distribution of the state

variables. For convenience, we also construct covariance matrices of noise in the system, we have

$$\mathbf{Q} = E(\mathbf{w}\mathbf{w}^T) \quad (10)$$

$$\mathbf{R} = E(\mathbf{v}\mathbf{v}^T) \quad (11)$$

We assume we can quantify \mathbf{Q} and \mathbf{R} and that they are time-invariant in our system.

Typically, the covariance of a column vector \mathbf{x} is computed as

$$cov(\mathbf{x}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T p(\mathbf{x}) dx_1 dx_2 \dots dx_N \quad (12)$$

So to compute the covariance of \mathbf{e}_k , we will need to know the true state and the state estimate. But how do we get \mathbf{P}_k if we don't have access to the true state \mathbf{x}_k ? Next we will derive an update rule for $\hat{\mathbf{x}}_k$ and \mathbf{P}_k , given initial condition \mathbf{x}_0 and \mathbf{P}_0 .

Derive the algorithm

Kalman filter has two steps, a predictor step and a corrector step. Suppose we have just finished time step $k-1$ and begun time step k . We have our last best estimate $\hat{\mathbf{x}}_{k-1}$ and \mathbf{P}_{k-1} , as well as the latest measurement \mathbf{z}_k . Our goal is to use these quantities to create an optimal estimate $\hat{\mathbf{x}}_k$ that minimizes \mathbf{P}_k .

Predictor step

First, in the predictor step, we use our model to calculate an *a priori* estimate at time step k ,

$$\hat{\mathbf{x}}_k^p = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} \quad (13)$$

where superscript p denotes "predict," and we assume $\mathbf{B}\mathbf{u}_k$ is known at all time. We also compute a predicted covariance using Eq. 7 and 9,

$$\begin{aligned} \mathbf{P}_k^p &= E((\mathbf{x}_k - \hat{\mathbf{x}}_k^p)(\mathbf{x}_k - \hat{\mathbf{x}}_k^p)^T) \\ &= E((\mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} - \mathbf{A}\hat{\mathbf{x}}_{k-1})(\mathbf{A}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} - \mathbf{A}\hat{\mathbf{x}}_{k-1})^T) \\ &= E((\mathbf{A}\mathbf{e}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1})(\mathbf{A}\mathbf{e}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1})^T) \\ &= \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q} \end{aligned} \quad (14)$$

Here we have used our assumption that $\mathbf{B}\mathbf{u}_{k-1}$ is known exactly, therefore, doesn't contribute to any covariance. Note that the term \mathbf{e}_{k-1} encapsulates all the error accumulated up to step $k-1$, and the term \mathbf{w}_{k-1} denotes the noise in between step $k-1$ and k , thus, the two terms are completely independent, and have covariance 0.

Corrector step

Next, if data becomes available, we combine the naive prediction $\hat{\mathbf{x}}_k^p$ and the measurement \mathbf{z}_k by

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^p + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^p) \quad (15)$$

The term in the parenthesis, $\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^p$, is called the *measurement residual*. It can be interpreted as the new information given by the measurement after subtracting what we think we know based on the predicted estimate $\hat{\mathbf{x}}_k^p$. \mathbf{K}_k is the *Kalman gain* that will balance our confidence in the *a priori* model prediction and in the measurements. Like what we have seen in the 1D case, by minimizing the cost function, we will have an expression for the Kalman gain.

Now we expand \mathbf{z}_k in Eq. 15 using Eq. 8, and get

$$\begin{aligned} \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^p + \mathbf{K}_k(\mathbf{H}\mathbf{x}_k + \mathbf{v} - \mathbf{H}\hat{\mathbf{x}}_k^p) \\ &= (\mathbf{1} - \mathbf{K}_k\mathbf{H})\hat{\mathbf{x}}_k^p + \mathbf{K}_k\mathbf{H}\mathbf{x}_k + \mathbf{K}_k\mathbf{v} \end{aligned} \quad (16)$$

Note that we don't know the true state \mathbf{x}_k , it just comes along for the ride in the math.

Since we want to minimize the error covariance, we plug Eq. 16 in to expression of error covariance, Eq. 9,

$$\mathbf{P}_k = E([(\mathbf{1} - \mathbf{K}_k\mathbf{H})(\mathbf{x}_k - \hat{\mathbf{x}}_k^p) - \mathbf{K}_k\mathbf{v}][(\mathbf{1} - \mathbf{K}_k\mathbf{H})(\mathbf{x}_k - \hat{\mathbf{x}}_k^p) - \mathbf{K}_k\mathbf{v}]^\top) \quad (17)$$

Given we have assumed Gaussian white noise, we can assume that the cross terms in the expression above are zero. Expand the equation above, with $\mathbf{R} = E(\mathbf{v}\mathbf{v}^\top)$, we have

$$\begin{aligned} \mathbf{P}_k &= E((\mathbf{1} - \mathbf{K}_k\mathbf{H})(\mathbf{x}_k - \hat{\mathbf{x}}_k^p)(\mathbf{x}_k - \hat{\mathbf{x}}_k^p)^\top(\mathbf{1} - \mathbf{K}_k\mathbf{H})^\top + \mathbf{K}_k\mathbf{v}\mathbf{v}^\top\mathbf{K}_k^\top) \\ &= (\mathbf{1} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^p(\mathbf{1} - \mathbf{K}_k\mathbf{H})^\top + \mathbf{K}_k\mathbf{R}\mathbf{K}_k^\top \end{aligned} \quad (18)$$

where \mathbf{P}_k^p is the error covariance calculated using Eq. 14.

Note the trace of the covariance matrix, $tr(\mathbf{P}_k)$, is the total variance, i.e. the sum of the population variances. We minimize $tr(\mathbf{P}_k)$ with respect to \mathbf{K}_k ¹, by setting the following to zero,

$$\frac{\partial tr(\mathbf{P}_k)}{\partial \mathbf{K}_k} = -2(\mathbf{H}\mathbf{P}_k^p)^\top + 2\mathbf{K}_k(\mathbf{H}\mathbf{P}_k^p\mathbf{H}^\top + \mathbf{R}) = 0 \quad (19)$$

We solve for an expression for \mathbf{K}_k ,

$$\mathbf{K}_k = \mathbf{P}_k^p\mathbf{H}^\top(\mathbf{H}\mathbf{P}_k^p\mathbf{H}^\top + \mathbf{R})^{-1} \quad (20)$$

Use Eq. 20 in Eq. 15, we get the corrected estimate, $\hat{\mathbf{x}}_k$. Just as in our intuition, we can see in the structure of \mathbf{K}_k , heuristically, if the covariance in measurements \mathbf{R} is large compared that in the predicted estimation, \mathbf{K}_k would

¹See wikipedia page for scalar-matrix differentiation.

reduce the contribution of the measurement to the corrected estimate, and vice versa.

Use Eq. 20 in Eq. 18, we get the corrected covariance. We can simplify by,

$$\begin{aligned}\mathbf{P}_k &= \mathbf{P}_k^p - \mathbf{P}_k^p \mathbf{H}^\top (\mathbf{H} \mathbf{P}_k^p \mathbf{H}^\top + \mathbf{R})^{-1} \mathbf{H} \mathbf{P}_k^p \\ &= \mathbf{P}_k^p - \mathbf{K}_k \mathbf{H} \mathbf{P}_k^p \\ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^p\end{aligned}\tag{21}$$

The expression for covariance in Eq. 21 is computationally efficient compared to the full formula Eq. 18. However, due to factors including finite precision arithmetic, it's prone to numerical errors and loss of symmetry. In these instances, the full formula should be used.

Now we have completed one iteration by deriving the quantities needed for the next time step, $\hat{\mathbf{x}}_k$ and \mathbf{P}_k . If data is not available in this time step, we can simply skip the corrector step,

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^p; \quad \mathbf{P}_k = \mathbf{P}_k^p\tag{22}$$

Final remarks

Kalman filter is a method within the broader category of linear-quadratic problems. A linear-quadratic problem is one where the system dynamics can be described by linear differential equations, and the cost to be optimized is a quadratic function.

Kalman filter is a linear-quadratic estimator, it estimates the state of the system. There are other well-known methods that follow similar ideas we have seen in the derivation of Kalman filter, e.g. linear-quadratic regulator (LQR). Combining LQR with Kalman Filter gives us linear-quadratic Gaussian control (LQG). For those who are interested, there are much more information about Kalman Filter, LQR, LQG on the web. For starters, I particularly like Prof. Steve Brunton's YouTube series called "Control Bootcamp."