An Introduction to Kalman Filtering

RF Tracking UAV for Sports Video Capturing

By Connor Lansdale, ECE '18

Abstract

The Kalman Filter is a powerful data fusion algorithm that attempts to predict the next state of a system (e.g. an objects position or temperature). This article will provide a high level overview, with the aim to describe the motivation and theory behind Kalman filtering in an intuitive way.

Introduction

The Kalman filter is a recursive estimation algorithm that attempts to predict the next state of a system by combining information in the presence of noise. Given any dynamic system that takes in uncertain data, the Kalman filter can be used to make an educated guess about what the system is going to do next (Babb). The resulting prediction will be more accurate than the estimate found from any one source alone, as it takes into account all indirect measurements, as well as the affect of control inputs and external factors to make its final estimation. The combination of this data results in a powerful prediction tool that has many important applications in modern technology. From vehicle navigation, to signal processing, to economics, the Kalman filter can be used to fuse data in any system that contains uncertainty. This article will focus on using the filter as a means of optimizing position data, such as from a GPS, to most accurately know the location of an object of interest.

Automated Car Example

The theory behind how the Kalman filter works is best explained with a running example.

Say we are building an autonomous car to travel in a straight line 100m and it must finish within 2m of the finish line.

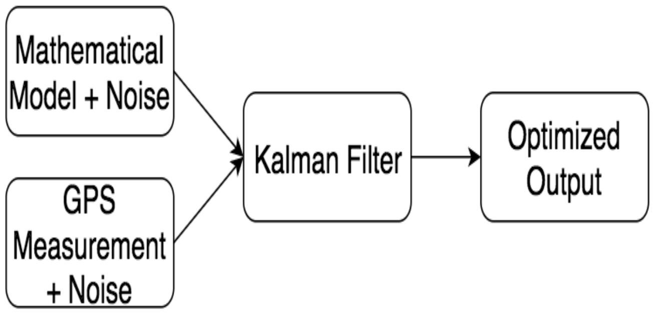
A Kalman filter can be used here to most accurately measure the cars position so that it knows when to stop. The car is equipped with a GPS sensor that directly measures the car's position, but this measurement is noisy and can only be trusted to ±5m. The model assumes the only input to the car is the gas pedal, which changes the car's velocity. The whole system has additional noise caused by external factors like wind or wheel slipping that will add to the uncertainty in the cars position. Now back to the example, the GPS measurement on its own will not be accurate enough to pass the test, but this measurement can be greatly improved by combining other indirect data sources about the cars position. In this case, we have a measurement of the acceleration of the car because we know its input, the throttle. The position can then be calculated by taking its second derivative. Again, this calculation also contains noise and will not produce a position estimate accurate enough to stay within the threshold. For this example the Kalman filter is applied only to the GPS measurement and the throttle input, but a real system could have hundreds of sensor measurements to be combined.

Theory

The Kalman filter works in a two-step process: a prediction step, and an update step.

The prediction step produces an estimate based on the current state variables, and the update step adjusts this estimate based on measurements from the system. In the automated car example, we would create a mathematical model of the cars position, and then send the input (acceleration from throttle) through to create an initial guess. An overview of this system can be seen in Figure 1. The information from each of these steps is crucial for the filter to work.

Without an estimation step, the prediction will be noisy as the measurements are the only data we have. If there is no measurement data the prediction will drift away from the true value because small errors in position are continually fed back into the model.



In the implementation of the vanilla Kalman filter, noise is modeled as Gaussian distributed random variables. This is an important assumption, allowing the algorithm to make estimations based on the average properties of the noise. A Gaussian random variable models the error in measurement as a classic bell curve centered on zero. It is highly likely that the measurement is off by a smaller amount (close to zero), and not as likely that the measurement error is something further away from zero. This does not mean that the magnitude of the noise is small necessarily, but the majority of the noise will be centered around zero. In the car example, there are three important pieces of data all with their own mean and variance: the initial state estimate, the predicted state estimate, and the GPS measurement. The initial state estimate is put through the mathematical model to get a predicted state estimate. The GPS measurement is then taken and these two probability functions are multiplied together to produce an optimal state estimate. functions is another Gaussian function (Faragher). The output of the filter is then found by scaling the probability density function and taking the mean, and just like that we have found an optimized estimate of the position.

The Kalman filter also introduces an important scaling factor called the Kalman gain. The gain changes the weights of the prediction estimate and the measurements from the system. If it is known that a measurement will contain a lot of noise, that data will carry less weight i.e. have a smaller effect on the output, in the final state estimate and vice versa. Going back to the car example, if the velocity of the car is high then it is likely that the GPS data can be weighed less due to a decrease in its reliability at high speeds. The weights are calculated from each covariance matrix, which is a measure of the uncertainty of the prediction of the system's state. Then, the gain is calculated each iteration to minimize the covariance of the final state estimate. The resulting output will be the most accurate state prediction for the given set of data. Another important property of the gain, K, is that it is updated in real time. The beauty of this is that these calculations c[illegible] require data from the previous step, so position updates come very quickly and are relatively cheap computationally.

Application

In our senior project we used a Kalman filter in an autonomous drone project that can track and record an athlete during a sports game using computer vision and a radio frequency transmitter/receiver system. The Kalman filter was an integral part of the drone localization, as well as object tracking in the computer vision module. The project aimed to provide high quality game footage in an inexpensive, easy to use package. Because the drone flies autonomously above the sports game, it needs to know where it is in space at all times. This is where the Kalman filter comes in. Similar to the car example, the drone itself has a GPS, accelerometers, a lidar detector (for altitude), as well as a computer vision system that uses anchor points in the corners of the field to feed position data to the drone. The Kalman filter combines all of these measurements and form a best guess of where the drone is relative to the player and its surroundings. After much experimentation and optimization a Kalman filter was built from scratch and the algorithm allowed successful and safe flight as well as more accurate object tracking.

Conclusion

The Kalman filter has been in use for more than 60 years, and still today remains an immensely important algorithm in many applications in engineering and beyond. Any system that handles noisy data and can be approximated can benefit from the use of a Kalman filter. Additionally, because it only requires data from the previous state, it is a very fast algorithm to implement and does not require a lot of computational power. It is for these reasons that the Kalman filter has gained

References

1. Faragher, Ramsey. “Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation” Illinois University, https://courses.engr.illinois.edu/ece420/sp2017/UnderstandingKalmanFilter.pdf

2. Kovvali, Narayan; Banavar, M. (2014). An Introduction to Kalman Filtering with MATLAB Examples.

3. Civera, Javier; Davison, A. (2012). Structure From Motion Using the Extended Kalman Filter.

4. Kalman & Richard S. Bucy (1961). New Results in Linear Filtering and Prediction Theory, Journal of Basic Engineering.