TTK4250 ESKF

Ludvig Løken Sundøen, Nadia Garson Wangberg

August 12, 2021

1 Abstract

In this assignment the Error state Kalman Filter (ESKF) was implemented and tuned both for a simulated and a real dataset for a fixed wing UAV with IMU and GNSS measurements.

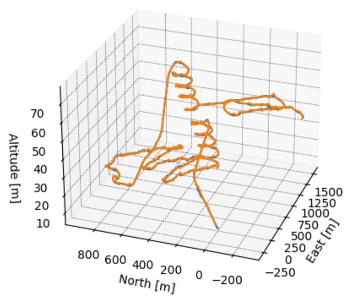

Figure 1: 3D path of simulated data

2 Tuning on simulated data

2.1 The default parameters

We started with the set of parameters given out, which were already quite good. The GNSS covariance is set higher in z than in x and y, which is done due to the GNSS being worse at measuring altitude than latitude and longitude. This is because the GNSS is modelling the earth as a spheroid (WGS84) while the earth is in fact an ellipsoid. A pressure sensor could be added to improve the altitude position.

The IMU acceleration and gyro noise terms were not tuned as they were given in the STIM300 data sheet. If the IMU data sheet was not provided, the IMU noise terms could be estimated.

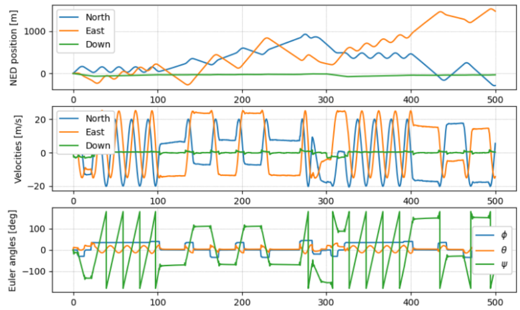

Figure 2: Pose estimates of simulated data

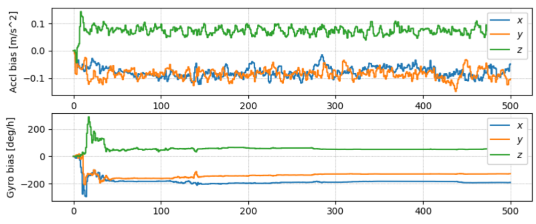

Figure 3: Bias estimates of simulated data

2.2 Tuning initial covariances

The process model and measurement model noise can be tuned in reference to NEES and NIS respectively. Both NIS and NEES should preferably be 1 in order to have an error consistent with the size of the covariance. If either is too large its corresponding model covariance matrix should be increased.

As seen in figure 4 the NEES graphs showed too low values at the beginning. This was fixed by tuning the initial process covariance matrices $P_{pred,0}$ down until the NEES was inside its confidence interval as seen in figure 8.

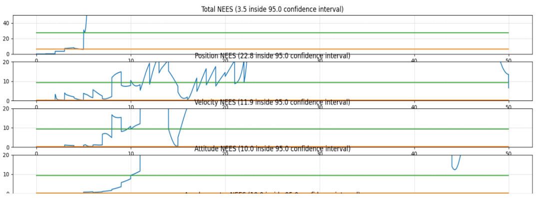

Figure 4: Initial NEES of simulated data

2.3 Tuning the bias covariance

The bias terms' default parameters gave quite good results, but attitude and gyro bias NEES was slightly too high. These were highly connected as the attitude is a direct function of the gyro bias, so only one had to be changed. One thing noticed was that changing the bias terms highly changed the NEES values for everything else. This truly shows how coupled all of our state equations really are, which made tuning pretty difficult.

The errors and the NEES was highly affected by changes in the gyro bias standard deviation. While a change in p_{acc}, of the same magnitude barely made a difference. This shows that the ESKF is sensitive to changes in bias values but not to p_{acc}.

2.4 Attitude estimation error

As can be seen in figure 6 the attitude error is $\pm 1°$, which is quite low. In the body frame, the gravity vector affects the y-axis when rolling and x-axis while pitching. This is sufficient information for the gyro to find roll and pitch.

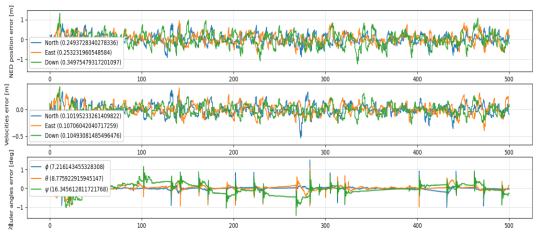

Figure 5: Pose errors of simulated data

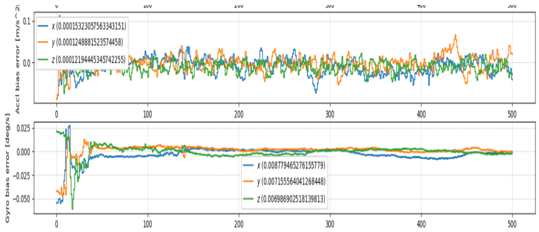

Figure 6: Bias errors of simulated data

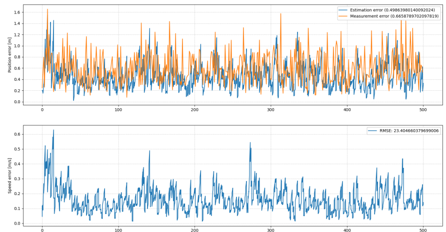

Figure 7: Position and speed errors of simulated data

Heading is more difficult to find, as you cannot measure it absolutely. Only rate measurements is found from the IMU, so integration and knowing the initial heading is required.

No matter the accuracy of the gyro and the bias estimation, drift will always happen. As we know, small noise in the gyro measurement can cause huge problems as the error explodes quickly in the integration process. This makes correction from GNSS measurements necessary. As the Gyro drifts, increasing the attitude error, the prediction covariance increases which overall increases the Kalman gain. This

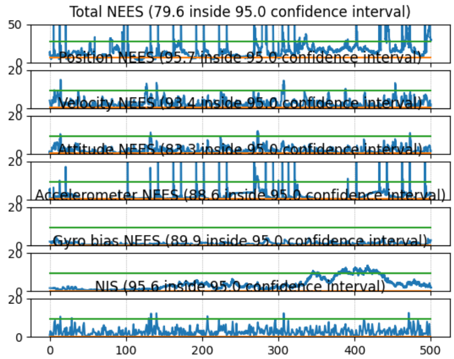

Figure 8: NIS and NEES of simulated data

weights the correction measurement from the GNSS highly and the error drops.

Heading is estimated by using several GNSS measurements to estimate the air-plane velocity which is used to find heading. This heading only observable while in motion, which is not a huge issues since air-planes always have a velocity while in the air. That means we can not trust the heading estimates while standing still.

Different methods can also be used to find heading. For example a gyro compass or using two GPS receivers on each end of the vehicle, these methods are commonly used for ships. Both of these techniques finds the heading even when stationary. Disadvantages are that the gyro compass is rather heavy and the two GPS receivers should have a large distance between them or be expensive RTK GPSs. The NTNU UAV-lab fixed wing used to create the real-dataset actually has two GPS receivers. Using data from both could improve the heading estimate and even estimate heading when stationary.

2.5 IMU misalignment matrix

When simulating with $steps = 20000$ the position error was $\pm 1m$ and the attitude error around ± 1 deg. The position error was not affected by changing S_a, S_g to identity, but the attitude error became slightly worse, around ± 4 deg. The NEES and NIS values also became worse, meaning tuning could make this error even less. All in all the error is not too significant, so disregarding this matrix in real life could be reasonable.

The misalignment matrix consist of the mounting error, orthogonality errors, scale error matrices. The orthogonality errors arise from faulty manufacturing, resulting in the IMU axis not being exactly perpendicular to each other. With lower quality IMUs, the scale errors and orthogonality errors will be larger, so the misalignment matrix may in this scenario be needed.

3 Tuning the ESKF on real data

3.1 Trying the simulation parameters

The parameters was initially set to those found when tuning for the simulated data set. These worked quite well due to really focusing on finding parameters that would generalise to real world data.

When tuning for the simulated dataset we noticed that the performance acquired with few steps $steps = 1000$ was not the same as with many $steps = 20000$. This seemed similar to over-fitting in supervised learning, as the model performed well for a small dataset, but failed to generalise. We therefore started by tuning with fewer steps, but to avoid over-fitting increased steps near the end. By doing this our parameters were able to generalise really well to our real world data. Another observation that can be made is that the trajectory in the simulated dataset is quite extreme. This can lead to our parameters not only working in a stable calm flight, but can also work in more extreme scenarios.

3.2 How real world data is different

It was however not perfect, so a simple change was done to the covariance of the GNSS measurement model, $R_{GNSS} = accuracy_{GNSS}[GNSSk]^2 I$. This was done due to having the accuracy of the GNSS available at each timestep. This accuracy was found from GNSS metadata such as dilution of precision. The initial value of P was already given in *run_INS_real.py*. The first time we ran it the results were satisfying.

An important thing to notice is that the NEES values are given as a function of the true state x. This means that ground truth must be know, which is not the case. NIS values are therefor the only ones used for tuning. This makes it important that the parameters are tuned well for the simulated dataset, as these values are used as a starting point in the real data set. It is worth to mention that the first half the data set the UAV was not flying. This is also the part where the perfomance was worst, as can be seen in figure 9.

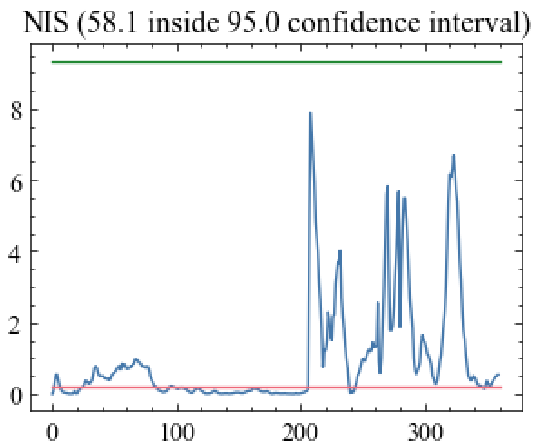

Figure 9: NIS of real data

3.3 Sensitivity analysis

Sensitivity analysis is particularly useful to test the robustness of the system and to increase our own understanding of how the system works. R_{GNSS} was scaled up to 10 times the former value. This is the same as $accuracy_{GNSS}[GNSSk]$ being scaled up by approximately 3.16. The result was as expected. The NIS value dropped to 0.46. Which would result in a huge inconsistency in our model. The estimates however remained mostly similar.

3.4 Simplified IMU misalignment matrix

Setting the IMU misalignment matrix to its simplified version does not really change the estimates in position, velocity and attitude that can be observed. Nevertheless it affects the biases of both the gyro and accelerometer in both scale and in the axis (x and y switched). This could be given as an indicator that IMU misalignment has not been corrected for. It seems like the biases catches the error given the simplified misalignment matrix, kind of like when using integral action in a control loop.

In this case it does not seem to matter that much for the output of the model, but in other cases where the misalignment is higher, it could be the cause greater errors. Another effect observed is that the NIS inside the CI of the model dropped from 58.1 to 23.9. Which testifies to a consistency issue. Therefore having a simplified IMU misalignment matrix may require other tuning parameters.