Department of Engineering Cybernetics

Sensor fusion

Graded assignment 2

Group:
Sensor fusion Group 65

Authors:
Eivind Heldal Stray (eivindhs)
Magnus Dyre-Moe (magnudyr)

October 30, 2020

1 Introduction

The error state Kalman filter (ESKF) uses IMU measurements as input to the process model and it updates predictions with other measurements, for example GNSS. For inertial navigation we can not use a regular KF or EKF because of attitude. Hence, the motivation to use an ESKF. However, the ESKF inherits the same weaknesses as the EKF due to the structure of the model. More specifically, the necessary linearizations needed to run the filter. ESKF also comes with some additional weaknesses, namely, sensitivity to initialization, lever arm compensation and IMU misalignment. Lever arm compensation is handled through the code, while initialization and IMU misalignement will be discussed throughout this paper.

The error state Kalman filter is assumed to be known and can be found in the course book in chapter 10.

1.1 Intuitive tuning strategy

Throughout this paper an assumption is that the STIM3000 is the IMU of choice for both the simulated and real data. The values for continuous bias standard deviation is given as $\sigma_{gyro} = 0.15°/s = 4.36{\cdot}10^{-5}\frac{rad}{s}$ and $\sigma_{acc} = 0.06\frac{m}{s^2} = 1{\cdot}10^{-5}m/s^2$. These will be discretized in the ESKF algorithm for each time step, and thus continuous values will suffice[1].

On the simulated data the values for $\mathbf{R}_{GNSS}$ were used, giving good results. Calculating the variance between ground truth and the measurements however, gave even better results and were a bit lower than the one handed out. The values for $\mathbf{R}_{GNSS}$ in the real data set were initially copied from simulated data but finally used as the accuracy estimates provided. The GNSS is relatively precise, and one should expect quite accurate measurements although altitude usually has a higher variance than the planar coordinates measured. This seems to be true for our data sets.

A general strategy has been to lower $\mathbf{R}_{GNSS}$ to increase the low trend of NIS and NEESes while still keeping it close to what is known to be close to true values. Furthermore, $\mathbf{R}_{GNSS}$ has been assumed to be a diagonal matrix, with no elements on the off-diagonal.

The final tuned values ended up as follows:

Simulated data	σ_{acc} 4.36*10^{-5}	σ_{gyro} 1*10^{-3}	p_{acc} 0	p_{gyro} 0	R_{GNSS} diag([0.09 , 0.09 , 0.27])
Real data	σ_{acc} 16*4.36*10^{-5}	σ_{gyro} 16*10^{-3}	p_{acc} 10^{-16}	p_{gyro} 10^{-16}	R_{GNSS} accuracy*diag([0.3, 0.3 , 0.4]) [2]

As a simplification to make the computational complexity less extensive, a second order Taylor approximation has been made for exponential matrix computations.

2 Run ESKF on simulated data x

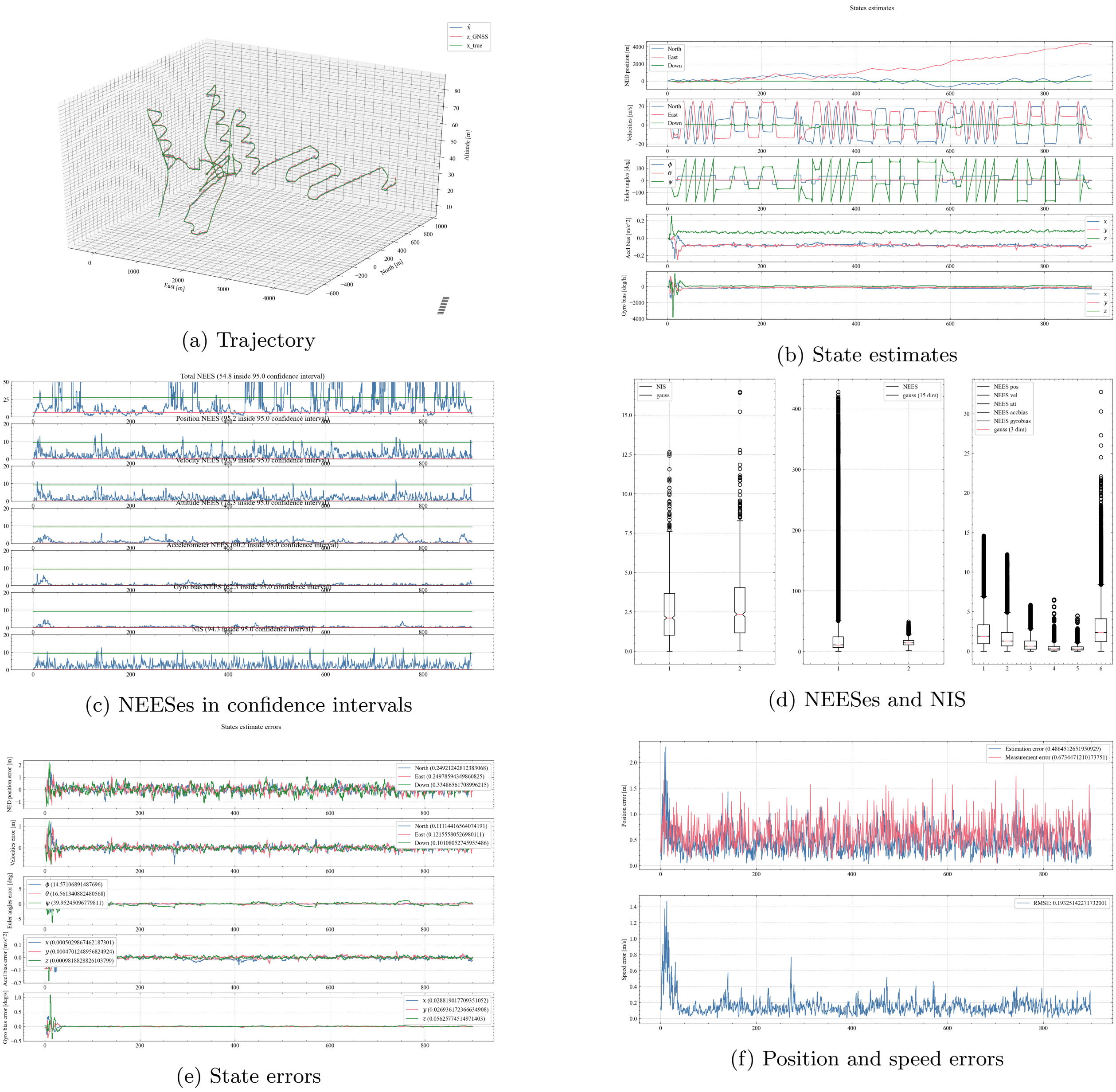
2.1 Problem a)

As observed in figure 1e a good initialization will yield an accurate filter, with a maximum error in position at 2 meters. Unsurprisingly, the largest error is at the start of flight. This makes sense as the uncertainties will decrease as time progresses when the filter manages to correct for errors. On the other hand, a bad initialization would result in far greater errors over a bigger time interval.

In general, regardless of initial bias, the trajectory will remain close to the true trajectory as long as initialization for position, velocity and attitude remain close to the true trajectory. This comes as no surprise as the error in bias will be corrected. However, it will greatly impact NEES and NIS. This is as expected as the bias will change rapidly in the beginning, making NEES and NIS changing rapidly.

Tuning initial position, velocity and attitude was done through knowledge about the data set. The starting position of the UAV is at zero velocity with a position and attitude. Hence low covariance was used for all three, however, not zero. By avoiding zeros singular covariance matrices and unwanted uncertainties transferred between states are avoided. For attitude it is also important to use a low covariance, as a large uncertainly in attitude may lead to trouble when following the desired trajectory. This is due to the heading not being observable for low velocities. This is further discussed in section 2.2.

[1] Real data ran nicely with initial bias values, and these values were therefore kept
[2] Accuracy for GNSS given in the dataset as accuracy_GNSS[i]



(a) Trajectory

(b) State estimates

(c) NEESes in confidence intervals

(d) NEESes and NIS

(e) State errors

(f) Position and speed errors

Figure 1: Plots from simulated data

From figure 1c and 1d a larger NEES in acceleration bias is presented, compared to the other NEESes. This indicates more measurements noise which can be solved through extensive tuning. However, this is shown it as is, as it may indicate potential problems when the IMU is misaligned in section 2.3.

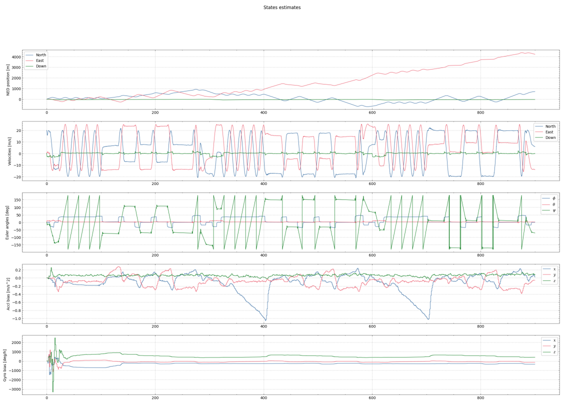
One important concept of the Kalman filter is observability, which refers to the ability to observe the different states in the system. Hence, why GNSS measurements are valuable. With measurements in position we are able to observe velocity and acceleration. The opposite is not necessarily true. Hence, with GNSS measurements, the UAV is able to correct its position, also altering acceleration, velocity and attitude in the process.

2.2 Problem b)

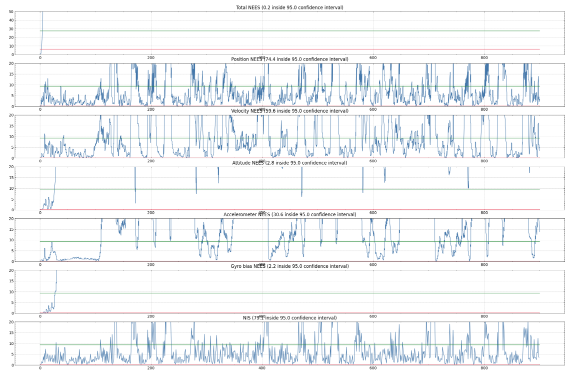
Heading is observable as long as the generalized speed is larger than zero. When the velocity is greater than zero we can calculate heading by geometric approximations. In NED, this approximation is $tan^{-1}(\frac{\Delta East}{\Delta North})$, more specifically, $\psi[k] = tan^{-1}(\frac{y[k]-y[k-1]}{x[k]-x[k-1]})$, where y and x are positions in NED. This may not yield a perfect result as position is a discrete variable prone to noise. However, for low speed, we know the result to be far from perfect as the position are close to each other, making the measurement significantly more prone to measurement noise. For a vehicle standing still, heading will not be defined as it results in tangens inverse of infinity. Velocity is making the heading observable. A way of improving this problem with heading observability can be to install a compass.

This can be observed in figure 1e as the Euler angles tend to fluctuate more at the start of the simulation, before the UAV picks up speed.

2.3 Problem c)



(a) State estimates with misalignment



(b) NEES with misalignment

Figure 2: State estimates and NEES from simulated data when $S_a = S_g = I_{3x3}$

Setting the S_g and S_a matrices to identity implies that the IMU is placed perfectly with the body axis and that there are no systematic error. This is, however, rarely the case since it needs to be placed in a practical manner. Effectively, another bias introduces itself when the IMU is not perfectly aligned in body, which is perhaps the most important factor. Orthogonality will most likely be close to achieved with little need for correction. This will, however, in the case of an imperfect IMU cause a slight drift in predictions based on the measurements. Scaling errors will not be possible to eliminate completely (we have factors such as temperature change, humidity, wear and tear), but presumably quite well by compensating with the S-matrices. IMUs are not sold with perfect scaling, and thus some tuning of the device through such a matrix is in order in addition to a rotation to the body frame.

When testing we observed an increase in bias from the accelerometer and gyroscope. As observed in figure 2a, the acceleration bias changed rapidly during excessive turning. Naturally, this made for an increase in bias error. Furthermore, we observed an overall increase in position and velocity error, as well as RMSE. This comes as no surprise, as it is a direct result of increased bias. Naturally, the NEES was also worse. Overall, the NEES values increased, indicating a less conservative filter. Never the less, the trajectory managed to stay close to the ground truth with a maximum error in position at roughly 4 meters. For this particular case the performance took a slight dip. However, the performance was still acceptable as the trajectory remained close to the ground truth. Generally, the S_a and S_g matrices should be accounted for in real life as it is a simple and cheap way of making the IMU measurements more reliable even if the device is mounted upside-down or with some other obscure orientation.

To conclude with, due to the misalignment that will most likely occur, this S-matrix will in most cases have to be included to get sensible measurements. One could argue that there are cases where the IMU is mounted up to 90° for a better fit in the reduced space available in a vehicle such as the drone.

3 Run ESKF on real data

3.1 Problem a)

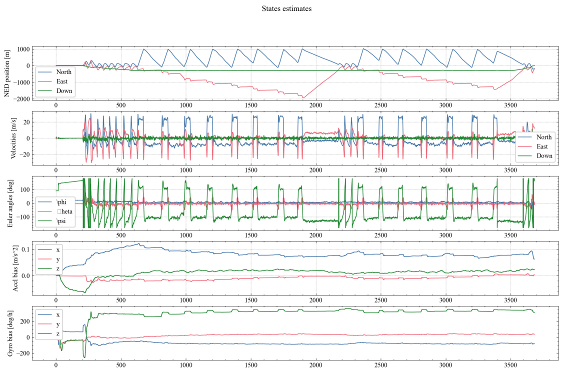
The physical drone in problem 4 seems to be quite similar to the drone simulated in problem 3. The variables from problem 3 allowed for a quite smooth trajectory, although NIS values were off quite badly with the filter being conservative. Letting R_{GNSS} based on each assumed accuracy squared downscaled quite aggressively gave better results.

Notice in 3b how NIS increases to a very high value after roughly 200 seconds and at the end. This is when the drone is being launched and landed, providing a violent acceleration that the model cannot keep up with in a precise manner. Therefore, the innovation norm at this point in time will be very high regardless of how one chooses to tune the parameters.

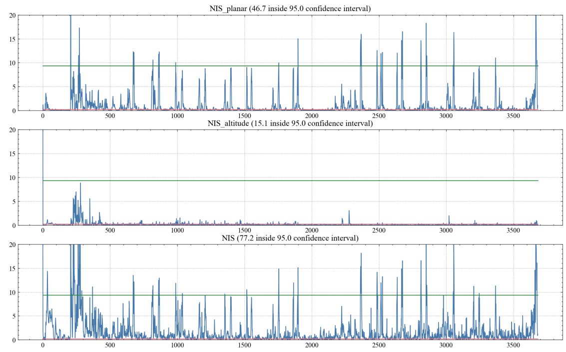
Another thing worth mentioning is the generally low tendency of NIS in 3b. Since the GNSS measurements are quite precise, little correction will be done as the predicted position will end up close to that of the GNSS if the drone is following a straight line. Although a low NIS can be associated with underfitting, one may argue that this effect comes from a straight flightpath close to GNSS measurements although the filter is clearly conservative.

Generally, the spikes in NIS in figure 3b are related to turning the UAV. During these phases of flight our linearlized model is a bad fit as the angles differ greatly from level flight and nonlinearities surface as errors. These errors are modelled as bias, and correction from measurements is resulting in the visible spikes. This is a problem for the ESKF and Markov models in general. Observe how the NIS is low for the straight paths, indicating low innovation. This stands to reason, as the linearlization is a better approximation for small angles.

When tuning the initial values for position, velocity and attitude, we used the same strategy as in 2. More specifically, this was assuming initial position, velocity and attitude to be known with a relatively small uncertainty modelled in the covariance matrices.



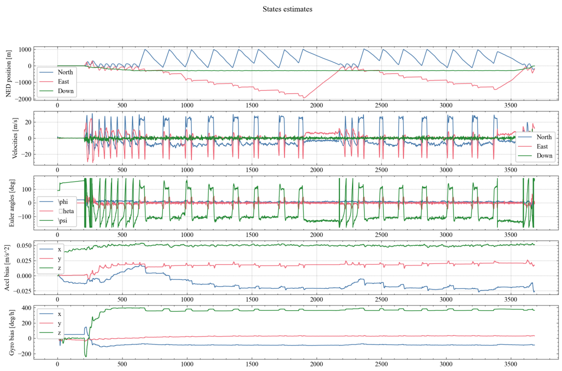
(a) State estimates



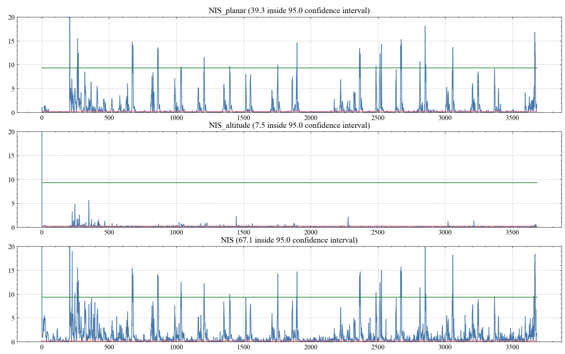
(b) Planar, altitude and total NIS

Figure 3: Plots from real data

3.2 Problem b)



(a) State estimates



(b) Planar, altitude and total NIS

Figure 4: State estimates and NIS from real data when only compensating for known mounting orientation – neglecting mounting, scaling and orthogonality errors.

Running on real data only compensating for mounting orientation yields quite similar results as with scaling and orthogonality errors accounted for using S_g and S_a. This tells us that mounting, scaling and orthogonality errors does not play a major role for the real data. However, disregarding mounting orientation may lead to a similar response to the one observed in 2.3. Connecting this to real world applications it is apparent that mounting orientation is of great importance. For this specific case; mounting orientation, scaling and orthogonality errors did not impact the result as much. However, these errors may be troublesome for other cases.

4 Conclusion and reflection on the results

With the simulated data, many things were known. Sensible values for $\mathbf{R}_{GNSS}$ and bias were handed out, and using this as a foundation for tuning as well as an argumentation about why these values were good yielded quite satisfying results. These values could be applied to the real data set and not much tuning was needed. In general, the two filters are conservative, with a NIS of 94.3% within the 95% confidence interval for simulated data and 77.2% within the 95% confidence interval for the real data. Furthermore, the trajectories in both cases look realistic and the RMSE error in the simulated data is low with errors in estimated position, measured position and velocity at $0.49m$, $0.67m$ and $0.19m/s$ respectively.