The Need for Kalman Filter to Estimate a Drones Sensor Data

Alejandro Colina-Valeri, ECE '21

Introduction

Drones use sensors. These sensors help drones acquire data of their surroundings to help the drone calculate the current position and velocity of the drone. To improve the data, multiple sensors can be combined. However, the data acquire from the sensors might be inaccurate due to noise created by the sensors. This makes the drone go berserk since each sensor reading might be giving data that contradicts one another. If so, how does the drone know where it is at, and where to go? To fix this problem, a Kalman Filter (KF) is used by taking the different data given by the sensor, analyzing the probability distribution of each sensor, and combining that data to come up with the best estimate of the drone's current position and velocity. To improve the filter, we can use more robust types of KF that perform better estimates. A more robust type is the extended Kalman Filter (EKF), which considers nonlinear data. The EKF will be the algorithm implemented in our Senior Design AR Parrot 2.0 drone project to get rid of the unwanted noise.

Why Drones Need Sensors?

Type of Sensors

Drones have sensors to perceive their current state, know more about their surroundings, and know how to act in return. Since we are dealing with an autonomous drone these sensors will act as the drones' eyes and hears during the entire flight. This sensor gives the drone two attributes. Self-awareness by given the telling the drone its location and position in time, and situational awareness which helps the drone detect and track objects around it. For our drone project, the AR 2.0 Drone will be using 4 different types of sensors:

<u>Accelerometer:</u> It reads the linear movement done by the drone. It read any changes in velocity perform by the drone, which includes the constant downwards acceleration due to gravity.

<u>Gyroscope:</u> It reads the angular movement done by the drone. This sensor helps determine the drone rate of rotation, degree of tilt, and angular velocity.

<u>Barometer:</u> It measures the air pressure surrounding the drone. As air pressure decreases proportionally with altitude, the drone can determine its height by calibrating the sensor with respect to sea level.

<u>Camera:</u> This sensor will help the drone be aware of its surroundings. So far, the other sensors mentioned above helped with the drone's self-awareness. This is the only sensor that will be used to determine the drone's situational awareness.

Sensors Fusion

By looking at the functionalities of each sensor we can fuse them together. What do I mean by fuse? This means that we can combine the accelerometer, gyroscope, and barometer to get more accurate and consistent data towards the overall location and position of the drone. Figure 1 shows how combining this sensor we can get a better perspective of the drone's direction, rotational velocity, and translational velocity.

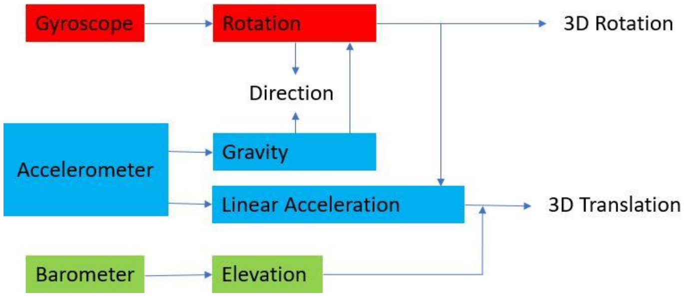

Figure 1: Sensor fusion of accelerometer, gyroscope, barometer, and camera

The Parrot drone will not use a GPS. Without it, the drone cannot determine its position with respect to the environment around it. To fix this problem, we will implement autonomous navigation using the front camera. Please read Michael Eve's technical paper, “Autonomous Navigation using SLAM / PTAM-based Mapping” to understand more how the autonomous navigation is implemented.

Combining this sensor seems to be fantastic! However, there is a problem. They are noisy. This can throw away the actual position and velocity since the corruption of the noise will be indicating different readings that will make each sensors data contradict one another. This means that before putting all the data through a PID controller, first this data must be filtered out to remove the unwanted noise. Therefore, the need of a KF.

Why Drones Need a Kalman Filter?

The KF is not exactly a filter, it is not an electronic device that filters out unwanted noise. The KF is a mathematical estimator. It estimates the actual data that comes in and outputs the best estimate to the PID controller. Therefore, the KF does not filter a signal, but it does filter the system. The KF works by calculating the best output from the measured data with the predicted data through a feedback system.

We will be using the KF in our drone to accurately estimate its position and velocity. The way it estimates the data is by considering the different noises introduced by each sensor. This means that both the measured and predicted data come with their own distinct noise. Figure 2 shows the noise model of both prediction and measured data.

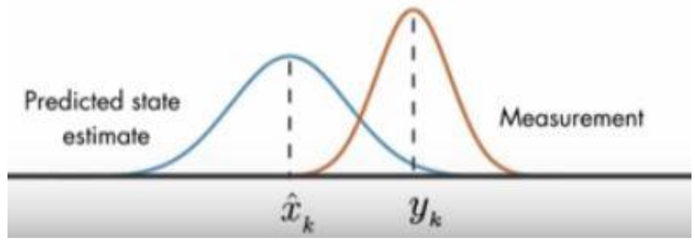

Figure 2: Noise models of the Predicted and measured data calculated by the KF

Will the KF lean more towards the prediction data, the measured data, or a combination of both? It depends on a series of equations within the feedback system that through a process of prediction and updates it can determine the best output. The KF will estimate the final value somewhere in between the two \ noise models by settling on the average of the two. This average depends on the shape of both noise models. Figure 3 depicts the optimal estimated state computed by finding an average between the two noise models.

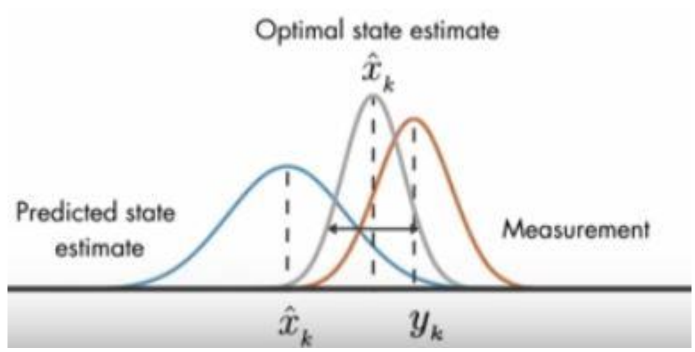

Figure 3: Optimal estimated state

This is what the KF does. Due to the noise generated by the sensors we need a KF to be implemented in our drone project. Each sensor will give different measurements of the drone’s position and velocity, so the KF must consider all these measurements. The KF calculates the noise that comes with these measurements, and from its calculations it finds an optimal estimated value that represents the most accurate measurement. For example, the accelerometers say the drone is 15 degrees from vertical, while the gyroscope says the drone has not rotated away from vertical. Without the KF, this would confuse the drone and through it off balance, but with the KF the system will find happy medium between all the data given by the sensors.

Extended Kalman Filter

An improvement over the KF is the EKF. The problem with the KF is that it only works for linear systems. However, real-world is nonlinear. For example, gusts of wind pushing against the drone. The EKF considers the non-linear system to calculate the best estimate between the noise models. The idea behind the EKF is basically the same as the KF, but to consider non-linear systems a derivative is applied to the KF's mathematical model.

Other than that, the EKF is the same as the KF. The only downfall with this implementation is that it increases the computational complexity by a little. Still, due to the advantage of solving non-linear systems is it preferable to make the switch. However, the EKF is not the only other upgrade that can be done to the KF. There is also the Unscented Kalman Filter (UKF) and the Particle Filter (PF). However, these other algorithms increase the computational complexity even more, which is unnecessary for what we need out of the drone. Therefore, the EKF will be the algorithm that will be used to help stabilize the AR 2.0 Drone.

Conclusion

From what we learned, the drone uses multiple sensors to be aware of its orientation and its surroundings. Since the sensors create a lot of noise, the KF must be used to account for the noise and estimate the output. Then a better mathematical estimator called the EKF, considers non-linear systems that originate from the unpredictability of living in the real world.

References

1. Flynt, Joseph, et al. "What Sensors Do Drones Use?" 3D Insider, 18 Apr. 2019.

2. Naiya, Pavel. "Key Trend: Sensor Fusion to Create Sensor as a Platform for Delivering Solutions." Counterpoint Research, 29 Dec. 2017.

3. Ulusoy, Melda. "Understanding Kalman Filters, Part 3: An Optimal State Estimator ." MATLAB & Simulink, Math Works, 27 Mar. 2017.

4. MATLAB, Tech Talk, director. Understanding Sensor Fusion and Tracking, Part 2: Fusing a Mag, Accel, & Gyro Estimate. YouTube, YouTube, 22 Oct. 2019.

5. Kim, Youngjoo, et al. "Introduction to Kalman Filter and Its Applications." IntechOpen, IntechOpen, 5 Nov. 2018.