Graded assignment 2: Error-state kalman filter

Jacob H. Dahl, Viktor Korsnes

November 13, 2020

1 Introduction

This report discusses and displays the result of a vessel using a error-state Kalman filter (ESKF) in combination with an inertial measurement unit (IMU) and a global navigation satellite system (GNSS) to estimate its own position in two different data sets, one simulated and one from a real UAV. The IMU gives high frequency relative data on the forces acting on the vessel, which can be integrated to find the position. The GNSS gives low frequency absolute data that can be used to correct for the drift when integrating a noisy sensor. The filter performance is analyzed using mainly normalized estimation error squared (NEES) and normalized innovation squared (NIS).

2 Simulated Data

In order to tune the system, the noise covariance matrices $\mathbf{Q}_{IMU}$ and $\mathbf{R}_{GNSS}$ has to be carefully chosen. Usually the data sheet and empirical data is a good starting point, however, for this data set the ground truth is available, which enables us to use real data in order to estimate the covariance.

By looking at the mean squared error between the GNSS measurement and the ground truth the $\mathbf{R}_{GNSS}$ can be estimated. For the IMU this is not as trivial. A method of estimating the IMU noise is to fix it in place, and sample data over a long period of time. Then you can easily extract the parameters from that data. In this example that is not possible.

The IMU has some misalignment and to compensate for this two matrices $\mathbf{S}_g$ for the gyroscope measurement and $\mathbf{S}_a$ for the accelerometer measurement is given. Using these matrices give good results, however when ignoring them the filter performed better. This was unexpected, however looking at the data sheet for the IMU the misalignment is set to 1. Most states remained within 1-5% of the NEES value with the misalignment matrices, but the gyro bias got a 40% better NEES value without $\mathbf{S}_a$ and $\mathbf{S}_g$. This is strange since the only difference in misalignment was in the accelerometer, not the gyro. However since the system is coupled, these effects are somewhat to be expected.

3 Real-world Data

Using the filter on real-world data is different from simulated in the simple sense that we lack a ground truth. This makes the task of tuning the filter more difficult. Using the GNSS values and comparing them to our estimates is a start, but one fall into the trap of tuning the filter to be exactly like the GNSS, which is not the desired effect of an ESKF. Seeing as we have the noise values for the process model from the INS data sheet it is possible to get good estimates from the process model immediately. The remaining task is to weight the process model versus the GNSS-updates to get an optimal estimate.

In the real-world data set a similar test to the misalignment test ran in the simulated case. Running the filter on the same data with and without the basic misalignment correction. The difference is subtle, but the x- and y-estimates of the acceleration bias flipped in sign. Seeing as they were close to zero it did not have a noticeable effect on the state estimates as a whole, but with a less accurate IMU with a higher bias-value it could cause issues with the rest of the estimates as well. By testing the IMU with some controlled movement and reading off the data, these misalignments can be found to some degree, but if high accuracy is desired then it is tricky to find the appropriate values for $\mathbf{S}_g$ and $\mathbf{S}_a$

4 Tuning

One of the benefits of using an ESKF with IMU data as control-inputs is the lack of having to tune the process model. The overall tuning strategy was to start with a small set of data-points (500-1000) and increase up to the whole data set as the results improved. The covariance of the process model is given by the covariance of the IMU data which is normally available in the IMU datasheet [1]. Using the values available for the STIM3000 gave a continuous gyroscope standard deviation $\sigma_{gyro} = 4.35 \times 10^{-5}[\frac{rad/s}{\sqrt{Hz}}]$, and a continuous accelerometer standard deviation $\sigma_{acc} = 1.17 \times 10^{-3}[\frac{m/s^2}{\sqrt{Hz}}]$. Raising these values an order of magnitude or two on the simulated data gave an overall worse performing filter and raising them up to $1-10$ gave divergence. On the real-world data raising the values to 10^{-2} and 10^{-1} gave an underconfident filter with NIS 0 which means all the estimates are identical to the GNSS measurements when available, and works as a poor interpolation between the GNSS updates when they are not available. Since it discards all the information available from the process model it is considered a sub-optimal tuning. Decreasing the noise values two orders of magnitude gave no difference in performance in the simulated case or the real-world case. Decreasing it another two orders made the filter overconfident and caused drift in the estimated values. However seeing how low the IMU noise had to be set to experience this effect, one can conclude that the IMU measurements are highly reliable, given adequate tuning on the biases.

The acceleration- and gyroscope biases driving noises were tuned first by looking at the suggested values in the range of 10^{-3} to 10^{-5}, however these gave sub-optimal NEES-values. Decreasing them further gave even worse results. Increasing them an two orders of magnitude gave the best results in the simulated case with $a_w = 0.4[\frac{m^4}{s^4}]$ and $\omega_w = 0.02[\frac{rad^2}{s^2}]$. These values are much higher than anticipated, but were necessary for a consistent result in the simulated case. Normally these values should be lower than white noise, but this might come from cancelling out a poor tuning in the initial gyro- and/or acceleration bias covariance. In the real-world case the values were tuned similarly and were much lower as expected with $a_w = 10^{-5}[\frac{m^4}{s^4}]$ and $\omega_w = 10^{-5}[\frac{rad^2}{s^2}]$.

The GNSS covariance was tuned by looking at the ground truth available in the simulated data and gave a resulting $\mathbf{R}_{GNSS} = 0.09[m^2] \times \mathbf{I}_{3\times3}$. Increasing the covariance puts less emphasis on the GNSS updates and makes the filter overconfident, the opposite action makes the filter underconfident.

One weakness of the ESKF is the need to initialize the filter correctly. Given ground truth this is easy, but the initial covariances also need initial tuning to ensure fast convergence. Erring on the side of caution, it is preferable to set the covariances too high rather than too low. Sacrificing a quicker convergence for the safety of assured convergence. In the simulated case the values set to $10^{-4} \times \mathbf{I}_{3\times3}$ in all states except the acceleration bias which was set to $10 \times \mathbf{I}_{3\times3}$.

Finally the Gauss-Markov time constants

NEES	In 95% CI	Average
Position	89%	4.10
Velocity	93%	2.96
Attitude	76%	2.22
Acc bias	80%	0.94
Gyro bias	80%	0.56
Total	56%	35.7
NIS	90%	3.80

Table 1: Resulting NEES values and NIS value for the simulated data set within the 95% confidence interval and the averaged value

were tuned to $T = 10^{16}[s]$ in both simulated- and real-world data, however, no real difference in filter performance was seen in the tuning of this between 10^2 and 10^{32}.

5 Results

The resulting NEES-values for the simulated data is presented in table 1 and the plots of NEESes and NIS are presented in fig 1. State estimates are presented in fig 2 and the state errors in fig 3. For the real-world data, the lack of ground truth made the tuning more dependant on the NIS values presented in fig 4.

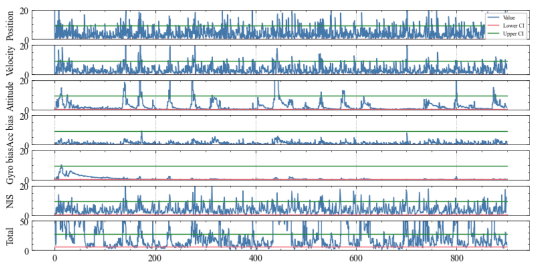

Figure 1: NEES and NIS values for simulated data set

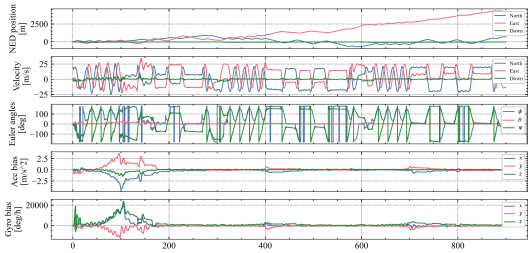

Figure 2: State values for simulated data set

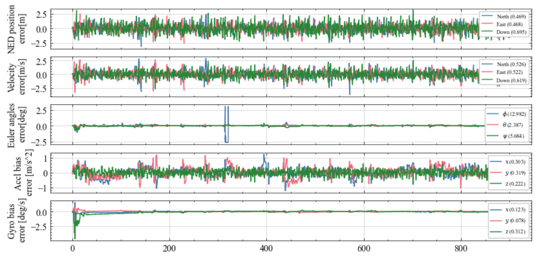

Figure 3: State errors for simulated data set with corresponding RMSE values in brackets

6 Discussion

Looking at the state errors in fig 3, most states have a small error, however there is some estimation error in the gyro bias at the start. This is the result of a poor initial estimate of the gyro bias and its covariance. However it does not converge quickly enough to not cause too much harm to the other states. There is also some non-physical spiking of the roll angle error. This is caused by some irregularities in the roll estimate, however the reason for this has not been investigated, and is to the authors, not obvious from the data.

The real-world data set is tuned mainly based on the NIS value shown in fig 4. Some tuning was made to make the NIS 0, however this is not desirable as explained in the tuning section. This final tuning

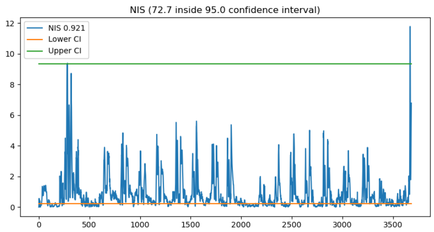

Figure 4: NIS of real data set with RMSE (in brackets) and percentage within confidence interval

was seen as a fitting middle-ground for the real-data. As no ground truth is present, it is a difficult task to say if this gives a good filter. The optimal filter is given when the noise values for all sensors and models are known exactly, but without ground truth this is not available. However since the model noise is given by the IMU and the GNSS noise was found in the simulated task, this resulting tuning should give a better estimate than the model- or the GNSS alone.

When estimating the heading with one IMU and one GNSS measurement the system is dependent on the previous measurement in order to *see* which direction the vessel is headed in. If the vessel is standing still the heading will be based on the noise of the GNSS measurement, and will not be reliable at all. That is why the system needs to move in order to converge on the true state. This is most notably for the yaw angle, as the gravity force makes the roll and pitch observable. This effect can be seen in fig 3 where the yaw-angle has the highest RMSE-value (ignoring the spiked up roll angle which inflates its RMSE). The yaw can be seen to spike up at several points which correspond to constant velocity in the UAV.

One way to mitigate this problem is using dual GNSS measurements on the vessel with a known vector between them, however, while standing still and depending solely on the GNSS measurements the system will converge slowly due to the low sampling frequency. Another solution is to add a magnetometer or similar devices that are able to measure the heading directly.

References

[1] *Datasheet STIM300*. TS1524 rev.9. Sensornor AS. May 2013.