INS - ESKF

Harald Minde Hansen and Peter Hatlebrekke Husebø

October 30, 2020

Abstract

This is the report for the Graded Assignment 2 in the course TTK4250 - Sensor fusion. The report consists of two sections, where section 1 answers Task 3 of the assignment and section 2 answers Task 4 of the assignment. We have no common conclusion for the whole assignment due to the restriction of four pages, but we make some comparisons with section 1 in section 2.

Run ESKF on simulated data

Tuning of the parameters

In this task, we were to tune the parameters of an ESKF run on a data set simulating a UAV. Since we have the ground truth available on this simulated data set, we chose to set the initial state equal to the initial ground truth state, even though this is not a realistic approach on real data sets. Then, the initial covariance could be set to be small, since we are certain that at least our first state error is zero. However, we did not set this covariance to zero, as this would cause the predicted covariance to be extremely small for each new IMU measurement in the prediction step, almost only increasing given the covariance in the process disturbance matrix $\boldsymbol{Q}$. Then, in the update steps in the ESKF, almost all the uncertainty will be assumed to be in the GNSS measurements. But setting the initial predicted covariance relatively low yielded quite good results. But, only 66% of the NEES for the gyro bias was inside the confidence interval. Also, a lot of this NEES was above the upper bound in this confidence interval, which implied a too low predicted covariance matrix in the KF and hence an overconfident filter since the IMU measurements works as inputs. Therefore, we increased the driving noise standard deviation for the bias of the gyro to make the model prediction count less in the filter update steps. This yielded more NEES inside the confidence interval both for the gyro bias and overall as expected and it now seemed to be neither over- nor underconfident, as can be seen in fig. 4. The ANEESes ended up outside the confidence interval, but from the NEES plots we see that most of it is inside.

The GNSS measurement noise was kept as given, as it was in about the same order of magnitude as the difference between the ground truth and the GNSS measurements. Setting this higher would make us trust the predictions from the model to a larger extent causing an overconfident model, and in general make us less certain on our filter than necessary. Setting it lower would make us trust the measurements more, such that the estimates would be more sensitive to the GNSS noise. In addition, this would yield a more conservative model.

The IMU inverse bias time constants were set to be very small, since this is the reciprocal of a large time constant of the bias when modelling it as a Gauss-Markov model. This large time constant works as a damping of the bias error (from equation 10.50 in the book) and can be seen as a saturation for the bias when it reaches large values, such that the bias will not go towards infinity in reality due to this time constant. Setting the IMU constant larger implies a smaller time constant, which will limit the bias to a larger extent, which again can cause us to omit some of the bias in our filter. This is of course not considerable, so we kept the IMU inverse time constant at this high value, as increasing it further would cause the saturation effect almost not to be present at all, such that we instead would end up with a Wiener process for the bias that drifts indefinitely.

We observed the total NEES inside the confidence interval was significantly less than for each of the individual states alone (position, velocity, attitude and biases), due to the coupling elements between the states that is off-diagonal on the estimated covariance matrix, which will be more difficult to control as this dynamic is more complex. Therefore, we focused on getting most of the decoupled NEESes inside the confidence interval for the state elements, and then see if we got the total NEES to be reasonable out from this.

Observability of heading angle

The heading angle will be observable if we approximate it to be equal to the course angle, i.e. setting the sideslip angle to be zero. Then, the heading relative to the north axis in NED can be calculated as the inverse tangens of the east component of the velocity divided by the north component of the velocity. However, when the UAV is standing still, it will have no speed and hence zero velocity components, such that the heading will be undefined. Also, when the vehicle turns, it will take some time for the heading to get stable velocities from the derivative of the GNSS postition measurements since the sideslip angle is non-zero such that the heading is different from the course. Therefore, the estimated heading will drift from the correct over time, before the GNSS samples corrects it again during the turns. When the UAV is moving more straight again, the GNSS will be able to make it converge towards its true values after some time.

Simplification of misalignment matrix

When we sat the S_a and S_g matrices to identity and hence omitting the mounting, scaling and orthogonality errors, we got larger errors than before, as expected. The most significant lack of performance was observed to be in the NEES and RMSE-values of both position and speed. $\mathbf{RMSE_{pos}} = 0.4141m$ and $\mathbf{RMSE_{speed}} = 0.1269m/s$ for corrected misalignments changed to $\mathbf{RMSE_{pos}} = 0.9183m$ and $\mathbf{RMSE_{speed}} = 0.5181m/s$ for uncorrected. As the state estimates will be based on violated assumptions of the axes standing perfectly orthogonal to each other, no scaling errors present and the IMU mounted perfectly aligned with the body axes. However, the NIS did not get the same lack of performance, as this is the innovation step where the measurements from the GNSS manage to correct some of the large state estimate errors to some degree. Due to this, the ESKF still manages to track the UAV, yet with a poorer accuracy. Still, the NIS for the $\mathbf{S_a} = \mathbf{S_g} = \mathbf{I_3}$ was not as consistent as for the corrected one, due to the propagation of errors in the predicted $\mathbf{x_{nominal}}$-state every time step using uncorrected IMU-inputs, which effects the calculated innovation, and thus the NIS.

Run ESKF on real data x

Tuning of the parameters

In this task, we were to tune the parameters from the previous task on a real data set with no ground truth available. The tuning parameters were discussed in the previous task, so we would expect this tuning to yield an appropriate tracking. The intial attitude was given in the code, and we sat the intial prediction of the position state equal to the first GPS measurement, while the rest was set to zero. Assuming small values in the predicted covariance matrix in the beginning worked well due to the low initial NIS, as seen in fig. 5. The huge spike at 250 seconds is due to the launching of the UAV. The biases has been erroneously estimated $\implies$ error in prediction $\implies$ spike in NIS. Following the hint, we started with decomposing the NIS into planar and altitude to see how it behaved for each of these to make things more tangible. Here, we omit the elements in the S matrix that represent the coupling between the new decompositions, but this anyway seems as an reasonable approach, as it is difficult to control these parameters. The total NIS seemed in total to be quite low, except from a couple of large spikes. This seem to partially propagate from the spikes in the given `accuracy_GNSS` -dataset we use as basis for the uncertainty in the GPS measurements. Also, the decomposed NIS in the North-East-plane and altitude were overall low-valued, which made us decrease the standard deviation of the GNSS measurement to make this correct the estimates to a larger extent each second. This decrease led to a higher NIS, as seen in fig. 5. It could be lifted higher by tuning in the same direction, but then the ANIS will increase above the upper bound due to the spikes. This leads to another problem, how to possibly deal with the occuring spikes. Maybe a starting point to do this would be to look more into the mentioned spikes in the `accuracy_GNSS` - dataset and other possible factors creating the spikes.

Simplification of the misalignment matrix

Setting the misalignment matrix to the given simplification only compensating for the mounting orientation, but not mounting, orthogonality and scaling errors yielded a total NIS of 72% inside the confidence interval. Also, the bias for the gyro increased, which we interpreted as that the minor errors from the simplification will cause a small drift from the correct frame, which will give much of the same drifting pattern as a Gauss-Markov process noise. Hence, the model could interpret this drift in the slightly rotated frame as a bias for the gyro.

State	Value	Conf-int
ANEES_pos	2.66	[2.98,3.02]
ANEES_vel	2.03	[2.98,3.02]
ANEES_att	1.55	[2.98,3.02]
ANEES_accbias	1.99	[2.98,3.02]
ANEES_gyrobias	0.98	[2.98,3.02]
ANEES	30.11	[14.96, 15.04]
ANIS	2.86	[2.84, 3.16]

Table 1: ANEESes and ANIS for simulated data

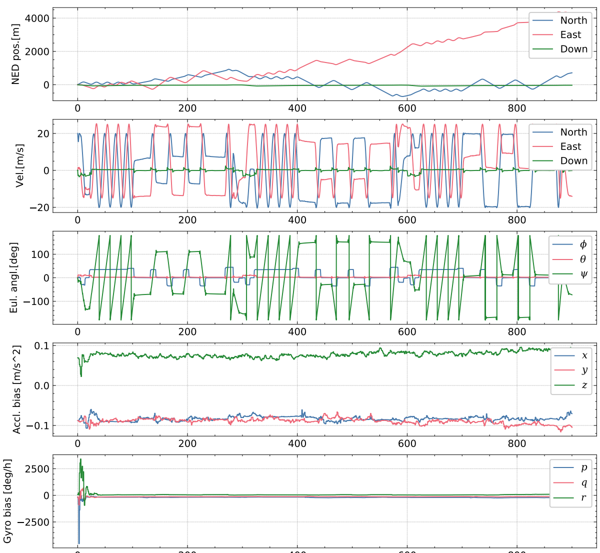

Figure 1: The simulated data state estimates

State	Value	Conf-int
ANIS(full correction)	2.11	[2.92, 3.08]
ANIS(basic correction)	1.45	[2.92, 3.08]

Table 2: ANIS for real data, with full correction and basic correction of mounting errors

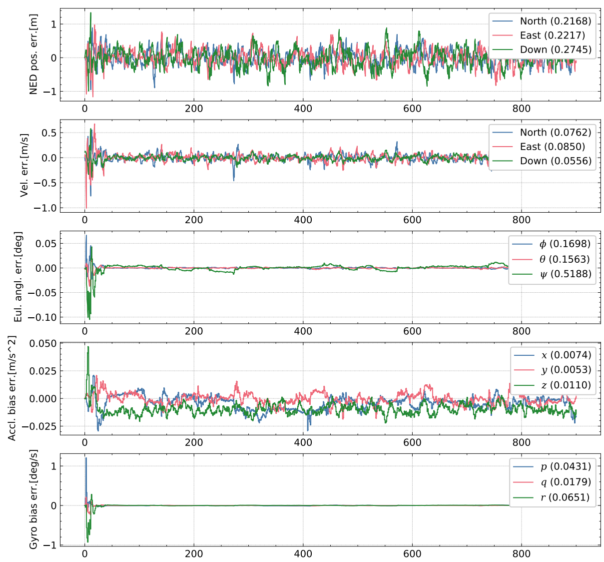

Figure 2: The simulated data state estimate errors

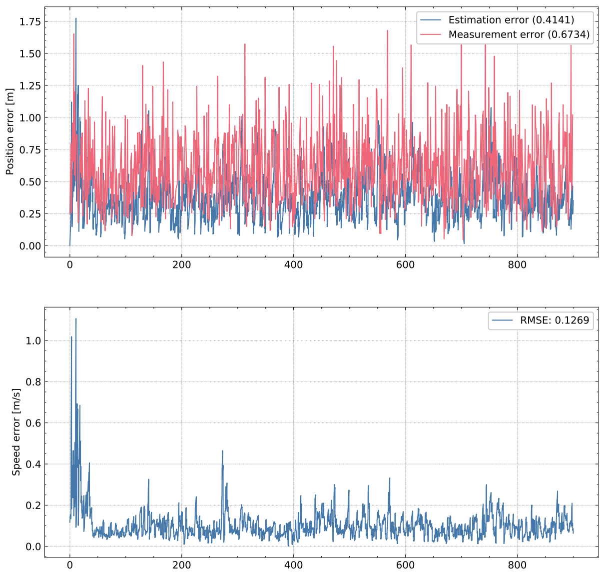

Figure 3: Measured and estimated position error and speed estimate error

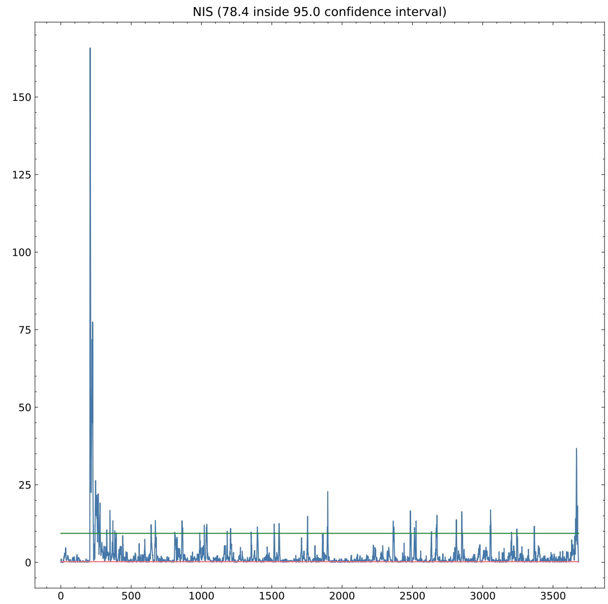

Figure 5: The real data NIS

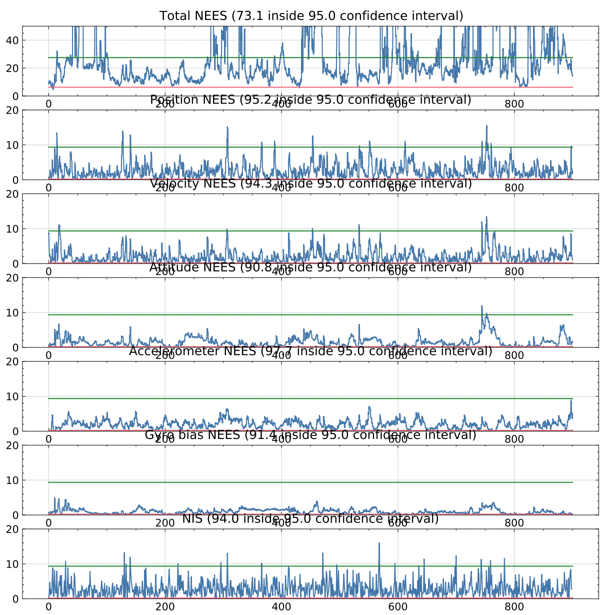

Figure 4: The simulated data NIS and NEESes

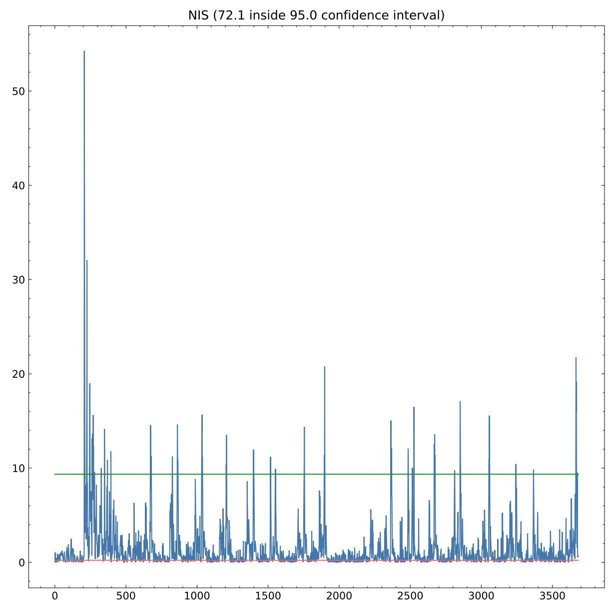

Figure 6: The real data NIS with basic mounting directions