TTK4250 Sensor Fusion
Graded Assignment 2

Group 63
Torbjørn Smith 521669
Isak Hammer 257390

November 8, 2021

Department of Engineering Cybernetics

1 Introduction

This report presents the application and tuning of an error-state Kalman filter (ESKF). The ESKF is used to estimate the position of an unmanned aerial UAV (UAV), using acceleration and angular velocity data from an inertial measuring unit (IMU) [1] and position data from global navigation satellite system (GNSS) data. The filter is tuned for two data sets: One simulated, where absolute position is known, and one with real recorded data. The basic idea behind the filter is to use the IMU-data and through integration produce the position estimate, while correcting for drift using the GNSS-data. Using mainly normalized estimation error squared (NEES) and normalized innovation squared (NIS), the filter performance was evaluated as the filter was tuned.

2 Simulated Data

When tuning the ESKF the tuning parameters for the IMU- and GNSS-data need to be chosen carefully. A decent starting point are the data sheets of the sensors and the raw data produced by the sensors. For this task however, simulated data for the IMU and GNSS with data for the ground truth of the UAV is provided. This allowed a more direct way of producing good tuning parameters for the error state and the GNSS measurements.

By extracting the ground truth points produced at the sampling rate of the GNSS sensor, the mean-squared error between ground truth and GNSS is calculated to produce the entries to the corresponding tuning parameters. This method is not as straight forward to apply to the IMU. A good method could be to record a long series of IMU measurements with the IMU at stand still, from which the covariance matrix for the IMU would be produced. Unfortunately, this method is not applicable with the provided data. Instead the sensor data sheet was used to produce the entries.

As gravity acts on the vertical axis (assuming a level UAV) the heading is not possible to estimate using just the IMU. Using the GNSS, the relative movement between sampling points of the GNSS makes estimating the heading of the UAV possible. If the UAV is still, the only differences would be the noise in the GNSS readings, making them useless for estimating the heading. The UAV needs to move using the current configuration with a single GNSS reading to estimate the heading. To solve this issue, a magnetometer could be added to the UAV to read the heading directly.

The IMU correction matrices S_a and S_g where given in the data set. These adjust for the error from mounting the IMU skewed compared to the body frame of the UAV, scale error and the orthogonality error between the different axis of the IMU. Disregarding these matrices was investigated by passing an identity matrix in their place. A drastic difference in all NEES values was observed, with orientation Θ and angular velocity bias ω_b going to zero percent inside the 90% confidence interval. With the current tuning discarding the correction matrices becomes a problem, but with updating the tuning some of the negative effects could be negated. It comes down to having even more modeling error in the filter and dealing with this through even more tuning.

3 Real Data

The fact that ground truth is no longer available, makes the tuning process more difficult. Initially, one can look at the raw GNSS data, but the goal is to fuse the IMU data and the GNSS measurements. The chosen method was to use the existing settings for the IMU tuning parameters, chosen based on the datasheet [1], and then adjust the GNSS tuning parameters and the initial values to obtain the best performance.

Like with the simulated data, the correction matrices were neglected, to test their influence on the final estimate. The result was surprising, as the NIS values were better without the correction matrices than they were with. This could be because the correction matrices for the real data are fairly close to identity matrices, and it could be that our tuning work's better without the provided correction matrices. This might show that the correction matrices can be compensated for with tuning.

Great care has to be taken when using IMU data as mistakes in for example mounting, sign conventions, or even units could prove catastrophic. If possible, static test should be conducted. Using just the (fairly) constant gravitational acceleration, and perhaps an accurate jig, the plant could be oriented in known attitudes to check the readings from the IMU. Furthermore, tuning becomes even more time consuming given smaller errors like the ones modeled in this experiment. Also letting the filter initialize and run for some time to get "up and running" before using its signal for control, could be useful to negate potential bad outcomes from wrong initial values.

4 Tuning

Using the simulator as a tuning framework was prioritized, as it provided a ground truth model to compare the filter to get good initial parameters for the filter model. It was crucial to simulate for a proper amount of time, to both negate the fluctuations from initialization of the filter and to be able to determine the long time effects of the filter.

Tuning the IMU was done by starting with the values from the datasheet. With these values the filter performed decently in most aspects of the simulation. Through perturbing these parameters by orders of magnitude, making them both smaller and larger, it was found that an order of magnitude larger bias std for both the gyro and the accelerometer was desired. This gave the final IMU filter parameters derived from the IMU datasheet.

Descriptor	Value
accm std	1.2×10^{-3}
accm bias std	4.9×10^{-3}
gyro std	4.36×10^{-5}
gyro bias std	1.45×10^{-3}

Table 1: Final filter parameters for the IMU derived from datasheet

To complete the IMU filter parameters, $p_{\mathbf{a}b}$ and $p_{\omega b}$ were set to really small: $p_{\mathbf{a}b} = p_{\omega b} = 10^{-6}$ as this gave the best performance.

The filter parameters for the GNSS were, as described earlier, found using the ground truth given in the simulator to calculate the mean-squared error between the ground truth and GNSS. This was done in Mathworks MATLAB using the *immse*-function. The following values were produced:

Descriptor	Value
gnss std ne	9.21×10^{-2}
gnss std d	2.693×10^{-1}

Table 2: Final filter parameters for the GNSS

The choice of initial states was crucial to be able to obtain long term convergence. It was quite clear that if the filter was not able to converge during the initialization procedure, it would perform poorly, even with good filter parameters. To find these the raw data was evaluated, and trial and error gave what was a decent guess. This was relatively straight forward in the simulator since you can accurately define the initial states and tune initial covariance matrices. However, for the real-world environment these should be chosen with caution. A rule of thumb is to preferably initialize the covariance matrix with large values to ensure a feasible convergence procedure sacrificing potential convergence speed.

5 Results

The overall position estimation of the filter, both in the simulator and with real data, can be seen in the figures 1 and 4. In the case of the simulation, where the ground truth is given, the NEES plot is provided in figure 2 and the following states in figure 3. For the real world case, where ground truth does not exist, the NIS plot is provided in figure 5 and the states of the filter in figure 6.

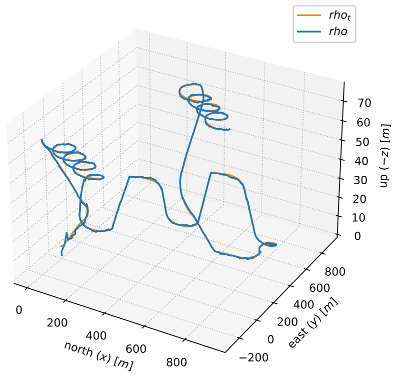

Figure 1: 3D plot of position of UAV in the simulation environment

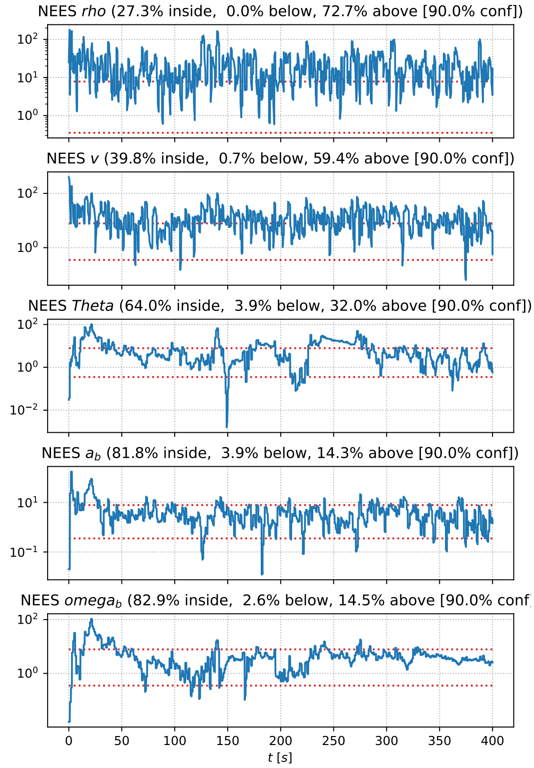

Figure 2: NEES plot for the simulation environment

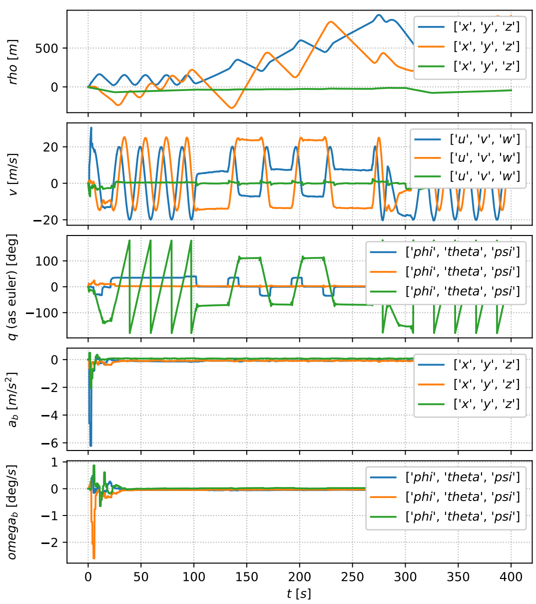

Figure 3: States in the simulator environment

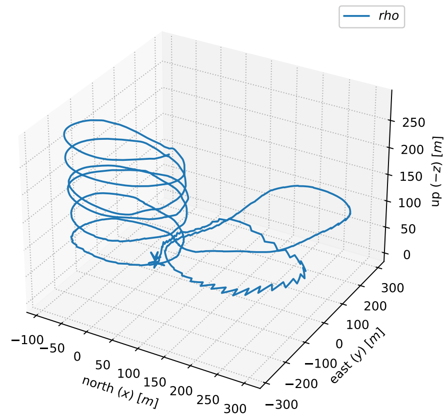

Figure 4: 3D plot of position of UAV in the real environment

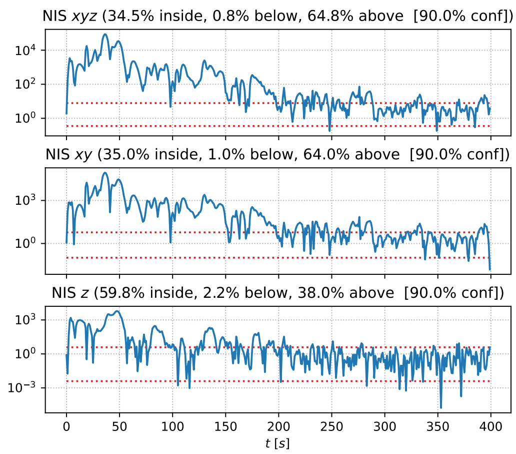

Figure 5: NIS plot for the real-world environment

6 Discussion

It takes significant time from $T = 0$ until the filter has converged to a solution in both the simulation environment and the real-world environment, but the simulator converges faster. This behaviour could be the result of initial states and biases that are poorly estimated, making the filter use more iterations to do major corrections before converging.

In the case of the simulation the NEES plots in figure 2 show both attitude and ve-

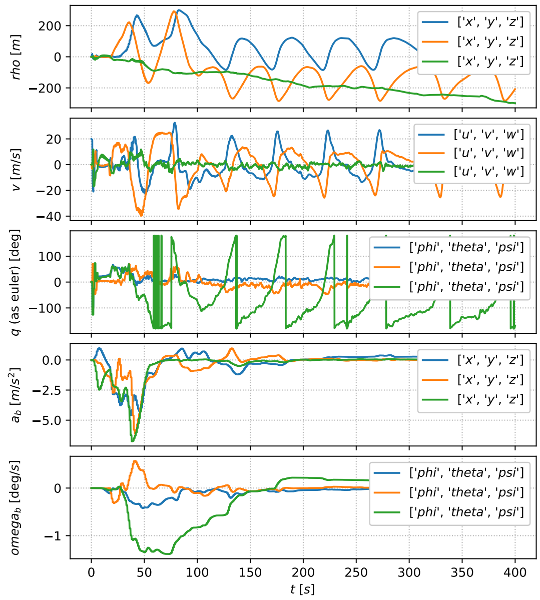

Figure 6: States in the real-world environment

locity are consistent in the long term, however they are both above the values described in the confidence interval.

Tuning on the real-world environment data set is mainly based on analyzing the NIS value shown in figure 5. Since no ground truth and exact noise models exist it is hard to validate the quality of the estimates. Estimating the heading using GNSS and IMU is dependent on several measurements. This makes it critical that the UAV is moving for heading estimation. In the case of standing still the filter is quite unstable. This is observed during the start phase at ground level, shown in figure 4, where the UAV is not moving.

References

[1] Sensonor AS. *Datasheet STIM300 Inertia Measurement Unit*. TS1524 rev.15. December 2013.