NTNU
Kunnskap for en bedre verden

Department of Engineering Cybernetics

TTK4250 - Sensor Fusion

Graded Assignment 2

Authors:
Even Eliassen
Rendell Cale
(Group 64)

October, 2020

1 Introduction

Error-state Kalman filter (ESKF) is a fundamental part of an inertial navigation systems (INS). It provides a way to fuse external and infrequent position measurements like GNSS with internal and frequent inertial measurements from inertial measurement units (IMU). In this project we have implemented an ESKF using the equations given in chapter 10 in [1]. We tune the implementation on simulated and real datasets, and investigate the effect of IMU mounting, scaling and orthogonality errors.

2 Simulated data

2.1 Tuning Process

After implementing the INS we looked at the tuning parameters, which are the IMU measurement noises, the IMU bias parameters and GNSS position noise, and the initial state and covariance matrices. For tuning we were given good noise and bias parameters and initial state. The goal was then to reduce the root mean square error (RMSE) by fine-tuning the given parameters and finding good initial covariances. We are especially interested in having a consistent normalized estimation error squared (NEES) and normalized innovation squared (NIS), since by getting those consistent with an appropriate χ^2 distribution, the error in the state estimate is well described by its covariance. This is perhaps more important than getting the best RMSE.

First, we set the elements of the initial covariance matrix. We assumed no correlations and used physical considerations to get realistic numbers. The position, velocity, accelerometer bias and gyro bias were simplified to $\sigma_{p0}\mathbf{I}$, $\sigma_{v0}\mathbf{I}$, $\sigma_{ab0}\mathbf{I}$, $\sigma_{gb0}\mathbf{I}$, respectively. We parametrized the angle error covariance by a diagonal matrix $\mathrm{diag}\left(\sigma_{\phi\theta 0}, \sigma_{\phi\theta 0}, \sigma_{\psi 0}\right)$. Then thought about how many meters the initial position was off, how many degrees the attitude was off, the maximum bias values and so on. The initial covariance mostly affects the filter in the beginning, and we saw that the NEES values were especially sensitive to these values in the beginning. If (A)NEES was big relative to its confidence interval, increasing the covariance pushed it down and vice versa. This reflects how NEES is defined mathematically. We had a big spike in NEES values in the beginning, and compensated by increasing the initial covariances. Changing the variance of one state affected not only the corresponding NEES but all of them. This made fine tuning a little difficult.

After setting the initial covariance matrix, the individual NEESes corresponding to position, velocity, attitude and biases were quite good, but the total NEES was way off. This indicated that there was some correlation (off diagonal elements) in the covariance matrix that only affects total NEES. We noticed that the total NEES jumped when the heading went from 180 to -180 degrees. To figure out where this correlation was, we plotted NEES values of various combinations of states, and found that especially the NEES of attitude error and the gyro bias was causing the jumps. Zooming in on the plot of the attitude error, we saw that when the heading jumped, the attitude error switched sign. Thinking about how the correlated part of NEES is calculated, it makes sense that this contribution to the NEES would switch sign, causing jumps. It was strange that the attitude error would change sign, as our quaternion estimates were almost smooth. However, taking a look at the true attitude state showed that while our estimate was smooth, the true state was not. A positive and a negative quaternion represent the same rotation, and it is common to restrict the quaternions to only having positive real part. This means that when the estimated quaternion had negative real part, the error attitude switched sign. To account for this in the calculation of NEES, we adjusted the calculation of the true error state, so that the attitude error always had positive real part. This made the total NEES reflect all of the individual NEESes.

To investigate the tuning sensitivity to GNSS noise matrix $\mathbf{R}_{\mathrm{GNSS}}$ we decomposed NIS into x, y, z and xy. Our thought being that the planar measurements have similar noise, and that the vertical measurement from GNSS is less accurate. By perturbing the values corresponding to planar uncertainties, we found that the planar NISes (x,y, and xy) changed, and the effect on NIS z was small. By modifying $\mathbf{R}_{\mathrm{GNSS}}$ we were able to get more consistent NIS, but this lead to less consistent NEES. We found that the values given in the assignment were good so we didn't modify them.

We visualized the NEESes and NISes using a boxplot and saw that the NEES of the gyro bias was especially high. To change this, we increased the driving noise of the bias. This greatly improved the NEES. We did not see the need to adjust the accelerometer biases since the boxplot showed their NEES was already consistent. Trying different bias time constants showed little change in results, so we kept them as they were.

After getting decent NEESes, we noticed that ANEES was too low. To try to combat this, we decreased IMU measurement noises. This increased the NEESes and ANEESes.

2.2 Results and Discussion

The tuning we ended up with is given in table 1. We note that the error state process noise matrix is parametrized as $\tilde{\mathbf{Q}} = \text{blkdiag}\left(\sigma_{\text{acc}}\mathbf{I}, \sigma_{\text{gyro}}\mathbf{I}, \sigma_{\text{acc,bias}}\mathbf{I}, \sigma_{\text{gyro,bias}}\mathbf{I}\right)$. Some of the process parameters seem oddly specific, because we scaled the parameters that were handed out which in turn are based on the datasheet of STIM300, but to replicate our results it is probably not necessary to specify the parameters to the same number of decimals as given in table 1. The corresponding results are given in figs. 1 to 3, and table 2. We see that the gyro biases fluctuate quite a bit before converging after about 50 seconds. This is reflected in all the states and can also be seen in the RMSE values. Comparing the estimate to GNSS, we found that the estimate has a mean position error of 0.389, while GNSS has a mean error of 0.673, which means that using IMU in addition to GNSS has improved our performance over using only GNSS.

Since gravity is aligned with the (NED) z-axis, there is no information about heading coming from the IMU alone. What makes heading observable is non-zero velocity and that we update the prediction with GNSS measurements. This means that the heading will not be correctly estimated when standing still.

Looking at table 2, we see that most of the ANEES values are a little below their confidence intervals, which probably could be improved by decreasing uncertainties. However, it is better that they are too low than too high, since too high values indicates that the filter is too confident about its estimates. There are still some correlation effects causing the total (A)NEES to be more outside (and above) its confidence interval than the ones corresponding to different states.

After viewing and analyzing the results we looked at the IMU's misalignment matrices. We set them to identity matrices and found that RMSE values increased significantly, and the NISes became inconsistent. We believe the

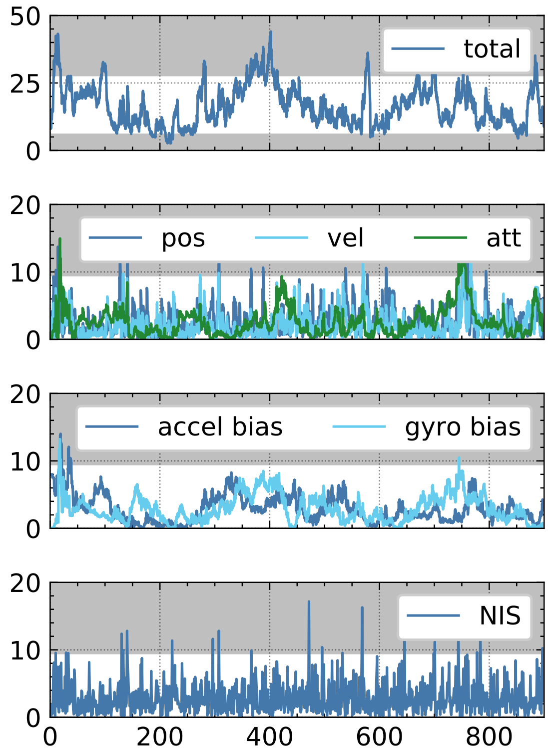

Figure 1: NEES and NIS over time. Grey region is outside 95% confidence interval.

NISes could be improved by increasing noise parameters, but did not try this. It makes sense that the results would be worse since the IMU data that is fed into the filter will essentially be less accurate. This was apparent when looking at the accelerometer bias. Without the misalignment matrices, the bias estimates would fluctuate presumably to compensate for the misalignment. In a real system we could disregard this, but would then have to accept more errors. The vehicle also becomes more dependent on GNSS measurements.

3 Real data

3.1 Tuning Process

In this task the data was provided by a STIM300 IMU giving 250 Hz measurements and a Ublox-8 GNSS giving 1 Hz measurements. For the INS, we started with the tuning from the above and adjusted for different sample time. The initial position, velocity and attitude was kept as in the hand out code. For the GNSS noise, we used a diagonal matrix multiplied with the GNSS accuracy, which is estimated by the GNSS.

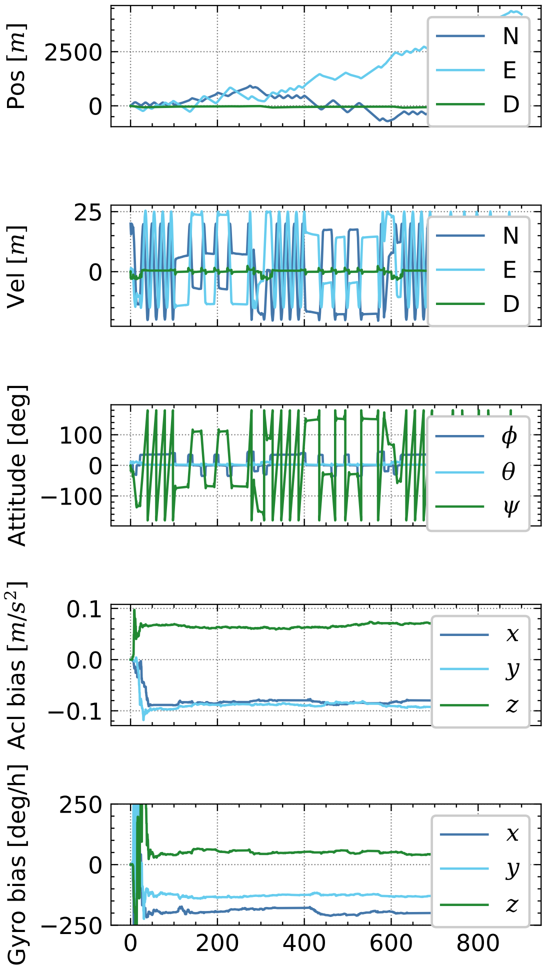

Figure 2: State estimates on simulated data.

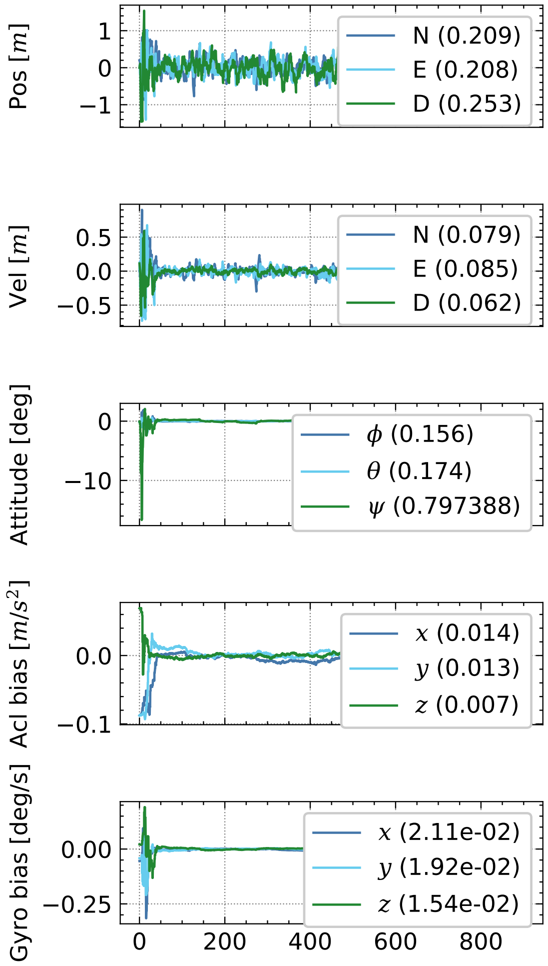

Figure 3: State estimate errors with RMSE values in parenthesis.

Parameter	Value
p_{acc}	10^{-16}
p_{gyro}	10^{-16}
σ_{p0}	7.5
σ_{v0}	7.5
σ_{ab0}	0.05
σ_{gb0}	0.005
$\sigma_{\phi\theta 0}$	$\pi/30.0$
$\sigma_{\psi 0}$	$\pi/3.0$
σ_{gyro}	0.0001744
σ_{acc}	0.0116700
$\sigma_{\text{gyro,bias}}$	0.0001744
$\sigma_{\text{acc,bias}}$	0.0004360
$\mathbf{R}_{\text{GNSS}}$	$\text{diag}([0.3^2, 0.3^2, 0.5^2])$

Table 1: Tuned parameters for simulated dataset

	Inside	Averaged	CI
NEES	87.0%	16.533	(14.964, 15.036)
NEES pos	95.8%	2.754	(2.984, 3.016)
NEES vel	94.2%	2.243	(2.984, 3.016)
NEES att	93.1%	2.604	(2.984, 3.016)
NEES gyro bias	94.5%	2.923	(2.984, 3.016)
NEES acc bias	96.5%	2.907	(2.984, 3.016)
NIS	94.2%	2.933	(2.984, 3.016)

Table 2: Consistency evaluation for simulated dataset

As the true state is unknown, we only had NIS to tune after. The NIS was very low, except for a huge spike at around 210 seconds when the UAV takes off. We were not able to find a way to remove the spike while at the same time increasing the rest of the NIS. We therefore decided to focus on NIS values after 1000 seconds, accepting inconsistent estimates during take off. To increase the NIS, we suspected lowering the GNSS noise would work. This was indeed the case. We also tried lowering the IMU noise. This also increased NIS. We settled for a combination of these two strategies.

To get more information when tuning, we did as with the simulated data and calculated NIS for each axis, as well as the planar axes. Here we could see that the vertical NIS was higher than the planar NISes for the same GNSS noise. This indicated that there was more uncertainty in the vertical than in the planar GNSS measurements. This could be compensated for by increasing the GNSS accuracy in the vertical axis.

3.2 Results and Discussion

The tuning we ended up with is the same as in section 2.2 except for the values given in table 3, and the corresponding results is shown in fig. 4 and table 4.

To investigate the spike in NIS just after take off, we looked at the norm of the estimated covariance, which increased on takeoff, but not enough to keep NIS within the confidence interval. We tried to compensate by using higher noise parameters here, but this led to inconsistent results in the rest of the flight. This is especially clear in the NIS starting at mid flight at 1000 seconds, which is also given in table 4. The total (A)NIS is much higher than the confidence interval, but after 1000 seconds the A(NIS) was much lower than the confidence interval.

We also noticed that the biases converged while the UAV was not in motion, but with higher covariance norm than when in motion. We believe this happens because the bias estimation rely on orientation, and heading cannot be estimated when standing still.

We tried setting the IMU's misalignment matrices to simplified rounded matrices. The effect of this was hard to notice when we did not have the ground truth to compare with. The NIS values changed, but could probably be improved by tuning. As with the simulated data, we noticed fluctuations in the biases especially during attitude changes. We think this is the filters way of compensating for the mounting, orthogonality and scaling errors, and these fluctuations could be noticed in a real world application.

Parameter	Value
σ_{gyro}	0.0002180
σ_{acc}	0.0058350
$\sigma_{\text{gyro,bias}}$	0.0002180
$\sigma_{\text{acc,bias}}$	0.0002180
$\mathbf{R}_{\text{GNSS}}$	$(0.5(\text{GNSS acc}))^2\mathbf{I}$

Table 3: Parameter changes for real data compared to simulated data

	Inside	Averaged	CI
NIS	77.0%	15.148	$(2.995, 3.005)$
NIS after 1000s	81.9%	1.29	$(2.995, 3.005)$

Table 4: Consistency evaluation for real dataset

4 Conclusion

We tuned the ESKF for both simulated and real data, with the main goal of getting consistent performance. The tuned filter gave low NEES and NIS, indicating that the filters were underconfident. In the real data, the NIS was very different during takeoff and flight. We were not able to mitigate this, but decided that consistency during flight was more important. By neglecting the IMU misalignment matrices, we obtained significantly worse performance on both datasets, so these matrices are important to account for in real systems.

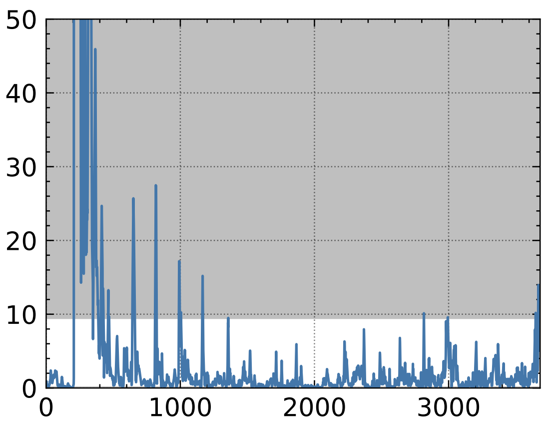

Figure 4: NIS over time. Grey region is outside 95% confidence interval.

Bibliography

[1] Edmund Brekke. *Fundamentals of Sensor Fusion*. 2020.