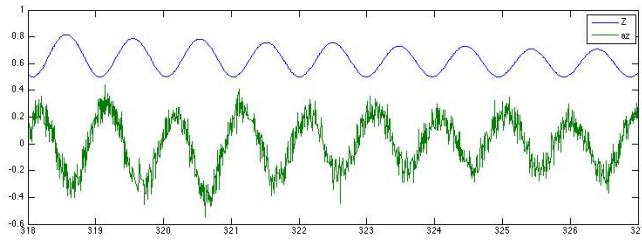


Semester Thesis

Sensor Fusion / State Estimation for a Kite Power Plant



Authors:

Roman Patscheider
Fabian Hilti

Advisers:

Aldo Zgraggen
Corey Houle

Contents

Abstract	v
1 Introduction	1
1.1 Kite Power in General and the SwissKitePower Project	1
1.2 Motivation and Methods	2
1.3 Related Work	2
2 IMUs	4
2.1 Centrifuge Test	5
2.1.1 Set-up	5
2.1.2 Result	8
2.1.3 Conclusion	16
3 State Estimation	17
3.1 Extended Kalman Filter	17
3.2 Algorithm for State Estimation	20
3.3 State Estimation Model	26
3.4 Sensor Estimation Model	29
4 Results - Vicon Test	31
4.1 Set up	31
4.2 Evaluation	33
4.2.1 "Pendulum"-estimator vs. "free mass model"-estimator	33
4.2.2 Noise of the GPS	40
4.2.3 GPS outage	45
4.2.4 The MTi-G unit versus the PX4 unit	48
5 Conclusion and Outlook	50
5.1 Sensor Selection	50
5.2 State Estimation	50
A Data Sheets of the PX4 sensors	54
A.1 Data Sheet of the Accelerometer	54
A.2 Data Sheet of the Gyroscope	56

A.3 Data Sheet of the Magnetometer	58
B Data Sheets of the MTi-G	60
C Data Sheets of the x-IMU sensors	67
C.1 Data Sheet of the Gyroscope	67
C.2 Data Sheet of the Accelerometer and Magnetometer	69

List of Figures

2.1 Picture of the centrifuge.	6
2.2 Picture of the box with the IMUs inside.	7
2.3 GPS Position	10
2.4 GPS velocity	11
2.5 Accelerometer	12
2.6 Gyrometer	13
2.7 Magnetometer	14
2.8 A closer look in time on the three IMU sensors.	15
3.1 A simple structure of the algorithm [1].	18
3.2 The prediction and the correction step with their corresponding equations [1].	20
3.3 The prediction and correction step of the EKF with their corresponding equations [1].	21
3.4 The definition of the inertial and body frame as well as the convention for the angles in the pendulum model.	22
3.5 The graphical representation of the algorithm's structure.	25
3.6 The black dot represents the sensor, while r is the distance between the center of mass of the pendulum and the sensor.	30
4.1 Pendulum set up.	32
4.2 Box with sensors.	32
4.3 Estimation of the position using the "Pendulum"-estimator compared to the ground truth obtained from the Vicon system.	35
4.4 Estimation of the orientation using the "Pendulum"-estimator compared to the ground truth obtained from the Vicon system.	36
4.5 Estimation of the position using the "free mass model"-estimator compared to the ground truth obtained from the Vicon system.	37
4.6 Estimation of the orientation using the "free mass model"-estimator compared to the ground truth obtained from the Vicon system.	38
4.7 Error in position and orientation for the "free mass model"-estimator.	39
4.8 GPS signal with a SNR of -5 dB.	40
4.9 GPS signal with a SNR of -25 dB.	42

4.10	Error in position and orientation using a GPS signal with a SNR of -5 dB.	43
4.11	Error in position and orientation using a GPS signal with a SNR of -25 dB.	44
4.12	Estimation of the states without GPS signal.	45
4.13	Error in position and orientation only using IMU measurements and no GPS data.	46
4.14	"Free mass model" estimator during a GPS outage from $t = 75\text{ s}$ to $t = 78\text{ s}$.	47
4.15	Error of the estimation of the states without GPS using only the PX4 IMU.	48
4.16	All mean errors. Where fmm stands for free mass model which is used only in the first error calculation. All errors were calculated using the MTi-G IMU. Only the last one is using the PX4 Unit.	49

Abstract

For a successful automated control of a kite, a precise estimation of its position and orientation at all times using GPS and Inertial Measurement Unit (IMU) measurements is essential. In this work we investigated the use of an Extended Kalman Filter that makes use of the information available for the system. To this purpose we developed an algorithm that estimates the state of a spherical pendulum. Using a physical model of this system we were able to reduce the error in the position estimation by 43

Chapter 1

Introduction

1.1 Kite Power in General and the SwissKitePower Project

At a time when windmills were already quite commonly used for power generation, Miles Loyd came up with the idea to use kites to convert wind energy to electricity. In 1980 he wrote a seminal paper exploring the possibility of generating electrical power using the pulling force of tethered airfoils, i.e., kites [2]. He describes his concept as follows:

A kite's aerodynamic surface converts wind energy into motion of the kite. This motion may be converted into useful power by driving turbines on the kite or by pulling a load on the ground. [...] Not simply facing into the wind, such kites would fly a closed path downwind from the tether point. The kite's motion would be approximately transverse to the wind, in the same sense that a wind turbine's blade moves transverse to the wind. The crosswind airspeed of a kite with this trajectory is increased above the wind speed by the lift-to-drag ratio L/D. The resultant aerodynamic lift is sufficient to support a kite and to generate power. [2]

Today, several research groups around the world are investigating this subject and working on prototypes. For example the University of Torino have already tested a prototype [3] and Massimo Ippolito has founded a company named KiteGen that is also located in Torino. At the University of Delft the group of Prof. Dr. Wubbo Oeckels is developing kites to produce energy with "Laddermills" [4] and at the K. U. Leuven the group of Prof. Moritz Diehl is working on the Highwind project [5]. Furthermore the SkySails company is already using wind power to pull large cargo ships [6] and Ampyx Power, a spin-off from T.U.Delft is working on their Power-Plane device. These European research efforts are all pursuing the approach where power is generated on the ground. In the United States however, the more popular strategy is to generate the electricity up in the air. Two examples of companies trying to establish this technology are Makani Power [?] and Joby Energy [?].

In Switzerland the research efforts are coordinated in the SwissKitePower Project [7]. SwissKitePower is a collaborative research and development project between FHNW (University of Applied Sciences Northwestern Switzerland), ETHZ (Swiss Federal Institute of Technology Zurich), EMPA (Swiss Federal Laboratories for Materials Testing and Research) and Alstom Switzerland AG. The goals of the project are to develop 'novel wind energy extraction technology' using tethered airfoils, or kites, and to promote this innovative new technology to the world.

1.2 Motivation and Methods

To successfully implement an automated control algorithm on a kite, it is essential that a precise and fast position estimation is available. Due to the slow update rate of the GPS units and their limited reliability, a Kalman Filter will be used to integrate IMU measurements, which consist of acceleration, rate of turn and magnetic field, with the GPS position and velocity estimations. A standard Kalman Filter, one that could also be used in an airplane for example, assumes no knowledge about forces acting on the body. Such an estimator will already enhance the position estimation from the raw GPS input since it incorporates more measurements. However, a kite can only move on a restricted surface due to it being tethered to the groundstation. An extended Kalman Filter could take advantage of that knowledge and further improve the estimator's performance. Further integrating an aerodynamical model of the kite could give more information about the forces acting on it and would reduce the estimator's dependency on precise sensor output.

In order to be able to compare the performance of the different estimators, it is necessary to know the exact location and orientation of the kite at all times. In practice it is nearly impossible to obtain such a ground truth for a kite flying around in the air. Therefore we decided to investigate the benefits of an accurate physical model on the performance of a Kalman Filter using a box suspended by a string. (See figure ...) This setup can be modeled as a spherical pendulum which has the advantage that all the forces acting on it are known. In addition, we are able to test the algorithms indoors using a Vicon System (add reference!!) that gives us a very accurate ground truth of the box's position and orientation. Since there is obviously no GPS reception indoors, the GPS output needs to be simulated using the Vicon's position output, reducing its frequency and adding an appropriate level of noise.

1.3 Related Work

There are many publications on the topic of IMU/GPS data fusion for navigation purposes. As a starting point of our work we used the Master Thesis of András Héjj "Kalman-filter based position and attitude estimation algorithms for an Inertial Measurement Unit" [8]. In an other Master project at the University of Southern Denmark Ushanthan Jeyabalan implemented a Kalman Filter using a spherical

pendulum model [9]. Several other publications investigated the use of a Kalman Filter for IMU/GPS data fusion in different applications. Sukkarieh applied it to land vehicles [10], Kim used it for unmanned areal vehicles in highly dynamic flight situations [11] and Crassidis used a Sigma Point Kalman Filter and compared it to an EKF. This list of course is not exhaustive. A much broader overview over the various approaches to multisensor data fusion in target tracking can be found in the survey article of Smith and Singh [12].

Chapter 2

IMUs

For estimating the kite's state measurements from different sensors are needed. An inertial measurement unit (IMU) contains several sensors that measure physical quantities like acceleration, rate of turn, a magnetic field or ambient air pressure. There are many different such IMU's available, which may be distinguished by which sensors are embedded, at what rate they provide the data, what is their range and sensitivity and how much signal processing is done internally.

The Swiss Kite Power Project has the following options of IMUs to choose:

MTi-G The MTi-G development kit is a commercial product from the Dutch company Xsens. The unit has an accelerometer, gyroscope, magnetometer, pressure sensor and GPS with antenna included. It exist two output data formats. It can be chosen wether the output is raw data or calibrated data. The calibrated data from the sensors without GPS gives the data in common units and uses a calibration to compensate offsets. In addition, the GPS and IMU data can be processed with an Extended Kalman Filter resulting in a filtered output of position and attitude.

PX4 The PX4 is a open-source/open-hardware IMU used and developed by the PIXHAWK Project of the Computer Vision and Geometry Lab of ETH Zurich. The unit contains a temperature and pressure sensor, an accelerometer, a gyroscope and a magnetometer. The IMU additionally provides a counter for each sensor to ensure an easy handling of the different output frequencies of the included sensors. The output of the sensors is raw data. In the estimation algorithm (see chapter 3) only the embedded accelerometer a BMA180 from Bosch (see the data sheet in appendix A.1), the gyroscope a L3GD20 from STMicroelectronics (see the data sheet in appendix A.2) and the magnetometer a HMC5883L from Honeywell(see the data sheet in appendix A.3) are used.

x-IMU The x-IMU is a product of the company x-io Technologies. It also has a temperature sensor, an accelerometer, a gyroscope and a magnetometer. The

configuration of the x-IMU was done by the FHNW. It provides only raw data. The data sheets of the sensors can be found in the appendix C.

2.1 Centrifuge Test

There are several performance requirements on an IMU installed on a kite. One of them is the dependency on acceleration because a kite experience acceleration of multiple times g. Therefore a centrifuge test was carried out in corporation with the Fachhochschule Nordwestschweiz (FHNW)

2.1.1 Set-up

The setting of the centrifuge is described in figure 2.1. It was provided by the FHNW. An arm is rotated by a motor. The IMUs are mounted in a box at the end of the arm. The sensors are put next to each other to have similar measurements in all sensors. A psicture of the Box can be seen in figure 2.2. Obviously, the motion of the box is describing a circle with a radius (the distance between the rotation axis and the center of the box) of 0.75 m. The motor is able to rotate the arm with a velocity of about 10 m/s. Which corresponds to a maximum acceleration of about $a = \omega * r^2 = 8 g$.

Even though a sensor on the centrifuge was measuring the rate of turn, there was no software for the data collection implemented. Therefore we have no ground truth to compare the sensor data with. During the test, the motor was driven in order to have 5 steps between 1.62 g and 7.8 g (ca. 1.6 g; 2.4 g 3.7 g; 5.4 g; 6.9 g; 7.8 g) of centripetal acceleration.

The PX4 and MTi-G are connected together. Both, the PX4 and the MTi-G measurements are written on the SD card of the PX4 Unit. The timestamps for the sensor readings are taken from the GPS clock for the MTi-G as well as for the PX4 at the time when they are logged on the SD card. This results in easy and accurate synchronization between the two IMUs. To get the time synchronized with the third IMU, the x-IMU is synchronized to the computer's time. Additionally the 3 sensors are hit for having a clearly distinguishable signal aiding in the estimation of how accurate the synchronization is.

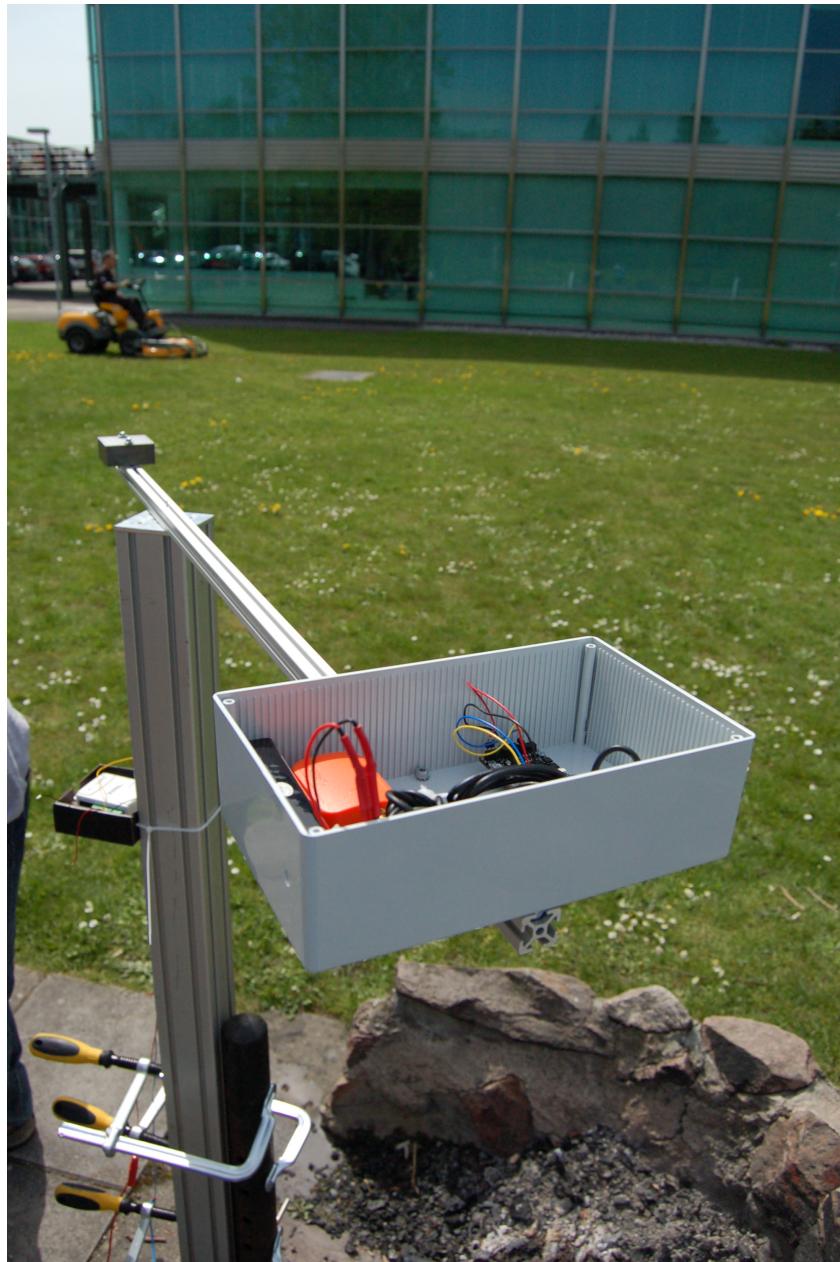


Figure 2.1: Picture of the centrifuge.

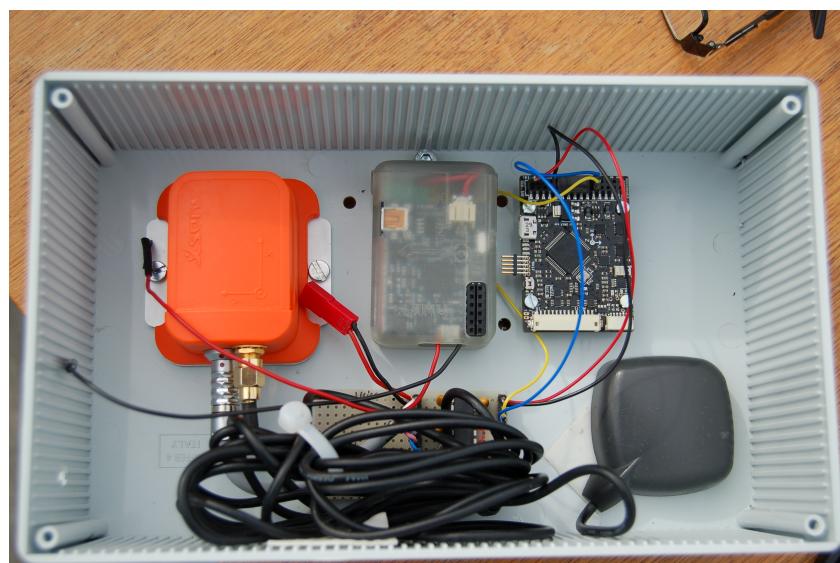


Figure 2.2: Picture of the box with the IMUs inside.

2.1.2 Result

The raw data from the x-IMU is scaled by Raphael Mueller bringing it to the common units m/s^2 for the accelerometer, deg/s for the gyroscope and $gauss$ for the magnetometer.

The PX4 was set by the group of the PixHawk Project. How the output of the sensors have to be scaled to bring them in the required units is shown in table 2.1. The scale factor describes by how much the output has to be multiplied to get the data converted in the units described in 4th row. The MTi-G has also raw output

Sensor	Sensitivity	Scale Factor	Unit	Comment
Accelerometer BMA180	2048 LSB/g	$output * 9.81/2048$	m/s^2	Sensitivity range: $+/- 4 g$
Gyroscope L3GD20	17.5 $mdps/digit$	$output * 17.5/1000$	deg/s	Sensitivity range $+/- 500 dps$
Magnetometer HMC5883L	1370 LSB/g	$output/1370$	gauss	Sensitivity range $+/- 8 gauss$

Table 2.1: This table shows how the raw data output of the PX4 has to be scaled to get the right units.

data but no clear explanation in the data sheet in appendix B on how to get to the required units. Therefore, were the MTi-G data scaled to bring it more or less to the same order as the other two IMUs are. In figure 2.3 we can see the plot of the GPS data. Until 4.83×10^4 s is the centrifuge not in motion. The mean value of this period is taken as the ground truth to calculate the average error of the GPS signal. The mean value of the north component is 47.48 deg for the east component 8.2138 deg and for the down component 415.74 m. The coordinates of Winidsch, where the test takes place is 47.4806 deg, 8.2222 deg and 357 m. By taking the average absolute difference between the GPS data and the mean value calculated before, an error of the GPS can be estimated: 1.08×10^{-5} deg, 1.37×10^{-5} deg and 2.49 m for north, east, down respectively. After 4.83×10^4 s the noise is increasing. At this time the centrifuge starts rotating. The calculated errors in this period are: 1.68×10^{-5} deg, 2.27×10^{-5} deg and 2.89 m for north east and down respectively. After 4.87×10^4 s the noise looks even higher. This is when we have a centripetal acceleration of more than 6.7 g. The estimated noise is: 1.04×10^{-4} deg, 1.02×10^{-4} deg and 13.01 m. Since in none of this periods a cosine of sine behavior of the north and east component can be seen, all three components should be stable in the ideal case. The error values are summarized in table 2.2. Due to the fact that the velocity is the derivative of the position of the GPS the amplitude of the error behaves similar. The periods with the correspondent average errors are shown in table 2.3. In figure 2.5 and figure 2.6 the accelerometer and gyroscope measurements are shown. As expected, we do have the highest acceleration in x direction and the fastest rotational rate in z direction. In the accelerometer x axis and the gyroscope z axis measurements can be seen the 5 steps of different rotational speed. The very first step is a test rotation

Period	Average error (north/east/down)
$4.80 * 10^4 \text{ s} - 4.83 * 10^4 \text{ s}$	$1.08 * 10^{-5} \text{ deg}/1.37 * 10^{-5} \text{ deg}/2.49 \text{ m}$
$4.83 * 10^4 \text{ s}4.87 * 10^4 \text{ s}$	$1.68 * 10^{-5} \text{ deg}/2.27 * 10^{-5} \text{ deg}/2.89 \text{ m}$
$4.87 * 10^4 \text{ s}4.91 * 10^4 \text{ s}$	$1.04 * 10^{-4} \text{ deg}/1.02 * 10^{-4} \text{ deg}/13.01 \text{ m}$

Table 2.2: The average error of the position in different time segments

Period	Average error(x/y/z)
$4.80e4 \text{ s} - 4.83e4 \text{ s}$	$5.54 \text{ m}/6.05 \text{ m}/6.57 \text{ m}$
$4.83e4 \text{ s}4.87e4 \text{ s}$	$101.86 \text{ m}/138.96 \text{ m}/27.36 \text{ m}$
$4.87e4 \text{ s}4.91e4 \text{ s}$	$157.20 \text{ m}/123.56 \text{ m}/58.78 \text{ m}$

Table 2.3: The average error of the velocity in different time segments

followed by the hit on the sensors which react with a peak. Since the sensitivity of the PX4 accelerometer is set to $+/- 4 \text{ g}$, the output of accelerometer in x direction is limited to 39.24 m/s^2 . The MTi-G Unit is as well limited to $+/- 5 \text{ g}$. This is not obviously seen in the plots. The acceleration in y direction is showing small steps which can be accounted for by the placement of the sensors slightly off the center of the box. The gyroscope from the MTi-G shows a limitation in measuring the rate of turn (see figure 2.6. It is unable to resolve more than 403 deg/s . By the choice of the PX4's gyroscope setting the range is limited to $+/- 500 \text{ dps}$. But on the plot this limitation is not seen up to 520 dps . The x-axis and y-axis should be zero for the gyroscope since the rotation is only two dimensional. Obviously, the noise again increases with higher rates of rotation. This can be clearly seen in the acceleration in z direction which should be constant at -9.81 m/s , the gravitation. It looks like the noise level of the PX4 is the highest before the x-IMU and the MTI-G which has the lowest. This can be traced back on the high sensitivity set up of the PX4 accelerometer.

In figure 2.7 the magnetometer output is plotted. There is a misalignment between the x-IMU,MTi-G and the PX4 which cannot be explained by the authors. The range of the maximum value and the minimum value of the PX4, MTi-G and the x-IMU is in x and y direction 0.4 gauss . This is the expected range in the region of Zurich. In this sense, if the the output is added with an offset in order to get the range to $+/- 0.2 \text{ gauss}$, the measurements should be correct. The PX4 has a mean value of 0.3893 gauss while the ground truth is 0.4268 gauss . The noise of the magnetometer is in the x-IMU higher than in the other two IMUs.The PX4 unit has limited range of $+/- 8 \text{ guass}$ what not appear in this test and therefor

In figure 2.8 a closer look on the x axis of the magnetometer and accelerometer and on the y axis of the gyroscope is presented. The form of the curve has sine behavior, what we would expect again.

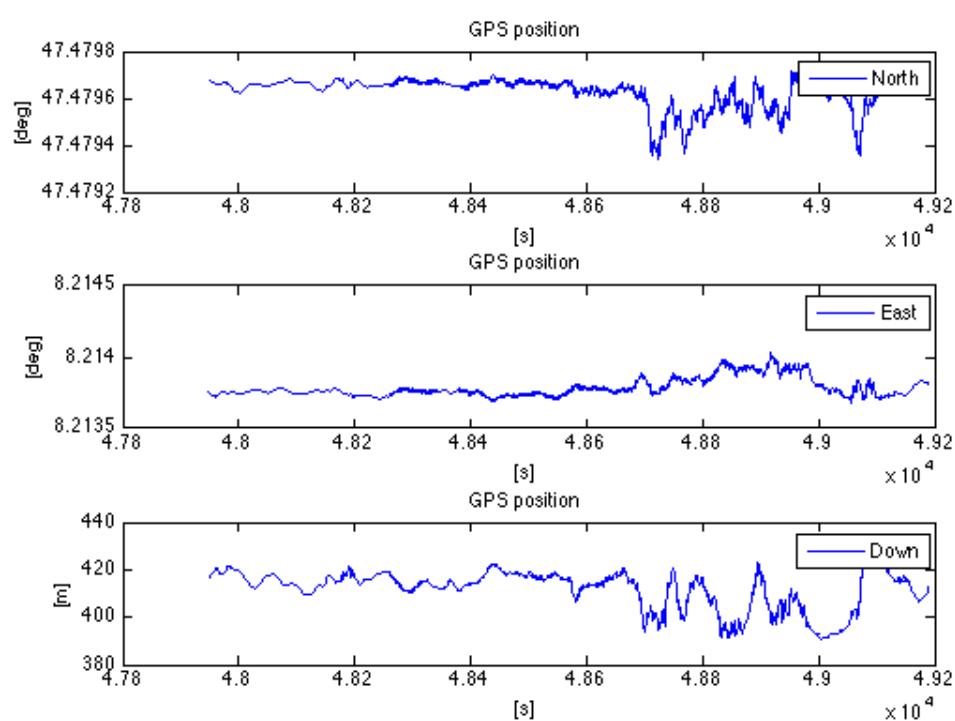


Figure 2.3: GPS Position

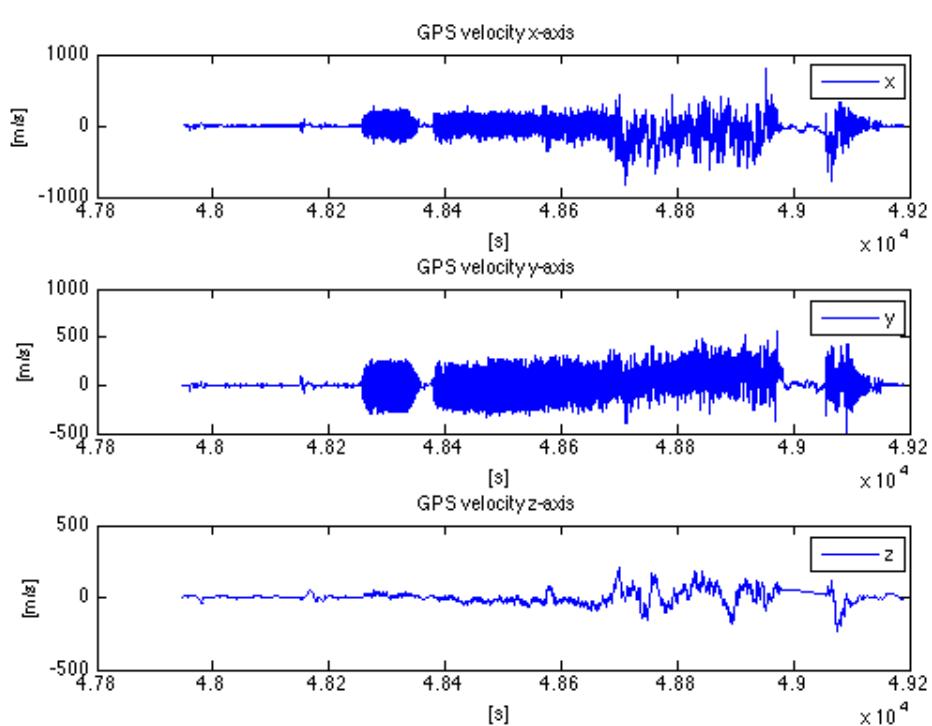


Figure 2.4: GPS velocity

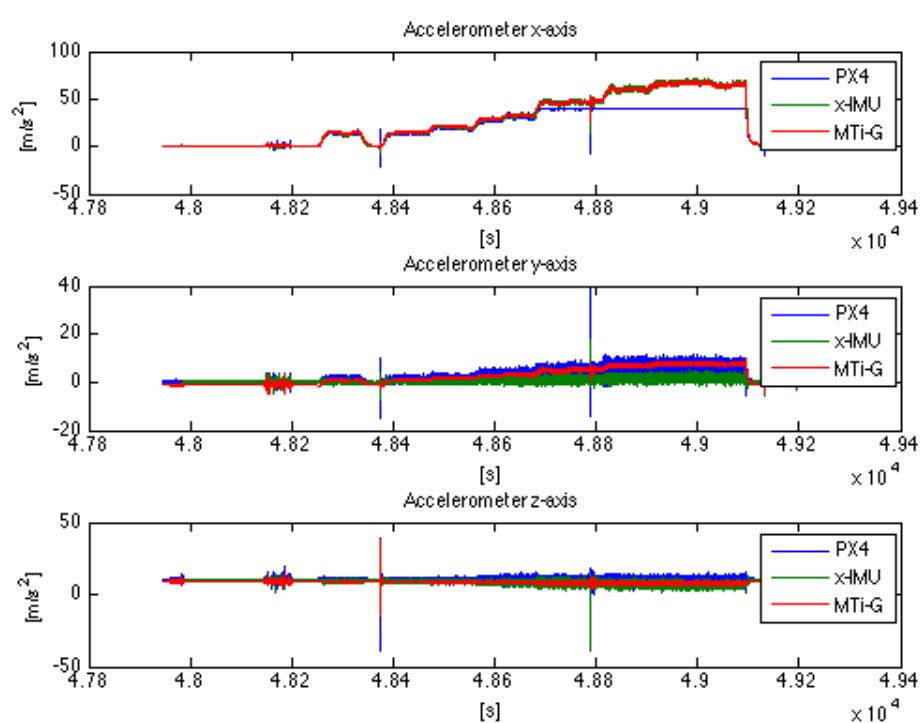


Figure 2.5: Accelerometer

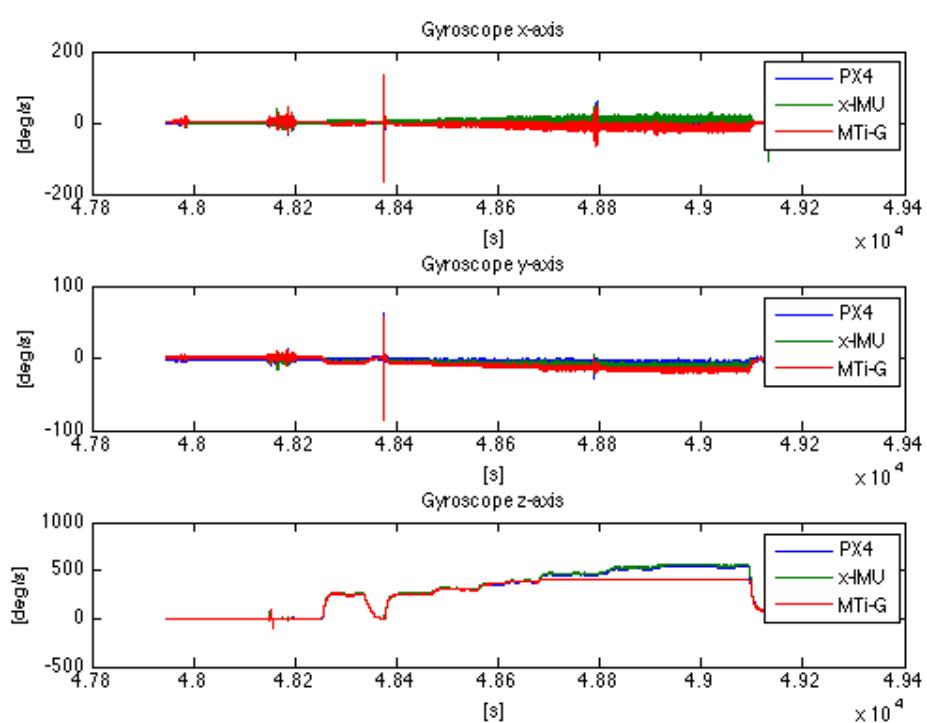


Figure 2.6: Gyrometer

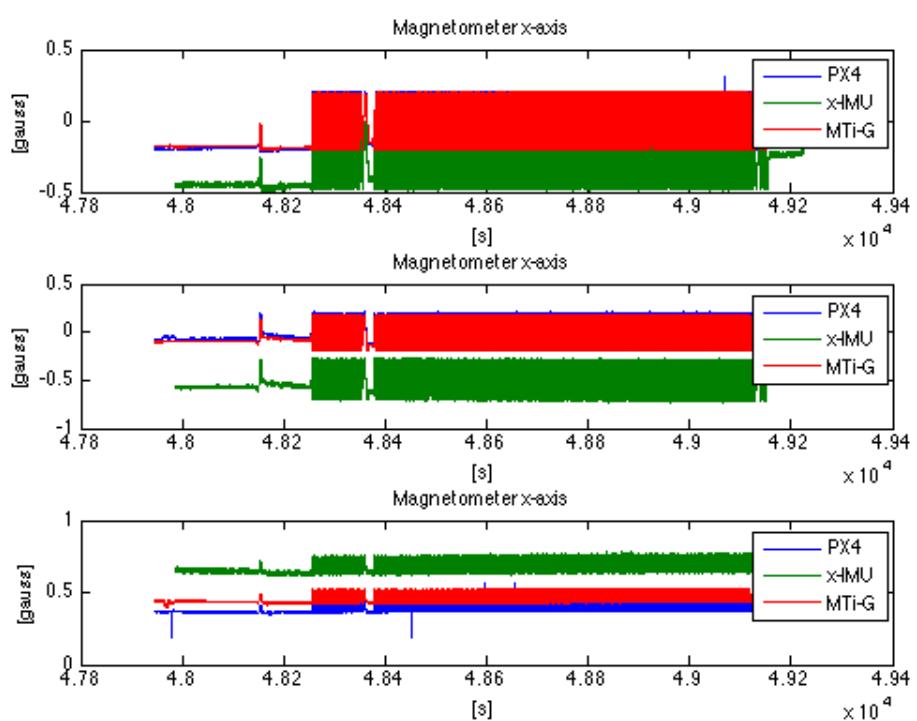


Figure 2.7: Magnetometer

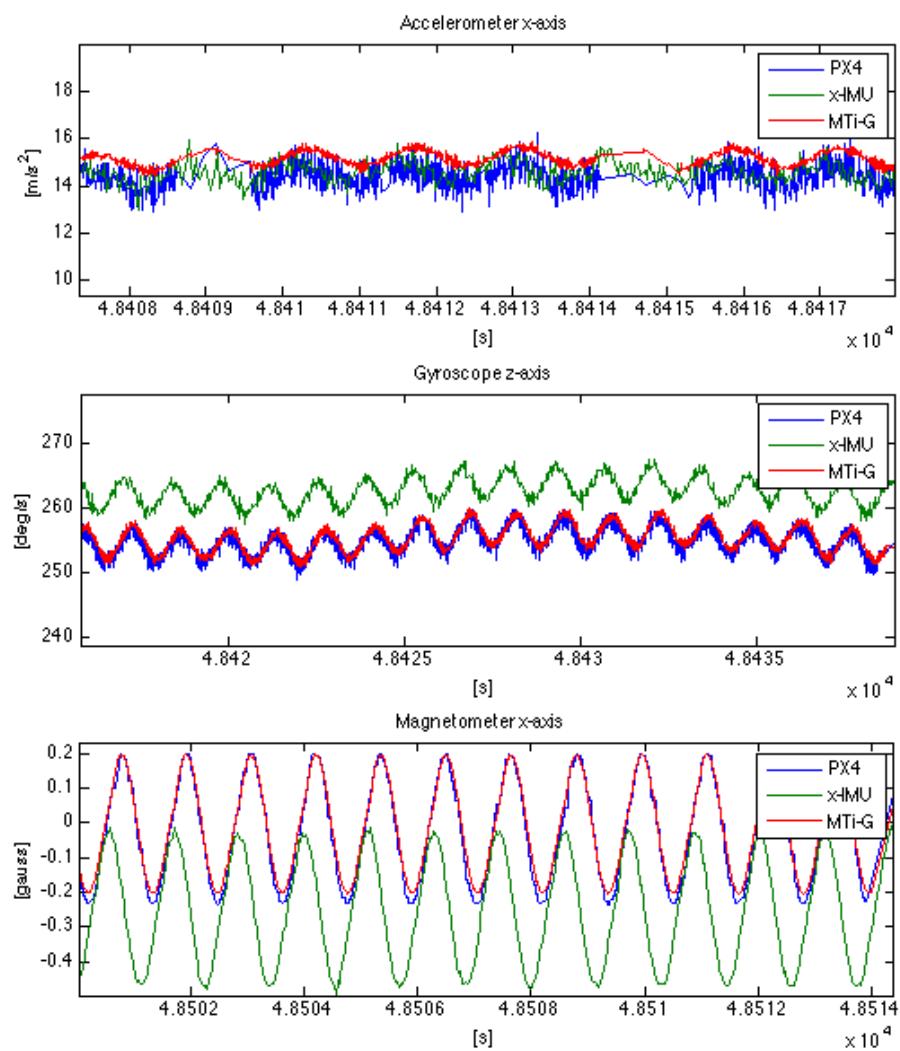


Figure 2.8: A closer look in time on the three IMU sensors.

2.1.3 Conclusion

All IMUs work in the same range of noise level. With an increasing rotational velocity and therefore a higher centripetal acceleration, the average error is increasing in all sensors, except in the magnetometer. There could not be found a dependency between noise in the measurement of the magnetic field and the acceleration. In a future experiment it would be interesting to also record the rotational rate of the centrifuge in order to have a more accurate ground truth that could be used to determine the accuracy of the IMU measurements. This could lead to a more founded comparison of the the different sensors' accuracy. Finally the sensitivity of the accelerometer of the PX4 unit should be set in a way to be able to observe the whole range of the applied centripetal acceleration and the MTi-G would need to be set to record the data in the calibrated output mode. An arbitrary scaling will then not be necessary anymore.

Chapter 3

State Estimation

In this chapter an introduction to the Kalman Filter and the modified version called Extendend Kalman Filter is given. In a second step the state estimation algorithm, that was developed during this thesis, will be explained. This gives an overview over the whole code and describes how the parameters were set and how a sensor data outage is handled. Sections 3.3 and 3.4 will focus on the two essential parts of the algorithm: The propagation of the states and how the measurements are estimated.

3.1 Extended Kalman Filter

Discrete Kalman Filter

We will first give a short introduction to the Kalman Filter in general and its mathematical justification.

The Kalman filter is an extremely effective versatile procedure for combining noisy sensor outputs to estimate the state of a system with uncertain dynamics.[13]

It was published in 1960 by R.E.Kalman. The Filter gained a strong impact in the area of autonomous or assisted navigation [1]. Systems in which the filter is estimating the state $x \in R^n$ can be described by the following linear stochastic difference equation.

$$x_{k+1} = A_k * x_k + B_k * u_k + w \quad (3.1)$$

The next equation shows the relation between the state and the measurement $z \in R^n$.

$$z_k = H_k * x_k + D_k * u_k + v \quad (3.2)$$

Where the random variables w and v stand for the process and the measurement noise, which are assumed to be independent, white and normal probability distributions.

$$p(w) \sim N(0, Q), \quad p(v) \sim N(0, R) \quad (3.3)$$

The $n \times n$ matrix A in equation 3.1 relates the state at time step k to the state at step $k + 1$, in the absence of either a driving function or process noise. The $n \times l$ matrix B relates the control input $u \in R^l$ to the state x . The $m \times m$ matrix H in the measurement equation 3.2 relates to the state to the measurement z_k .[1]

In most navigation applications, the noisy sensors are a GPS receiver and an inertial navigation system or other sensors like for example speed sensors. The states of a system are described by the position, velocity, attitude and attitude rate [13].

Mathematical Background

In figure 3.1 the structure of the algorithm is sketched for a better understanding of the filter's mathematical background. In a first step the state at $k + 1$ is estimated

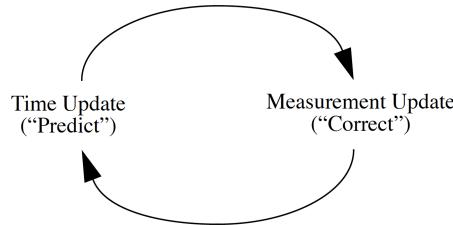


Figure 3.1: A simple structure of the algorithm [1].

based on 3.1. This estimation is called a priori state estimation and written as \hat{x}_k^- . In a second step the a priori state estimation is corrected by the knowledge of measurements z_k . This corrected estimation is called the a posteriori estimation and written as \hat{x}_k .

Now two error can be defined with it's an error covariances. First the error of the a priori estimation

$$e^- = x_k - \hat{x}_k^- \quad (3.4)$$

and second the error of the a posteriori estimation

$$e = x_k - \hat{x}_k \quad (3.5)$$

with the a priori estimate error covariance

$$P_k^- = E[e^- e^{-T}] \quad (3.6)$$

and the a posteriori error covariance.

$$P_k = E[ee^T] \quad (3.7)$$

The Kalman Filter is calculating the a posteriori estimation, the one we are finally looking for. This calculation is a linear combination between the a priori estimated state and the difference of the the estimated measurement $x_k * H$ and the actual measurement z_k weighted with the Kalman gain K . This difference $z_k - H_k \hat{x}_k^-$ is called residual and tells us how accurate the estimation of the measurements is. This is summarized in equation 3.8.

$$\hat{x}_k^- * K * (z_k - H_k \hat{x}_k^-) \quad (3.8)$$

The Kalman gain K is the heart of the Kalman Filter. K , a $n \times n$ Matrix is chosen in a way to minimize the a posteriori error covariance shown in equation 3.7. With some calculations the following

$$K = \frac{P_k^- * H_k^T}{(H_k * P_k^- * H_k^T + R_k)} \quad (3.9)$$

can be derived for the Kalman gain K [1] This concept uses the information of a normal gaussian distribution of the noise mentioned in equations 3.3. P_k is minimized using the Maximum Likelihood Estimation and is called the Gaussian Maximum-Likelihood Estimator. A more detailed derivation of the mathematics is shown in [14] and [13].

Algorithm

With equations described in the section above, the estimation and correction step can be summarized as shown in figure 3.2. For a fine tuning of the filter often the measurement error covariance matrix R_k and process noise Q_k can be used. R_k contains the information how accurate the sensors and therefore the measurements are expected to be. This can be estimated in a first round with some static sensor test or can be found in the data sheets of the sensors. The matrix Q_k describes how correct the propagation model described by the matrix A_k is believed to be. In this sense as higher the values and therefore the noise of R_k and Q_k the less influence do the measurements or the propagation model respectively have on the estimated state. Having a diagonal matrix with constant values over time, does not represent the reality, but brings it to a form which is fast stable and easier tunable [1].

Extended Kalman Filter (EKF)

In reality the process to estimate the new state and the measurement relationship to the process is non-linear.

$$x_{k+1} = f(x_k, u_k, w) z_k = h(x_k, v_k) \quad (3.10)$$

One of the solution to overcome this problem is the Extended Kalman Filter shortly EKF. The EKF is linearizing the functions with a first order Taylor-Series around the current state. This can be summarized with the following equation:

$$x_{k+1} \approx \hat{x}_{k+1} + A(x_k - \hat{x}_k) + w \quad z_k \approx \hat{z}_k + H(x_k - \hat{x}_k) + v \quad (3.11)$$

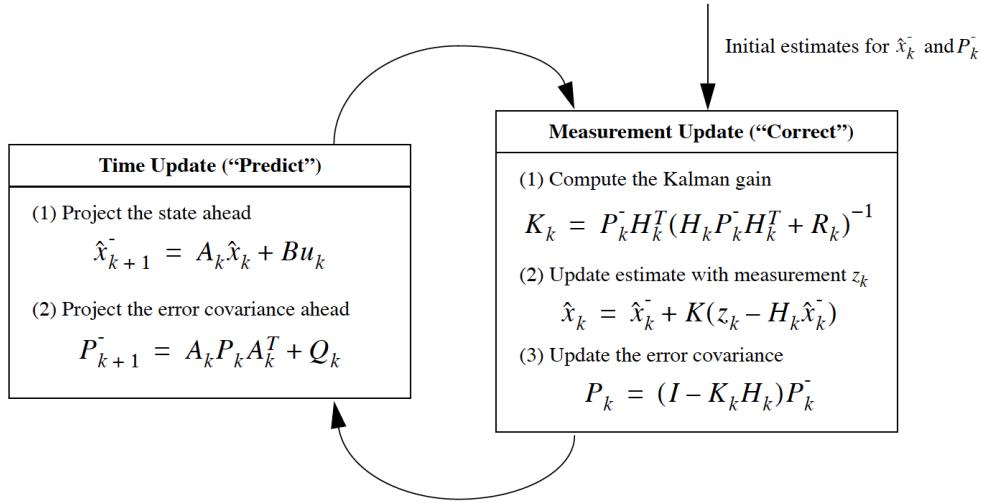


Figure 3.2: The prediction and the correction step with their corresponding equations [1].

Where A and H are the Jacobian matrix of the function f and h .

$$A_{i,j} = \frac{\partial f_i}{\partial x_j}(\hat{x}_k, u_k, 9) \quad H_{i,j} = \frac{\partial h_i}{\partial x_j}(\hat{x}_k, u_k, 9) \quad (3.12)$$

This brings us to the summarized equations of the EKF in figure 3.3.

3.2 Algorithm for State Estimation

In this chapter we will present the structure of the state estimation algorithm as we implemented it in MATLAB. As mentioned before, the goal is to get an estimation of the system's state as precise as possible using an Extended Kalman Filter. The Algorithm can be split in several main tasks. First the physical relations between the state x_k and the state x_{k+1} has to be defined and summarized in a propagation model. This is needed to calculate the a priori state estimation as described in chapter 3.1. Since this step is crucial for the accuracy of the algorithm, it is further explained in chapter 3.3. A second key role plays the measurement estimation which corrects the a priori state estimation and calculates the a posteriori state estimation. A more detailed explanation of this correction can be found in chapter 3.4. A third question to be elaborated is how the algorithm has to behave if no sensor data is available or if not all sensors provide their outputs with the same frequency. In the last paragraph we will explain how the error covariances were chosen.

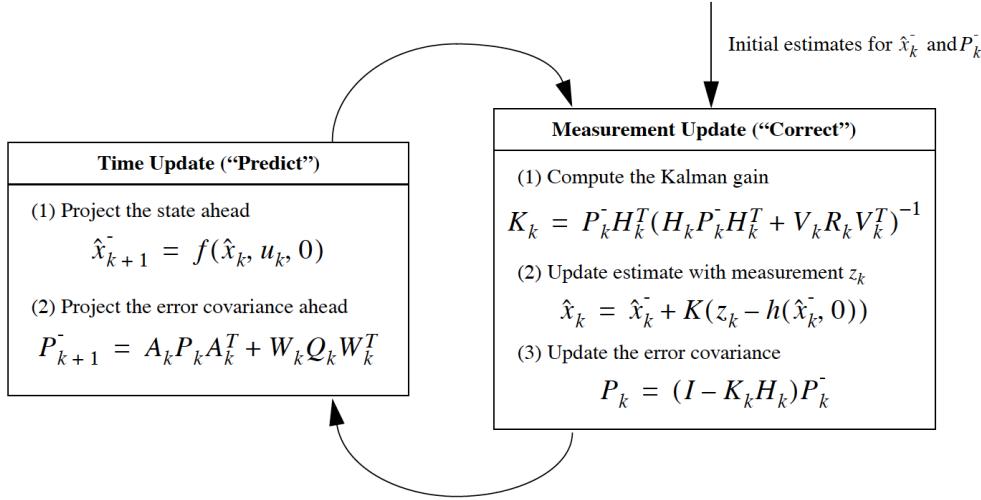


Figure 3.3: The prediction and correction step of the EKF with their corresponding equations [1].

Orientation

The estimator algorithm has to handle two different coordinate frames. One is the inertial frame and the other the body frame. The inertial frame is the earth fixed frame. The two measurements position and the velocity both provided by the GPS are in the inertial frame. Also the states of the system are in the inertial frame. The origin of the inertial frame lies at the suspension point of the pendulum/kite. The body frame is a coordinate system on the kite. The propagation model needs the body frame to calculate the forces acting on the body. Measurements form the IMU are as well as in body coordinates since they represent the forces. Its z-axis always points in the direction of the line and its origin lies at the center of gravity of the kite. The sensor measurements from the IMU are all in the body frame. Figure 3.4 contains an illustration of these two coordinate systems.

The euler angles describe how the inertial frame has to be rotated to bring into the body frame. With other words, the euler angles tell us the orientation of the body frame. With the direct cosine matrix (*DCM*) we can transform a vector from the inertial frame into the body frame and vice versa. For example can the accelerometer data which is given in body frame coordinates be transformed into the inertial frame, where the propagation of the states is calculated [13].

Degrees of Freedom

As stated in the introduction, we implemented two different versions of the state estimation algorithm. A standard implementation assuming a free mass and one version that exploits the constraints given by the knowledge that the mass is oscil-

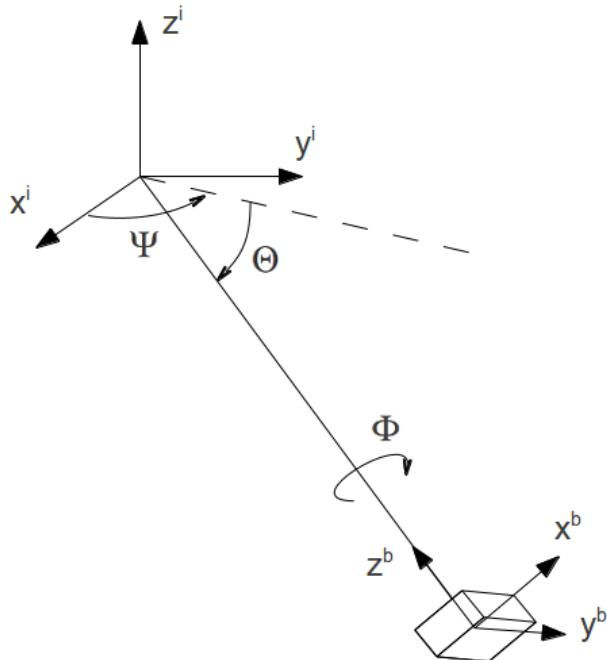


Figure 3.4: The definition of the inertial and body frame as well as the convention for the angles in the pendulum model.

latingly suspended at a fixed point. These two versions are quite similar and share a lot of code, but one crucial point where they differ from each other is the number of degrees of freedom (DoF). The free mass model uses 12 DoF, namely position, velocity, attitude and attitude rate. Thus the state vector consists of 12 variables. The pendulum model uses only 6 true degrees of freedom: Three angles representing the position and attitude as well as their respective rates representing velocity and attitude rate. The definition of these angles can be seen in figure 3.4.

States and Measurements

The state vector in the free mass model with the dimension of 12 looks as follows:

$$x = \begin{pmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \phi \\ \theta \\ \psi \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \quad (3.13)$$

The first six variables are position and velocity in the inertial frame. The angles and angular rates are Euler angles with the convention ZY'X''. This means that in order to get from the inertial frame to the body frame the following three rotations have to be performed: a rotation with angle ψ around the z-axis in the inertial frame, a rotation with angle θ around the new y-axis and a last rotation with angle ϕ around the new x-axis.

The state vector in the pendulum model has eight elements. Even though only 6 DoF are assumed, two additional variables, the radius and the change in radius, are included in the state to ensure the possibility of loosening the constraint of a fixed line length and implementing a spring model as suggested later on in the outlook section. Thus the state vector is defined as follows:

$$x = \begin{pmatrix} \Phi \\ \Theta \\ \Psi \\ r \\ \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \\ \dot{r} \end{pmatrix} \quad (3.14)$$

For the definition of the angles Φ , Θ and Ψ see figure 3.4.

The measurement vector is the same for the free mass model as well as for the pendulum model. It consists of the position (pos) and the velocity (vel) of the GPS, the acceleration (acc) from the accelerometer and the rate of turn (gyro) from the gyroscope and the orientation (mag) from the magnetometer. Each of them in all

three dimensions give us the measurement vector z with a dimension of 15.

$$z = \begin{pmatrix} pos\ x \\ pos\ y \\ pos\ z \\ vel\ x \\ vel\ y \\ vel\ z \\ acc\ x \\ acc\ y \\ acc\ z \\ gyro\ x \\ gyro\ y \\ gyro\ z \\ mag\ x \\ mag\ y \\ mag\ z \end{pmatrix} \quad (3.15)$$

Structure of the Algorithm

In 3.5 the schematic of the algorithm is shown. Each of the blocks will now successively explained.

Block 1, totalTime: The filter estimates with constant rate the state of the system. This rate is independent on any output rate of the sensors. In the first block the time at which the state is estimated is set. It is the old time total-Time added with the constant time step t .

Block 2, Propagation: During the second block the a priori state of the system is estimated with the help of the physical model. How and why the physical model is defined is explained in chapter 3.3. After this step the vector x_{est} is up to date.

Block 3, Finding newest Data: If there is no new data available until the time of the a priori estimated state, the correction step is not executed. This allows the algorithm to run the EKF with a higher rate, then the IMU and GPS provide data. If there is new data available, the algorithm is finding the newest or closest one to the a priori estimation time(totalTime). To summarize: If we have no new data from the sensors the system just keeps propagating based on the physical model. If there is at least one new measurement from any sensor block 4 will be executed. If we have new data, the algorithm takes the newest one for the correction.

Block 4, measurement selection: Mostly not all sensors of an IMU provide an output with the same rate. Here it is tested, which sensors actually have

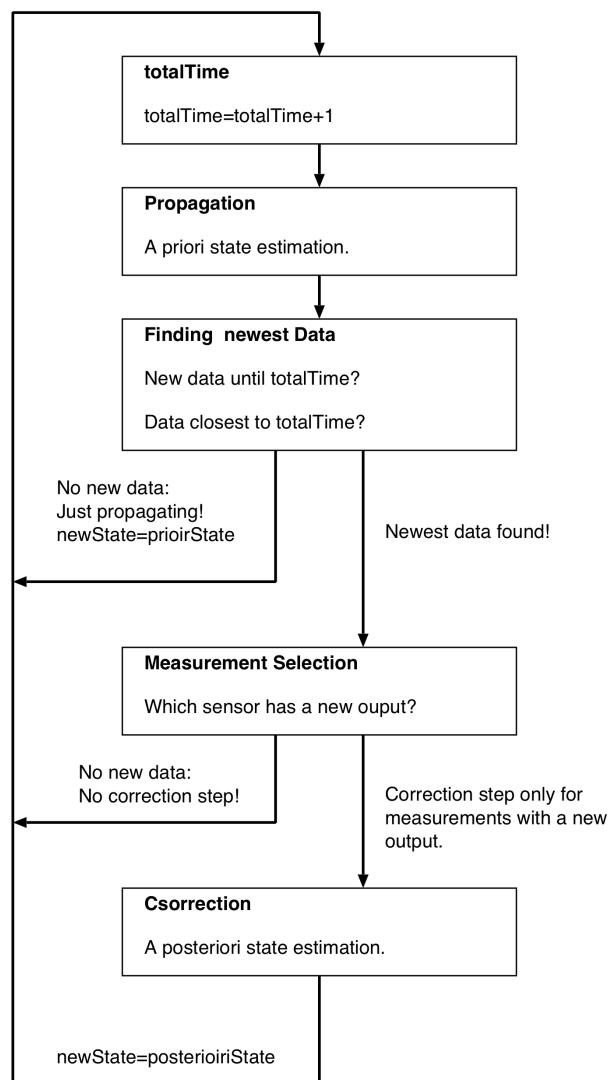


Figure 3.5: The graphical representation of the algorithm's structure.

a new output value and which has still it's output value from the previous estimation. The correction step in block 5 is only executed for measurements with new data and therefore only these have an influence updating the a priori estimation to the a posteriori estimation.

Block 5, correction: As last step the correction of the a priori estimation takes place and the posteriori estimation is calculated. How the relationship between the state and the measurements look like is explained in chapter 3.4

Error Covariance

For the measurement error covariance the noise of the different sensors are taken from the data sheets. But they were manually scaled according to have more accurate and stable filter. For the propagation model the covariance was estimated on how accurate the equations are. But also in this case were they manually adjusted afterwards. For an easier handling only the diagonal elements have a value. The off diagonal values are all zero.

Jacobian Matrix

In chapter 3.1 the matrixes H and A were derived. Since the physical model and relations between state and the measurements are non linear, H and A are the Jacobian matrixes of the functions f and h . For having an accurate as possible solution the Jacobian was in a first version calculated analytically with symbolic toolbox from MathWorks. The Jacobian matrix then has to be evaluated in every iteration step. Simulating 0.01 seconds took the filter much longer than 10 minutes. It was then decided to rewrite the algorithm calculating the Jacobian matrixes numerical. The function `ekf.m` [..ekfwebsite..]from MathWorks was restructured to match the requirements of this algorithm. It uses a complex step differentiation to calculate the derivatives of the function f and h [..papCompDiff..].

3.3 State Estimation Model

In order to propagate the state of the system x_k to the a priori state estimation \bar{x}_{k+1} the filter needs to be able to predict in what state the system will be at time t_{k+1} based on its state at time t_k and the control inputs, if there are any. In the most basic case, the assumption is that there exists no knowledge about the forces acting on the body. This leads to the so called "free mass model". However, in the case of a kite tethered to a ground station, the body is not able to move freely in three dimensions, but its movement is in a first approximation limited to a sphere with a given radius. This means, that one of the main forces acting on the body, the centripetal force, can be accounted for in the prediction. Furthermore an aerodynamical model of the kite might be able to predict also other forces than the centripetal force. In order to explore the benefits of such a model, we decided

to use a spherical pendulum for the reasons described in the introduction. In that case, all the forces acting on the body are known and the degrees of freedom are further reduced due to the bridled suspension. In the following sections, these two models are going to be described in detail.

Free Mass Model

The dynamics of the body are described by a system of first order differential equations describing the change in the state variables.

$$\dot{x} = g(x, t) \quad (3.16)$$

In this case the model is time invariant and thus g is not time dependent.

$$\dot{x} = g(x) \quad (3.17)$$

Since there is no knwoledge about the forces acting on the body, this set of equation looks rather simple:

$$g(x) = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ 0 \\ 0 \\ 0 \\ x_{10} \\ x_{11} \\ x_{12} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.18)$$

Pendulum

Again the dynamics are described by a system of time invariant first order differential equations.

$$\dot{x} = g(x) \quad (3.19)$$

Due to the gravitation and the centripedal force being the only forces acting on the body, we can derive an exact set of differential equations that govern the motion of a spherical pendulum. Therefore we use the Euler-Lagrange-Equation to derive the differential equations as suggested in [15, 156ff].

With the kinetic energy

$$T = \frac{1}{2}mr^2(\dot{\Theta}^2 + \dot{\Psi}^2 \cos^2 \Theta) \quad (3.20)$$

and the potential energy

$$V = -mr \sin \Theta \quad (3.21)$$

the Lagrangian is defined by:

$$L = T - V = \frac{1}{2}mr^2(\dot{\Theta}^2 + \dot{\Psi}^2 \cos^2 \Theta) + mr \sin \Theta \quad (3.22)$$

Substituting the Lagrangian into Euler-Lagrange-Equation

$$\frac{\partial L}{\partial \Psi} - \frac{d}{dt} \frac{\partial L}{\partial \dot{\Psi}} = 0 \quad (3.23)$$

and solving for $\ddot{\Psi}$ results in:

$$\ddot{\Psi} = 2 \tan \Theta \dot{\Theta} \dot{\Psi} \quad (3.24)$$

And similarly for $\ddot{\Theta}$:

$$\frac{\partial L}{\partial \Theta} - \frac{d}{dt} \frac{\partial L}{\partial \dot{\Theta}} = 0 \quad (3.25)$$

$$\ddot{\Theta} = \frac{\cos \Theta(g - r \sin \Theta \dot{\Psi}^2)}{r} \quad (3.26)$$

The third degree of freedom, the body's rotation about its z-axis, is assumed to be constant. However, due to the convention of the euler angles, a change in phi contributes an additional term to the z component of the rotation vector in the body frame. (For further explanation see section ...) To compensate for that, the second derivative of Φ is not zero but defined as follows:

$$\frac{d}{dt} w_z^{body} = \frac{d}{dt}(\dot{\Phi} - \sin \Theta \dot{\Psi}) = 0 \quad (3.27)$$

$$\ddot{\Phi} = \cos \Theta \dot{\Theta} \dot{\Psi} + \sin \Theta \ddot{\Psi} \quad (3.28)$$

With the radius assumed to be constant, the fourth and eighth state variable (r and \dot{r}) are not changing and thus their time derivatives are zero.

Therefore the set of differential equations is given by:

$$g(x) = \begin{bmatrix} x_5 \\ x_6 \\ x_7 \\ 0 \\ \cos x_2 \dot{x}_2 \dot{x}_3 + \sin x_2 \ddot{x}_3 \\ \cos x_2(g - r \sin x_2 \dot{x}_3^2) \\ \frac{r}{2 \tan x_2 \dot{x}_2 \dot{x}_3} \\ 0 \end{bmatrix} \quad (3.29)$$

Solving the Differential Equations

So far we have derived the sets of differential equations that contain all the knowledge we have about the systems. To calculate the a priori state estimation \bar{x}_{k+1} from the state of the system x_k we now only need to solve these sets of differential equations. This is done numerically using the classical fourth-order Runge-Kutta method [16].¹ Note that in our case $g(x)$ is time invariant.

$$k_1 = hg(t_n, x_n) \quad (3.30)$$

$$k_2 = hg\left(t_n + \frac{1}{2}h, x_n + \frac{1}{2}k_1\right) \quad (3.31)$$

$$k_3 = hg\left(t_n + \frac{1}{2}h, x_n + \frac{1}{2}k_2\right) \quad (3.32)$$

$$k_4 = hg(t_n + h, x_n + k_3) \quad (3.33)$$

$$x_{n+1} = x_n + \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4 + \mathcal{O}(h^5) \quad (3.34)$$

3.4 Sensor Estimation Model

An illustration of the relation between the position and velocity in (y, x, z) or (dx, dy, dz) and the two angles Φ, Θ and Ψ is shown in figure 3.4. With this sketch, the assumption of a constant radius and the fact that the velocity is the derivative of the position we get to the equations:

$$pos\ x = R * \cos(\Theta)\cos(\Psi) \quad (3.35)$$

$$pos\ y = R * \sin(\Psi)\cos(\Theta) \quad (3.36)$$

$$pos\ z = R * \sin(\Theta) \quad (3.37)$$

$$vel\ x = -\sin(\Theta)\cos(\Psi) * R * \dot{\Theta} - \cos(\Theta)\sin(\Psi) * R * \dot{\Psi} \quad (3.38)$$

$$vel\ y = -\sin(\Theta)\sin(\Psi) * R * \dot{\Theta} + \cos(\Theta)\cos(\Psi) * R * \dot{\Psi} \quad (3.39)$$

$$vel\ z = -\cos(\Theta) * R * \dot{\Theta} \quad (3.40)$$

$$(3.41)$$

The centripetal acceleration always acting in z direction of the sensor can be calculated by the formula:

$$a_{cp} = \frac{v^2}{R} \quad (3.42)$$

We then have the an additional acceleration of the earth gravitation. The earth gravitation is in the inertial frame only in z direction and has to be transformed

¹This is the same method that is used in Matlab's built in ODE45 differential equation solver, with the difference that it only uses one timestep. This leaves the possibility to analytically derive the Jacobian matrix.

with the DCM_{bi} from the inertial in the body frame. Finally the acceleration can be written as:

$$acc = \begin{bmatrix} 0 \\ 0 \\ \frac{v^2}{R} \end{bmatrix} + DCM_{bi} * \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (3.43)$$

The magnetic field is a constant value in Zurich. By transforming that into the body frame again with the help of the DCM_{bi} , the expected measurement from the magnetometer is calculated:

$$mag = DCM_{bi} * \begin{bmatrix} 0.2145 \\ 0.0060 \\ 0.4268 \end{bmatrix} [gauss] \quad (3.44)$$

The gyroscope measurement is directly represented by the rate of the euler angles $\dot{\Phi}, \dot{\Theta}$ and $\dot{\Psi}$. Since the euler angular rate has to be transformed into the body frame coordinates and additinally has a converstion about which axes is rotated the first, what the gyroscope has not,a transformation matrix has to be used as follows:

$$gyro = \begin{bmatrix} 1 & 0 & -\sin(\Theta) \\ 0 & \cos(\Phi) & \sin(\Phi)\cos(\Theta) \\ 0 & -\sin(\Phi) & \cos(\Phi)\cos(\Theta) \end{bmatrix} * \begin{bmatrix} \dot{\Psi} \\ \dot{\Theta} \\ \dot{\Phi} \end{bmatrix} \quad (3.45)$$

Because the position of the senors is not at the center of mass of the pendulum, an additional displacement vector has to be added to the position (compare figure 3.6). With transforming the distance between the center of mass and the sensor into the inertial frame, the position is adjusted. The displacement gives an additional velocity which is the crossproduct of the rate of rotation and the displacement from the center of mass. By the derivative of this additional velocity the additional acceleration is calculated. The displcement has no influence on the gyrometer measurement and the magnetometer.

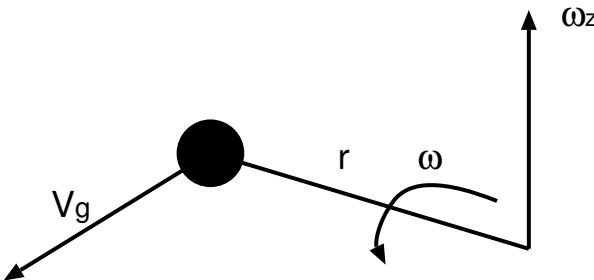


Figure 3.6: The black dot represents the sensor, while r is the distance between the center of mass of the pendulum and the sensor.

Chapter 4

Results - Vicon Test

In this chapter the algorithm from chapter 2 is tested under different conditions. In a first part (see chapter 4.1) the set up of the test is presented. In the second part (see chapter 4.2). The different test conditions are described and the results shown.

4.1 Set up

To verify the performance of the Kalman filter algorithm described in chapter 3 a ground truth is needed. By using the motion capture system from the English company "vicon" some tests were performed. The vicon system provides the position of the sensors and the orientation. The convention of the three Euler Angles describing the orientation is different from any of the conventions used in this text so far. The rotation sequence is XY'Z". This means that Ψ (the rotation about the Z" axis) now corresponds to the box's rotation about the line. Because this test has to be performed indoors no GPS signal is available. The vicon data for position added with some noise is simulated as GPS measurement and as such handed to the algorithm. The same is done with the derivative of the vicon position to provide a velocity measurement. The impact of the noise level added to the signals can be seen in chapter 4.2.2. The pendulum setup can be seen on the picture in figure 4.1. The box with the two sensors PX4 and MTi-G is showed in figure 4.2. How the algorithm's state estimation differs depending on wether the measurements from the PX4 IMU or from the MTi-G IMU are used, can be read in chapter 4.2.4.

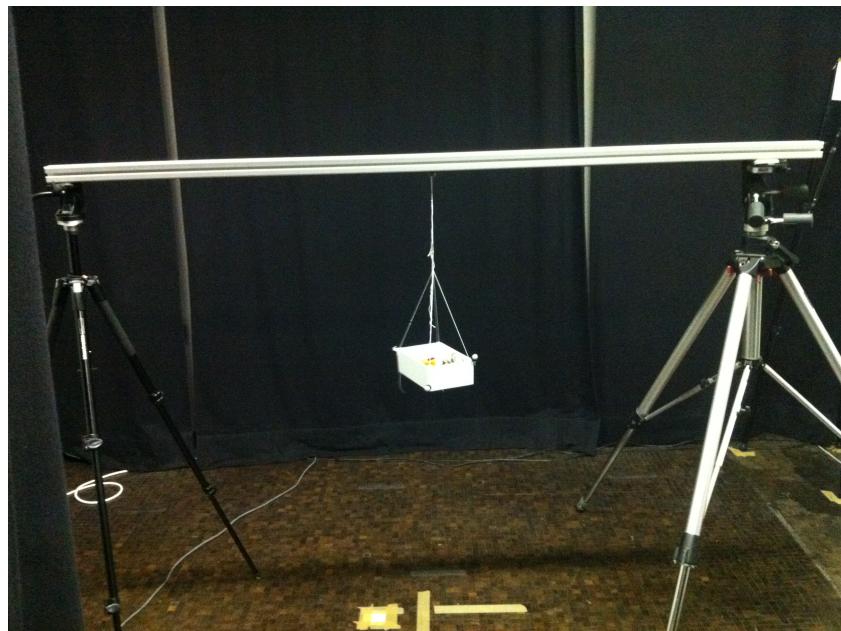


Figure 4.1: Pendulum set up.

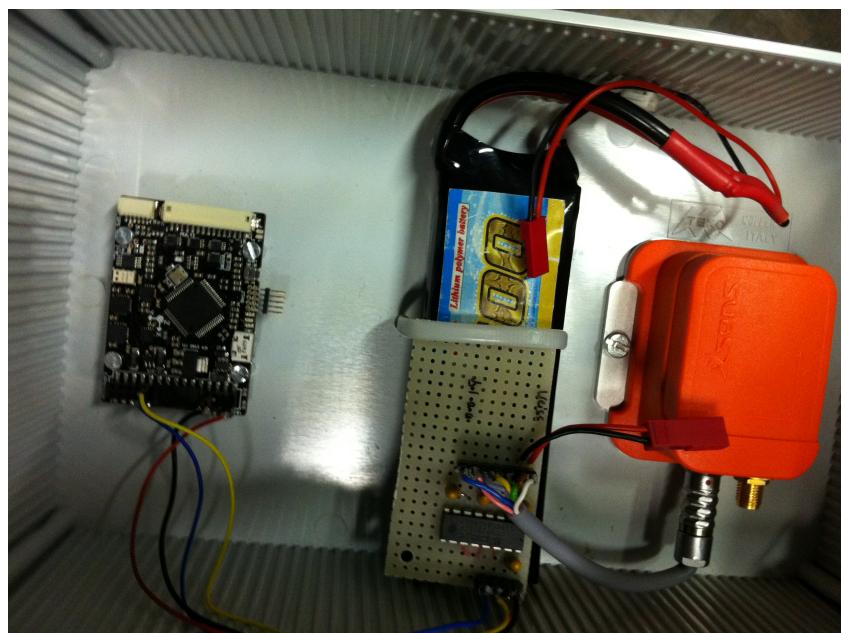


Figure 4.2: Box with sensors.

4.2 Evaluation

Four comparisons are made to get an overview of the estimators performance. This chapter shows how different test conditions influence the performance. First it has to be verified if an estimator using a model of the pendulum's physical behavior is more accurate than a conventional Extended Kalman Filter using the free mass model for estimating. The results of this comparison can be seen in section 4.2.1. In chapter 4.2.2 the estimator is tested once with a low noise GPS signal and a second time with a higher noise GPS signal. Since a GPS signal is not always available or has at least a lower frequency than the IMU measurements, the behavior of the estimator in between two GPS fixes is a main performance criterion for an estimator. In chapter 4.2.3 this condition is tested. How the estimator performs with the measurements of the two IMUs is shown in section 4.2.4. The following tests are all using the MTi-G IMU except the last one in chapter 4.2.4.

4.2.1 "Pendulum"-estimator vs. "free mass model"-estimator

In order to investigate the performance difference of the two different estimators, they were both tested with the same inputs and their respective outputs were compared. In both cases a GPS signal with a signal to noise ratio (SNR) of $-5dB$ was used. SNR is the ratio between the desired signal and the background noise. It behaves like the smaller the ratio the higher the noise. The results are shown in Figures 4.3, 4.4, 4.5 and 4.6. The estimator output is plotted together with the ground truth obtained from the Vicon system.

From these plots it can be seen, that the accuracy of the position estimation is not very much dependent of the estimator. However, due to its reduced number of degrees of freedom, the "Pendulum"-estimator clearly outperforms the "free mass model"-estimator in the accuracy of the orientation. To quantify these observations, the errors in position and orientation are calculated. The position error is calculated by the norm of the difference between estimated position and the effective position:

$$error = \left\| \begin{bmatrix} vicon_{pos\ x} \\ vicon_{pos\ y} \\ vicon_{pos\ z} \end{bmatrix} - \begin{bmatrix} state_{pos\ x} \\ state_{pos\ y} \\ state_{pos\ z} \end{bmatrix} \right\| \quad (4.1)$$

For the orientation only the error of the Ψ angle is calculated since the errors in the other two angles are represented by the error in position. The plots of these errors can be seen in Figures 4.10 and 4.7. When calculating the mean value of the error, starting at the time, when the estimator is settled down, the following results are

obtained:

$$\text{meanError}_{\text{pos}}^{\text{pendulum}} = 0.0162 \text{ m} \quad (4.2)$$

$$\text{meanError}_{\text{pos}}^{\text{fmm}} = 0.0280 \text{ m} \quad (4.3)$$

$$\text{meanError}_{\Psi}^{\text{pendulum}} = 0.1440 \text{ rad} \quad (4.4)$$

$$\text{meanError}_{\Psi}^{\text{fmm}} = 0.3227 \text{ rad} \quad (4.5)$$

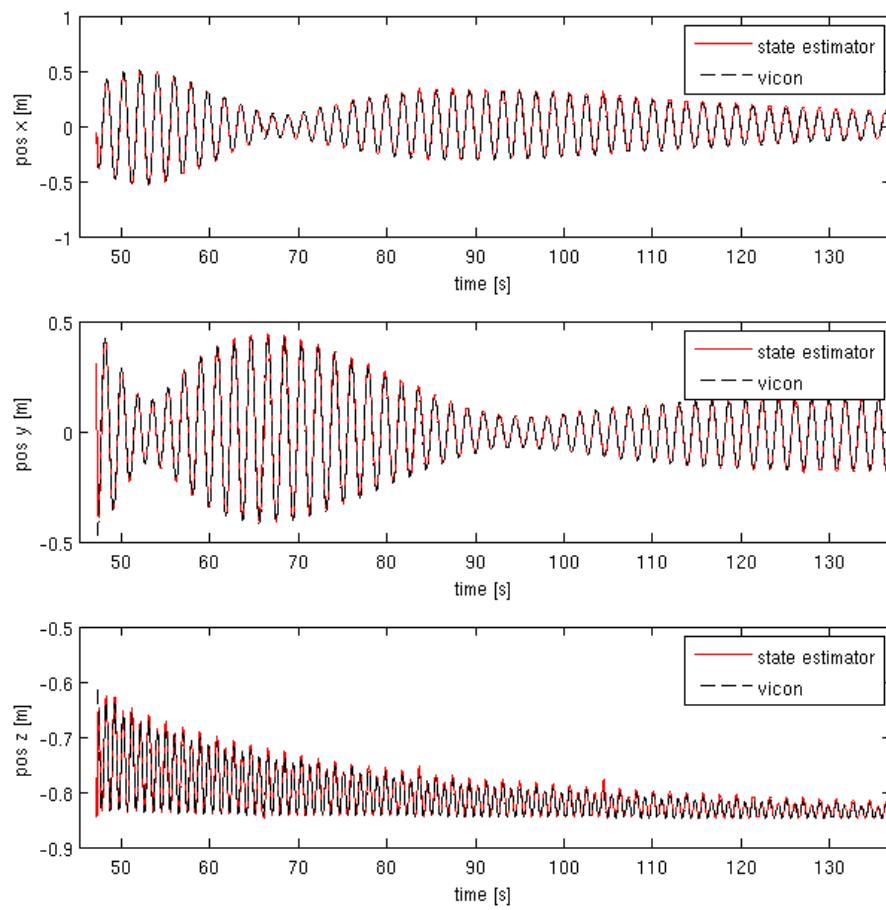


Figure 4.3: Estimation of the position using the "Pendulum"-estimator compared to the ground truth obtained from the Vicon system.

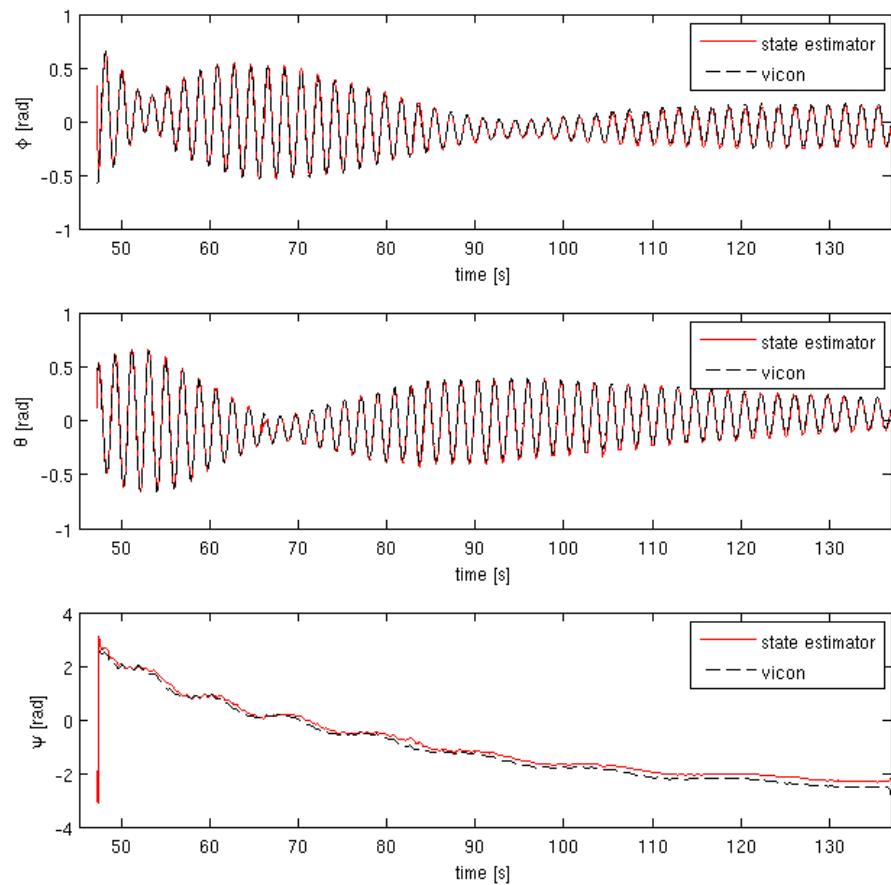


Figure 4.4: Estimation of the orientation using the "Pendulum"-estimator compared to the ground truth obtained from the Vicon system.

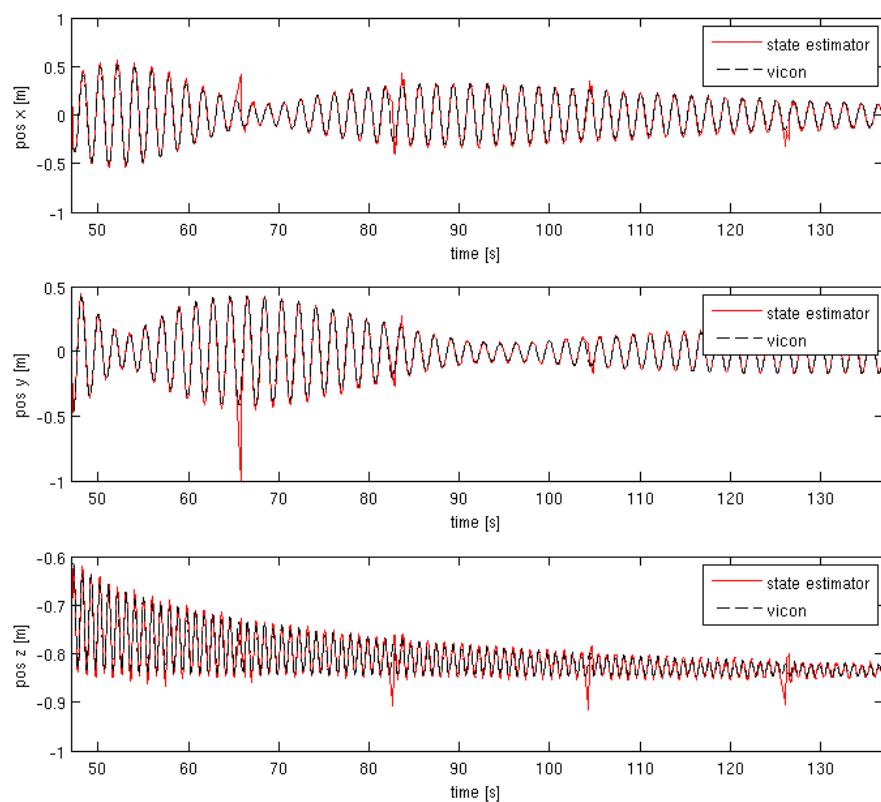


Figure 4.5: Estimation of the position using the "free mass model"-estimator compared to the ground truth obtained from the Vicon system.

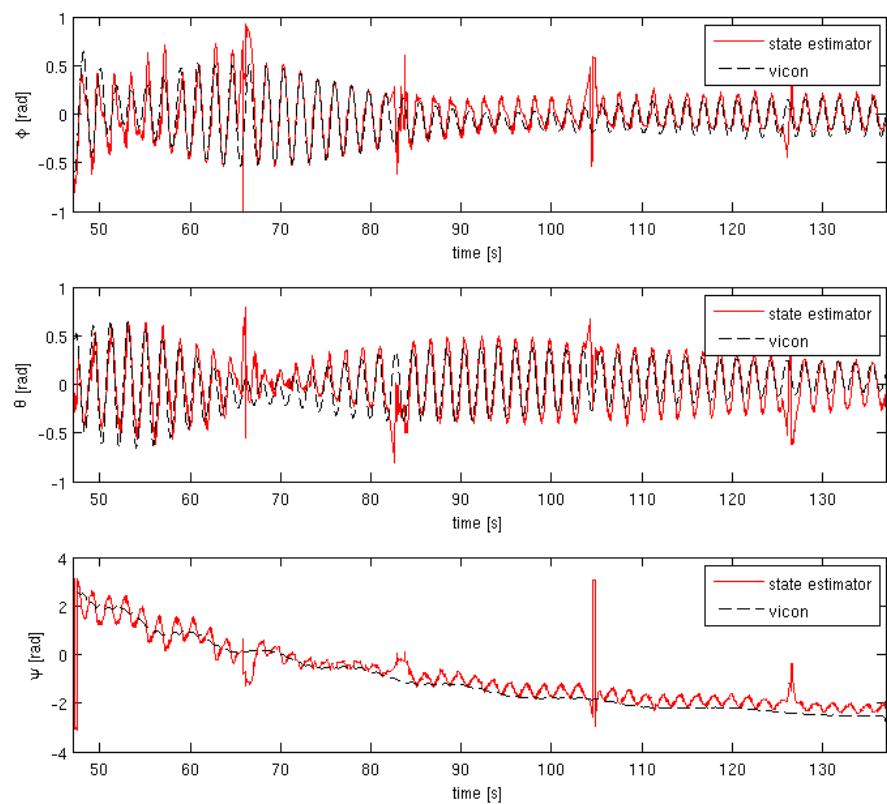


Figure 4.6: Estimation of the orientation using the "free mass model"-estimator compared to the ground truth obtained from the Vicon system.

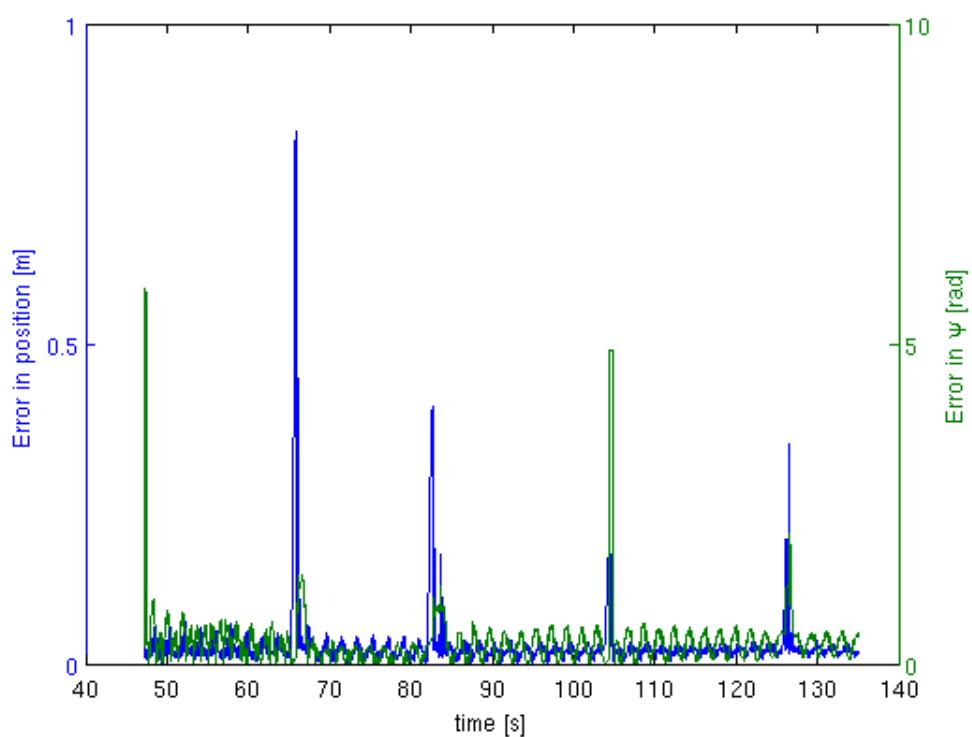


Figure 4.7: Error in position and orientation for the "free mass model"-estimator.

4.2.2 Noise of the GPS

The quality of the received GPS signal can vary according to the GPS unit used or due to different circumstances. The weather conditions or an acceleration as we have seen in chapter 2 can have an influence on the signals quality. As mentioned before is the GPS signal in the vicon test only simulated according the vicon data. This has the advantage of being able to use a GPS with different SNR since this is artificially added on the vicon's position data. The estimator output somehow

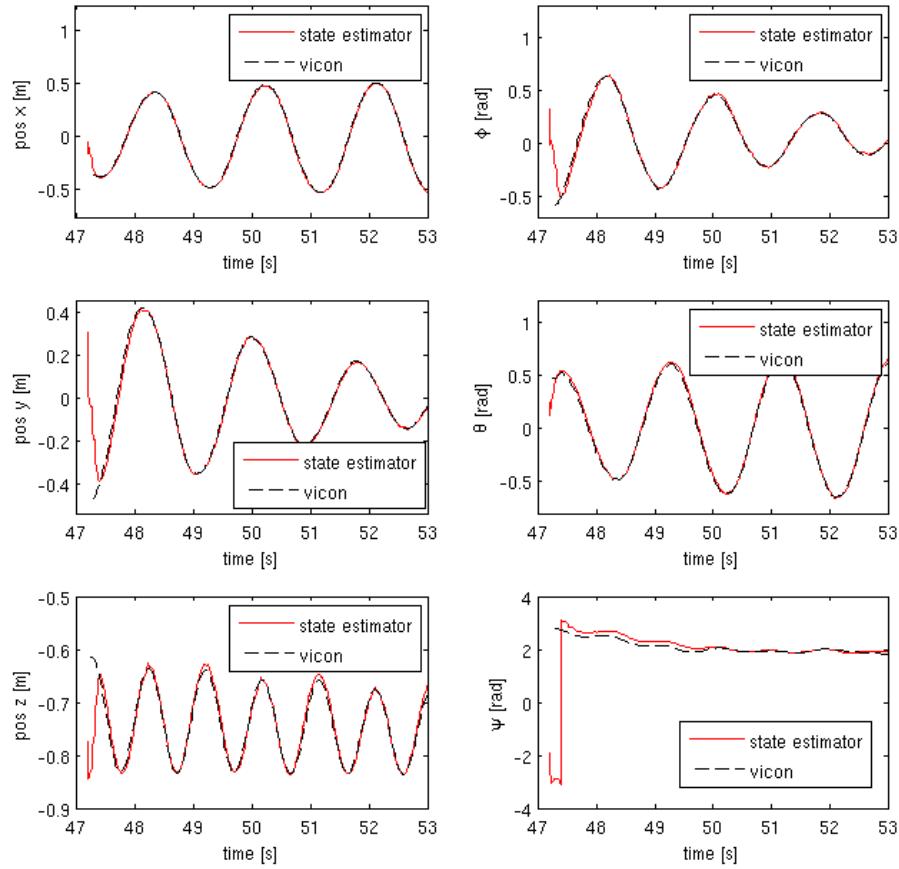


Figure 4.8: GPS signal with a SNR of -5 dB.

differs, but the noise level does not have a strong impact. In the algorithm, the noise covariance matrixes are adjusted depending on the input signal's SNR. In the first case of a -5 dB SNR the variance of the GPS position and velocity are defined as 0.001 m and 0.01 m/s respectively. In the second case with a SNR of -25 dB the variance in the algorithm is increased to 0.01 m for the position and 0.05 m/s for

the velocity. In figure 4.10 and figure 4.11 the errors in position and orientation are plotted for the -5 dB and -25 dB GPS signals respectively. For the orientation only the error of the Ψ angle is calculated since the errors in the other two angles are represented by the error in position. When calculating the mean value of the error, starting at the point, when the estimator is settled down, the following results are obtained:

$$\text{meanError}_{\text{pos}}^{-5 \text{ dB}} = 0.0162 \text{ m} \quad (4.6)$$

$$\text{meanError}_{\text{pos}}^{-25 \text{ dB}} = 0.0223 \text{ m} \quad (4.7)$$

$$\text{meanError}_{\Psi}^{-5 \text{ dB}} = 0.1440 \text{ rad} \quad (4.8)$$

$$\text{meanError}_{\Psi}^{-25 \text{ dB}} = 0.1584 \text{ rad} \quad (4.9)$$

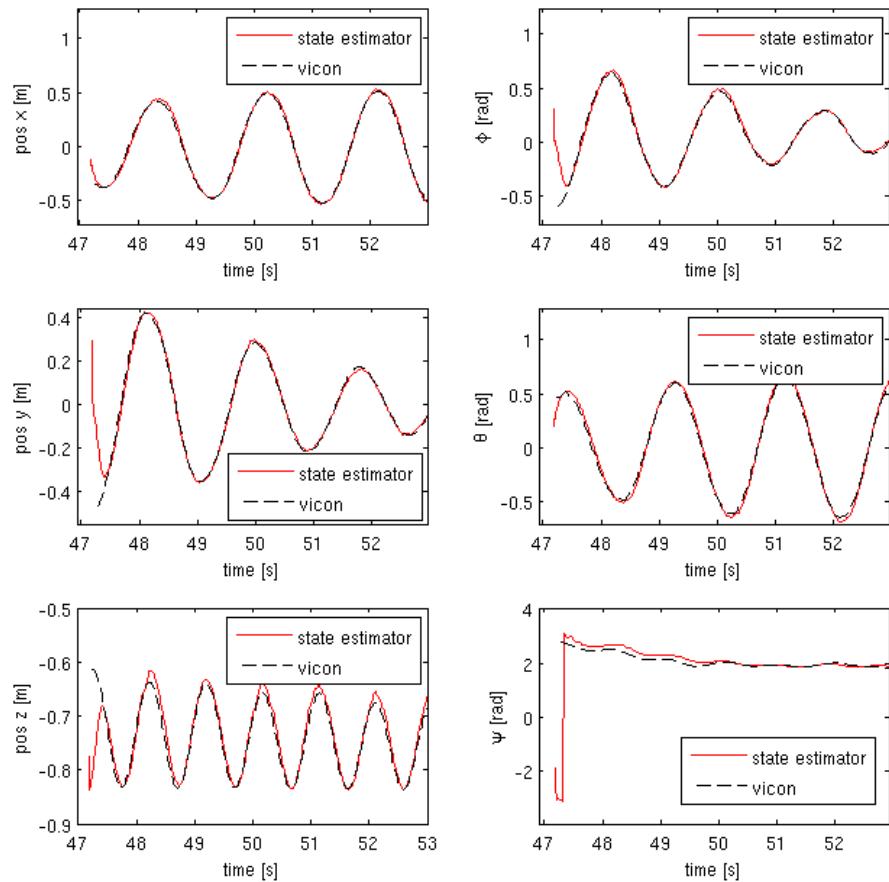


Figure 4.9: GPS signal with a SNR of -25 dB.

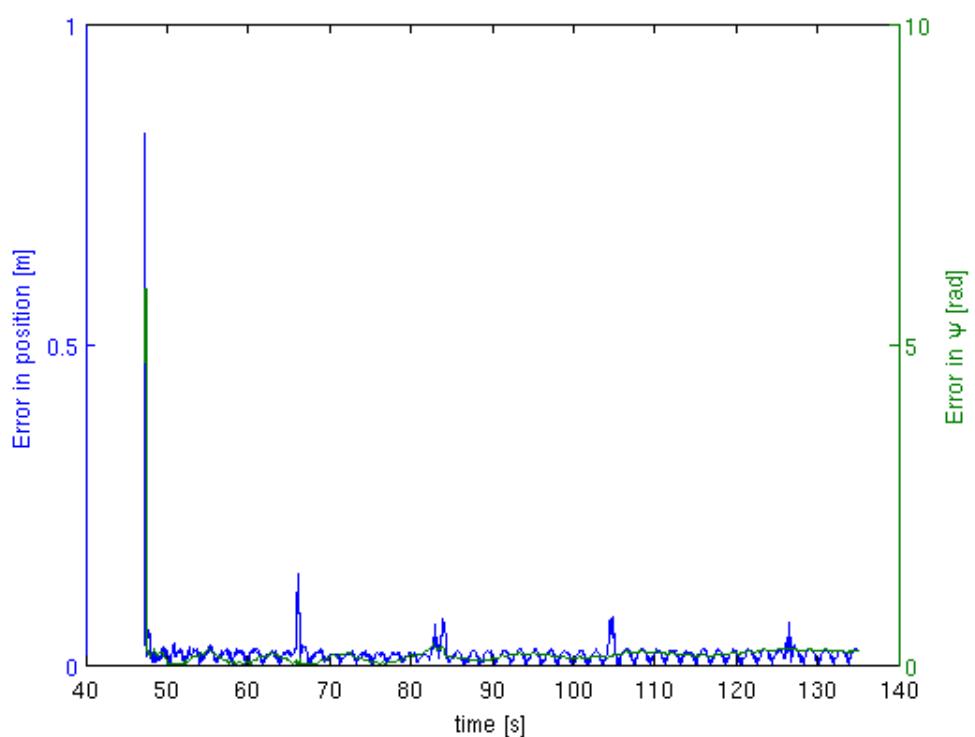


Figure 4.10: Error in position and orientation using a GPS signal with a SNR of -5 dB.

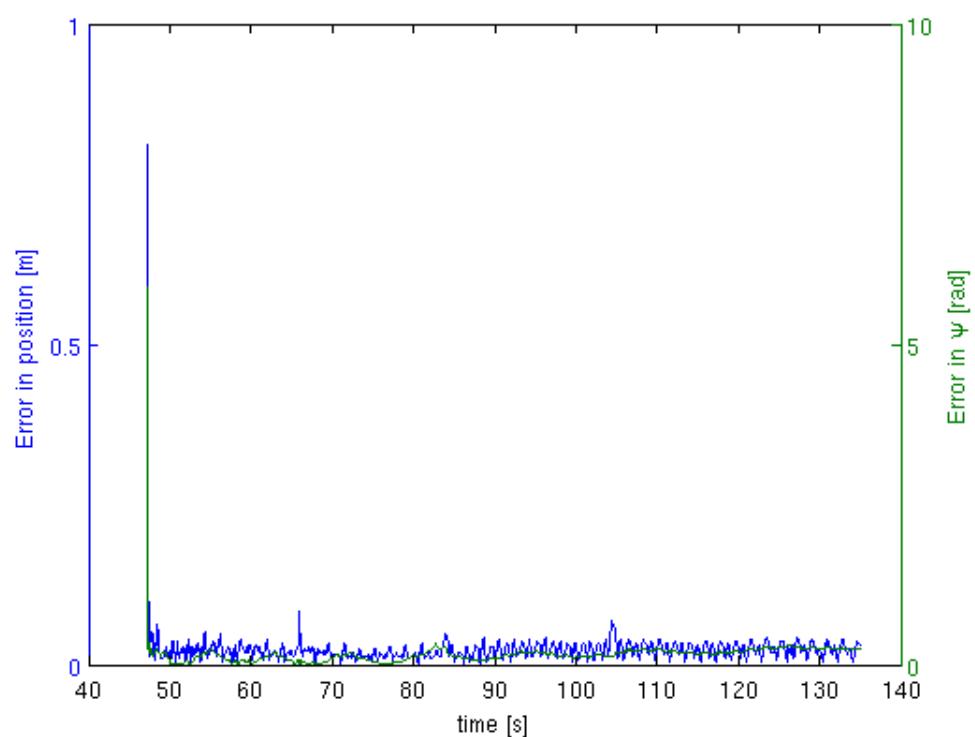


Figure 4.11: Error in position and orientation using a GPS signal with a SNR of -25 dB.

4.2.3 GPS outage

As explained earlier in this chapter, the performance of a estimator without having a GPS signal is an important criterion. For this purpose, the algorithm is run without any position or velocity measurements in the input. In figure 4.12 a detail of the plot is shown. As can be seen, the estimator is able to very accurately estimate

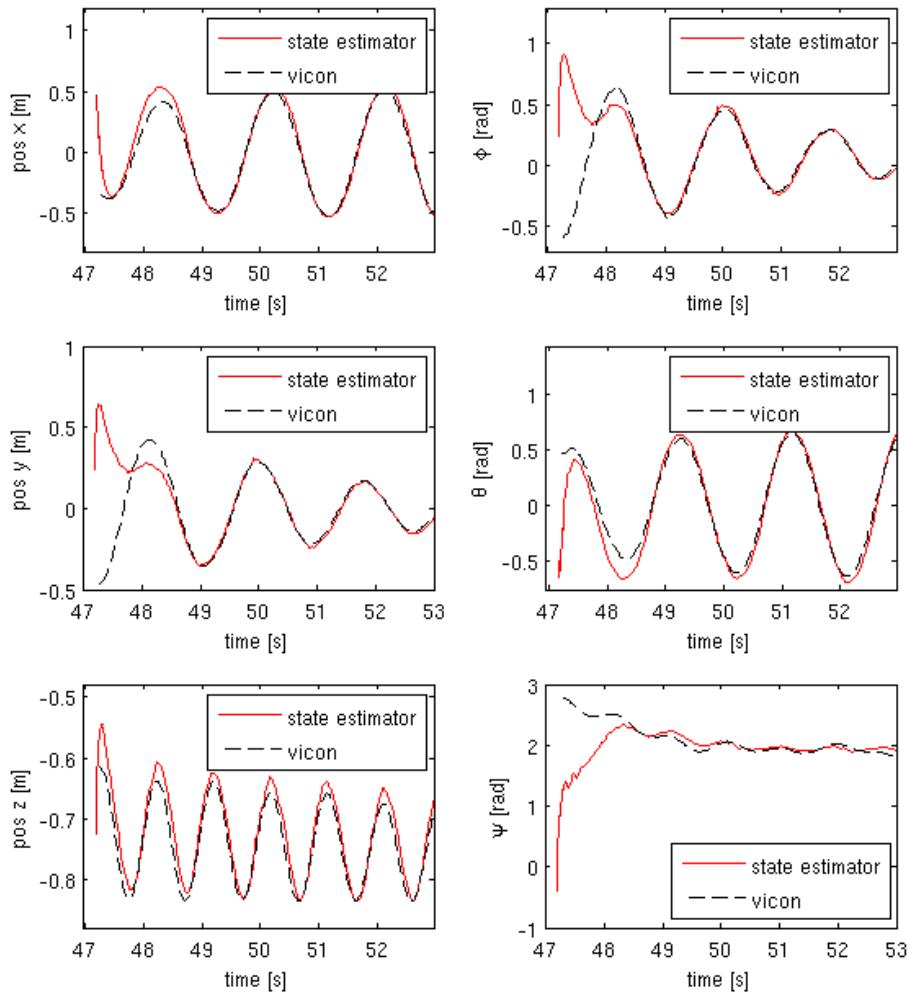


Figure 4.12: Estimation of the states without GPS signal.

the state of the system only using the IMU measurements. In this case the MTi-G data was used. Again can the error of the estimated position compared to the true position, as well as the difference between the estimated angle of Ψ and the true

angle can be calculated and plotted. This is done in Figure 4.13. The mean value of this error is calculated in the same way as the error between the different noise levels in chapter 4.2.2. The fact that no GPS data is available has an influence on the time

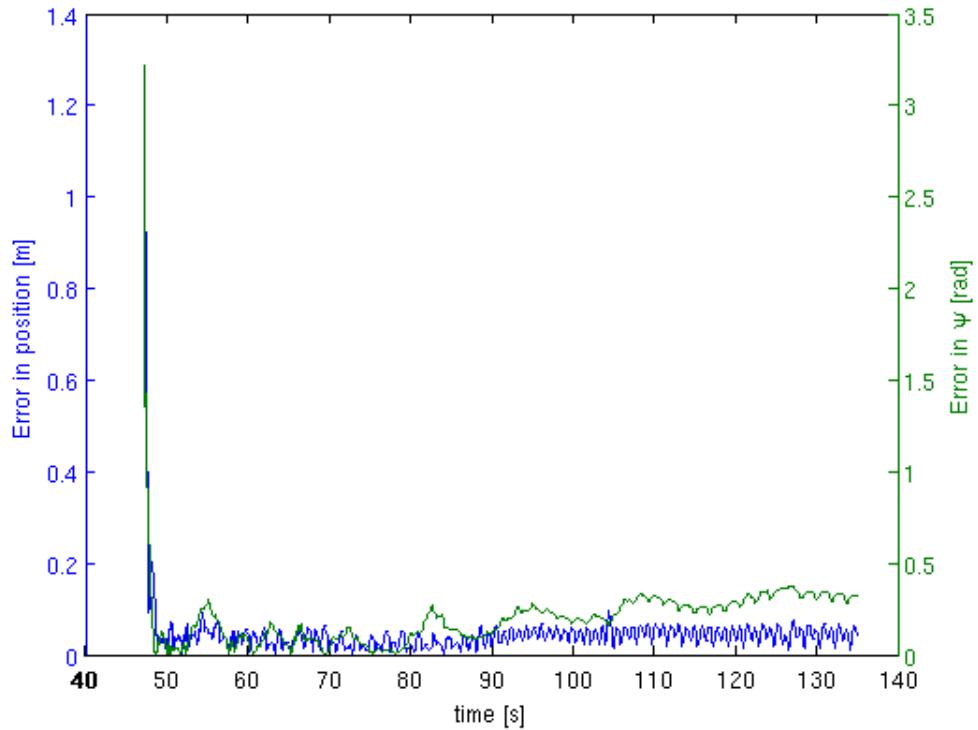


Figure 4.13: Error in position and orientation only using IMU measurements and no GPS data.

it takes the estimator to settle. Without the accurate GPS signal, it takes between one and two seconds while the accurate information on position and velocity drag the estimator to its correct state in a few tenths of a second. Furthermore, the overall error levels are slightly higher without the GPS data (0.0393 m compared to 0.0162 m for the position and 0.1947 rad compared to 0.144 rad for Ψ).

This behaviour is also the big advantage of the pendulum model compared to the free mass model. While the first does not dramatically change its performance, the latter is very much dependent on a GPS signal. The "free mass model"-estimator was used on a GPS signal that contained a simulated GPS outage from $t = 75\text{ s}$ to $t = 78\text{ s}$. During that time, the position and orientation both drift away from the true values. However, upon reacquiring a GPS fix, the estimator returns very fast to its correct state, as can be seen in Figure 4.14.

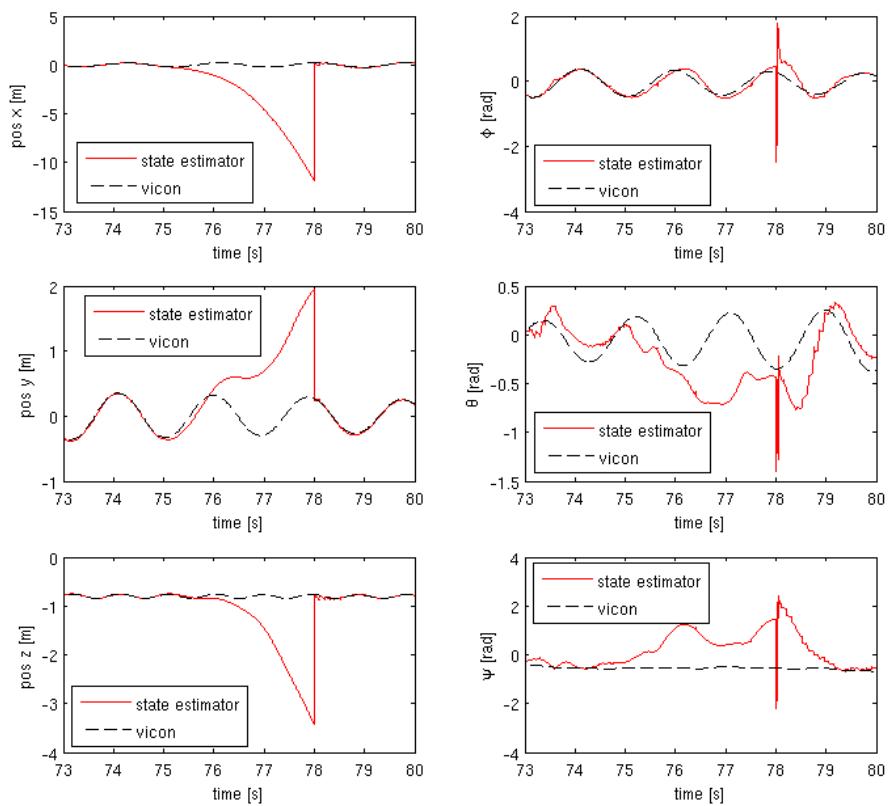


Figure 4.14: "Free mass model" estimator during a GPS outage from $t = 75\text{ s}$ to $t = 78\text{ s}$.

4.2.4 The MTi-G unit versus the PX4 unit

In this section the influence of the different IMU's on the estimator is presented. The above test of the estimator without having a GPS signal is carried out again. But this time using the PX4 IMU data instead of the MTi-G data. Quite surprisingly, the performance of the estimator degrades significantly compared to the MTi-G. The algorithm takes a lot longer to settle (between five and ten seconds) and the overall error is higher ($0.0743m$ compared to $0.0393m$ for the position and $0.241rad$ compared to $0.1947rad$ for Ψ). The plot of the errors can be found in figure 4.15. In figure all mean errors are summarized.

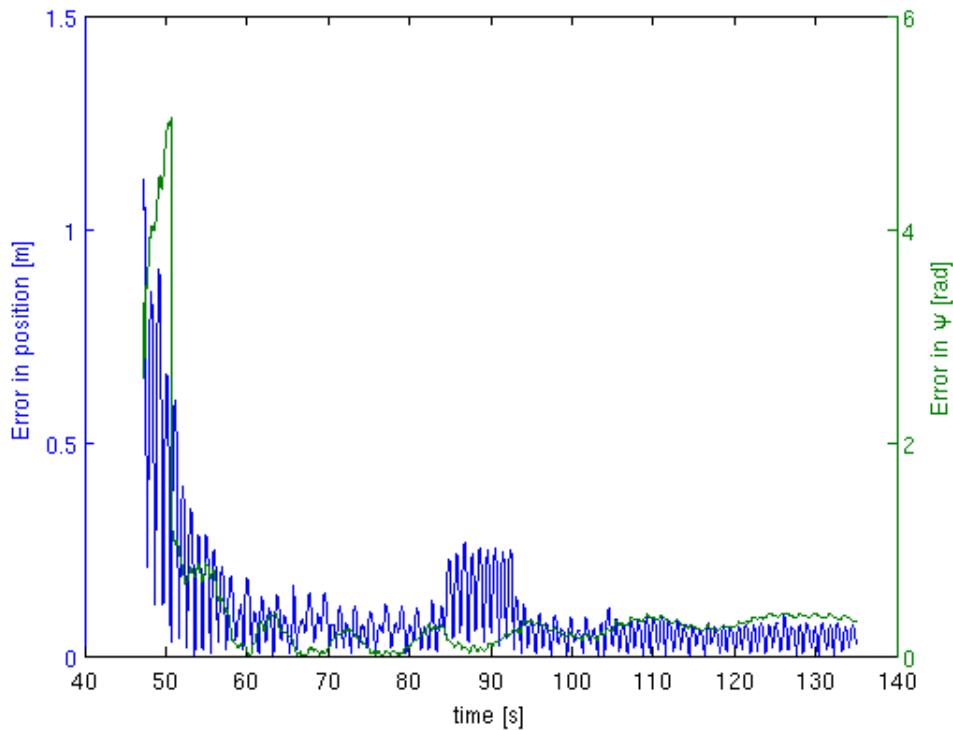


Figure 4.15: Error of the estimation of the states without GPS using only the PX4 IMU.

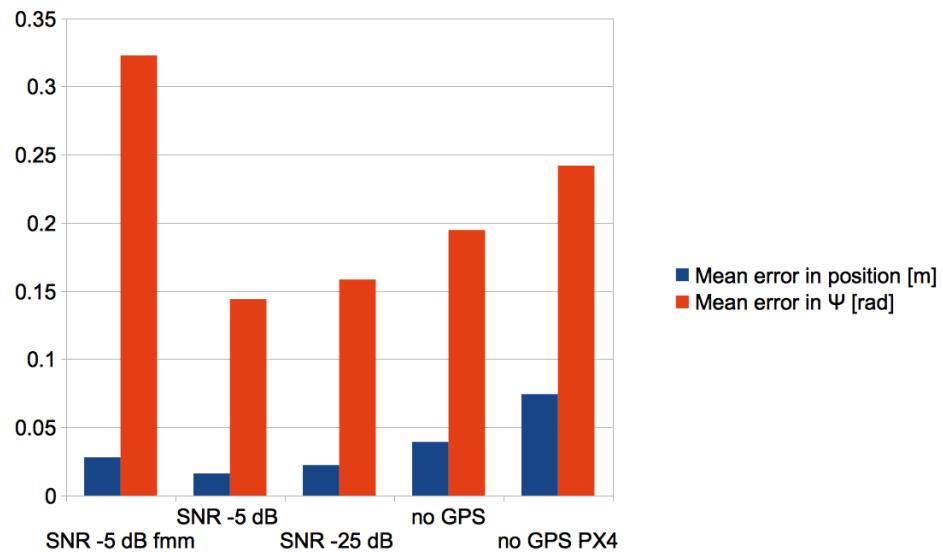


Figure 4.16: All mean errors. Where fmm stands for free mass model which is used only in the first error calculation. All errors were calculated using the MTi-G IMU. Only the last one is using the PX4 Unit.

Chapter 5

Conclusion and Outlook

5.1 Sensor Selection

The centrifuge test was not able to give a meaningful conclusion about the quality of the different sensors. There seems to be a correlation between the acceleration acting on the sensors and their accuracy, but this could also be due to more vibrations at higher speeds. However, the test showed some general limitations in the range of the different IMU's. The PX4 was not able to cover the whole acceleration range and the MTi-G could not measure rates of turn larger than 403 deg/s. In order to be able to better compare these different units, a test would have to be done where a precise ground truth would be available. Furthermore, the MTi-G should be set to output the calibrated measurements, as this spares the conversion to conventional units.

5.2 State Estimation

Based on the Vicon test, we were able to show an improvement of accuracy in the state estimation when using the "pendulum model"-estimator. Especially in the absence of a GPS signal, this algorithm was still able to estimate the true position while the conventional Kalman Filter drifted off during such periods. This improvement is on one hand due to the reduction in degrees of freedom, giving more redundant measurements per state, and on the other hand due to the accurate prediction of the future states by using the exact knowledge of the forces acting on the body. These promising results make us confident, that position estimation of a kite with an Extended Kalman Filter could be improved by including as much knowledge of the system as possible. The complete independence of the GPS measurements in the pendulum case, suggests that also in the more complex system of the kite, the dependency on an accurate GPS signal could be reduced.

For future investigations in this subject, we suggest to also investigate the use of models with 8 or more degrees of freedom, allowing for example the radius to vary. Furthermore, the ideal combination of Covariance matrices would have to be

found either by finding the exact uncertainties in the sensors and the prediction, or by minimizing the output error. Considering the application of the algorithm to kites, it would be interesting to find out how sensitive the filter is to errors in the prediction of the forces acting on the body. If an very accurate propagation model, as the one of the pendulum that was used in this work, is needed, the algorithms application to a real kite system could prove to be difficult since these systems are a lot more complex. However, investigating the use of additional sensors, like for example line angle sensors, the possibilities of an exact estimation of the kites position and orientation without relying on a GPS signal could be determined.

Bibliography

- [1] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, Department of Computer Science, University of North Carolina at Chapel Hill, NC 27599-3175, 1997. Available from ens-cachan.fr.
- [2] Miles L. Loyd. Crosswind kite power. *J. Energy*, 4(3):106–111, 1980.
- [3] M. Canale, L. Fagiano, M. Ippolito, and M. Milanese. Control of tethered airfoils for a new class of wind energy generator. In *Decision and Control, 2006 45th IEEE Conference on*, pages 4020 –4026, dec. 2006.
- [4] R. Schmehl. Large-scale power generation with kites. *J. Soc. Aerospace Engineering Students*, 1:21–22, 2012.
- [5] M. Vukov S. Gros J. Swevers K. Geelen, H. Ahmad and M. Diehl. An experimental test set-up for launch/recovery of an airborne wind energy (awe) system. In *In Proceedings of the 2012 American Control Conference, Montreal, Canada*, 2012.
- [6] Michael Erhard and Hans Strauch. Control of towing kites for seagoing vessels. *CoRR*, abs/1202.3641, 2012.
- [7] Swiss kite power. <http://www.swisskitepower.ch>. accessed 06-2012.
- [8] András Héjj. Kalman-filter based position and attitude estimation algorithms for an inertial measurement unit.
- [9] Ushanthan Jeyabalan. The ring on the hook.
- [10] S. Sukkarieh, E.M. Nebot, and H.F. Durrant-Whyte. A high integrity imu/gps navigation loop for autonomous land vehicle applications. *Robotics and Automation, IEEE Transactions on*, 15(3):572 –578, jun 1999.
- [11] Jong-Hyuk Kim, Salah Sukkarieh, and Stuart Wishart. Real-time navigation, guidance, and control of a uav using low-cost sensors. In ShinâÄZichi Yuta, Hajima Asama, Erwin Prassler, Takashi Tsubouchi, and Sebastian Thrun, editors, *Field and Service Robotics*, volume 24 of *Springer Tracts in Advanced Robotics*, pages 299–309. Springer Berlin / Heidelberg, 2006. 10.1007/10991459₂9.

- [12] D. Smith and S. Singh. Approaches to multisensor data fusion in target tracking: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 18(12):1696 –1710, dec. 2006.
- [13] Angus P. Andrews Mohinder S. Grewal, Lawrence R. Weill. *Global positioning systems, inertial navigation and integration*. Hoboken, NJ : Wiley, 2007.
- [14] Chapter 6 design and performance analysis of kalman filters. In Peter S. Maybeck, editor, *Stochastic models, estimation, and control*, volume 141, Part 1 of *Mathematics in Science and Engineering*, pages 289 – 367. Elsevier, 1979.
- [15] L. Debnath. *Nonlinear Partial Differential Equations for Scientists and Engineers*. Birkhäuser, 2005.
- [16] W. Kutta. Beitrag zur näherungweisen Integration totaler Differentialgleichungen. *Z. Math. Phys.*, 46:435–453, 1901.

Appendix A

Data Sheets of the PX4 sensors

A.1 Data Sheet of the Accelerometer

BMA180

Digital, triaxial acceleration sensor

Bosch Sensortec



BOSCH
Invented for life

General description

The BMA180 is an ultra high performance digital triaxial acceleration sensor, aiming for low power consumer market applications.

The BMA180 allows high accurate measurement of accelerations in 3 perpendicular axes and thus senses tilt, motion, shock and vibration in cell phones, handhelds, computer peripherals, man-machine interfaces, virtual reality features and game controllers.

BMA180 target applications

- ▶ Navigation (INS/Dead Reckoning)
- ▶ High accurate tilt detection
(e.g. tilt compensated compass)
- ▶ Pointing and menu scrolling
- ▶ Display profile switching (portrait/landscape)
- ▶ Gaming
- ▶ Drop detection for warranty logging
- ▶ Shock detection
- ▶ Step-counting

Sensor operation

The BMA180 represents a new generation of digital acceleration sensors with a unique performance and feature set within 3 mm x 3 mm x 0.9 mm standard LGA package.

Key features BMA180

- ▶ All parameters and features user programmable
- ▶ 7 user programmable g-ranges and 10 selectable bandwidth settings
- ▶ Low-power consumption
- ▶ Ultra-low-power self-wake-up mode
- ▶ User programmable interrupt engine
- ▶ 0g offset regulation and in-field offset re-calibration with ultra-high accuracy
- ▶ SPI (4-wire)/I²C interface
- ▶ RoHS compliant, halogen-free

The BMA180 is highly configurable in order to give the designer full flexibility when integrating the sensor into his system. All features can be set by software via the digital interface. Here the user can choose between an I²C and an SPI (4-wire) interface mode.

Technical data		BMA180
Sensitivity axes		x/y/z
Measurement range (switchable via SPI/I ² C)		±1g, ±1.5g, ±2g, ±3g, ±4g, ±8g, ±16g
Sensitivity (calibrated)		1g: 8192LSB/g 1.5g: 5460LSB/g 2g: 4096LSB/g 3g: 2730LSB/g 4g: 2048LSB/g 8g: 1024LSB/g 16g: 512LSB/g
Resolution		14bit ⇒ 0.244mg (±2g range) (switchable 12 bit option)
Nonlinearity		±0.15% FS (±2g range)
Zero-g offset (ex-factory)		±15mg
Zero-g offset (after offset fine tuning)		±5mg
Zero-g offset temperature drift		±0.5mg/K
Noise density		150µg/√Hz
Bandwidth (switchable via SPI/I ² C)	low pass	10Hz ... 1200Hz
	high pass	1Hz
	band pass	0.2 ... 300Hz
Digital input/output		SPI & I ² C, interrupt pin
Supply voltages	V _{DD}	1.62 ... 3.6V
	V _{DDIO}	1.20 ... 3.6V
Current consumption		650µA
Temperature range		-40°C ... +85°C

A.2 Data Sheet of the Gyroscope

2 Mechanical and electrical specifications

2.1 Mechanical characteristics

@ Vdd = 3.0 V, T = 25 °C unless otherwise noted.

Table 3. Mechanical characteristics⁽¹⁾

Symbol	Parameter	Test condition	Min.	Typ. ⁽²⁾	Max.	Unit
FS	Measurement range	User-selectable		±250		dps
				±500		
				±2000		
So	Sensitivity	FS = 250 dps		8.75		mdps/digit
		FS = 500 dps		17.50		
		FS = 2000 dps		70		
SoDr	Sensitivity change vs. temperature	From -40 °C to +85 °C		±2		%
DVoff	Digital zero-rate level	FS = 250 dps		±10		dps
		FS = 500 dps		±15		
		FS = 2000 dps		±75		
OffDr	Zero-rate level change vs. temperature	FS = 250 dps		±0.03		dps/°C
		FS = 2000 dps		±0.04		dps/°C
NL	Non linearity	Best fit straight line		0.2		% FS
Rn	Rate noise density			0.03		dps/ (Hz)
ODR	Digital output data rate			95/190/ 380/760		Hz
Top	Operating temperature range		-40		+85	°C

1. The product is factory calibrated at 3.0 V. The operational power supply range is specified in [Table 4](#).

2. Typical specifications are not guaranteed.

A.3 Data Sheet of the Magnetometer

HMC5883L

SPECIFICATIONS (* Tested at 25°C except stated otherwise.)

Characteristics	Conditions*	Min	Typ	Max	Units
Power Supply					
Supply Voltage	VDD Referenced to AGND VDDIO Referenced to DGND	2.16 1.71	2.5 1.8	3.6 VDD+0.1	Volts Volts
Average Current Draw	Idle Mode Measurement Mode (7.5 Hz ODR; No measurement average, MA1:MA0 = 00) VDD = 2.5V, VDDIO = 1.8V (Dual Supply) VDD = VDDIO = 2.5V (Single Supply)	- -	2 100	- -	µA µA
Performance					
Field Range	Full scale (FS)	-8		+8	gauss
Mag Dynamic Range	3-bit gain control	±1		±8	gauss
Sensitivity (Gain)	VDD=3.0V, GN=0 to 7, 12-bit ADC	230		1370	LSb/gauss
Digital Resolution	VDD=3.0V, GN=0 to 7, 1-LSb, 12-bit ADC	0.73		4.35	milli-gauss
Noise Floor (Field Resolution)	VDD=3.0V, GN=0, No measurement average, Standard Deviation 100 samples (See typical performance graphs below)		2		milli-gauss
Linearity	±2.0 gauss input range			0.1	±% FS
Hysteresis	±2.0 gauss input range		±25		ppm
Cross-Axis Sensitivity	Test Conditions: Cross field = 0.5 gauss, Happlied = ±3 gauss		±0.2%		%FS/gauss
Output Rate (ODR)	Continuous Measurement Mode Single Measurement Mode	0.75		75 160	Hz Hz
Measurement Period	From receiving command to data ready		6		ms
Turn-on Time	Ready for I2C commands Analog Circuit Ready for Measurements		200 50		µs ms
Gain Tolerance	All gain/dynamic range settings		±5		%
I ² C Address	8-bit read address 8-bit write address		0x3D 0x3C		hex hex
I ² C Rate	Controlled by I ² C Master			400	kHz
I ² C Hysteresis	Hysteresis of Schmitt trigger inputs on SCL and SDA - Fall (VDDIO=1.8V) Rise (VDDIO=1.8V)		0.2*VDDIO 0.8*VDDIO		Volts Volts
Self Test	X & Y Axes Z Axis		±1.16 ±1.08		gauss
	X & Y & Z Axes (GN=5) Positive Bias X & Y & Z Axes (GN=5) Negative Bias	243 -575		575 -243	LSb
Sensitivity Tempco	T _A = -40 to 125°C, Uncompensated Output		-0.3		%/°C

General

ESD Voltage	Human Body Model (all pins) Charged Device Model (all pins)			2000 750	Volts
Operating Temperature	Ambient	-30		85	°C
Storage Temperature	Ambient, unbiased	-40		125	°C

Appendix B

Data Sheets of the MTi-G

MTi-G

MINIATURE AHRS WITH INTEGRATED GPS



XSENS

The MTi-G is a miniature size and low weight 6DOF Attitude and Heading Reference System (AHRS). The MTi-G contains accelerometers, gyroscopes, magnetometers in 3D, an integrated GPS receiver, a static pressure sensor and temperature sensor. Its internal low-power signal processor provides real time and drift-free 3D orientation as well as calibrated 3D acceleration, 3D rate of turn, 3D earth-magnetic field, 3D position and 3D velocity data.

The MTi-G is an excellent measurement unit for stabilization and control of air and ground objects, even during situations of long term accelerations.

Highlights

- Real-time computed GPS-enhanced attitude/heading and inertial enhanced position/velocity data
- GPS integration overcomes typical IMU challenges
- Integrated AHRS, GPS and static pressure sensor
- On board DSP, running sensor fusion algorithm
- High update rate (120 Hz), inertial data at max 512 Hz
- Individually calibrated for temperature, 3D misalignment and sensor cross-sensitivity
- UTC referenced output

Compact design

- Compact and robust design
- Easy integration in any system or application (OEM)
- Low weight, ultra-low power consumption

High performance

The MTi-G is a combination of a MEMS IMU, GPS and barometer. Yet, the MTi-G is more than just a sensor assembly. The IMU, GPS and barometric information is blended together in Xsens' sensor fusion algorithm to estimate the most accurate orientation and position possible. Because of this fusion, the output is more accurate than the output from the IMU or GPS receiver only. For example, the MTi-G copes with transient accelerations; a typical error source for any AHRS using the gravity as its reference estimating roll and pitch. The loose coupling works both sides: double-integrating the accelerometers for short periods, the MTi-G is able to calculate position and velocity even during short GPS outages. There are several more corrections realized to aid the IMU functionality and to enhance the GPS measurements.

User friendliness

The MTi-G is a sensor which can be used in a wide range of applications. Because of the specific requirements for all these applications, the MTi-G uses different filter settings and constraints, implemented in scenarios. Among others, there are scenarios for use in automotive and aerospace applications.

Output

- 3D orientation (360°)
- 3D position and velocity (aided and unaided by inertial sensors)
- 3D acceleration, 3D rate of turn
- 3D magnetic field



TYPICAL USAGE APPLICATIONS



Automotive

- Vehicle dynamics analysis
- Racing cars and motorbikes
- Performance testing

Full access to valuable data is available for engineers working in any level of sports car or motorbike competitions. The MTi-G outputs are suitable to test and analyze the dynamic behavior of e.g. automobiles. Non-holonomic constraints can be used to further enhance accuracy. The roll and pitch remains accurate even during long term accelerations typical for this application.



Marine

- In-competition optimization of racing yachts
- State estimation of leisure yachts
- Backup system for high-grade GPS systems

For racing yachts, the MTi-G provides orientations of several parts of the ship, such as the roll angle of the hull or the movement of the mast. This allows a sailing team to refine the ship's performance during a race, or include the data in the autopilot. Another major application is commercial shipping. Measuring roll and pitch as well as heading is important for container ships, cargo ships and surveying vessels. Installing the MTi-G together with high grade GPS systems is a logical choice to enhance accuracy and to reduce costs, especially in situations with limited GPS reception.



Unmanned ground vehicles and robotics

- Autonomous control for driving and walking robots
- Military and civil ground vehicles
- Camera/LIDAR stabilization and correction

The MTi-G is an excellent sensor for driving and walking robots. The main functionality of the MTi-G in robotics applications is attitude control, even under dynamic conditions. Accurate position and orientation makes autonomous navigation possible for Unmanned Ground Vehicles (UGV's), even on rough terrain or during short GPS outages. The MTi-G has been used in DARPA Grand Challenges for these purposes. These applications are typically out of reach for conventional MEMS IMU'S.



Aerospace

- Autonomous attitude and navigation control
- Dynamics of (aerobatics) planes
- Camera/LIDAR stabilization and correction
- Head-up display

The MTi-G is the ideal choice for control and stabilization for any type of small to medium sized fixed-wing and rotary-wing aircraft. Because of the low latency, autonomous control can be designed in a simple and robust manner. The MTi-G provides a wide variety of dynamic data, suitable for dynamics analysis of (un)manned airplanes. An easy software/hardware interface makes data processing possible post-flight or even real-time. The MTi-G is easy to install and computes all the data required for e.g. an accurate artificial horizon or a digital map.

MTi-G DEVELOPMENT KIT

The MTi-G DK contains the following:

- MTi-G (any configuration)
- Antenna
- USB cabling
- MT Software Development Kit (see below)
- Hardcopy documentation
- Optional: serial cabling
- Suitcase

MT Software Development Kit (MT SDK)

The MT SDK is an extensive set of tools for every level of interfacing, which allows configuring the MTi-G to the user needs, reading out and storing data and (re-)processing MTi-G data previously recorded. It also allows the user to extend own user source code with the MTi-G communication, using commands and code examples provided.

The MT SDK contains:

MT Manager Software

A specially developed, easy-to-use graphical user interface with possibilities to configure Xsens' sensors, reading out, store and show data in real-time charts and visualizations.

MT COM-object API and DLL API for Windows

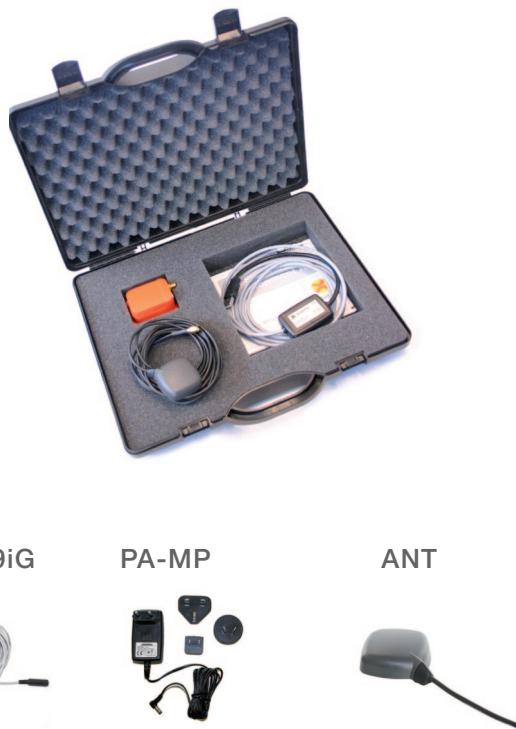
Integrating the MTi-G in Windows programs, such as Matlab, C++ and Excel is made easy with the MT COM-object API and the DLL API. User-modifiable example code for programs Matlab, C++ and Excel (VBA) is included.

C++ Class and binary communication for any (RT)OS

A C++ class is available for users who want to use the MTi-G on a binary level. Direct communication without using the C++ class is possible, following the fully documented communication protocol.

Magnetic Field Mapper plug-in

The Magnetic Field Mapper plug-in enables compensation for hard and soft iron effects.



ACCESSORIES

Cable options

CA-USB2G



USB cable

CA-SERi-2,5



Serial cable
RS232 + pigtail

CA-DB9iG



RS232, DB9
power

PA-MP



Power adapter
(for CA-DB9iG only)

ANT



Antenna



ABOUT XSENS TECHNOLOGIES

Xsens is a leading global supplier of 3D motion tracking products based upon miniature MEMS inertial sensor technology.

Since its inception in 2000, several thousands of motion sensors and motion capture solutions have successfully been deployed in areas such as 3D character animation, rehabilitation and sports science, and robot and camera stabilization. Customers include Electronic Arts, Sony Pictures Imageworks, INAIL Prostheses Centre, Daimler, Saab Underwater Systems, Kongsberg Defence & Aerospace and many other companies and institutes throughout the world.

Xsens' research department has created unique intellectual property in the field of multi-sensor data fusion algorithms, combining inertial sensors with aiding technologies such as GPS and RF positioning and biomechanical modeling. The company and its products have received several awards, amongst which four consecutive entries in Deloitte's ranking of fastest growing technology companies in Europe.

Xsens is headquartered in Enschede, The Netherlands and has a subsidiary, Xsens North America Inc. in Los Angeles, California, US.



XSENS

Xsens Technologies B.V.

phone +31 88 97367 00
fax +31 88 97367 01
e-mail info@xsens.com
internet www.xsens.com

Xsens North America Inc.

phone +1-866-973-6787 (toll-free)
fax +1-866-973-6701
e-mail info@xsens.com
internet www.xsens.com

Appendix C

Data Sheets of the x-IMU sensors

C.1 Data Sheet of the Gyroscope

C.2 Data Sheet of the Accelerometer and Magnetometer

