Understanding the Kalman Filter

A Simple, Verbose, 1D Guide

Caleb Adams
UGA Small Satellite Research Laboratory

February 28, 2019

The use of the Kalman Filter Algorithm is common in robotics as a control method [1]. More importantly, it can be used to control spacecraft - and who doesn't love a good spaceship? Preferably one with an invisibility feild... but we won't be discussing spaceships, or invisibility feilds. For this case we will imagine a humble cart on a track. The cart can control its movment by applying a force. The cart will also have the ability to detect it's distance along the track using a lazer.

1 A Motion Model

It is simple to understand a linear motion model. For every time step, let's call the time step Δt, we move some distance, let's call the distance Δx, across the track. We will imagine that this track is the x axis. The amount of distance (Δx) that the cart moves in a certain amount of time (Δt) is its velocity [2]. We will say the the robots instantaneous change in position, the derivative that results in velocity, is represented as $\dot{x}$. If we remember the form definition of the derivative [3] we see:

$$\dot{x} = x(a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}$$

We can see that any movement that our cart makes along our track is given by some change in time multiplied by the instantaneous change in position over that time, so $\Delta x = \dot{x}\Delta t$. If we recall from the great teacher Newton, we can write an equation of motion [4] as follows:

$$x = x_0 + \dot{x}\Delta t + \frac{1}{2}\ddot{x}(\Delta t)^2$$

Here the x_0 value represents the starting position of the cart along the track. In the simplest cast, the starting position can be 0. We can also see that the term $\ddot{x}$ is the acceleration term a. I will be helpful to again remember $F = ma$. We can rearrange this to see $F = ma \rightarrow F = m\ddot{x} \rightarrow \ddot{x} = \frac{F}{m}$, thus:

$$x = x_0 + \dot{x}\Delta t + \frac{1}{2}\frac{F}{m}(\Delta t)^2$$

We can then view the velocity in the same way. With the term $\dot{x}_0$ the intial velocity of the system:

$$\dot{x} = \dot{x}_0 + \frac{F}{m}\Delta t$$

Next, we want to view the system in a more compact way. To do this we see that the cart has a state s at any given time t. In otherwords, at any given time the state of the cart can be discribed by its position and velocity. We can represent this state with the existing funcitons for x and $\dot{x}$:

$$s = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

And if we continue to expand this equation we get...

$$s = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} x_0 + \dot{x}\Delta t + \frac{1}{2}\frac{F}{m}(\Delta t)^2 \\ \dot{x}_0 + \frac{F}{m}\Delta t \end{bmatrix}$$

Now We apply some basic linear algebra [5]:

$$s = \begin{bmatrix} x_0 + \dot{x}\Delta t + \frac{1}{2}\frac{F}{m}(\Delta t)^2 \\ \dot{x}_0 + \frac{F}{m}\Delta t \end{bmatrix} \rightarrow s = \begin{bmatrix} x_0 + \dot{x}\Delta t \\ \dot{x}_0 \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\frac{F}{m}(\Delta t)^2 \\ \frac{F}{m}\Delta t \end{bmatrix}$$

$$\rightarrow s = \begin{bmatrix} x_0 + \dot{x}\Delta t \\ \dot{x}_0 \end{bmatrix} + \begin{bmatrix} \frac{1}{2m}(\Delta t)^2 \\ \frac{1}{m}\Delta t \end{bmatrix} F \rightarrow s = \begin{bmatrix} x_0 + \dot{x}\Delta t \\ \dot{x}_0 \end{bmatrix} + \begin{bmatrix} \frac{(\Delta t)^2}{2m} \\ \frac{\Delta t}{m} \end{bmatrix} F$$

$$\rightarrow s = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ \dot{x}_0 \end{bmatrix} + \begin{bmatrix} \frac{(\Delta t)^2}{2m} \\ \frac{\Delta t}{m} \end{bmatrix} F$$

Thus,

$$s = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ \dot{x}_0 \end{bmatrix} + \begin{bmatrix} \frac{(\Delta t)^2}{2m} \\ \frac{\Delta t}{m} \end{bmatrix} F$$

It is useful to think of our system in this way. It allows us to have a single force F acting upon the cart and allows us to derive its state given an initial x_0 and $\dot{x}_0$ with an arbitrary time step Δt.

The last step is to realize that, in practice, nothing is perfect. To make our motion model more practical we must assume some error in our ability to move in a purely deterministic way. If we could, after all, move in a perfectly deterministic way, then there would be no need for the Kalman filter or complex computational physics. The world would be deterministic and boring. But reality is stochastic, and we want our model to reflect reality. This is discussed more in section 3.2 below.

To start this, we characterize the true accuracy of our motion. For this case we will assume a normal distribution of error - our friend the bell curve. This is represented by the equation [6]:

$$\mathcal{N}(\mu, \sigma) = f(y|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}$$

We imagine that our cart, when it moves, has some small error in movment that is within the bell curve above. Let's call a randomly sampled value from this bell curve ϵ. Each time we move we are effectively selecting a random ϵ value of error. To represent this in our motion equations, we simply add the ϵ terms (simply errors in position and velocity):

$$s = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ \dot{x}_0 \end{bmatrix} + \begin{bmatrix} \frac{(\Delta t)^2}{2m} \\ \frac{\Delta t}{m} \end{bmatrix} F + \begin{bmatrix} \epsilon \\ \dot{\epsilon} \end{bmatrix}$$

2 A Sensor Model

The cart as a lazer that is shoots to a wall and it receives a distance to that wall. For convenience we allow this distance to allign perfectly with the axis of movment. We will call the measurement returned by the lazer z and, because we are getting the distance to the wall at the start of the track, we *should* get the position value x from this measurement. Thus, $z = x$. So if, like above, we add some error term ot the measurement - let's say δ this time - we sould be good. right?

As you may have suspected, it is not that simple. But, don't worry, it still is pretty simple. Instead of considering only the x position value we want to consider the entire state of the cart in the sensor model. To do this we will have to use the state vector s. We do this because, in some models, we may have sensor that detect velocity rather than position. Or, we may have an accelerometer that detects acceleration rather than position or velocity. To make our model general, so that we can truely understand the kalman filter, we use the vector for with s:

$$z = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \delta \rightarrow z = \begin{bmatrix} 1 & 0 \end{bmatrix} s + \delta$$

notice that the $[1 \quad 0]$ matrix results in a scalar. We are using this to ignore the velocity component of the state.

3 Final Terms & Considering Covariance

3.1 Motion & Sensor Matrices

Let's look back at our motion model. Notice that it is of the form:

$$As_0 + BF + \epsilon \quad = \quad \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ \dot{x}_0 \end{bmatrix} + \begin{bmatrix} \frac{(\Delta t)^2}{2m} \\ \frac{\Delta t}{m} \end{bmatrix} F + \begin{bmatrix} \epsilon \\ \dot{\epsilon} \end{bmatrix}$$

Then we see that our sensor model is of the form:

$$z = Cs + \delta \quad = \quad \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \delta$$

This gives us the matrices:

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} \frac{(\Delta t)^2}{2m} \\ \frac{\Delta t}{m} \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

3.2 Covariance Matrices

Next we will want to view ϵ and δ with some matrix representation. To do this we will find their covariance matrices. Let's let Remember that covariance is the measurement of the *joint* variability of two random numbers [6]. In otherwords it describes how correlated two random distributions are.

See that, for a finite set of random numbers $x_i \in \mathbb{R}$, the variance σ^2 with expected value v_x is:

$$\sigma^2 = \frac{1}{n-1} \sum_{n}^{i=1} (x_i - v_x)^2$$

We can actually do the same thing for two variables, let's say x and y. This is what we call the covariance, and for a finite set of random numbers $x_i \in \mathbb{R}$ and $y_i \in \mathbb{R}$ we get the equation:

$$\sigma(x, y) = \frac{1}{n-1} \sum_{n}^{i=1} (x_i - v_x)(y_i - v_y)$$

Notice that we can also find the variance of x if we just find its covariance with iteself. Covariance $\sigma(x, x)$ would just give us the variance of x.

The Covariance Matrix C is a matrix representation of the calcuated covariances of all variables in the system. It is a square, semetric matrix, which contains the possible covariances. If the matrix is diagonal, then the valiables used are independent. If it is not, then then the variables outside of the diagonal are not independent.

$$K = \begin{bmatrix} \sigma(x, x) & \sigma(x, y) \\ \sigma(y, x) & \sigma(y, y) \end{bmatrix}$$

In practice, we would discover this experimentally using the steps above, but for the purposes of this example I will assume that the motion and sensing model have reasonably compact covariance matrices that are diagonal. The Kalman filter is optimal if the covariance matrices are known prefectly, but it will still give good results

if diagonal estimates of the covariance matrices are used. So, I will give ϵ the covariance matrix C_ϵ and δ the covariance matrix C_δ:

$$C_\epsilon = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad C_\delta = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$$

This is me assuming that we have a decent model for our movment model and that our sensor is relativley inaccurate in comparison. Notice that our sensor model's covariance matrix has higher variance for its entries than the motion model's does.

4 Creating the Kalman Filter

4.1 The Kalman Gain

The main idea with the Kalman filter is to combine the predicted measurement, predicted motion, and real measurement. Then, adjust the state so that so that the better model or measurements is trusted more. This becomes more clear when the following equation is considered:

$$\hat{s} = s + K(\hat{z} - z)$$

Here the $\hat{s}$ term is the estimated state of the cart system. The s term is generated from or motion model, $\hat{z}$ is the measured distance and z is the predicted measurement of distance. Notice that all of these terms would be known. The K term is known as the Kalman Gain.

Note, that if the z values are exact, there is no Kalman Gain. This is intitive, because if our movement was perfectly modeled then we would not need any adjustment. However, if this was not the case - and it usually isn't - then there will be some optimal Kalman gain K generated that *corrects* the equation. The K *corrects* the estimations. This is the core idea. In the folowing section, we will show how this becomes more complicated, but in essence is the same.

4.2 The Kalman Filter Algorithm

First, Recall the matrices $A, B, C, C_\epsilon, C_\delta$ that we are using:

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} \frac{(\Delta t)^2}{2m} \\ \frac{\Delta t}{m} \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$C_\epsilon = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad C_\delta = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$$

The Kalman Filter Algorithm also requires two inputs. These inputs are the initial state s_0 and the covarience Σ. The covarience Σ is estimated to start and, for our purposes, we will treat it as the identity matrix and let the algorithm modify it. Remember the equation for a gaussian distribution: $\mathcal{N}(\mu, \sigma)$, note that this requires a mean and variance as input. It just so happens that the state s_0 and the covariance Σ are these values! This is because the Kalman filter algorthm is - at its core - a way to modify gaussian distributions. These gaussian distributions are modified based on what estimates / measurements (which are also gaussian distributions) can be trusted more. The measure for trust is K, the Kalman gain, explained in section 4.1 above.

The first step of the Kalman Filter algorthm is to generate the prediction of the state, this is done with our motion model. In this step, we take in the initial state s_0. This also relies on the force of the system, which we can assume to be an arbitrary constant for simplicity:

$$(1) \qquad \bar{s} = As_0 + BF$$

The second equation takes the covariance Σ that is given as an input and we attempt to predict what the new covariance should be. This step takes our previous covariance and estimates what our new one should be, we call this state $\bar{s}$.

$$(2) \qquad \bar{\Sigma} = A\Sigma A^T + C_\epsilon$$

The thrid step calculates the Kalman Gain for the current iteration of the system. Notice tha the larger the varience in the sensor measurements C_δ, the small the Kalman Gain will be. This equation for the Kalman Gain *desides* how much to trust the given variences based on their characterizations.

$$(3) \qquad K = \bar{\Sigma}C^T(C\bar{\Sigma}C^T + C_\delta)^{-1}$$

Just as before (in 4.1), we use the Kalman Gain to update the state prediction. Here the $\hat{s}$ term is the newest estimated state. The s term is the state previously estimated by the motion model. The $\hat{z}$ term is the actual measurement make by the sensor and the z value is the measurement predicted by the sensor model. This is done as follows:

$$(4) \qquad \hat{s} = s + K(\hat{z} - z)$$

The final step is to update covariance to represent the new estimation. Notice that if we had made a perfect prediction (which never happens) we would just have the identity matrix times the predicted covariance. Otherwise, the variance is updated based on the Kalman Gain:

$$(5) \qquad \Sigma = (I - KC)\bar{\Sigma}$$

All together this is the Kalman Filter Algorthm:

Predict:

$$\bar{s} = As_0 + BF$$
$$\bar{\Sigma} = A\Sigma A^T + C_\epsilon$$

Update:

$$K = \bar{\Sigma}C^T(C\bar{\Sigma}C^T + C_\delta)^{-1}$$
$$\hat{s} = s + K(\hat{z} - z)$$
$$\Sigma = (I - KC)\bar{\Sigma}$$

The output of the Kalman filter is another state s and another covariance Σ, which are meant to be iterativley updated. This makes sense, because we want to improve the certianty of our measurements and movments. These measurements and movments are represented by gaussians, which are functions mean and variance.

5 Double Iteration Example Question

Assume the initial contidions, constants, and above variables:

$$s_0 = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad F = 1$$

Preform 2 iterations of the Kalman Filter Algorthm

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[2] T. L. Heath and Euclid, *The Thirteen Books of Euclid's Elements, Books 1 and 2*. New York, NY, USA: Dover Publications, Inc., 1956.

[3] M. Spivak, *Calculus*, 4th ed. Publish or Perish, 2008.

[4] A. N. Whitehead and B. Russell, *Principia Mathematica*. Cambridge University Press, 1925–1927.

[5] G. Strang, *Introduction to Linear Algebra*, 4th ed. Wellesley, MA: Wellesley-Cambridge Press, 2009.

[6] W. Feller, *An Introduction to Probability Theory and Its Applications*. Wiley, January 1968, vol. 1.