INSGPS Algorithm

The following pdf is a little old and not well edited, but it describes most of the theoretical approach behind the Extended Kalman Filter (EKF) used the INS/GPS implementation.

[INSGPSAlg.pdf](http://wiki.openpilot.org/download/attachments/950387/INSGPSAlg.pdf?version=3&modificationDate=1315830505000&api=v2)

System state estimation with Extended Kalman Filter (EKF)

The sensors on the OpenPilot hardware, or any common Flight Controller, cannot observe the system state directly. This means the flight computer cannot see the orientation, position and movement speed of the craft as is. Instead it has sensors similar to the human "inner ear" that provide it with rotation and acceleration measurements, a compass, a barometer, and possibly a global positioning system (GPS) from which that state information has to be computed.

The measurements taken by these sensors however are noisy, inaccurate and affected by temperature dependant offsets. Therefore a **filter** needs to be used to fuse all these sensors into a consistent view of the actual situation, the so called **state estimation**. This process is known as **sensor fusion**.

**Complementary filter**

CC and CC3D use a simple **complementary filter**, to fuse CC's accelerometer and gyroscopes into an estimate of the orientation in space or **attitude**. This complementary filter however gets skewed by centripetal forces (aka when flying in circles for a longer time) and it can not calculate speed and position. It cannot even estimate which way CC is facing, since CC has no compass.

**Extended Kalman Filter**

**State:**

The OpenPilot Revolution system runs an Extended Kalman Filter to compute a 13 dimensional system state, including a complete covariance in real time. The following 13 state variables are computed:

1. Position in space - N axis
2. Position in space - E axis
3. Position in space - D axis
4. Movement velocity - N axis
5. Movement velocity - E axis
6. Movement velocity - D axis
7. Orientation in space - Quaternion - Q0
8. Orientation in space - Quaternion - Q1
9. Orientation in space - Quaternion - Q2
10. Orientation in space - Quaternion - Q3
11. Gyroscope drift bias - X axis
12. Gyroscope drift bias - Y axis
13. Gyroscope drift bias - Z axis

**Prediction model:**

The EKF runs iteratively at an update rate of 1000 updates per second (1kHz). It has a simple movement model of the craft, that simply computes the change in estimated velocity and orientation during the 1/1000 second based on the current measurement of the gyroscope (corrected by the estimated gyroscope drift bias) and accelerometers. The movement model takes the gravity vector into account to adjust the acceleration accordingly. The following sensors are used for this:

* Accelerometer measurement - X axis
* Accelerometer measurement - Y axis
* Accelerometer measurement - Z axis
* Gyroscope measurement - X axis
* Gyroscope measurement - Y axis
* Gyroscope measurement - Z axis

This is called the prediction step. The new estimation then gets updated in a correction step.

**Correction step:**

The correction uses the following sensors to improve the state estimation:

In each step the following sensors can be measured to improve the estimation of the orientation in space.

* Magnetometer measurement - X axis
* Magnetometer measurement - Y axis
* Magnetometer measurement - Z axis

Other sensors have a slower update rate and are not available in every step but at a slower rate:

* Barometric altitude measurement (altitude in meters, corrected by a slowly adjusted pressure offset) - D axis
* GPS position (latitude/Longitude recomputed into NED coordinates relative to home location) - N axis
* GPS position (latitude/Longitude recomputed into NED coordinates relative to home location) - E axis
* GPS position (altitude) - D axis
* GPS velocity - N axis
* GPS velocity - E axis
* GPS velocity - D axis
* Airspeed (if available)

**Variances**

An extended Kalman filter tracks not only the most likely system state in the way the complementary filter does, but also how different components of the state relate to each other, and how sure the filter is about each part of the state. This allows the EKF to adjust and improve all state variables from any single sensor measurement. The necessary relationship information is stored in the**covariance matrix** of the filter.

In every **prediction step**, the system overall variance increases, measuring the (un)certainty in the system. Every **correction step**usually decreases the overall uncertainty. However the sensor measurements themselves are not 100% certain, as each sensor has its own sensor **variance**. By definition the variance is simply the average error squared. For example, lets assume the GPS altitude is by average off by 3 meters, then this sensor's variance would be 3²=9m.

With most sensors (accelerometers, gyroscopes and magnetometers) the most obvious and easy to measure error is **noise**. This is a zero mean deviation, that can easily be estimated for example by the sensor noise calibration plugin in the GCS config widget.

However this easy to measure noise is not the only error, there might also be other errors such as magnetic disturbances in flight coming from the engines or non 100% correct calibration of sensors, leading to offset sensor readings and the like. Therefore it might be necessary to set the variance higher than the noise calibration suggests.

**Sensor contradiction**

Further more, the relationship of different sensors' variance settings tell the EKF which sensor to trust most.

For example, both accelerometers and magnetometers are providing information about the orientation of the craft in space. If the local magnetic field is not as expected, the filter gets fed contradictory information. If this is the case, the filter must choose which sensor to trust and which to discard as wrong, and it will do so based on the variance setting if the variances for these sensors differ significantly (aka by a factor of at least 2 !!! )

If two sensors provide contradictory information and have similar variances, the EKF can start to oscillate between state estimations that fit either of the sensors. Instead of a stable attitude estimation, it might provide a "drunken" artificial horizon.

If this is the case, it is an indicator for sensor contradiction as well as non clear priorities which sensors to trust. To fix this, the variances should be adjusted by increasing or decreasing the variance of one of the affected sensors as well as fixing the cause for the contradiction (for example an incorrect magnetometer calibration).