Implementation of ESKF

Carl Fredrik Haakonsen and Marcus Notø Hansen

October 2020

Abstract

Navigation systems has become an essential tool in modern society. Its application stretches from all types of vehicles to robotics. Estimating ones own position is done by using interoceptive and exteroceptive senors. In this report we will be looking at the performance of an ESKF using IMU and GNSS measurements. The ESKF will first be used on a simulated data set with a ground truth, before it will be implemented on a real data set using real sensor not containing ground truth. The simulated data set shows how the ESKF can use multiple sensor to estimate the trajectory better than the GNSS, but also showing the sensitivity of the initialization values. The implications of implementing the ESKF on a real data set was explored as no ground truth is available along with real life factors as wind, temperature etc. Despite this, a filter maintaining track with decent performance was implemented on the real data.

Introduction and problem setup

The goal of this report is to is to implement an error-state Kalman filter (ESKF) using an inertial measurement unit (IMU) as control input and complementary measurements from a global navigation satellite system (GNSS). The IMU consists of a gyroscope and an accelerometer. In the first part of the report simulated data is used, making tuning easier as ground truth is available. Results are presented and discussed based on the tuning parameters. Then the ESKF has been applied to a real data set from a UAV flight, and further analyses are made. In the conclusion the main results from the report are presented and discussed.

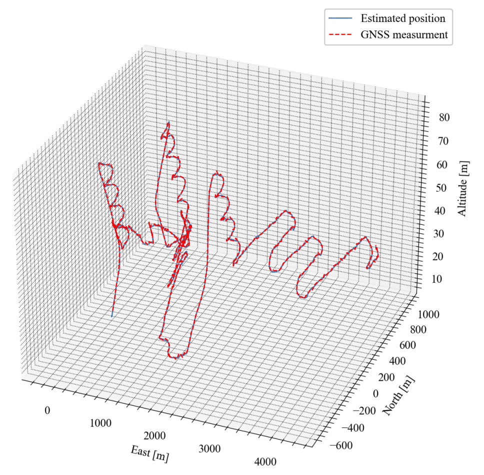

Figure 1: Estimated UAV trajectory with corresponding GNSS measurments in 3D.

ESKF for simulated data set

Before tuning an ESKF on real data it can be useful to first implement and tune the ESKF on simulated data. The ESKF has been implemented using the theory described in [1]. The tuning mainly consists of finding reasonable values for the IMU measurement noises, bias time constants and driving disturbance for the IMU as well as the GNSS position noise. The initial parameters were based on common sense and typical values for IMU and GNSS measurement noise. They were then tuned based on RMSE, NEES and NIS values. The parameters after tuning are presented below in Table 1. The estimated trajectory using the parameters in Table 1 is shown in Figure 1 along with the GNSS measurements.

Table 1: Accelerometer and gyroscope parameters

σ_a	σ_{ab}	σ_ω	$\sigma_{\omega b}$
$5x10^{-4}$	$6x10^{-3}$	$2.9x10^{-3}$	$3.3x10^{-6}$

σ_a and σ_{ab} are respectively the accelerometer measurement noise standard deviation and the bias driving noise standard deviation. The same applies for σ_ω and $\sigma_{\omega b}$ for the gyroscope. The bias instability values p_{ab} and $p_{\omega b}$ were set to 10^{-16}, while the GNNS noise covariance was set to a diagonal matrix corresponding to the vector $[0.3^2, 0.3^2, 0.5^2]$.

The most sensitive parameters when tuning was the IMU bias driving covariances. The $\sigma_{\omega b}$ was especially sensitive to changes. If set too low, the bias estimate do not converge, resulting in larger errors in the filter. A converging bias estimate is essential for a stable filter, and tuning the bias driving noises should be done appropriately. Other parameters sensitive to tuning is the measurement noise for the GNSS. In a real case the accuracy estimation in standard deviation will however be provided by the GNSS receiver.

We see that the filter maintains track throughout the entire flight. The estimated states at given time steps of the flight in Figure 1 are shown in Figure 2.

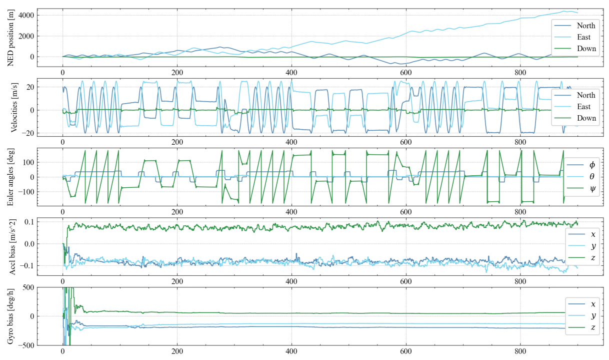

Figure 2: Estimated UAV states for simulated data set.

As ground truth is available for the simulated data, the error states can be calculated. These are shown in figure 3.

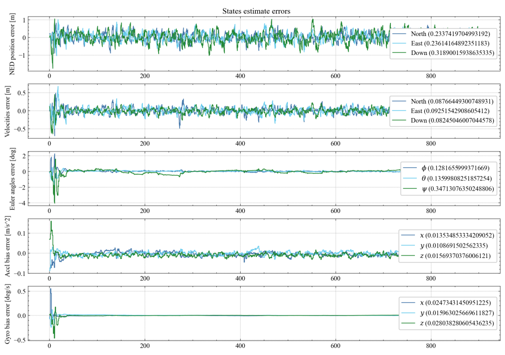

Figure 3: Error state between estimation and ground truth.

First thing to note is that position RMSE for Down is slightly larger than for North and East. This is to be expected as the GNSS measurements are less accurate in Down direction. Further we notice that the yaw angle ψ tends to have periods where the estimate error is significantly larger than that of ϕ and θ, resulting in over 2.5 times the RMSE value in ψ compared to θ and ϕ. The reason for the periodical error in the heading lies in the physical world and not necessarily the tuning. When the UAV is flying straight both roll and pitch can be observed because of the gravitational force. This is not true for the heading, and the estimate drifts of. In Figure 3 we see that for some periods the heading estimate error is similar to that of roll and pitch. These periods correspond to when the UAV is maneuvering in Figure 1. It is therefore possible to predict the heading error when maneuvering compared to when moving straight or standing still.

To further assess the performance, the positional and velocity errors for the estimate are plotted along with the positional error for the GNSS measurements in figure 4.

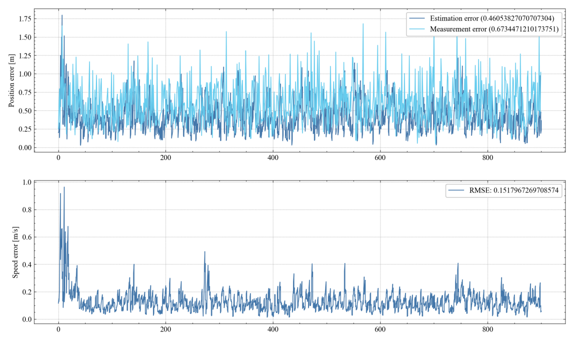

Figure 4: Showing measured and estimated position error compared to the ground truth, and velocity error for the estimated measurements. Achieved RMSE = 0.152 for the tuned parameters.

We note that for the positional RMSE is lower for the estimate (0.46m) than for the GNSS measurement (0.57m). The estimated speed RMSE is 0.15m/s. From these measures we see that the estimated trajectory is close to the true trajectory.

We proceed to look at the consistency of the filter. As we have access to ground truth, both NEES and NIS can be used to judge the filter consistency. The total NEES and NIS are plotted along with their confidence interval bounds in figure 5 [1].

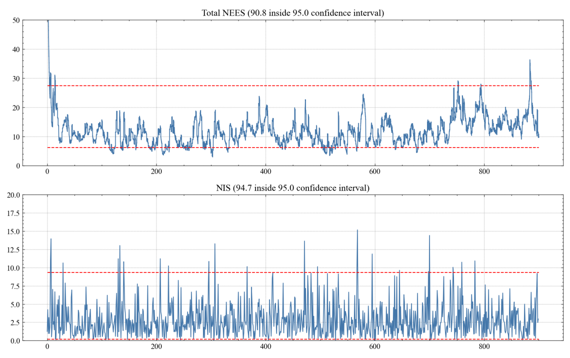

Figure 5: Total NEES and NIS with corresponding 95% confidential interval.

The resulting total ANEES was 11.86 with CI = $[14.96, 15.04]$, while ANIS was 2.86 with CI = $[2.98, 3.02]$: This means that the filter is a bit under-confident, and further tuning can be done to improve the filter. Both when tuning and validating the performance it can be useful to look at NEES for position, velocity, attitude, accelerometer and gyroscope separately. These are plotted in figure 6.

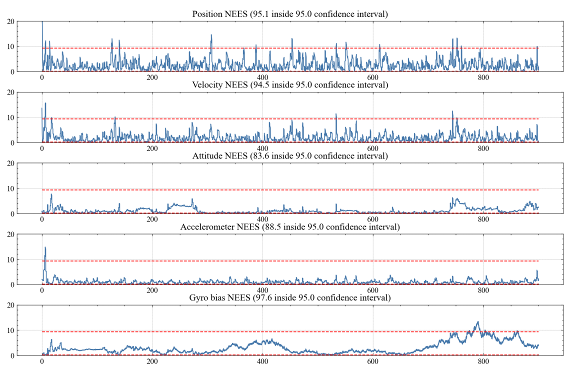

Figure 6: NEES value for position (95.1%), velocity (94.5%), attitude (83.6%), accelerometer (88.5%), gyro bias (97.6%). Most parameters are over 90% inside the confidence interval of 95% expect from Attitude and Accelerometer.

The filter yielded ANEES$_{pos}$ = 2.63, ANEES$_{vel}$ = 1.84, ANEES$_{att}$ = 1.08, ANEES$_{gyrobias}$ = 3.02 and ANEES$_{accbias}$ = 0.92, all with CI = $[2.98, 3.02]$. We note that all of the values are below the confidence interval except for gyroscope bias. This had to be tuned down at the cost of a lower overall ANEES. A value above the confidence interval means that the filter is too confident in the gyroscope measures, and in worst case it can lead to divergence in the filter estimate.
An interesting aspect when looking at the filter stability is to see what happens when remove the IMU misalignment corrections. That is done by setting the sensor correction matrices equal to the identity matrices. One critical consequence is that the gyroscope estimated bias diverges quickly, while the estimated accelerometer bias is fluctuating and not converging to a constant value. The biases in the measurements are thus not removed and used in the filter. The filter still maintains a track of the UAV, but that is mainly due to the GNSS measurements. RMSE value increases and the NEES values drop significantly. The state estimates are off, especially attitude, accelerometer and gyro bias. As these are essential for a stable filter, this emphasises the importance of misalignment corrections.

ESKF for the real data set

The real data set is from a real fixed wing UAV flight where a STIM300 IMU gives 250Hz measurements, and two Ublox-8 GNSS receivers gives 1Hz measurements [2]. In order to find initial values for tuning the data sheets for the sensors were used [3] [4]. The GNSS noise covariance was initially sat to a diagonal matrix the diagonal corresponding to the vector $[0.4^2, 0.4^2, 0.4^2]$, and scaled by a gain parameter equal to 0.255 and the time-varying accuracy of the GNSS sensor. For the IMU values, the biases and errors was tuned from the values given in the STIM300 datasheet [4]. The values obtained from this datasheet are given as Allan variance and is not compatible with our filter. They had to be recalculated to the correct unit. The accelerometer the noise (σ_a) is given as 0.06 m/s$\sqrt{h}$ and bias noise (σ_{ab}) as 0.05 mg. The gyroscope the noise (σ_ω) is given as 0.15 deg/$\sqrt{h}$ and bias noise($\sigma_{\omega b}$) 0.5 deg/h. The resulting parameters after tuning are presented in table 2. The estimated trajectory using the parameters in table 2 is shown in figure 7 along with the GNSS measurement. To easier visualize the trajectory the plot is shown in 2D.

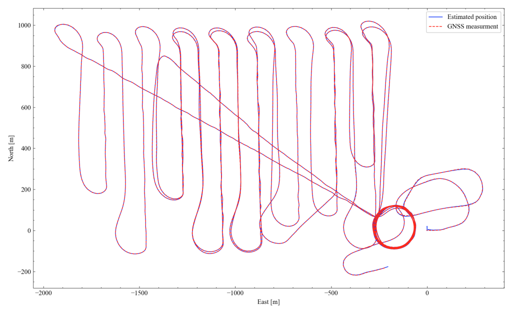

Figure 7: Estimated UAV trajectory with corresponding GNSS measurements in 2D.

Table 2: Accelerometer and gyroscope parameters for real data set

σ_a	σ_{ab}	σ_ω	$\sigma_{\omega b}$
$1x10^{-3}$	$4.9x10^{-4}$	$4.4x10^{-5}$	$1.6x10^{-4}$

From the figure 7 we see that the estimated trajectory and the GNSS behave approximately the same, but for more information it is helpful to look at the estimated states. The estimated states are presented in figure 8. Note that nothing is happening with the UAV for until 205-210 seconds (ca. 50000 steps). This is because the UAV is not launched until this point.

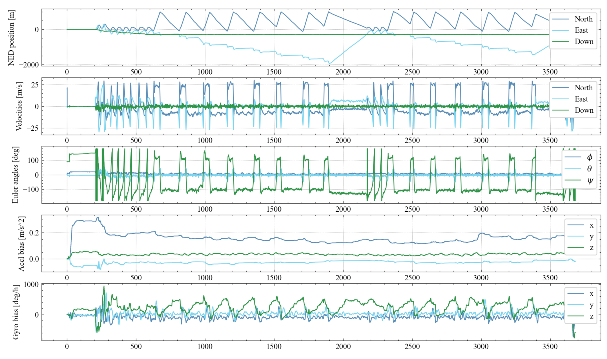

Figure 8: Estimated UAV states for the real data set.

When comparing figure 8 and figure 3, it is not surprising to see that transition between states are more harsh in the real case than for the simulated case. This becomes evident when looking at the bias estimate. In the simulated case the the Gyro bias converges over time, but for the real case it has problem converging and it has corrections throughout the flight, especially at the turns. For the accelerometer it has much larger bias values, especially x and y direction. This is not strange since it for the most part stays on the same altitude. When plotting the Euler angles we see that it changes between two almost opposite values. This is not strange because, as we see in Figure 7, the UAV flies in a pattern going back and forth between north and south. This indicate that the bias estimates work as intended, as the right heading is given and general state estimates seems to be correct. To evaluate the performance even further total NIS is presented in figure 9, with a value of 80.4% within the confidential interval.

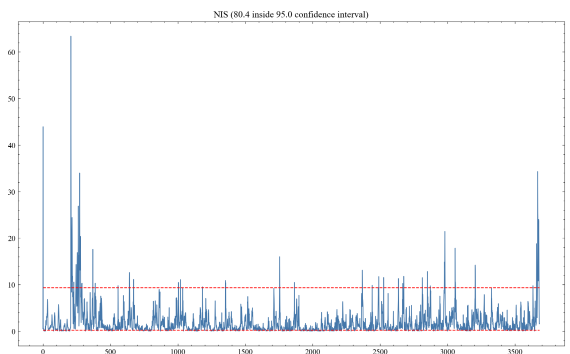

Figure 9: Total NIS value for the estimated UAV with corresponding 95% confidential interval.

The average NIS was approximately 1.89, Which was below the confidential interval. Comparing this performance with the simulated data set shows that, as expected, it's more difficult to tune an ESKF for a real life problem than for simulated one. This is because for the real life case we are lacking the ground truth, and we thus can't calculate the estimated error and NEES. This makes tuning harder. It is however manageable, and further tuning should be done to improve the filter.
As for the simulated data set, neglecting the misalignment correction was done for the real data. Like for the simulated data the filter still maintains a track of position and speed due to the GNSS measurements. As expected, the bias values converge to other values due to the different starting points. An important observation is that the heading estimate is the opposite of the true heading, which makes it easy to see that something is wrong. Correcting this should however be achievable in a real world application. If the heading estimate says that the UAV is traveling north while it in fact is traveling south, misalignment should be accounted for.

Conclusion

In this report we have evaluated how an ESKF performs in combining data from an IMU and GNSS. The IMU measurements are used as control inputs for prediction and GNSS as complementary measurements for the update of the estimation. In addition to treating the IMU as the control input we also estimate the IMU biases. This makes the ESKF more robust, if tuned for bias estimate convergence, since it minimises the error propagating in the system. We have observed that the filter is not perfect, and in cases where heading estimates are important, one might want to include a compass or other sensors in the filter. By neglecting misalignment corrections we saw its important in order to obtain proper estimates, as the bias estimates might not converge or the attitude estimates can turn out wrong. We have also seen that the ESKF is sensitive to different initialisation, and poor attitude initial values can affect the estimates for a long time and keep the biases from converging. To show the benefits and limitations of the ESKF it was first implemented on a simulated data, before it was then tuned and tested on a real data set. As ground truth was available on the simulated data set, the ESKF could be tuned to yield good performance estimation. This is because NEES and RMSE gives good indications on the tuning. In contrast, only measurements with corresponding uncertainty are available in the real data set. Thus, tuning is more complicated, and results are generally worse. The report did however show that an ESKF using IMU measurements can keep a track of a real UAV flight and be reasonably stable. This shows that the filter assumption of linear error state dynamics is sufficient in real world cases.

References

[1] E. Brekke: *Fundamentals of Sensor Fusion*

[2] TTK4250 Sensor Fusion: *Graded Assignment 2*

[3] GNSS datasheet: *u-blox 8 / u-blox M8 Receiver description*

[4] IMU datasheet: *STIM300 Inertia Measurement Unit*