ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/251783951

Integration of a GPS aided Strapdown Inertial Navigation System for Land Vehicles

Article · March 2006

CITATIONS
22

READS
537

1 author:

Adrian Schumacher
Swisscom

5 PUBLICATIONS **37** CITATIONS

SEE PROFILE

All content following this page was uploaded by Adrian Schumacher on 15 January 2016.

The user has requested enhancement of the downloaded file.

Integration of a GPS aided Strapdown Inertial Navigation System for Land Vehicles

ADRIAN SCHUMACHER

KTH Electrical Engineering

Master of Science Thesis
Stockholm, Sweden March 2006

XR-EE-SB 2006:006

Abstract

The estimation accuracy of a low-cost inertial navigation system (INS) is limited by the accuracy of the used sensors and the imperfect mathematical modeling of the error sources. By fusing the INS data with GPS data, the errors can be bounded and the accuracy increases considerably. In this project, a low-cost in-house constructed inertial measurement unit (IMU) and an off-the-shelf GPS receiver are used for the data acquisition. The measurements are integrated with a loosely coupled GPS aided INS approach. For the assessment of the results, one data set with real data obtained from a field test is available.

The tuning of the covariance matrices is a delicate adjustment and does not always provide convergence. Values for acceptable results could be found and two implementations of inertial navigation systems are compared. The use of nonholonomic constraints showed a dramatic increase in the accuracy.

An analysis of the importance and influence of different IMU sensor errors provides a foundation for the modeling and inclusion of further error states in the extended Kalman filter.

Acknowledgment

This Master Thesis project was carried out at the Signal Processing Laboratory, School of Electrical Engineering, at the Royal Institute of Technology. I would like to thank my examiner Peter Händel for allowing me to do this thesis, and his confidence and support.

Furthermore, I want to thank my adviser Isaac Skog for his help, support, and all the discussions throughout this thesis project. It was a pleasure to work with him.

Thanks also go to René Widmer from the School of Engineering and Information Technology, at Berne University of Applied Sciences for the help with the GPS delay measurement, and to my friends and fellow students at KTH for the enjoyable and nice time here.

Contents

1 Introduction **1**
- 1.1 Background and Objective . . . 1
- 1.2 System Overview . . . 2
- 1.3 Thesis Outline . . . 3

2 Inertial Navigation **5**
- 2.1 Coordinate Frames . . . 6
- 2.2 Coordinate Transformations . . . 7
- 2.3 Navigation Equations . . . 9
- 2.4 Error Dynamics . . . 10
- 2.5 Global Positioning System . . . 13
  - 2.5.1 Error Sources . . . 14
  - 2.5.2 Dilution of Precision . . . 17
  - 2.5.3 Differential GPS . . . 18

3 GPS/INS Integration **19**
- 3.1 Coupling Approaches . . . 19
- 3.2 Discretization . . . 20
- 3.3 The Extended Kalman Filter . . . 22
- 3.4 Observability Analysis . . . 23
- 3.5 Numerical Considerations for the Covariance Matrix $\mathbf{P}$ . . . 24
- 3.6 Filter Tuning . . . 24
- 3.7 Lever-Arm Correction . . . 26
- 3.8 Calibration . . . 26
  - 3.8.1 Accelerometers . . . 27
  - 3.8.2 Gyros . . . 28
- 3.9 Accuracy Improvement . . . 28
  - 3.9.1 Nonholonomic Constraints . . . 28
  - 3.9.2 Field Calibration . . . 29
  - 3.9.3 Zero-Velocity Update . . . 29

4 TRÖGE Platform for Inertial Navigation **31**
- 4.1 Hardware . . . 31
  - 4.1.1 Testbed . . . 32
  - 4.1.2 Inertial Measurement Unit . . . 32
  - 4.1.3 GPS Receiver . . . 33
- 4.2 Software Implementation . . . 34
  - 4.2.1 Data Acquisition . . . 34

4.2.2 Navigation System Implementation 35

5 Results and Analysis 39
5.1 Field Test 39
5.1.1 Equipment 39
5.1.2 Trajectory 40
5.1.3 Data Pre-Processing 41
5.2 Experimental Results 42
5.2.1 Basic GPS aided INS 42
5.2.2 GPS aided INS with Nonholonomic Constraints 44
5.3 Error Analysis 46
5.3.1 Accelerometers 46
5.3.2 Gyros 47
5.3.3 GPS 48
5.3.4 Summary 49

6 Conclusions and Further Work 51
6.1 Conclusions 51
6.2 Recommendations and Further Work 52

Bibliography 55

Notations

Matrices are denoted in bold upper case letters

Vectors are denoted in bold lower case letters

$(\cdot)^b$	variable in the *body*-frame
$(\cdot)^e$	variable in the *Earth-Centered Earth-Fixed*-frame
$(\cdot)^i$	variable in the *inertial*-frame
$(\cdot)^t$	variable in the *tangent*-frame
$\mathbf{R}_a^b$	discrete cosine matrix to transform from a- to b-frame
$(\cdot)^T$	matrix transpose
$(\cdot)^{-1}$	matrix inverse
$\mathbf{I}_n$	identity matrix of order n
$\mathbf{0}_n$	zero matrix of order n
x_k	value at instant k: $x(k)$
$\dot{x}$	first order derivative of x
δx	error of x
$\hat{\cdot}$	estimated value
$\tilde{\cdot}$	measured value
$(\cdot)^-$	predicted state
$E\{\}$	expectation
$\lvert\cdot\rvert$	magnitude
$\lVert\cdot\rVert$	norm

Chapter 1

Introduction

1.1 Background and Objective

To know the exact position of an object on the Earth can be important. For outdoor applications e.g. for navigation or geodetic measurements, there exist *Global Navigation Satellite Systems* (GNSS). With expensive equipment, it is possible to determine one's position on the Earth's surface as accurate as a few centimeters. However, consumer grade receivers have an accuracy in the order of 15-100 meters. Furthermore, it implies a line-of-sight connection to the satellites. In urban areas with high buildings or in forests, the quality of the position estimates degrades or even leads to a signal blocking (definitely in tunnels). Another drawback is the slow update rate of usually once a second. In applications such as car racing, driver training, driving exercises, or performance measurements, a more frequent position estimation is required in order to record an accurate trajectory. In addition to that, more information such as the velocity and attitude of the vehicle is desired. *Inertial Navigation Systems* (INS) can provide the estimates of the desired information. Based on accelerometers and gyroscopes (gyros), and Newton's laws of motion, it is possible to determine the position, velocity, and attitude if the initial values are known. The sensors can be sampled with a higher frequency (typically 100 Hz for consumer grade devices). It is a so-called self-contained system. The inertial sensors are classified as dead reckoning sensors since the current evaluation of the state is formed by the relative increment from the previous known state. Due to this integration, errors such as low frequency noise and sensor biases are accumulated, leading to a drift in the position with an unbounded error growth. A combination of both systems, a GNSS (e.g. the *Global Positioning System* GPS) with the long-term accuracy and an INS with the short-term accuracy but high update rate, can provide a good accuracy with frequent updates. The GNSS makes the INS error-bounded and on the contrary, the INS can be used to e.g. identify and correct GNSS carrier phase cycle slips.

In the past decades, the GNSS and INS integration has been extensively studied [1, 2, 3] and successfully used in practice. Concerning the INS, there are several quality categories. Navigation grade INS have a position error of around 30 m/min but cost over US$ 100'000. Consumer grade INS in turn, have a position error of up to 3000 m/min and cost under US$ 1'000. Recently, the advances in *Micro Electro Mechanical Systems* (MEMS) led to very inexpensive sensors, compared to navigation or tactical grade sensors. However, their accuracy, bias, and noise characteristics are of an inferior quality than in the other categories. It is attempted to improve the accuracy with advanced signal processing.

In a previous thesis [4], a model for a GPS aided INS was developed and simulated. Furthermore, a low-cost *Inertial Measurement Unit* (IMU) was developed. The objective of this thesis is to include a calibration method for the inertial sensors, the development of a software to communicate with the IMU and store the sampled data, the integration and test of the whole system using real-world data, the Kalman filter tuning, and finally the implementation of the algorithms in C++ for the use in a real-time system, e.g. on the TRÖGE testbed (see 4.1.1).

1.2 System Overview

As described in the previous section, a combination of an INS with GPS is advantageous. It provides position estimates at a higher update rate and a smaller position error than a stand-alone GPS receiver. Usually, the INS acts as the main navigation system since it is self contained, and if GPS position estimates are available, they are used to correct the errors in the INS. Therefore the name GPS aided INS.

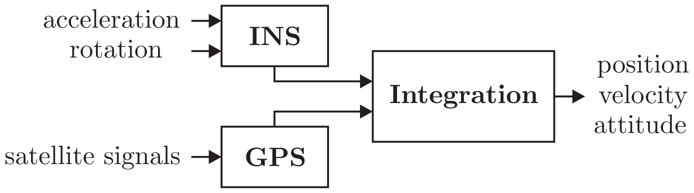


Figure 1.1: Block diagram of the loosely coupled GPS aided INS system.

Several coupling methods are described and compared in Section 3.1. For this thesis, the loosely coupled approach is used. Figure 1.1 shows a block diagram of the system. Before the data from the different sensors can be fused, they need to be transformed into a common coordinate frame, which in this case is the Earth-Centered Earth-Fixed (ECEF) frame. This coordinate system rotates with the Earth and has its origin at the Earth center of mass. Hence, the GPS position estimates are already provided in an ECEF frame and only need to be converted to the metrical xyz coordinates. This makes it relatively flexible and allows the use of any off-the-shelf GPS receivers. To integrate the INS sensor data, more transformations are needed. Since the sensors do not truly span a three-dimensional space, this misalignment has to be compensated first. Secondly, the measurements obtained in the inertial frame of reference for both the accelerometers and gyros need to be transformed into a common frame of reference, e.g. the platform or body frame. At last, the coordinates in this frame referring to the IMU can be transformed to ECEF coordinates.

Before the IMU data can be fed to the system, it needs to be pre-processed. Since the sampled data consists of integer values, they need to be dequantized.

The pre-processing will also correct the misalignment and scaling, and roughly compensate the bias. These parameters have to be obtained first through a calibration process. The sensor biases are moreover contained in the state-space model to compensate the slowly varying nature e.g. due to temperature drift.

In the GPS aided INS, a position error is calculated from the difference in the position estimation from both systems. This error signal is then used to drive an extended Kalman filter, which contains a model for the error dynamics. These error states are then used to compensate the INS states, which results in a closed loop system, depicted in Figure 1.2.

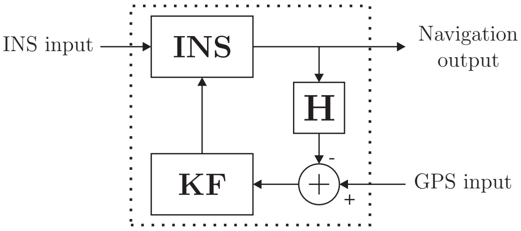


Figure 1.2: A loosely coupled position aided closed loop implementation. KF denotes the Kalman filter, and H the observation matrix. The dashed frame indicates the extended Kalman filter.

Finally, since this signal processing part is run on the testbed, the navigation output can be plotted on the display to show the trajectory. Further information such as the velocity and attitude of the vehicle can be shown as well.

1.3 Thesis Outline

Chapter 2 reviews the basic principles and fundamentals of inertial navigation and GPS. It starts with a description of different coordinate frames that are used, followed by the derivations and equations needed to transform a point between the different coordinate systems. The navigation equations and error dynamics for the INS are derived. The last section is devoted to GPS, where the different error sources are presented and analyzed.

The GPS/INS integration is described in Chapter 3. After the description of different coupling methods, the navigation equations from the previous chapter are discretized for the use in a time discrete digital signal processing system. Then, the extended Kalman filter is presented with its equations. Following that, some additional aspects for the system are discussed, such as the observability, the covariance matrices, and the lever-arm correction. Finally, a calibration process for the inertial sensors is presented and some methods for an accuracy improvement are proposed.

In Chapter 4, the hardware involved in this project is described. The section about the software shows the procedures to obtain measurement data from the IMU and describes the design of the C++ implementation of the navigation algorithm.

The results are presented in Chapter 5. First, the setup of the equipment for the field measurements is described, including a brief explanation of the trajectory, and some notes about the data pre-processing. Second, the results using field data are presented. At last, an analysis of the various errors is given, based on the MATLAB simulation (i.e. with simulated sensor data).

Chapter 6 contains conclusions, gives some recommendations formed through this thesis, and provides suggestions for future development.

As mentioned in the introduction, this thesis is based on a previous thesis [4] and continues the work therein. Further contributions can also be found there (see the list of references in [4]).

Parts of this work have been presented as:

> Isaac Skog, Peter Händel, and Adrian Schumacher, "The TROGE Project", KTH and FOI workshop on navigation, Stockholm, Sweden, April 9 2006. Proceedings IR-EE-SB 2006:009, Royal Institute of Technology, Stockholm, Sweden.

and have been submitted for publication and presentation as:

> Isaac Skog, Adrian Schumacher, and Peter Händel, "A Versatile PC-Based Platform For Inertial Navigation," *7th Nordic Signal Processing Symposium*, 7.-9. Jun. 2006.

Chapter 2

Inertial Navigation

This chapter briefly describes the principles of the inertial navigation and the incorporation of a satellite navigation system.

Inertial navigation is based on the principle that an object will resist in its current state (stationary or in uniform motion), unless it is disturbed by a force which causes an acceleration. Measuring this acceleration allows us to determine its motion. A mathematical integration with respect to time leads us to the velocity, and through a second integration, we obtain the relative position. To determine the direction it is crucial to consider the actual attitude.

Basically, there are two physical implementations of inertial sensors: the gimbaled arrangement, and the strap-down arrangement. In the first one, the sensors are mounted on a gimbaled mechanized platform that always remains aligned with the navigation frame. In the later arrangement, the sensors are directly mounted to the vehicle. This however, requires a higher bandwidth and dynamic range, and a higher computational load. The advantages are the smaller size, decreased weight, low power consumption and a significantly reduced cost. Therefore, strap-down arrangements are preferred for low-cost applications.

The acceleration is measured with an accelerometer. These sensors not only measure the acceleration due to an external force, but also the acceleration due to the local gravity. Given that we know the attitude of the accelerometers, we can mathematically remove the local gravity component.

In order to determine the attitude, an angular velocity sensor, i.e. a gyroscope can be employed. Mathematically integrating these values allows us to determine the rotation of the platform and therefore the change in its attitude.

Our inertial measurement unit (IMU) (see Section 4.1.2) comprises of three orthogonal aligned accelerometers and three orthogonal aligned gyros. This arrangement is referred to as a *six degree-of-freedom* (6-DoF) IMU.

The integration of the sensor values has a low-pass characteristic that attenuates undesired high-frequency noise. In contrast, a small bias in the sensed values leads to a drift in the estimated position. This is a common problem and one of the major drawbacks in inertial navigation. One can combat this problem by aiding the system e.g. with a positioning system such as the *Global Positioning System* (GPS).

2.1 Coordinate Frames

In an inertial navigation system, several information sources usually come from different coordinate systems. Therefore, it is necessary to transform the measured quantities into a common coordinate frame in order to process all the data. There are basically three classes of coordinate frames; Earth centered systems, local systems, and vehicle-centered systems.

Accelerations are measured in their inertial frame of reference, resolved in their instrumental frame and transformed into the platform frame (vehicle centered). The same applies to the angular rates from the gyros. For the data fusion with other systems, e.g. GPS, the coordinates are transformed into the navigation frame (e.g. Earth centered).

A coordinate expressed in an arbitrary a-frame is denoted with the superscript a. Figure 2.1 illustrates the relations between different frames that are used in this thesis and they are briefly described here.

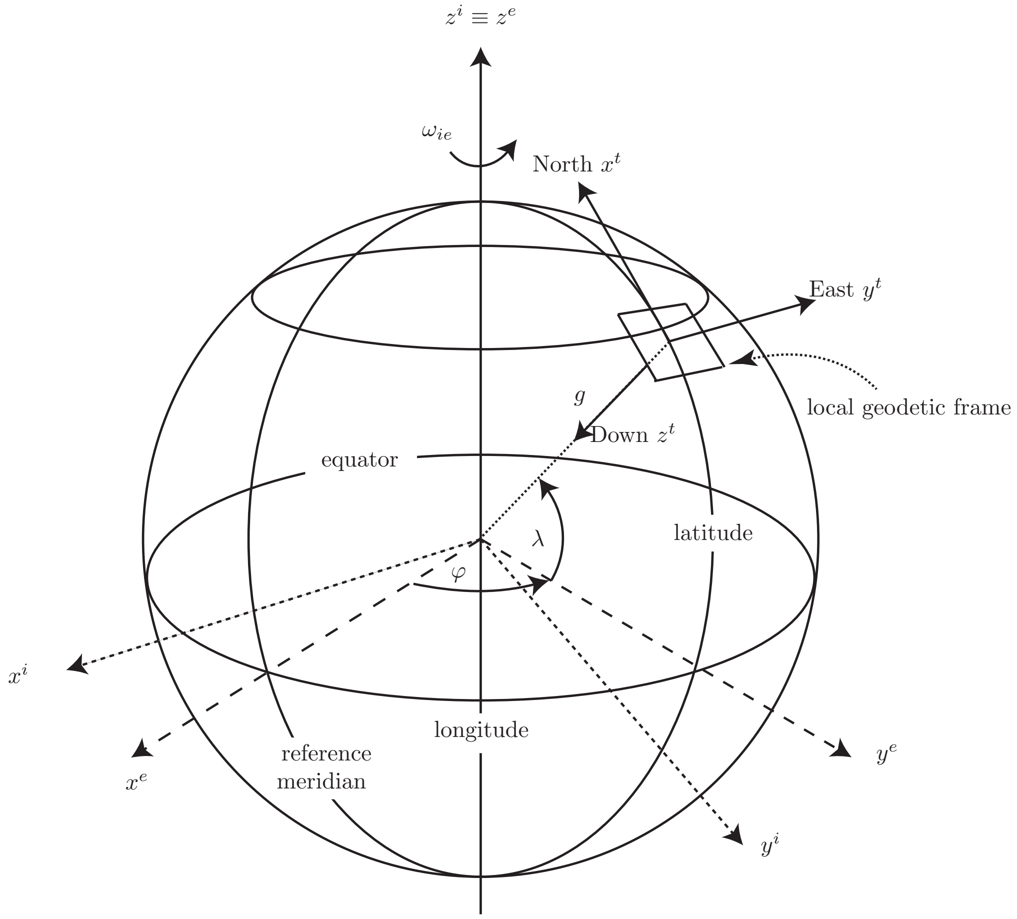

$z^i \equiv z^e$
ω_{ie}
North x^t
East y^t
local geodetic frame
g
Down z^t
equator
λ
latitude
φ
x^i
longitude
reference meridian
x^e
y^e
y^i

Figure 2.1: Relations between ECEF-frame (e), local geodetic-frame (t) and inertial-frame (i) [4].

Earth-Centered Earth-Fixed Frame (ECEF, e-frame): As the name indicates, this frame has its origin at the center of mass of the Earth and rotates with it. The z-axis is parallel to the mean spin axis of the Earth, the x-axis points to the reference meridian and the y-axis completes the right-handed orthogonal frame.

Local Geodetic Frame (t-frame): This coordinate system has its origin coinciding with that of the instrumental frame, but its x-axis always points towards geodetic north, the z-axis towards the origin of the e-frame, and the y-axis completes the right-handed orthogonal frame on the geodetic reference ellipse. This is also referred to as the north-east-down (NED) system.

Inertial Frame (i-frame): In this coordinate frame Newton's laws of motion apply and it is not accelerating. It could be chosen arbitrary but it is convenient to let its origin coincide with the one of the e-frame.

Body Frame (b-frame): This coordinate system has its origin at the center of gravity of the vehicle with its x-axis pointing in the forward direction, the z-axis down through the bottom of the vehicle and the y-axis completes the right-handed orthogonal system as depicted in Figure 2.2. In case the instrumentation platform is not aligned with the body frame, an additional transformation has to be done.

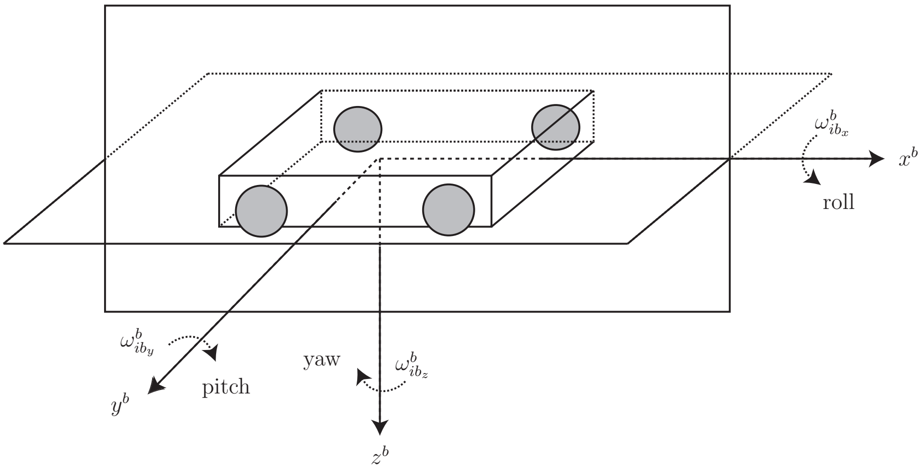


Figure 2.2: Body coordinate frame [4].

2.2 Coordinate Transformations

One transformation method uses the *direct cosine matrix* (DCM). We consider an arbitrary vector written in terms of the orthogonal unit vectors $\{\mathbf{i}_a, \mathbf{j}_a, \mathbf{k}_a\}$

$$\mathbf{p}^a = x_a\,\mathbf{i}_a + y_a\,\mathbf{j}_a + z_a\,\mathbf{k}_a \tag{2.1}$$

and want to transform from one coordinate system (a) to another (b) that uses the same origin. Using projection, we find the matrix in Equation 2.2 (derivation in [4]). $\alpha_{\mathbf{i}_k,\mathbf{j}_l}$ denotes the angle between the unit vectors $\mathbf{i}_k$ and $\mathbf{j}_l$.

$$\mathbf{R}_a^b = \begin{bmatrix} \cos(\alpha_{\mathbf{i}_a,\mathbf{i}_b}) & \cos(\alpha_{\mathbf{j}_a,\mathbf{i}_b}) & \cos(\alpha_{\mathbf{k}_a,\mathbf{i}_b}) \\ \cos(\alpha_{\mathbf{i}_a,\mathbf{j}_b}) & \cos(\alpha_{\mathbf{j}_a,\mathbf{j}_b}) & \cos(\alpha_{\mathbf{k}_a,\mathbf{j}_b}) \\ \cos(\alpha_{\mathbf{i}_a,\mathbf{k}_b}) & \cos(\alpha_{\mathbf{j}_a,\mathbf{k}_b}) & \cos(\alpha_{\mathbf{k}_a,\mathbf{k}_b}) \end{bmatrix} \tag{2.2}$$

The matrix multiplication

$$\mathbf{p}^b = \mathbf{R}_a^b \, \mathbf{p}^a \tag{2.3}$$

is then used to transform one vector $\mathbf{p}^a$ in the a-frame to a vector $\mathbf{p}^b$ in the b-frame.

The DCM $\mathbf{R}_a^b$ has only three degrees of freedom and can be uniquely described with the three Euler angles ϕ (roll), θ (pitch) and ψ (yaw) (see [1]). They can in return be obtained from the transformation matrix as

$$\phi \quad = \quad \operatorname{arctan2}(r_{32}, r_{33}) \,, \tag{2.4}$$

$$\theta \quad = \quad -\tan^{-1}\left(\frac{r_{31}}{\sqrt{1 - r_{31}^2}} \right) , \tag{2.5}$$

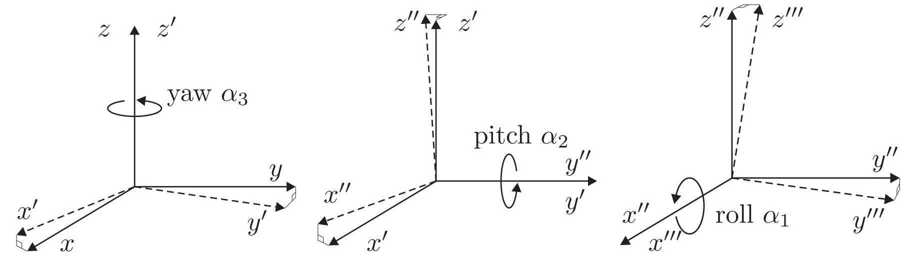
$$\psi \quad = \quad \operatorname{arctan2}(r_{21}, r_{11}) \,, \tag{2.6}$$

with r_{ij} being the (ith, jth) element of the DCM and arctan2(y,x) representing the four quadrant inverse tangent function.

Another method is to subsequently rotate a vector in the three planes, e.g. first in the plane spanned by the x- and y-axis, then the one spanned by the x- and z-axis, and finally in the plane spanned by the y- and z-axis (Figure 2.3). Mathematically, this sequence of rotations can be composed as follows to a complete DCM

$$\mathbf{R}_a^b = \mathbf{R}_x(\alpha_1)\mathbf{R}_y(\alpha_2)\mathbf{R}_z(\alpha_3) \,, \tag{2.7}$$

where α_1, α_2, and α_3 correspond to the three Euler angles in case of a transformation from the t- to the b-frame.



(a) Rotation by yaw (α_3) **(b)** Rotation by pitch (α_2) **(c)** Rotation by roll (α_1)

Figure 2.3: Rotations by yaw, pitch, and roll.

It is important to note the following properties of the DCM, due to the fact that it is orthonormal ($\mathbf{I}$ denotes an identity matrix):

$$\mathbf{R}_a^b \, {\mathbf{R}_a^b}^T = {\mathbf{R}_a^b}^T \, \mathbf{R}_a^b = \mathbf{I} \qquad \text{and} \qquad \mathbf{R}_b^a = (\mathbf{R}_a^b)^{-1} \,. \tag{2.8}$$

The general DCM $\mathbf{R}_b^a$ is deduced in [4]. Applied to the transformation from the t- to the b-frame leads to

$$\mathbf{R}_t^b = \begin{bmatrix} c(\psi)\,c(\theta) & s(\psi)\,c(\theta) & -s(\theta) \\ -s(\psi)\,c(\phi) + c(\psi)\,s(\theta)\,s(\phi) & c(\psi)\,c(\phi) + s(\psi)\,s(\theta)\,s(\phi) & c(\theta)\,s(\phi) \\ s(\psi)\,s(\phi) + c(\psi)\,s(\theta)\,c(\phi) & -c(\psi)\,s(\phi) + s(\psi)\,s(\theta)\,c(\phi) & c(\theta)\,c(\phi) \end{bmatrix} \tag{2.9}$$

where ϕ, θ, and ψ correspond to the roll, pitch, and yaw, respectively. $c(\cdot)$ and $s(\cdot)$ denote the cosine and sine operator. Similar, the rotation matrix for the t- to e-frame transformation reads

$$\mathbf{R}_e^t = \begin{bmatrix} -s(\lambda)\,c(\varphi) & -s(\lambda)\,s(\varphi) & c(\lambda) \\ -s(\varphi) & c(\varphi) & 0 \\ -c(\lambda)\,c(\varphi) & -c(\lambda)\,s(\varphi) & -s(\lambda) \end{bmatrix}. \tag{2.10}$$

Having the two transformation matrices $\mathbf{R}_t^b$ and $\mathbf{R}_e^t$, they can be combined into one DCM that reads $\mathbf{R}_e^b = \mathbf{R}_t^b\,\mathbf{R}_e^t$.

If a vector $\mathbf{p}$ is rotated only by very small angles around the coordinate axes with the angles $\delta\boldsymbol{\theta}_{ba}^a = [\delta\theta_x, \delta\theta_y, \delta\theta_z]^T$, the resulting rotation matrix can be approximated with $\mathbf{R}_a^b \approx \mathbf{I} + \delta\boldsymbol{\Theta}_{ba}^a$ where $\delta\boldsymbol{\Theta}_{ba}^a = (\delta\theta_{ba}^a \times)$ is the skew symmetric matrix representation of the rotation angles $\delta\theta$:

$$\delta\boldsymbol{\Theta}_{ba}^a = \begin{bmatrix} 0 & \delta\theta_z & -\delta\theta_y \\ -\delta\theta_z & 0 & \delta\theta_x \\ \delta\theta_y & -\delta\theta_x & 0 \end{bmatrix}. \tag{2.11}$$

Since the attitude of the body frame will change, also some transformation matrices will change. The gyros provide measurements in terms of the angular rates. To express the rate of change of a rotation matrix, we take the time derivative of the transformation matrix $\mathbf{R}_b^a$. According to [4], $\dot{\mathbf{R}}_b^a$ can be expressed as

$$\dot{\mathbf{R}}_a^b(t) = \mathbf{R}_a^b(t)\,\boldsymbol{\Omega}_{ba}^a\,. \tag{2.12}$$

Using the properties of the cross product, one can show that $\boldsymbol{\omega}_{ba}^a \times \mathbf{p} = \boldsymbol{\Omega}_{ba}^a\mathbf{p}$, where $\boldsymbol{\Omega}_{ba}^a$ is the skew symmetric matrix representation of $\boldsymbol{\omega}_{ba}^a$ as shown in Equation (2.13). The matrix $\boldsymbol{\Omega}_{ba}^a$ from Equation (2.12) consists of the three angular rates obtained from the gyros ($\omega_{ba_x}^a$, $\omega_{ba_y}^a$, $\omega_{ba_z}^a$) and is composed according to (2.13). The subscripts indicate a rotation of the a-frame relative to the b-frame and the superscript indicates that the rotation is measured in the a-frame.

$$\boldsymbol{\Omega}_{ba}^a = \begin{bmatrix} 0 & -\omega_{ba_z}^a & \omega_{ba_y}^a \\ \omega_{ba_z}^a & 0 & -\omega_{ba_x}^a \\ -\omega_{ba_y}^a & \omega_{ba_x}^a & 0 \end{bmatrix} \tag{2.13}$$

2.3 Navigation Equations

Newton's first law states that an object resists in its attitude and uniform motion as long as no force is applied. Whereas the second law states, that the acceleration of an object is proportional and in the same direction as the applied inertial force, and inversely proportional to its mass. Considering a position vector $\mathbf{r}^i$, the gravity acceleration $\mathbf{g}^i$, the specific force $\mathbf{f}^i$, and Newton's laws, we can deduce an equation for the kinematical acceleration

$$\ddot{\mathbf{r}}^i = \mathbf{g}^i + \mathbf{f}^i\,. \tag{2.14}$$

Assuming a position $\mathbf{r}^a$ in an arbitrary frame, we find the related position in the inertial frame by transforming the position using the DCM and adding the vector $\mathbf{q}^i$ pointing from the origin of the i-frame to the origin of the a-frame

$$\mathbf{r}^i = \mathbf{R}_a^i\,\mathbf{r}^a + \mathbf{q}^i\,. \tag{2.15}$$

By differentiating Equation (2.15) twice with respect to time, using Equation (2.12) and its derivative, the result can be used to substitute $\ddot{\mathbf{r}}^i$ in (2.14). To simplify the integration of the INS with the GPS system, we will use the ECEF-frame as the common coordinate frame. Therefore, we substitute the arbitrary a-frame with the ECEF-frame and get

$$\mathbf{g}^e + \mathbf{f}^e = \ddot{\mathbf{r}}^e + 2\,\boldsymbol{\Omega}_{ie}^e\,\dot{\mathbf{r}}^e + \dot{\boldsymbol{\Omega}}_{ie}^e\,\mathbf{r}^e + \boldsymbol{\Omega}_{ie}^e\boldsymbol{\Omega}_{ie}^e\mathbf{r}^e + \ddot{\mathbf{q}}^e\,. \tag{2.16}$$

Since the e- and i-frame only differ in a rotation around the z-axis, the vector $\ddot{\mathbf{q}}^e = 0$ and the skew-symmetric matrix $\boldsymbol{\Omega}_{ie}^e$ only has a non-zero element in both occurrences of $\omega_{ie_z}^e$. This equals the Earth rotational rate relative to the i-frame and amounts to $\omega_{ie} \approx 7.292115 \cdot 10^{-5}$ rad/s [1].

It is possible to further simplify Equation (2.16). The third term of the right hand side is the Euler acceleration and can be omitted (see [3]). The second last term, the centripetal force, can be neglected as well, because of the low sensitivity of the used sensors. The simplified navigation equation now reads

$$\ddot{\mathbf{r}}^e = -2\,\boldsymbol{\Omega}_{ie}^e\,\dot{\mathbf{r}}^e + \mathbf{g}^e + \mathbf{f}^e\,. \tag{2.17}$$

The sensors measure the specific force $\mathbf{f}$ and the rotation rates $\boldsymbol{\omega}_{ib}^b$ in the body frame. Therefore, we need to transform them to the ECEF-frame in order to integrate them in the navigation equation. The continuous-time version of the two first order differential equations for the implementation is

$$\begin{bmatrix} \mathbf{v}^e \\ \dot{\mathbf{v}}^e \\ \dot{\mathbf{R}}_b^e \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{r}}^e \\ -2\boldsymbol{\Omega}_{ie}^e\,\mathbf{v}^e + \mathbf{g}^e + \mathbf{R}_b^e\,\mathbf{f}^b \\ \mathbf{R}_b^e\,\boldsymbol{\Omega}_{eb}^b \end{bmatrix}, \tag{2.18}$$

where $\mathbf{v}^e$ is the velocity vector. The skew-symmetric matrix $\boldsymbol{\Omega}_{eb}^b$ for the rotation rates between the ECEF- and body frame consists of the angular rates $\boldsymbol{\omega}_{ib}^b$ measured by the gyros and the Earth rotational rate. It can be assembled using the equation according to [5]

$$\boldsymbol{\omega}_{eb}^b = \boldsymbol{\omega}_{ib}^b - \mathbf{R}_e^b\,\boldsymbol{\omega}_{ie}^e\,. \tag{2.19}$$

The block diagram of the derived INS in ECEF coordinates is shown in Figure 2.4.

2.4 Error Dynamics

Through the calibration of the IMU, the bias and scaling in the accelerometers and gyros can be compensated. Nevertheless, these values still can vary over

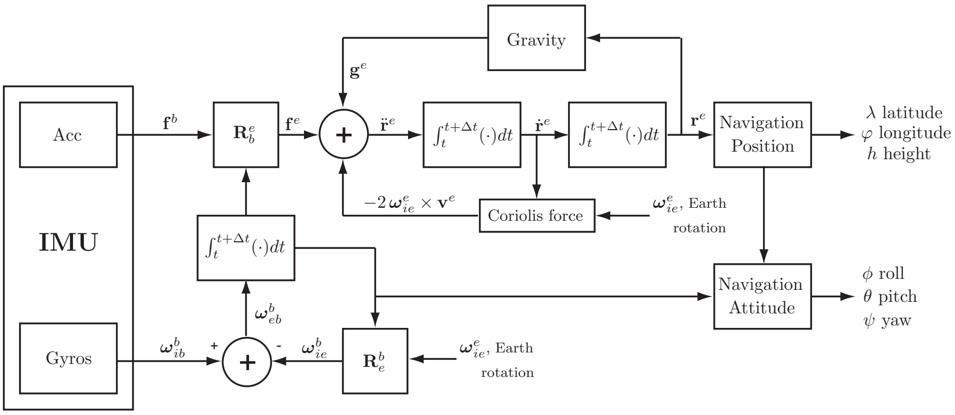

Gravity
$\mathbf{g}^e$
Acc
IMU
Gyros
$\mathbf{f}^b$
$\mathbf{R}_b^e$
$\mathbf{f}^e$
+
$\ddot{\mathbf{r}}^e$
$\int_t^{t+\Delta t}(\cdot)dt$
$\dot{\mathbf{r}}^e$
$\int_t^{t+\Delta t}(\cdot)dt$
$\mathbf{r}^e$
Navigation Position
λ latitude
φ longitude
h height
$-2\,\boldsymbol{\omega}_{ie}^e \times \mathbf{v}^e$
Coriolis force
$\boldsymbol{\omega}_{ie}^e$, Earth rotation
$\int_t^{t+\Delta t}(\cdot)dt$
Navigation Attitude
ϕ roll
θ pitch
ψ yaw
$\boldsymbol{\omega}_{eb}^b$
$\boldsymbol{\omega}_{ib}^b$
+
-
$\boldsymbol{\omega}_{ie}^b$
$\mathbf{R}_e^b$
$\boldsymbol{\omega}_{ie}^e$, Earth rotation

Figure 2.4: Block diagram of the ECEF INS [6].

time e.g. due to a change in temperature. Therefore, an error model is required to combat these effects. Furthermore, the position error being the difference between the GPS and INS estimates is also related to the velocity and attitude errors. We will now investigate these error equations.

Referring to Equation (2.18), the velocity error is simply $\delta\mathbf{v}^e = \delta\dot{\mathbf{r}}^e$. The acceleration equation from Equation (2.18) reads in the mechanized form

$$\dot{\hat{\mathbf{v}}}^e = \hat{\mathbf{R}}_b^e\,\tilde{\mathbf{f}}^b + \hat{\mathbf{g}}^e - 2\boldsymbol{\Omega}_{ie}^e\,\hat{\mathbf{v}}^e\,, \tag{2.20}$$

where $\tilde{\cdot}$ denotes a measured, and $\hat{\cdot}$ an estimated value. $\tilde{\mathbf{f}}^b$ represents the measured accelerations, $\hat{\mathbf{g}}$ the estimated local gravity, $\boldsymbol{\Omega}_{ie}^e$ is a known constant matrix, and $\hat{\mathbf{v}}^e$ the estimated velocity.

In [4] it is shown that writing the Equation (2.20) in terms of the true value and an error term, with expansion and the consideration of terms that cancel each other out, as well as neglecting second order terms, will lead to

$$\delta\dot{\mathbf{v}}^e = -\mathbf{S}_f^e\,\boldsymbol{\epsilon} + \mathbf{R}_b^e\,\delta\mathbf{f}^b + \delta\mathbf{g}^e - 2\,\boldsymbol{\Omega}_{ie}^e\,\delta\mathbf{v}^e\,, \tag{2.21}$$

where

$$\mathbf{S}_f^e = \begin{bmatrix} 0 & -f_z^e & f_y^e \\ f_z^e & 0 & -f_x^e \\ -f_y^e & f_x^e & 0 \end{bmatrix} \tag{2.22}$$

and $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \epsilon_3]^T$ is the attitude error vector denoting the small angle rotations of the DCM $\mathbf{R}_b^e$. To find the attitude errors, we start with the mechanized attitude equation from (2.18):

$$\dot{\hat{\mathbf{R}}}_b^e = \hat{\mathbf{R}}_b^e\,\boldsymbol{\Omega}_{eb}^b\,. \tag{2.23}$$

The skew symmetric matrix $\boldsymbol{\Omega}_{eb}^b$ can be totally described according to Equation (2.13) with the vector $\boldsymbol{\omega}_{eb}^b$. With this help, the equation

$$\dot{\boldsymbol{\epsilon}} = \mathbf{R}_b^e\,\delta\boldsymbol{\omega}_{eb}^b \tag{2.24}$$

can be formed. Further substitutions according to [4] lead to the final attitude error equation. To conclude, let us note all three navigation error equations:

$$\begin{bmatrix} \delta\mathbf{v}^e \\ \delta\dot{\mathbf{v}}^e \\ \dot{\boldsymbol{\epsilon}} \end{bmatrix} = \begin{bmatrix} \delta\dot{\mathbf{r}}^e \\ -\mathbf{S}_f^e\,\boldsymbol{\epsilon} + \mathbf{R}_b^e\,\delta\mathbf{f}^b + \delta\mathbf{g}^e - 2\,\boldsymbol{\Omega}_{ie}^e\,\delta\mathbf{v}^e \\ \mathbf{R}_b^e\,\delta\boldsymbol{\omega}_{ib}^b - \boldsymbol{\Omega}_{ie}^e\,\boldsymbol{\epsilon} \end{bmatrix} . \tag{2.25}$$

In our system, it is feasible to model the measurement errors as white Gaussian noise with a bias described by a random level. We further assume that the IMU sensors are the only noise sources in our system. Referring to the navigation equations (2.18), the noise enters the system through the last two equations, describing the velocity and attitude states. Rewriting the difference equations in a state space model and defining the error state vector as

$$\delta\mathbf{x}(t) = \begin{bmatrix} \delta\mathbf{r}^{eT} & \delta\mathbf{v}^{eT} & \boldsymbol{\epsilon}^T & \delta\mathbf{f}^{bT} & \delta\boldsymbol{\omega}_{ib}^{b\,T} \end{bmatrix}^T \tag{2.26}$$

and the measurement noise vector with the accelerometer and gyro noise $u_{\text{acc}}(t)$ and $u_{\text{gyro}}(t)$, respectively as

$$\mathbf{u}_c(t) = \begin{bmatrix} \mathbf{u}_{\text{acc}}^T(t) & \mathbf{u}_{\text{gyro}}^T(t) \end{bmatrix}^T , \tag{2.27}$$

we can define the navigation error state model as

$$\delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\,\delta\mathbf{x}(t) + \mathbf{G}(t)\,\mathbf{u}_c(t) , \tag{2.28}$$

where $\mathbf{F}(t)$ is the 15×15 matrix

$$\mathbf{F}(t) = \begin{pmatrix} \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & -2\boldsymbol{\Omega}_{ie}^e & -\mathbf{S}_f^e & \mathbf{R}_b^e & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & -\boldsymbol{\Omega}_{ie}^e & \mathbf{0}_3 & \mathbf{R}_b^e \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{pmatrix} \tag{2.29}$$

and $\mathbf{G}(t)$ the 15×6 matrix

$$\mathbf{G}(t) = \begin{pmatrix} \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{R}_b^e & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{R}_b^e \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \end{pmatrix} . \tag{2.30}$$

In the matrices (2.29) and (2.30), $\mathbf{I}_3$ and $\mathbf{0}_3$ denote the identity and zero matrix of order 3, respectively. Note that these error navigation equations are time varying, since $\mathbf{R}_b^e$ depends on the attitude and $\mathbf{S}_f^e$ on the acceleration of the vehicle.

The three accelerometers and three gyros used in the IMU are of the same type each and its noise models are assumed to be identical. However, the noises are assumed to be uncorrelated, since the sensors are independent devices. Denoting the variance of the accelerometer noise with σ_{acc}^2 and the variance of the gyro noise with σ_{gyro}^2, we find the covariance matrix $\mathbf{Q}_c(t)$ of the Gaussian measurement noise as

$$\begin{aligned} \mathbf{E}\{\mathbf{u}_c(t)\,\mathbf{u}_c^T(\tau)\} &= \begin{bmatrix} \sigma_{\text{acc}}^2\,\mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \sigma_{\text{gyro}}^2\,\mathbf{I}_3 \end{bmatrix} \delta(t-\tau) \\ &\triangleq \mathbf{Q}_c(t-\tau) \end{aligned} \tag{2.31}$$

where $\delta(t)$ is the Kronecker delta.

2.5 Global Positioning System

There are currently two *Global Navigation Satellite Systems* (GNSS), namely the widely used *Global Positioning System* (GPS) from the U.S. Department of Defense[1], and the *Russian Global Orbit Navigation Satellite System* (GLONASS)[2]. A third system named *Galileo*[3], which is launched by the European Union and the European Space Agency, will be in commercial operation phase in year 2008. Because the GPS is well established and a broad range of receivers (also some relatively inexpensive receivers) are available, this system was chosen for this project.

With the GPS, civilian users can only use the restricted *Standard Positioning Service* (SPS). The *Precise Positioning Service* (PPS) is only intended for the U.S. military and other authorized users.

There are at least 24 satellites on six equally spread orbits around the Earth. The six orbital planes are inclined with a 55° angle with respect to the equator. The orbits are located about 20200 km above the Earth's surface.

The GPS mainly provides a 3-dimensional position estimation and the accurate Coordinated Universal Time UTC. To estimate a 3-d position, the receiver needs to receive the signals of at least 4 satellites. The accuracy of the estimate depends on several factors that are discussed in 2.5.1.

In order to compute the position at the receiver, the position of the satellites has to be known. These coordinates can roughly be calculated from the GPS almanacs that are transmitted by the satellites. For a more accurate satellite coordinate calculation, the ephemeris data, also transmitted in the satellite message, is used.

Each GPS satellite transmits the data simultaneously on the two frequencies L1 (1575.42 MHz) and L2 (1227.60 MHz). The data (50 bit/s) is BPSK modulated and spread with binary codes (CDMA). The later are used to differentiate between the satellites. On L1, the signal is spread with a Precision or p-code and a Coarse/Acquisition or C/A-code. The signal on L2 is only spread with the p-code. While the C/A-code is publicly available (SPS), the p-code is further encrypted for a regulated access and is only known by authorized users (PPS). The C/A-code consists of 1023 samples, is transmitted at 1/10 of the fundamental GPS frequency and is repeated every 1 ms, whereas the p-code is transmitted at the fundamental frequency of 10.23 MHz and repeated every 267 days.

There are three observables, namely the pseudorange, the carrier phase, and the Doppler shift. The pseudorange represents the time shift needed to correlate the received and demodulated signal with a receiver-replicated code. Since the receiver- and satellite-clocks are not perfectly synchronized, and also due to other error sources, the description pseudorange rather than range is used.

Better performance for the position determination can be gained by measuring the carrier phase before the demodulation. This allows an accuracy down to a fraction of the wavelength, but there is a problem called integer ambiguity. Using the phase, only a fraction of a wavelength can be measured, but there remains an unknown number of whole wavelengths that fit between the satellite and the receiver. Due to high vehicle dynamics, satellite shading, and high ionosphere activity, a loss of phase lock can occur and results in a so-called cycle slip. A method to overcome this problem can be found in [7].

---

[1] USCG Navigation Center, *GPS*: http://www.navcen.uscg.gov/gps/

[2] Russian Space Forces, *GLONASS*: http://www.glonass-center.ru/

[3] European Space Agency, *Galileo*: http://www.esa.int/esaNA/galileo.html

Finally, a Doppler shift can be measured due to the movements of a receiver. However, a Doppler shift only occurs if the receiver is moving towards to or away from a satellite. Also the satellites itself move, but, since their trajectory is predictable, this influence can be compensated. The measured Doppler shift is used to determine the receiver's velocity. The position of a particular satellite

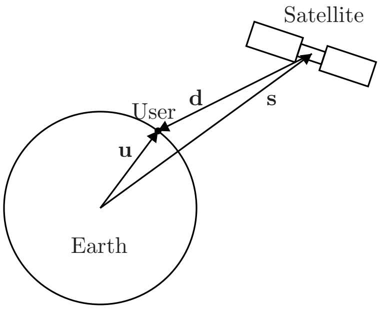


Figure 2.5: Vectors that define the position between the Earth, a satellite and a user.

is known from the ephemeris data (ECEF coordinates) and is denoted as **s** in Figure 2.5. The distance **d** is estimated using the pseudorange and phase measurements (the distance is obtained by multiplying the time delay with the speed of light). Adding this two vectors, one can find the user position as

$$\mathbf{u} = \mathbf{s} + \mathbf{d}\,. \tag{2.32}$$

The measured pseudorange estimated from each satellite (subscript k) can be expressed as

$$\rho_k = \|\mathbf{u} - \mathbf{s}_k\| + c \cdot (\delta t_{s_k} - \delta t_r) + \varepsilon_k\,, \tag{2.33}$$

where the difference $\delta t_{s_k} - \delta t_r$ multiplied with the speed of light is an error due to the clock drift in the satellite and receiver. The remaining error components are collected in ε_k and described in the following (see also Table 2.1). Disregarding ε_k, we see from Equation (2.33) that we have three unknowns for the position in ECEF coordinates, and one unknown for the clock difference. Therefore, we need four satellites to solve for the 4 unknowns. Usually, there are more than four satellites available leading to an over-determined system that can be solved in a least squares sense, and provide a more accurate solution.

2.5.1 Error Sources

Several error sources can deteriorate the quality of the position estimation. The following paragraphs name the important sources. The size of these typical errors is collected in Table 2.1. Figure 2.6 shows an example on how the errors influence the position estimation. The observation time was approximately 30 minutes and the receiver was kept stationary.

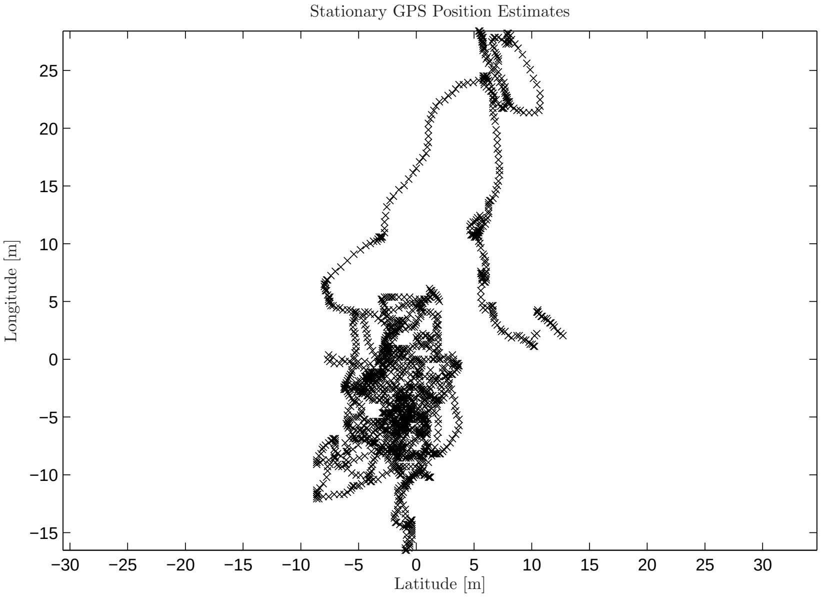


Figure 2.6: Error in stationary GPS position estimates during approx. 30 min. The measurement took place on 7 November 2005 on the roof of the department building.

Selective Availability

The U.S. Department of Defense has with *Selective Availability* (SA) a method, to add a clock error introduced by the satellites to affect the position determination for unauthorized users. In May 2000, this SA was removed [8] but there is no guarantee that it will not be introduced again. For civil users, this SA accounts for the largest errors. There are methods such as Differential GPS (DGPS) to reduce this error [9].

Atmospheric Effects

Atmospheric effects introduce further errors. The troposphere is relatively close around the Earth and extends between 6 to 18 km. It is electrically neutral and non-dispersive for frequencies up to 15 GHz [10, 11]. However, the presence of water vapor, the atmospheric temperature, and the pressure cause a delay, which is the same for both frequencies L1 and L2.

The ionosphere extends roughly from 50 to 1500 km and contains a large amount of free electrons and positive ions. These cause a group delay of the signal but can also cause refraction and diffraction effects [10]. The ionospheric activity is strongly dependent on the number of sunspots. Using appropriate models and DGPS, these atmospheric effects can be significantly reduced to improve the position accuracy.

Orbital Error

Another error is introduced due to the inaccuracy in the ephemeris data. The *Signal-in-Space User Range Error* (SIS UER) was 1.4 m RMS as of April 2001 across the constellation and is expected to decrease [12]. Replacing older Block II satellites with newer Block IIR satellites and the *Legacy Accuracy Improvement Initiative* (AII) contribute to a better performance. Data that are more accurate are available through the official Navel Surface Warfare Center together with the National Imagery and Mapping Agency, which are available a few weeks after the observation. A large user community uses orbit products from the International GNSS Service (IGS)[4], because they also provide predicted real-time IGS Ultra-Rapid orbit products with an accuracy of about 0.25 m. More information about how IGS works can be found in [13].

Clock Errors

The atomic clocks in the GPS satellites have to run synchronized with the GPS system time. The small difference is constantly monitored by the Master Control Station (MCS) and the errors are transmitted as coefficients of a second order polynomial [14]. The larger clock errors occur in the receiver. It varies depending on the clock quality between some μs to a few ms. Nevertheless, this clock-drift as well as the satellite clock error can be effectively removed by using DGPS [9].

Multipath and Noise

In wireless radio links, one has to deal with multipath reception. The best case is to have a direct line of sight between the transmitter and the receiver. Nonetheless, there are reflected signals that superimpose the direct signal (see Figure 2.7). These multipath reflections affect both the pseudorange and phase measurements in a GPS receiver. Methods to decrease the effects are known from e.g. cellular radio communications [10]. At last, there is also noise generated in the receiver itself and maybe in the environment. The receiver noise is depending on the antenna gain, amplifiers, receiver dynamics, and the code correlation methods. It can mostly be reduced by a careful and thorough hardware design.

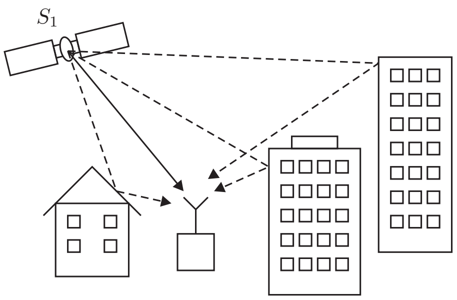


Figure 2.7: Multipath propagation; the solid arrow indicates the direct path, the dashed arrows indicate reflected path.

---

[4] International GNSS Service: http://igscb.jpl.nasa.gov/

Table 2.1: Typical errors that affect the pseudorange measurement [15, 11].

Type of error		Typical size (1σ)
Ephemeris error	δr_{orb}	2.1 m
Satellite clock error	$c \cdot \delta t_s$	2.1 m
Receiver clock error	$c \cdot \delta t_r$	0.5 m
Selective Availability on/off	δr_{SA}	25 m / 0 m
Ionospheric delays	δr_{ion}	4.0 m
Tropospheric delays	δr_{trop}	0.7 m
Multipath errors	δr_{MP}	1.4 m
Receiver measurement noise	v	0.5 m
User Equivalent Range Error	σ_{UERE}	5.3 m

2.5.2 Dilution of Precision

Another effect that affects the precision of a position estimation is caused by the satellite constellation. If e.g. all satellites happen to be on a straight line from the user's point of view, the precision is worse than if e.g. three satellites are placed equidistant on a large circle around the user and one satellite straight above. The *Dilution of Precision* (DOP) [1] expresses these effects together with the time bias errors and the other pseudorange errors. The pseudorange error $\delta\rho$ can be obtained from the position errors and the time error $\delta\mathbf{e} = [\delta x, \delta y, \delta z, c \cdot \delta t]^T$ through a linearized equation as

$$\delta\rho = \mathbf{H}\,\delta\mathbf{e} + \delta\boldsymbol{\epsilon}_\rho\,, \tag{2.34}$$

where $\mathbf{H}$ is a $(m \times 4)$ matrix of partial derivatives of the Equation (2.33), with respect to the four unknown variables. $\delta\boldsymbol{\epsilon}_\rho$ is a zero mean noise term. Assuming many simplifications, one can take the expectation of the estimated vector $\delta\hat{\mathbf{e}}$ leading to $E[\delta\hat{\mathbf{e}}\,\delta\hat{\mathbf{e}}^T] = \sigma_e^2(\mathbf{H}^T\mathbf{H})^{-1}$, see [15]. The matrix multiplication and inversion can be expressed as

$$(\mathbf{H}^T\mathbf{H})^{-1} = \begin{bmatrix} D_{11} & D_{12} & D_{13} & D_{14} \\ D_{21} & D_{22} & D_{23} & D_{24} \\ D_{31} & D_{32} & D_{33} & D_{34} \\ D_{41} & D_{42} & D_{43} & D_{44} \end{bmatrix}, \tag{2.35}$$

where each $D_{ij} \in \mathbb{R}$ represents a scale factor for the variance σ_e^2. Since the diagonal elements of that matrix relate the measurement errors with the computed position and time errors, one can find the following parameters:

$$HDOP = \sqrt{D_{11} + D_{22}} \tag{2.36}$$

$$VDOP = \sqrt{D_{33}} \tag{2.37}$$

$$TDOP = \sqrt{D_{44}} \tag{2.38}$$

These first three DOP parameters describe the *horizontal DOP*, *vertical DOP*, and *time DOP*, respectively. Two more general DOP parameters can be calculated, namely, the *position DOP* and the *geometric DOP*:

$$PDOP = \sqrt{D_{11} + D_{22} + D_{33}} \tag{2.39}$$

$$GDOP = \sqrt{D_{11} + D_{22} + D_{33} + D_{44}} \tag{2.40}$$

The total position, vertical, or time error magnitude can be estimated as the multiplication of the standard deviation σ_e with the desired DOP value. The GPS receiver used for this thesis only provides the PDOP. So, the position error magnitude is approximated as

$$\sigma_r = PDOP \cdot \sigma_e \,. \tag{2.41}$$

It becomes clear that a low value indicates a good accuracy. In practice, values below 8 can be considered as fairly good. In field tests on an open area with a clear line of sight and low multipath reflections, the PDOP value usually ranged between 2 and 5.

2.5.3 Differential GPS

In Section 2.5.1, the error sources with the most influence on the position accuracy were pointed out. Except for multipath and noise errors, it is possible to reduce or even remove their effect on the position estimation. There are several combination methods available. The most common one is the *single difference GPS* (or *differential GPS* (DGPS)) as depicted in Figure 2.8. Sophisticated methods such as double or triple difference incorporate measured differences from several satellites. Another technique is to use a second frequency, e.g. L2, in order to minimize the ionospheric effects. However, this method is not available to public, unauthorized users. The single difference GPS employs one mobile receiver e.g. on a vehicle and a stationary receiver (*base station*). The exact coordinates of the base station are known, therefore it is possible to compute the real errors in the pseudorange and form some correction parameters. They are transmitted to the mobile receiver over e.g. a radio link (dashed arrow in Figure 2.8). If the mobile receiver is relatively close to the base station, both received signals from the satellites experience almost the same atmospheric disturbances. The mobile receiver then uses the correction parameters to minimize or even eliminate the errors. However, since the thermal noise in the two receivers is uncorrelated, its variance increases with the factor two. Still, it is possible to get a position accuracy of a few centimeters.

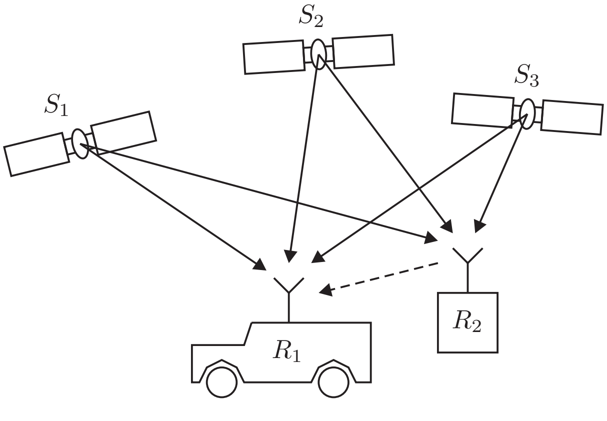


Figure 2.8: Illustration of a single difference GPS.

Chapter 3

GPS/INS Integration

There are several methods to integrate the INS and GPS data. The most common is done by means of a Kalman filter. In the first section, the coupling method INS-GPS-Kalman filter is described.

In the previous Section 2.3, the continuous-time navigation equations were derived. In order to apply the equations to the practical implementation, we first find the zero-order-hold sampling of the navigation and attitude equations. Thereafter, the model of the navigation error dynamics is discretized. In the next section, we describe the extended Kalman filter, which is used to fuse the IMU and GPS data together, and form the GPS aided INS.

In order to use the acquired IMU sensor data, they first need to be calibrated. Section 3.8 describes a calibration algorithm for the accelerometers and gyros. Since the gyro calibration requires further hardware that is not yet available, a very simple method to roughly estimate the bias and scaling of the gyros is described.

Finally, some methods and ideas are presented to increase the accuracy of the INS.

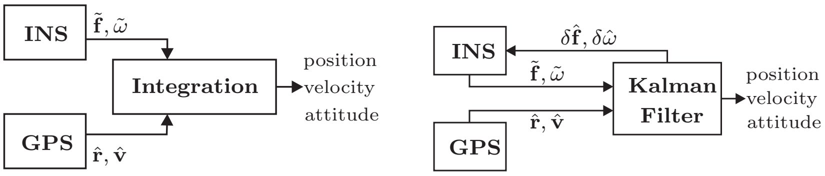
3.1 Coupling Approaches

The coupling of INS, GPS, and the integration algorithm can be done in different ways. The complexity of the integration and the requirements to the sensors and GPS receiver vary. There are three categories of coupling methods: uncoupled filter, loosely coupled filter, and tightly coupled filter (see Figure 3.1).

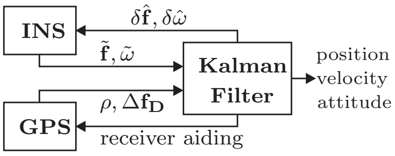
In the uncoupled filter as in Figure 3.1(a), the integration only consists of an integrator for the INS measurements $(\tilde{\mathbf{f}}, \tilde{\omega})$ and an algorithm to combine the solution with the GPS estimates $(\hat{\mathbf{r}}, \hat{\mathbf{v}})$. The overall complexity is relatively low. However, the accuracy of the IMU (and GPS receiver) has to be very high, since there is no access to internal states of the INS and GPS and no possibility to influence these two units.

A more accurate way is the loosely coupled filter depicted in Figure 3.1(b). The INS and GPS are still two separate units providing their preprocessed measurements/solutions. A Kalman filter is used to integrate and fuse the INS and GPS data. Estimated errors $(\delta\hat{\mathbf{f}}, \delta\hat{\omega})$ are used to correct the IMU measurements and bound its errors. Because the GPS receivers often include a filter to provide less noisy signals, they become correlated over time. Nevertheless, this scheme provides high flexibility, allows the use of off-the-shelf hardware and with a closed-loop INS implementation, also medium to low accuracy IMUs can be used. The loosely coupled approach is also used in this thesis.

Tightly coupled filters comprise of a centralized Kalman filter (Figure 3.1(c)). From the GPS receiver, the pseudoranges ρ, and Doppler-measurements $\Delta\mathbf{f_D}$ are directly used as inputs to the navigation Kalman filter. More error states can be used to correct the estimated values. Other outputs from the Kalman filter (e.g. receiver clock error) can be used to aid the GPS receiver (e.g. satellite tracking loops). From these three categories, the tightly coupled filter is the most efficient, accurate, but also complex implementation.


INS
$\tilde{\mathbf{f}}, \tilde{\omega}$
Integration
position
velocity
attitude
GPS
$\hat{\mathbf{r}}, \hat{\mathbf{v}}$
INS
$\delta\hat{\mathbf{f}}, \delta\hat{\omega}$
$\tilde{\mathbf{f}}, \tilde{\omega}$
Kalman Filter
position
velocity
attitude
$\hat{\mathbf{r}}, \hat{\mathbf{v}}$
GPS

(a) Uncoupled filter **(b)** Loosely coupled filter


INS
$\delta\hat{\mathbf{f}}, \delta\hat{\omega}$
$\tilde{\mathbf{f}}, \tilde{\omega}$
Kalman Filter
position
velocity
attitude
$\rho, \Delta\mathbf{f_D}$
GPS
receiver aiding

(c) Tightly coupled filter

Figure 3.1: Coupling approaches

3.2 Discretization

As mentioned above, we derived the continuous-time navigation equations in the previous chapter. Based on these ECEF differential equations from (2.18), we write the first two equations in state space form where we use

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & -2\mathbf{\Omega}^e_{ie} \end{bmatrix}, \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{I}_3 \end{bmatrix}, \tag{3.1}$$

and obtain

$$\begin{bmatrix} \dot{\mathbf{r}}^e \\ \dot{\mathbf{v}}^e \end{bmatrix}(t) = \mathbf{A} \begin{bmatrix} \mathbf{r}^e \\ \mathbf{v}^e \end{bmatrix}(t) + \mathbf{B}\left[\mathbf{g}^e + \mathbf{R}^e_b\,\mathbf{f}^b\right](t). \tag{3.2}$$

As described in [4, 16], one can find a solution for this state space equation at time instant $t > t_0$. In our case, we set $t = t_0 + T_s$. Integrating $\mathbf{A}$ by knowing it is time invariant, we get the exponential function $\exp(\mathbf{A} \cdot (t - t_0))$ [16]. Expanding this with the power series definition yields $\mathbf{I}_6 + \mathbf{A}\,T_s$, assuming that $\mathbf{A}^n = \mathbf{0}_6\ \forall n > 1$. $\mathbf{I}_6$ and $\mathbf{0}_6$ denote the identity matrix and zero matrix of order 6, respectively. To solve the remaining integration, we insert what we got from the previous integration and assume constant values over the integration time T_s. As a result, we obtain the zero-order-hold sampling as

$$\begin{bmatrix} \mathbf{r}^e_{k+1} \\ \mathbf{v}^e_{k+1} \end{bmatrix} = (\mathbf{I}_6 + \mathbf{A}\,T_s) \begin{bmatrix} \mathbf{r}^e_k \\ \mathbf{v}^e_k \end{bmatrix} + T_s \mathbf{B} \left[\mathbf{g}^e + \mathbf{R}^e_{b,k}\, \mathbf{f}^b_k \right] . \tag{3.3}$$

To discretize the third equation from (2.18), the attitude equation, one has to maintain the orthogonality of the DCM. We assume that $\mathbf{\Omega}^b_{eb}$ is constant over one sample period T_s. Then, the matrix to propagate the components of the DCM to the next sample iteration is $\exp(\mathbf{\Omega}^b_{eb}\,T_s)$. To preserve the orthogonality constraint, we use a Padè approximation for the exponential function [17]. A $(2, 2)$ Padè approximation is used as in [4] to finally obtain

$$\mathbf{R}^e_{b,k+1} = \mathbf{R}^e_{b,k} (2\mathbf{I}_3 + \mathbf{\Omega}^b_{eb}\,T_s)(2\mathbf{I}_3 - \mathbf{\Omega}^b_{eb}\,T_s)^{-1} . \tag{3.4}$$

The state space model for the error dynamics (2.28) contains the two time varying matrices $\mathbf{F}(t)$ and $\mathbf{G}(t)$. Assuming the sample rate is high so that the two matrices can be regarded as constant over the integration time T_s, the same method as above can be applied. The discrete time state space model reads

$$\delta\mathbf{x}_{k+1} = \mathbf{\Phi}_k\, \delta\mathbf{x}_k + \mathbf{u}_{d,k} . \tag{3.5}$$

The time discrete state transition matrix is approximated similar to $\mathbf{A}$ in (3.2) with

$$\mathbf{\Phi}_k \approx \mathbf{I} + \mathbf{F}(kT_s)T_s . \tag{3.6}$$

The term $\mathbf{u}_{d,k}$ denotes the discrete-time process noise. Its derivation is shown in [4]. Because it is a linear combination of Gaussian noise, its characteristics are still Gaussian and can be described by its first and second order moments. The mean of $\mathbf{u}_c(t)$ is assumed to be zero, therefore also $\mathbf{u}_{d,k}$ has zero mean. The discrete time noise $\mathbf{Q}_{d,k}$ is as in [1] approximated as

$$\mathbf{Q}_{d,k} \approx \mathrm{diag}(\mathbf{0}_3, \sigma^2_{\mathrm{acc}}\, \mathbf{I}_3, \sigma^2_{\mathrm{gyro}}\, \mathbf{I}_3, \mathbf{0}_6) , \tag{3.7}$$

where $\mathrm{diag}(\cdot)$ denotes a block diagonal matrix. This is due to the fact that $\mathbf{Q}_c(t)$ is a diagonal matrix and due to the orthonormality property of $\mathbf{R}^e_b$.

To conclude this section, the required equations are mentioned here. First, we note the state observation equation as in [4], where $\delta\mathbf{y}_k$ in this case only denotes the difference between the GPS and INS position estimates:

$$\delta\mathbf{y}_k = \mathbf{H}_k \delta\mathbf{x}_k + \mathbf{w}_{d,k} . \tag{3.8}$$

The observation matrix $\mathbf{H}$ is a zero-matrix if no observations are available and contains an identity sub-matrix for the observations:

$$\mathbf{H}_k = \begin{cases} [\mathbf{I}_3, \mathbf{0}_{3\times 12}], & \text{GPS estimates available} \\ \mathbf{0}_{3\times 15}, & \text{otherwise} \end{cases} . \tag{3.9}$$

$\mathbf{w}_{d,k}$ denotes the measurement noise i.e. the error in the GPS position estimates. The required equations for the navigation are now available in discrete time. The extended Kalman filter is presented in the next section.

3.3 The Extended Kalman Filter

In inertial navigation systems, the Kalman filter approach is widely used. Due to the underlying state space model it offers great flexibility e.g. to include further differential equations or to accommodate measurement updates from various sensors. We noticed that the navigation equations consist of non-linear systems. However, a Kalman filter is based on linear operations. One way to overcome this problem is to linearize the system around the output of the INS, i.e. around the filter estimates $\mathbf{z}_k$. This method is known as the *Extended Kalman Filter* (EKF).

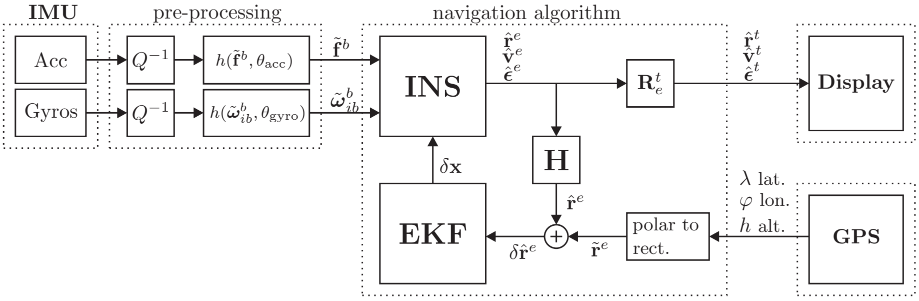

IMU
pre-processing
navigation algorithm
Acc
Gyros
Q^{-1}
Q^{-1}
$h(\tilde{\mathbf{f}}^b, \theta_{\mathrm{acc}})$
$h(\tilde{\boldsymbol{\omega}}^b_{ib}, \theta_{\mathrm{gyro}})$
$\tilde{\mathbf{f}}^b$
$\tilde{\boldsymbol{\omega}}^b_{ib}$
INS
$\hat{\mathbf{r}}^e$
$\hat{\mathbf{v}}^e$
$\hat{\boldsymbol{\epsilon}}^e$
$\mathbf{R}^t_e$
$\hat{\mathbf{r}}^t$
$\hat{\mathbf{v}}^t$
$\hat{\boldsymbol{\epsilon}}^t$
Display
H
$\delta\mathbf{x}$
$\hat{\mathbf{r}}^e$
EKF
$\delta\hat{\mathbf{r}}^e$
$\tilde{\mathbf{r}}^e$
polar to rect.
λ lat.
φ lon.
h alt.
GPS

Figure 3.2: Block diagram of the navigation system with an extended Kalman filter.

The derivation of the Kalman filter will not be given here, instead the reader is referred to Appendix A in [4] which follows the derivations in [16, 18].

As in [4], we define two state vectors. $\mathbf{z}_k$ contains the navigation system outputs: position, velocity, and attitude. Whereas $\mathbf{a}_k$ contains the navigation system inputs: accelerations, and angular rates. These two vectors read

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{r}_k^{eT} & \mathbf{v}_k^{eT} & \boldsymbol{\theta}_k{}^T \end{bmatrix}^T \tag{3.10}$$

$$\mathbf{a}_k = \begin{bmatrix} \mathbf{f}_k^{bT} & \boldsymbol{\omega}^b_{ib,k}{}^T \end{bmatrix}^T . \tag{3.11}$$

With the state vectors from above, we can write the non-linear navigation equations (3.3) and (3.4) as

$$\mathbf{z}_{k+1} = f(\mathbf{z}_k, \mathbf{a}_k) , \tag{3.12}$$

An estimate of the state vectors $\mathbf{z}_k$ and $\mathbf{a}_k$ can be expressed as

$$\hat{\mathbf{z}}_k = \mathbf{z}_k^- + \delta\hat{\mathbf{z}}_k \tag{3.13}$$

and

$$\hat{\mathbf{a}}_k = \mathbf{a}_k^- + \delta\hat{\mathbf{a}}_k , \tag{3.14}$$

where $(\hat{\cdot})$ denotes estimated states and $(\cdot)^-$ predicted states.

To conclude this section, the equations for the navigation algorithm are summarized here.

Prediction

The prediction of the covariances $\mathbf{P}_{k+1}$ is propagated according to the model

$$\mathbf{P}_{k+1}^{-} = \mathbf{\Phi}_k \, \mathbf{P}_k \, \mathbf{\Phi}_k^T + \mathbf{Q}_{d,k} \,, \tag{3.15}$$

with $\mathbf{\Phi}_k$ defined in (3.6) and $\mathbf{Q}_{d,k}$ in (3.7). The state error estimates $\delta\hat{\mathbf{z}}_{k+1}^{-}$ and $\delta\hat{\mathbf{a}}_{k+1}^{-}$ are obtained with

$$\begin{bmatrix} \delta\hat{\mathbf{z}}_{k+1}^{-} \\ \delta\hat{\mathbf{a}}_{k+1}^{-} \end{bmatrix} = \mathbf{\Phi}_k \begin{bmatrix} \delta\hat{\mathbf{z}}_k \\ \delta\hat{\mathbf{a}}_k \end{bmatrix} . \tag{3.16}$$

Kalman Gain

The Kalman gain is computed according to the theory as following:

$$\mathbf{K}_{f,k} = \mathbf{P}_k^{-} \, \mathbf{H}_k^T \, (\mathbf{H}_k \, \mathbf{P}_k^{-} \, \mathbf{H}_k^T + \mathbf{R}_{d,k})^{-1} \,. \tag{3.17}$$

$\mathbf{R}_{d,k}$ is the covariance of the measurement noise $\mathbf{w}_{d,k}$ that is mentioned in Equation (3.8).

Measurement Update

Finally, in the measurement update we use the previously defined matrices and vectors to update the states (for the derivation of $\mathbf{P}_k$ refer to [4]):

$$\begin{bmatrix} \delta\hat{\mathbf{z}}_k \\ \delta\hat{\mathbf{a}}_k \end{bmatrix} = \begin{bmatrix} \delta\hat{\mathbf{z}}_k^{-} \\ \delta\hat{\mathbf{a}}_k^{-} \end{bmatrix} + \mathbf{K}_{f,k} \left(\mathbf{y}_k - \mathbf{H}_k \begin{bmatrix} \mathbf{z}_k^{\mathrm{nom}} \\ \mathbf{a}_k^{\mathrm{nom}} \end{bmatrix} - \mathbf{H}_k \begin{bmatrix} \delta\hat{\mathbf{z}}_k^{-} \\ \delta\hat{\mathbf{a}}_k^{-} \end{bmatrix} \right) \tag{3.18}$$

$$\mathbf{P}_k = \mathbf{P}_k^{-} - \mathbf{K}_{f,k} \, \mathbf{H}_k \, \mathbf{P}_k^{-} \tag{3.19}$$

The implemented navigation algorithm has two branches. If no GPS data is available, Equations (3.14), (3.12), (3.16) where only $\delta\hat{\mathbf{a}}_{k+1}^{-}$ is updated with $[\mathbf{\Phi}_k]_{10:15,10:15}$, (3.15), and (3.4) are employed.

If GPS data is available, here every 100th sample, Equations (3.17), (3.18), (3.13), (3.14), (3.19), (3.12), (3.16), (3.15), (3.4) are employed.

3.4 Observability Analysis

A Kalman filter for an INS can easily contain many states. It is common to use 15 states, but the more errors are modeled, the more states are required. Usually, the number of states is higher than the number of observations and it might be possible that some states are not observable. This happens if some states do not affect certain observations or if they influence the observations in the same way. To determine the observability of a system, a matrix can be set up (see [1, 19]) as

$$\mathbf{O}(\mathbf{H}(k), \mathbf{\Phi}(k)) = \begin{bmatrix} \mathbf{H}(k-m+1) \\ \mathbf{H}(k-m+2)\mathbf{\Phi}(k-m+1) \\ \mathbf{H}(k-m+3)\mathbf{\Phi}(k-m+2)\mathbf{\Phi}(k-m+1) \\ \vdots \\ \mathbf{H}(k-1)\mathbf{\Phi}(k-2)\cdots\mathbf{\Phi}(k-m+2)\mathbf{\Phi}(k-m+1) \\ \mathbf{H}(k)\mathbf{\Phi}(k-1)\cdots\mathbf{\Phi}(k-m+2)\mathbf{\Phi}(k-m+1) \end{bmatrix} , \tag{3.20}$$

where $\mathbf{H}(k)$ denotes the observation matrix at the discrete time instant k, $\mathbf{\Phi}(k)$ the state transition matrix from the discrete time instant $k-1$ to k, and m the number of states. If the matrix has full rank, i.e. the rank of the matrix $\mathbf{O}$ is m, the system is fully observable.

The rank of the matrix $\mathbf{O}$ was evaluated in MATLAB. After the first iteration, the observability is 6, after the second 9, after the third 12, and after four iterations, the system becomes fully observable with rank 15. This is due to the fact, that there is no position measurement available at the first iteration. Since the position is obtained after integrating the velocity and this in turn after integrating the accelerations, the position can only be estimated after two iterations.

Considering the alignment of the INS, there are several publications that investigated the influence in connection with the observability. [20] deals with the ground alignment although they only consider a system with the horizontal velocity errors and the attitude errors. This simplification is reasonable since the vehicle is kept stationary. In [19] however, an in-flight alignment is described where it is proposed to use all states during the alignment process and later switch to reduced states, i.e. to damp the z accelerometer bias and velocity. The observability of the 18 error states position, velocity, acceleration, accelerometer bias, gyro bias, and lever-arm is discussed in [21] and may be considered in the future work.

3.5 Numerical Considerations for the Covariance Matrix P

The two requirements on the covariance matrix $\mathbf{P}_k$ are that it should be symmetric and positive semi-definite. With the Equation (3.19), it could happen that these two requirements cannot be met. There exist more sophisticated methods to guarantee symmetric and positive semi-definite covariance matrices. One method lies behind the idea of a *Square Root Algorithm.* [18] presents several types of this method. Since the square root of the covariance matrix is updated ($\mathbf{P}^{1/2}$), the span between the largest and smallest elements is decreased. Furthermore, it provides a higher accuracy with a finite numerical precision, but increases also the computational cost. Another method is called the *Symmetric Joseph Form* as explained in [22], and is defined as follows:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_{f,k}\mathbf{H}_k)\mathbf{P}_k^{-}(\mathbf{I} - \mathbf{K}_{f,k}\mathbf{H}_k)^T + \mathbf{K}_{f,k}\mathbf{R}_{d,k}\mathbf{K}_{f,k}^T \,. \tag{3.21}$$

The sum of the two symmetric matrices, the first being positive definite and the second positive semi-definite, is supposed to better prevent from an ill-conditioned matrix that could result in filter divergence.

3.6 Filter Tuning

The accuracy and behavior of a Kalman filter relies on the values placed in the different covariance matrices. The process of finding those values that lead to an optimal result is called tuning. It is a delicate adjustment of particularly the $\mathbf{Q}$ and $\mathbf{R}$ matrices whose influences are described in the next paragraphs.

The system noise or state covariance matrix **Q** provides the statistical description of the error model. A large value in **Q** indicates increased parameter uncertainty and results in noisy estimates. During the prediction, the uncertainty in the IMU data will grow. A GPS position estimate will then correct the INS more, independent of its accuracy. In other words, a large value in **Q** will cause the INS to closely follow the GPS position estimates. This in turn, will lead to an inaccurate trajectory if the GPS estimates are noisy. To prevent this, a small value could be chosen that will lead to smooth, but biased estimates.

How well the measurement noise is modeled, is determined by the measurement noise covariance matrix **R**. Imperfect modeling of the noise of the measurement observables, i.e. ignoring the fact that the noise has non-white properties, leads to a bad estimation quality. Unfortunately, the GPS receiver that is used for this work outputs already filtered data, which means that the noise is not white anymore. Currently, this fact is disregarded.

Choosing a large value for **R** reflects inaccurate and noisy measurements and might not correct the INS sufficiently. Otherwise, a small value implies an accurate measurement and will cause the system to rely more on the measured data than on the model.

The velocity terms are reflected in both the position and attitude evaluations [23]. Therefore, the less uncertainty in the velocity terms, the higher the dampening on the attitude errors. If the attitude is affected by large uncertainties, it will cause oscillatory corrections in the attitude.

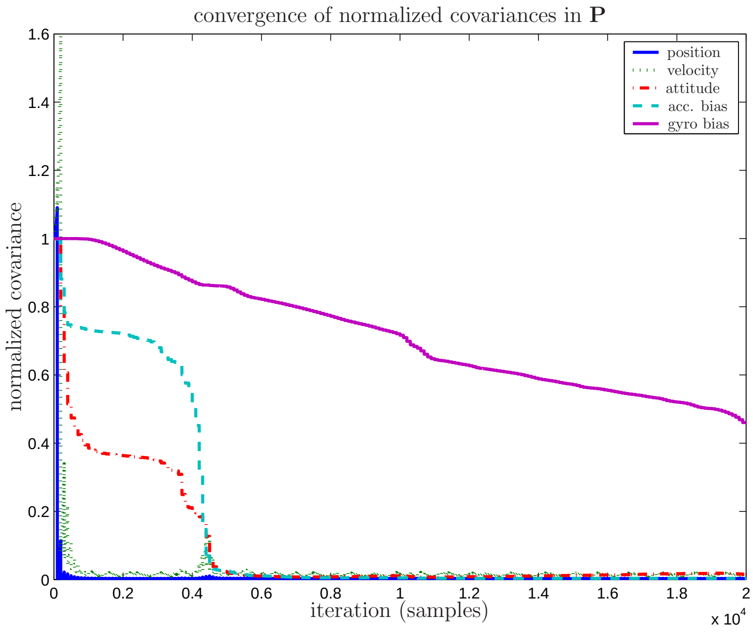


Figure 3.3: Convergence of error covariances for the different states. Each curve represents the norm of the three error state covariances, normalized with respect to the initial value. The data used for this graph is from the trajectory of the field test (see 5.1.2).

Finally, the error covariance matrix $\mathbf{P}$ is predicted and updated according to Equation (3.15) and (3.19). However, the initial values have to be preset. This *a priori* information in $\mathbf{P}_0$ is not uncritical. Even though $\mathbf{P}_0$ does not affect the Kalman filter transient duration and steady state conditions, it yields a different magnitude transient characteristic. It should also be noted that strongly observed states converge first (e.g. the position), whereas weakly observed states (e.g. sensor biases) take more time to converge, see Figure 3.3. As long as not all the states have converged, the estimation obviously leads to poor results. Therefore, a good choice of the initial values is important.

3.7 Lever-Arm Correction

In reality, the IMU and GPS antenna cannot be placed at exactly the same position on the vehicle. This spatial separation causes the IMU and GPS to sense slightly different positions and velocities, which is called the *Lever-Arm* effect [1]. If this separation is large, the position and velocity has to be corrected according to the following relations [24]:

$$\mathbf{r}^e_{\text{IMU}} = \mathbf{r}^e_{\text{GPS}} - \mathbf{R}^e_b\,\Delta\mathbf{r}^b_{\text{LA}} \tag{3.22}$$

$$\mathbf{v}^e_{\text{IMU}} = \mathbf{v}^e_{\text{GPS}} + \boldsymbol{\Omega}^e_{ie}\,\mathbf{R}^e_b\,\Delta\mathbf{r}^b_{\text{LA}} - \mathbf{R}^e_b\,\boldsymbol{\Omega}^b_{ib}\,\Delta\mathbf{r}^b_{\text{LA}} \tag{3.23}$$

The offset vector $\Delta\mathbf{r}^b_{\text{LA}}$ denotes the displacement of the GPS antenna to the IMU in the body frame, as illustrated in Figure 3.4.

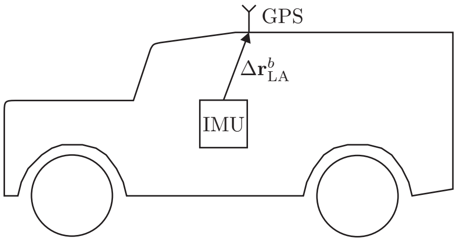


Figure 3.4: Spatial separation of the IMU and GPS antenna, known as the lever-arm effect.

3.8 Calibration

For the accuracy and convergence time of an INS, it is of great advantage to calibrate the MEMS sensors. However, low-cost sensors such as the ones used for this project, can have different biases and scale factors after every switch-on. Furthermore, the biases and scale factors can change during the calibration process and certainly do with a change in temperature. This limits the impact of an extensive lab calibration. As a complement, field calibration methods should be employed to increase the accuracy.

To asses the problem of the bias, we first consider the acceleration on the x-axis. To the measured force f_x we only add a small bias b_{a_x} and assume the gyro measurements to be error-free. By integrating $a_x = f_x + b_{a_x}$ twice with respect to time, we notice that the position error increases proportional to the square of time. Now, suppose that the acceleration measured on the x-axis f_x is error-free, there is no noise but a small and constant bias b_{ω_z} in the z-axis gyro and no angular rotation is performed. The resulting acceleration error for the x-axis will be

$$a_x = f_x \sin(b_{\omega_z} t)\,. \tag{3.24}$$

Assuming a small angle increment (i.e. high update rate), we can simplify the above equation and do the integration to get the position:

$$a_x = f_x b_{\omega_z} t \tag{3.25}$$

$$v_x = \frac{1}{2} f_x b_{\omega_z} t^2 \tag{3.26}$$

$$r_x = \frac{1}{6} f_x b_{\omega_z} t^3 \tag{3.27}$$

Hence, a bias in the gyro measurement causes a position error which grows with the cube of time. Furthermore, a noise term will lead to a random walk due to the integration. Therefore it is crucial for the system accuracy that the gyro biases are known and can be compensated and that the noise is modeled correctly.

A relatively simple method to calibrate the accelerometer and gyro parameters is presented in [25] and summarized bellow.

3.8.1 Accelerometers

First, an error model for the MEMS sensors is defined. The errors with the most influence are identified as misalignment, scale factor, and bias. The misalignment originates from the nonorthogonality of the hardware construction. Without precise instruments and tools, it is almost impossible to mount the sensors so that their axes are perfectly orthogonal. If the misalignment angles are small, a transformation matrix as in Equation (3.28) can be used to correct the sensor data [2]. The angles α_{ij} denote the rotation of the i-th accelerometer axis around the j-th platform axis.

$$\mathbf{T}_a^p = \begin{bmatrix} 1 & -\alpha_{yz} & \alpha_{zy} \\ \alpha_{xz} & 1 & -\alpha_{zx} \\ -\alpha_{xy} & \alpha_{yx} & 1 \end{bmatrix} \tag{3.28}$$

Here, the subscript a and superscript p indicate a transformation from the accelerometer to the platform frame. Further, the scale factors can be defined as $\mathbf{K}_a = \mathrm{diag}(k_{x_a}, k_{y_a}, k_{z_a})$, and the biases as $\mathbf{b}_a = [b_{x_a}, b_{y_a}, b_{z_a}]^T$. Using these parameters, the error model for the measured output of a sensor cluster can be written as

$$\tilde{\mathbf{f}}^a = \mathbf{K}_a (\mathbf{T}_a^p)^{-1} \mathbf{f}^p + \mathbf{b}_a + \mathbf{v}_a\,, \tag{3.29}$$

with additional measurement noise $\mathbf{v}_a$.

If the IMU is stationary, the gravity $\mathbf{g}_{\mathrm{stat}}^p$ is the only force sensed by the accelerometers. Its value can be assumed constant and approximately calculated e.g. with the Earth model WGS-84 [3]. With a nonlinear numerical least squares search over the cost function

$$\{\hat{\mathbf{T}}_a^p, \hat{\mathbf{K}}_a, \hat{\mathbf{b}}_a\} = \underset{\mathbf{T}_a^p, \mathbf{K}_a, \mathbf{b}_a}{\mathrm{argmin}} \left(\sum_{n=0}^{N-1} \left(\left\| \mathbf{g}_{\mathrm{stat}}^p \right\|^2 - \left\| \mathbf{T}_a^p \mathbf{K}_a^{-1} (\tilde{\mathbf{f}}_{\mathrm{stat},n}^a - \mathbf{b}_a) \right\|^2 \right)^2 \right)\,, \tag{3.30}$$

the calibration parameters can be found. Note that the search is performed in the proximity of the approximately known gravity. Since the norm of the vectors is used, this method does not require a precise mechanic platform. In a two-step procedure, first $\tilde{\mathbf{f}}^a_{\text{stat}}$ for nine different orientations is measured (totally N samples) and then the parameters are estimated using the described algorithm.

3.8.2 Gyros

The same calibration algorithm as described in Section 3.8.1 can be applied for the calibration of the gyros. However, a constant rotating platform is needed. This rotation velocity will correspond to the gravity force.

By the time this thesis was written, no appropriate instruments were available to perform a gyro calibration with this method. Instead, a very simple and relatively inaccurate method was used in order to find approximate calibration parameters. If the IMU is stationary, the gyros will not sense any rotation and therefore the output signals are assumed to reflect the biases $\mathbf{b}_g$. To find approximate scale factors $\mathbf{K}_g$, the IMU was rotated 90° for each axis. Then, these measured angular velocities were integrated over the time and compared with π rad. Finally, the misalignment matrix $\mathbf{T}^p_g$ was assumed to be an identity matrix.

3.9 Accuracy Improvement

This section will propose several methods to increase the precision of the estimated position from the GPS aided INS.

3.9.1 Nonholonomic Constraints

In a nonholonomic system, the controllable degrees of freedom are less than the total degrees of freedom. For example a car, considered as an object, could be moved in all three orthogonal axes. Although in almost all the cases, a car does not slide in the plane perpendicular to the forward direction or jump off the ground. Hence, it is possible to set up two nonholonomic constraints on the velocity in y- and z-direction, respectively. The method applying these constraints is presented in [26]. The velocities are first transformed into the body-frame:

$$\hat{\mathbf{v}}^b = \mathbf{R}^b_e \hat{\mathbf{v}}^e \,. \tag{3.31}$$

Now, it is possible to apply the constraints and obtain

$$\begin{bmatrix} v^b_y \\ v^b_z \end{bmatrix} = \begin{bmatrix} \hat{v}^b_y \\ \hat{v}^b_z \end{bmatrix} + \begin{bmatrix} \nu_y \\ \nu_z \end{bmatrix} \tag{3.32}$$

where the constraint violation is modeled as Gaussian white noise ν_y and ν_z with variance $\sigma^2_{\text{cons}_{v_y}}$ and $\sigma^2_{\text{cons}_{v_z}}$, respectively:

$$\begin{aligned} v^b_y - \nu_y &= 0 \qquad (3.33) \\ v^b_z - \nu_z &= 0 \,. \qquad (3.34) \end{aligned}$$

The forward speed, i.e. $\tilde{v}^b_x$ has to be obtained from a further sensor, either a wheel encoder, from the GPS Doppler measurements, or another source. An error signal can be generated as $v_x^b - \tilde{v}_x^b = \nu_x$ with variance $\sigma^2_{\mathrm{enc}_{v_x}}$. The measurement covariance matrix for this part is then

$$\mathbf{R}_{\mathrm{cons}} = \begin{bmatrix} \sigma^2_{\mathrm{enc}_{v_x}} & 0 & 0 \\ 0 & \sigma^2_{\mathrm{cons}_{v_y}} & 0 \\ 0 & 0 & \sigma^2_{\mathrm{cons}_{v_z}} \end{bmatrix} . \tag{3.35}$$

Since the ECEF-frame is used in the navigation algorithm, the velocity vector needs to be transformed back from the b- to the e-frame, and the measurement covariance matrix has to be transformed by

$$\bar{\mathbf{R}}_{\mathrm{cons},k} = \mathbf{R}^e_{b,k} \, \mathbf{R}_{\mathrm{cons}} \, {\mathbf{R}^e_{b,k}}^T , \tag{3.36}$$

where k indicates time instant of the corresponding time varying matrix.

Experimental results using nonholonomic constraints are presented in Section 5.2.2.

3.9.2 Field Calibration

As mentioned in Section 3.8, biases and scale factors from low-cost IMUs can change from one switch-on to the next. In other words, a lab calibration will not accurately reflect the parameters that will be present during a field run. Therefore, it may be appropriate to develop a field calibration for those parameters. Since a strapdown IMU can not easily be rotated, it will constrain the number of calibration parameters that can be obtained. Modeling and including more error states in the extended Kalman filter could provide a better accuracy in terms of the position and attitude estimation. However, care has to be taken with respect to the observability so that the system will not have unobservable states. This might imply the incorporation of more measurement sensors such as tilt or velocity sensors. The performance of the alignment in context of the trajectory is analyzed in [27] where the same 15 states are used as in this project, but in the navigation frame (*North-East-Down-system*). This could be used as a base for an in-motion alignment.

3.9.3 Zero-Velocity Update

For low-cost IMUs it is necessary to bound the errors to an acceptable level. In the past, good results could be obtained using *zero velocity updates* (ZUPT). The idea is to stop the vehicle from time to time and run a Kalman filter to update the error states and covariances accordingly. However, this method is not applicable for airborne systems, but can be used as a help for land vehicles especially in urban traffic situations.

Chapter 4

TRÖGE Platform for Inertial Navigation

As mentioned in the introduction, there exists an experimental platform called TRÖGE[1] at KTH S3. It allows studies, tests, and performance measurements in the automotive field. The whole system is flexible and expandable to accommodate the needed subsystems or new future subsystems for other applications. An overview of the system is also presented in [28].

This chapter will give a brief overview of the systems needed for the *GPS aided Inertial Navigation* application. The existing hardware consisting of the testbed, the IMU, and a GPS receiver is described. Further, the design of the developed software is documented to facilitate future development and extension.

4.1 Hardware

The following subsections describe the three relevant hardware units that are used for this project: testbed, IMU, and GPS. They are depicted in Figure 4.1 as a block diagram, where the arrows indicate the information flow.

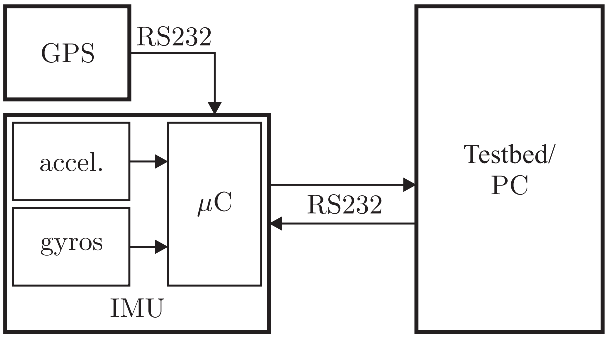


Figure 4.1: Block diagram of the hardware that is used for this project.

---

[1] TRÖghetsnaviGEring - Swedish for inertial navigation

4.1.1 Testbed

The TRÖGE testbed has been developed at KTH/S3. This platform consists of an industrial PC capable of real-time signal processing with a Pentium-M 1.7 GHz processor, 512 MB RAM, 60 GB HD, a 7” TFT touch-screen, a CD-ROM drive, several USB ports, a serial RS232 interface, as well as WLAN and GSM/GPRS for wireless communication. The system can run from three power sources, namely from 230 VAC mains, external 11 - 15 VDC supply, and from internal batteries allowing 2 hours of stand alone operation, enabling a fully mobile operation. A picture of the testbed is shown in Figure 4.2.

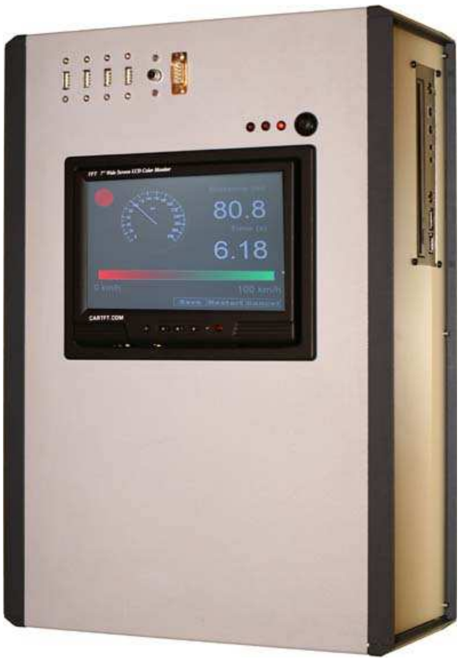


Figure 4.2: The TRÖGE testbed comprising an industrial PC, a touch-screen display, and several peripheral devices.

4.1.2 Inertial Measurement Unit

A six degree-of-freedom IMU has been constructed from scratch at KTH/S3. It comprises of state-of-the-art MEMS gyros (Analog Devices, ADXRS-150) and accelerometers (Analog Devices, ADXL-201). The three sensors span a quasi-orthogonal coordinate system. Figure 4.3 shows the developed hardware. The inevitable misalignment has to be corrected in the preprocessing of the sensor data, according to the values found with the calibration (see Section 3.8).

The heart of the IMU is an 8-bit microcontroller (Atmel, ATmega8) with a built-in 10-bit analog to digital converter. Using the also built-in analog multiplexer, it is possible to consecutively sample the six sensors. The sampling rate per sensor is 100 Hz, synchronized once every second based on a certain GPS message. The GPS is connected with the IMU over a RS232 interface. A second RS232 interface is used to allow an interaction with a host PC, whereas the PC

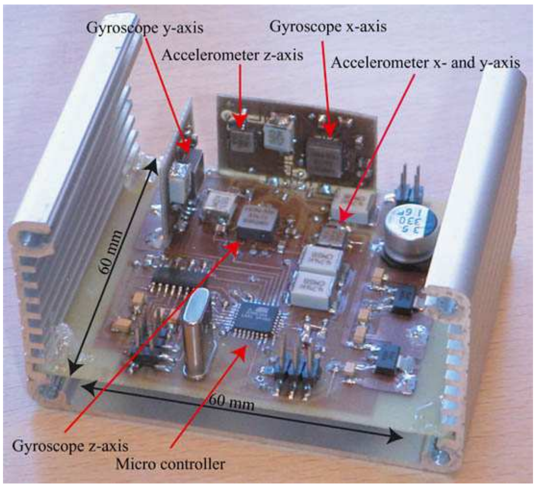

Gyroscope y-axis
Gyroscope x-axis
Accelerometer z-axis
Accelerometer x- and y-axis
60 mm
60 mm
Gyroscope z-axis
Micro controller

Figure 4.3: The IMU that was developed and constructed at KTH/S3.

acts as a master by sending commands to the IMU. Additional modifications were made to the firmware and hardware. The current behavior of the firmware is described in Section 4.2.1. In the hardware, three amplifiers were added to amplify the output signal of the gyros. The resulting change in the scale factors is accounted for in the data pre-processing.

4.1.3 GPS Receiver

The GPS receiver is a low-cost off-the-shelf product from the manufacturer GlobalSat (Figure 4.4). The BR304 is equipped with a SiRF Star II/LP chipset, handles 12-channel parallel processing, and outputs the NMEA 0183 (V2.2) protocol (commands: GGA, GSA, GSV, RMC, GLL, VTG) on a RS232 serial interface. The interface speed is 4800 baud and the used datum for the position calculation is WGS84. The receiver is powered over an additional power cable with 5 VDC and consumes approximately 80 mA.

Figure 4.4: GPS receiver BR304 from GlobalSat.

Since the IMU sampling is triggered with the GGA message that is output approximately every 1000 ms, the IMU trigger does not correspond with the actual time when the GPS position-fix was computed. This latency due to the used protocol and the interface speed affects the accuracy of the navigation system. Using an absolute time reference that is synchronous with UTC, it was possible to estimate this latency. As a time reference, the demodulated DCF77 time signal was used. The travel time that the radio signal experiences from Mainflingen (Germany) to Burgdorf (Switzerland) was shown to be negligible for the required precision. The measurement was done by logging the GPS time from the receiver module and measuring the time difference between the DCF77 impulse and the first character on the serial port. It turned out that the latency is in the order of 45 ms and corresponds to approximately 4 IMU samples. A constant delay in the GPS aided INS can lead to inaccurate results. A method to calibrate this measurement delay is described in [29]. However, such a calibration is beyond the scope of the current work.

4.2 Software Implementation

The implementation is done using the language C++ . In order to ease and simplify the implementation of the signal processing algorithms, the library IT++ is used. It is a collection of mathematical, signal processing, speech processing, and communications classes and functions that are commonly used and resemble the ones in MATLAB. IT++ is available for free on the Internet[2].

4.2.1 Data Acquisition

The IMU is accessed through a simple serial RS232 interface. By using the *GPS/IMU module Serial Communication Protocol* (GISCP) that is described in [30], the IMU can be controlled and the measurements be obtained. To understand how the IMU works, its state diagram is shown in Figure 4.5. The text along the arrows denotes what triggers the event to change the state. The abbreviation *cmd* indicates a command send from the host PC to the IMU. The data flow is seen from the IMU, i.e. a reception *rx* is a reception of data coming from the host PC, and a transmission *tx* is a transmission of data from the IMU to the host PC. Having Figure 4.5 as a reference, it is straightforward to

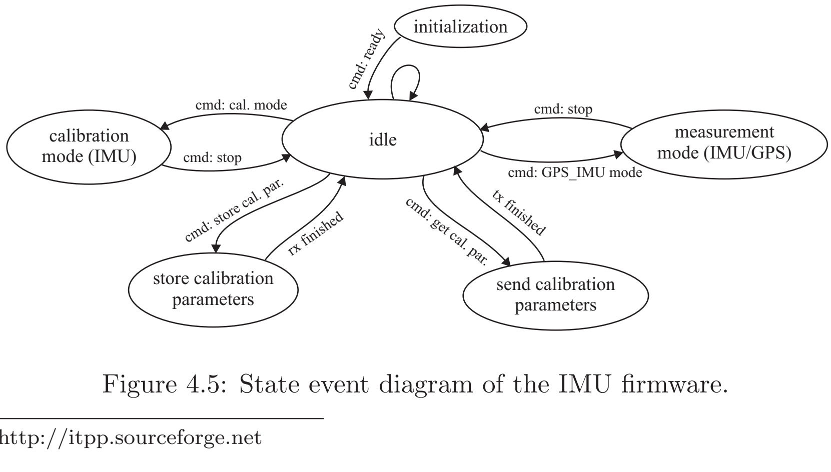


Figure 4.5: State event diagram of the IMU firmware.

---

[2] http://itpp.sourceforge.net

implement a data acquisition routine. At the start-up, a *ready* command has to be send from the host PC to the IMU. Then, to obtain IMU samples and GPS data, the command *GPS_IMU mode* has to be sent. The IMU will synchronize to the GPS messages, start sampling the IMU data and transmit both IMU and GPS data to the host PC. To stop the process, the host PC simply needs to send a *stop* command, and the IMU returns into its idle mode.

The message structure and data format of the data is described in [30]. The data acquisition routine buffers the data obtained from the IMU. Since needed data from the GPS is distributed over the three messages GGA, GSA, and RMC, they cause a latency in the processing. To simplify the design, it was agreed to constantly delay the processing with one second, so that all needed data are available by the time they are needed, i.e. the output is one second delayed compared with the input.

4.2.2 Navigation System Implementation

The INS algorithm was implemented in C++ using IT++ and is available as a class. In order to test the implementation but also to run the algorithm from MATLAB and speed up the processing time, a MEX[3] wrapper was developed, that serves as a gateway routine to interface the navigation algorithm with MATLAB. In the gateway, all input arguments are first checked and copied into the processing buffers of the C++ INS implementation. Then, the INS algorithm is called to process the available data. Finally, the MATLAB variables for the output are allocated and filled with the data. Figure 4.6 illustrates the interaction between the three components, the MATLAB environment, the MEX wrapper as a DLL, and the INS algorithm as a C++ class instanced in the DLL.

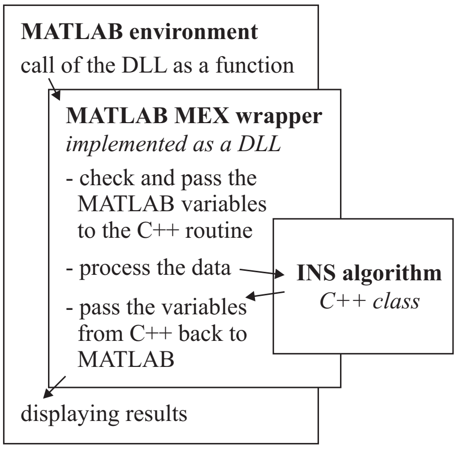

MATLAB environment
call of the DLL as a function
MATLAB MEX wrapper
implemented as a DLL
- check and pass the MATLAB variables to the C++ routine
- process the data
- pass the variables from C++ back to MATLAB
INS algorithm
C++ class
displaying results

Figure 4.6: Visualization of the MATLAB–MEX wrapper–processing algorithm interaction.

The INS algorithm implementation is rather simple and through the use of IT++ it resembles the MATLAB implementation. The flow chart in Figure 4.7 shows the order of the processing steps with the corresponding function names.

---

[3] Matlab EXternal, see the MATLAB help under *External Interfaces/API*

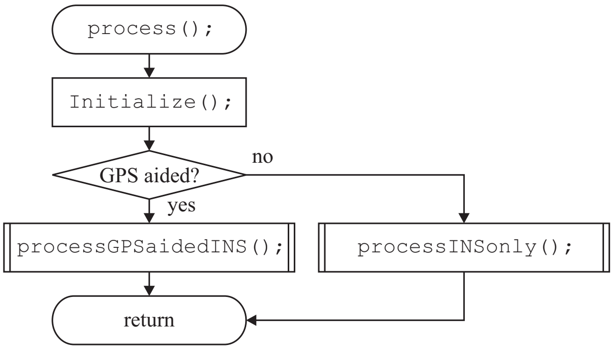

process();
Initialize();
GPS aided?
no
yes
processGPSaidedINS();
processINSonly();
return

Figure 4.7: Flow chart of the INS implementation.

The functions in the flow chart of Figure 4.8 perform the following computations. The function `BiasCorrection()` corresponds to Equation (3.14). In the `INSalgorithm()`, the state vector **z** is updated (3.12), the prediction according to Equation (3.15), and the DCM according to Equation (3.4). If no valid GPS

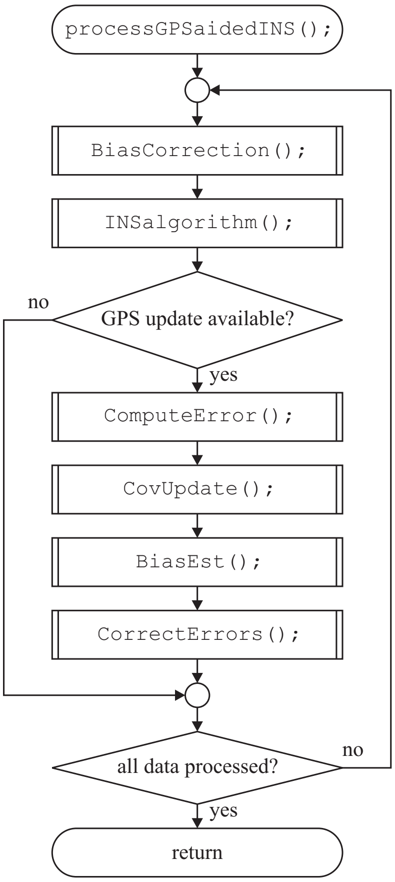

processGPSaidedINS();
BiasCorrection();
INSalgorithm();
no
GPS update available?
yes
ComputeError();
CovUpdate();
BiasEst();
CorrectErrors();
all data processed?
no
yes
return

Figure 4.8: Flow chart of the GPS aided INS implementation.

data is available, what can be determined according to the *working mode* in the message GSA, the loop ends here and iterates from the beginning. In case, valid GPS data is available, the Kalman innovation is computed in `ComputeError()` that implements the terms in the brackets of Equation (3.18). The Kalman Gain as in Equation (3.17) and the covariance update from Equation (3.19) are computed in `CovUpdate()`. The function `BiasEst()` updates the measurements with the predicted biases as in Equation (3.18), and finally `CorrectErrors()` computes Equation (3.13) and updates the DCM $\mathbf{R}_b^e$. If all available data was processed, the function returns here, otherwise it iterates again.

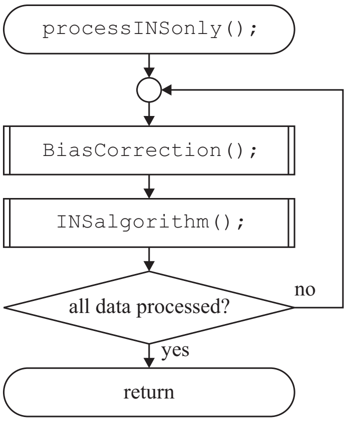


Figure 4.9: Flow chart of the INS only implementation.

In case only INS data should be processed without GPS updates, the function `processINSonly()` in Figure 4.9 is called. The two functions `BiasCorrection()` and `INSalgorithm()` that are called, are the same as described in the previous paragraph.

Chapter 5

Results and Analysis

The basic navigation algorithm for the GPS aided INS was simulated in [4]. Before the navigation algorithm provided reasonable results, many things had to be done. This chapter describes the analysis and results for the integrated system using real-world data from a field test. First, this field test is described, then the results are presented and compared, and finally, an analysis of error sources is provided to emphasize the influence of the different parameters on the output, for the future integration of more error models.

5.1 Field Test

Several field tests were conducted for the test and analysis with the hardware. During these tests, the data was only collected and stored for later processing and post-mission analysis. The last test from which also the data for the results in this chapter are used, took place on 9 December 2005. The following subsections report this field test.

5.1.1 Equipment

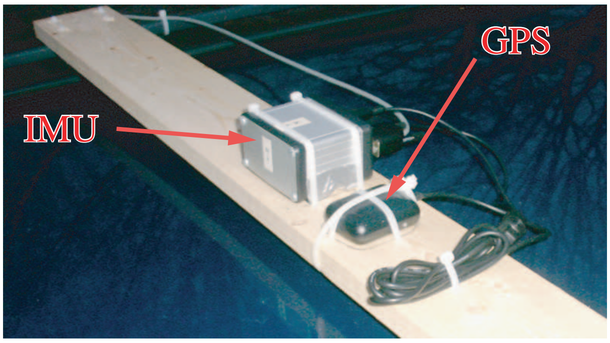


Figure 5.1: The IMU and GPS mounted on the roof of a car.

The low-cost IMU that was developed at KTH and shortly described in Section 4.1.2 was used together with the low-cost GPS receiver described in

Section 4.1.3. These two instruments were mounted on a wooden bar which was fixed to the roof railing of a car, see Figure 5.1. For the data collection, a Notebook was used with a self written software to log all the data to a file. Before the IMU was mounted on the car, a lab calibration of the accelerometers according to Section 3.8.1 was performed.

5.1.2 Trajectory

A map illustrating the area where the test took place is showed in Figure 5.2. The area is situated just north of the city Stockholm. Overlaid are the GPS receiver position estimates, marked with red crosses, and the estimated trajectory with a blue solid line. First, the vehicle was started in Bergshamra and kept stationary for the first 30 seconds. After a small left turn, a wider right turn was driven to approach a highway. Then, the vehicle was driven along the highway for approximately 2.5 minutes, finally heading southward. During that time, no GPS outages occurred, although the PDOP (see Section 2.5.2) reached values as high as 40 at a few points. During the other times, the value was below 5 for 72% of the whole observation time and can be considered as acceptable.

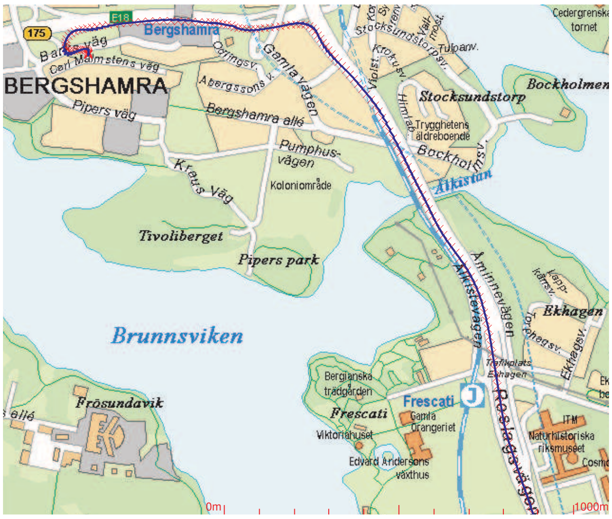

BERGSHAMRA
Bergshamra
E18
175
Carl Malmstens väg
Pipers väg
Abergssons v.
Bergshamra allé
Gamla vägen
Pumphus-vägen
Koloniområde
Kretsväg
Tivoliberget
Pipers park
Brunnsviken
Frösundavik
Stocksundstorp
Bockholmen
Trygghetens äldreboende
Bockholmsv.
Tulpanv.
Cedergrenska tornet
Ålkistan
Åminnevägen
Ålkistevägen
Ekhagen
Bergianska trädgården
Frescati
Gamla Orangeriet
Viktoriahuset
Edvard Andersons växthus
Naturhistoriska riksmuseet
Roslagsvägen
0m
1000m

Figure 5.2: Estimated trajectory (blue solid line) and GPS-receiver position estimates (red crosses) from the field test. (Map: © 2006 Stockholms stadsbyggnadskontor)

Due to the lack of appropriate equipment, it was not possible to log a "reference" trajectory, e.g. with DGPS. Therefore, an exact position error evaluation is not possible and no absolute conclusions can be drawn as long as the "true" trajectory is not available for comparison.

The velocity during the field test is shown in Figure 5.3. This forward velocity information is obtained from the GPS receiver and does not necessarily reflect the true velocity.

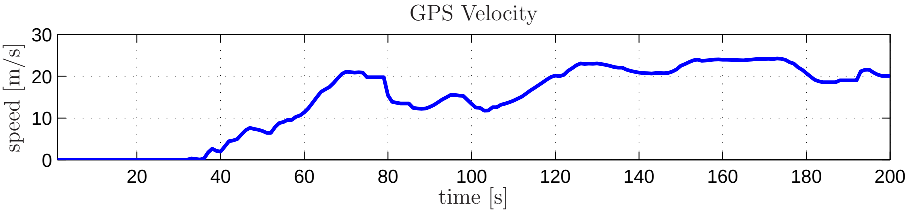


Figure 5.3: Estimated Velocity obtained from the GPS Receiver.

5.1.3 Data Pre-Processing

The obtained raw data from the IMU containing the samples from the accelerometers, gyros, but also GPS messages, needs to be pre-processed before it can be used for the navigation algorithm. The sensor data is represented with 10 bits, i.e. an integer value between 0 and 1023. These values have to be dequantized according to

$$\tilde{\mathbf{f}}'_b = \frac{5.0\,\mathrm{V}}{1024} \cdot g \cdot (\tilde{\mathbf{f}}''_b - 511) \quad [\mathrm{m/s^2}] \tag{5.1}$$

for the acceleration measurements $\tilde{\mathbf{f}}''_b$, where g equals the local gravity with the unit m/s^2/V. The dequantization equation for the angular rates $\tilde{\boldsymbol{\omega}}''_b$ reads

$$\tilde{\boldsymbol{\omega}}'_b = \frac{5.0\,\mathrm{V}}{1024} \cdot \frac{\pi}{180^\circ \cdot 12.5\,\mathrm{mV}/^\circ/\mathrm{s}} \cdot (\tilde{\boldsymbol{\omega}}''_b - 511) \quad [\mathrm{rad/s}]\,. \tag{5.2}$$

After the dequantization, the bias, scaling, and misalignment have to be corrected. For the accelerations, the parameters obtained through the algorithm described in 3.8.1 are used. Since it is not possible to get those parameters for the gyros, the bias is assumed to be the measured value when the IMU is at rest. The scale factors were corrected according to Section 3.8.2.

The received GPS messages first need to be parsed to extract the desired information. The used protocol is NMEA 0183. The latitude, longitude, and height above mean sea level are extracted from the GGA message, the PDOP and working mode from the GSA message, and the speed over ground from the RMC message. The latitude, longitude, and height together are converted into rectangular coordinates (x, y, z) and the speed needs to be converted from knots to meters per second.

In order to get a measure of how accurate a GPS position estimation is, the Position Dilution of Precision can be assessed. Figure 5.4 shows the PDOP (red solid line) and the number of used satellites (blue circles) in respect of the time. As expected, there is a correlation between the number of satellites and the PDOP value.

Around 80 s, the number of satellites reaches zero for two times. At the first time, the highway passes eastward under a large building complex, and the second time under a bridge. This can also be seen on Figure 5.2, just above and a bit to the right of the blue name *Bergshamra*. However, it is not clear why the number of satellites decreases to zero between 180 s and 200 s. One reason could be the high buildings of *Stockholm Universitetet* east of the trajectory and an adjacent forest on the other side.

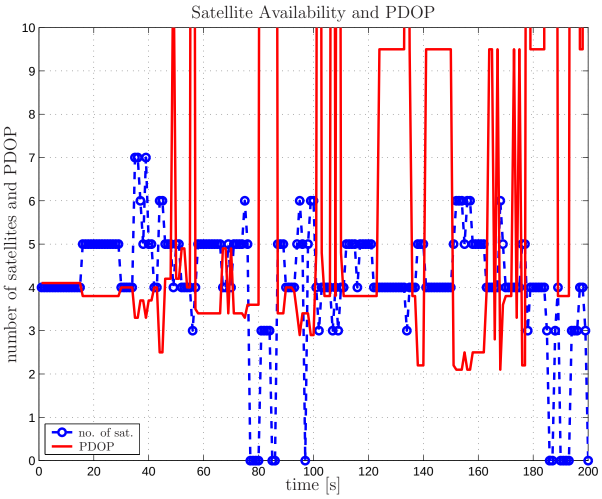


Figure 5.4: Satellite availability and PDOP value during the first 200 s of the field test.

5.2 Experimental Results

This section will present the results obtained with the real-world data from the field test. First, the basic algorithm that was used in [4] is evaluated. Following that, the results with the navigation algorithm enhanced with nonholonomic constraints are presented.

5.2.1 Basic GPS aided INS

Without appropriate tuning, it was not possible to obtain any results. The filter diverges relatively easy if the covariances are not in a specific range. Figure 5.5 shows the sensor biases with respect to time. The accelerometer biases converge after approximately 70 seconds while the gyro biases seem to converge after 100 seconds, which corresponds to the trajectory turn from east to south-east.

From the estimated yaw in Figure 5.6 it can be seen that there is a constant rotation after 100 seconds. This indicates that the estimated bias for the z-axis gyro is wrong. A possible explanation is that there is a delay between the time instant when the GPS position estimates are obtained and the time instant when the IMU samples are taken (see [29]).

Figure 5.7 shows the first 110 seconds of the resulting estimated trajectory. The blue crosses are the GPS position estimates used to aid the navigation system and the estimated GPS aided INS position is marked with a red solid line. Note that there is no reference ("true") trajectory available. The GPS position estimates can deviate from the true position.

Figure 5.8 shows another detail of the estimated trajectory between 138 s and 167 s. From 151 s until 153 s there is a simulated GPS outage (light gray crosses). Because of the large heading error, the estimated position is drifting away and reaches about 17 m in East direction and -5 m in North. But the high confidence in the GPS estimates brings the position back with the next available measurement.

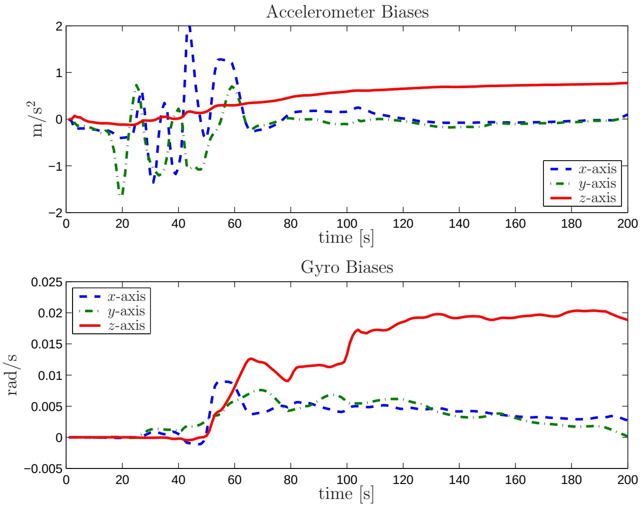


Figure 5.5: Estimated sensor biases using the basic navigation algorithm.

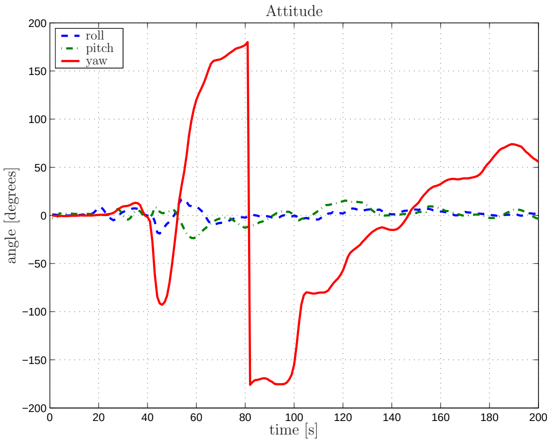


Figure 5.6: Estimated attitude from the basic navigation algorithm.

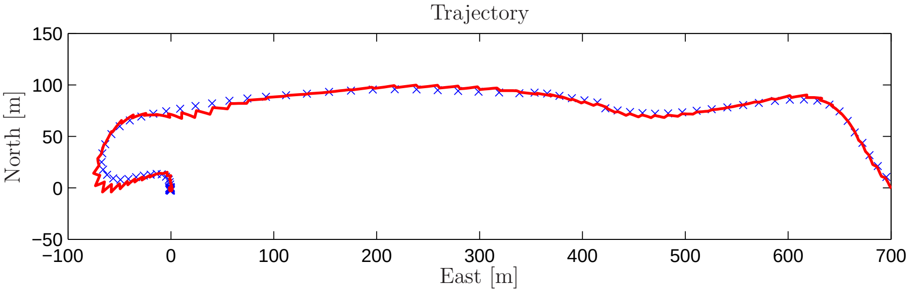


Figure 5.7: Estimated position for the first 110 s using the basic navigation algorithm.

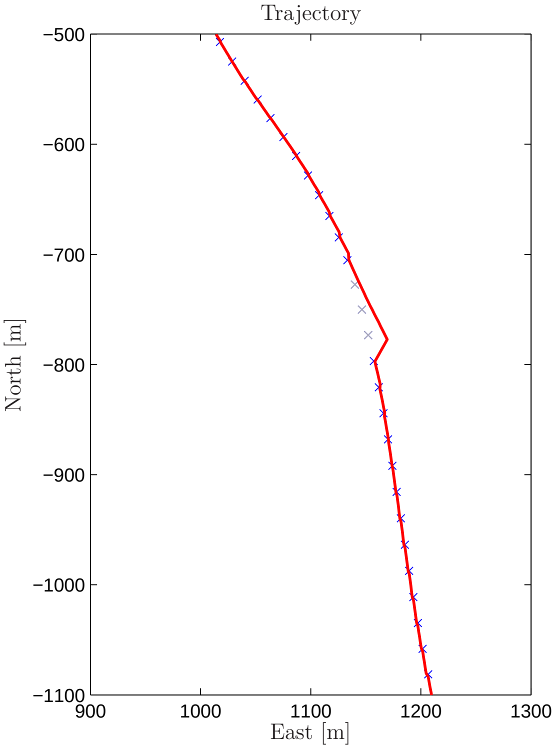


Figure 5.8: Estimated position of the basic navigation algorithm between 138 s and 167 s, with a simulated GPS outage (3 light gray crosses). The axes denote the position relative to the starting point.

5.2.2 GPS aided INS with Nonholonomic Constraints

As described in Section 3.9.1, it is possible to set constraints on the velocity vectors. These constraints have been implemented and after tuning the covariance matrices, interesting results could be obtained. Figure 5.9 shows the estimated sensor biases in function of the time. There is a variation in the gyro biases, although considering the y-axis scaling, the x- and y-bias converged to a value of approximately zero and the z-bias slightly below zero.

The estimated attitude angles in Figure 5.10 are reasonable. After the start, the turn causes a change in the heading of -90° at the first turn and +90° after the second turn. After the turn around 105 s, the heading fluctuates around +150°.

At the beginning of the trajectory, the estimated position deviates from the GPS position as it can be seen in Figure 5.11. But as soon as the attitude has stabilized, a smooth trajectory results. If the deviation between the INS estimate and the GPS positions in the turn from east to south-east direction reflects the true trajectory, is not assessable.

A simulated GPS outage does not influence the INS position estimation much, provided that the attitude is correct. The result with a 3 seconds outage is shown in Figure 5.12.

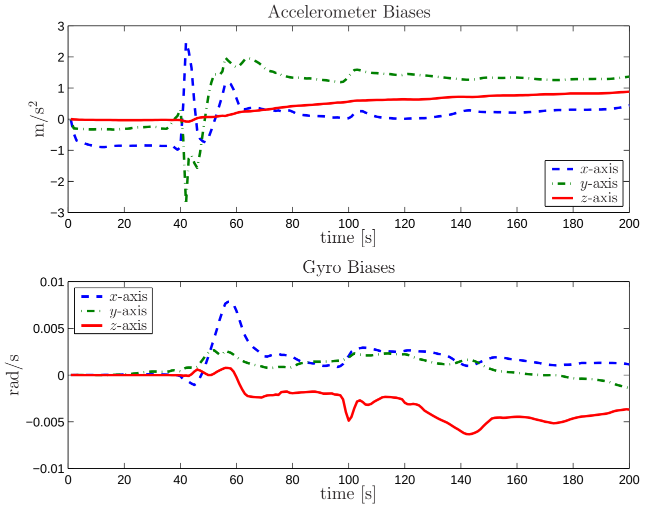


Figure 5.9: Estimated sensor biases using the navigation algorithm with nonholonomic constraints.

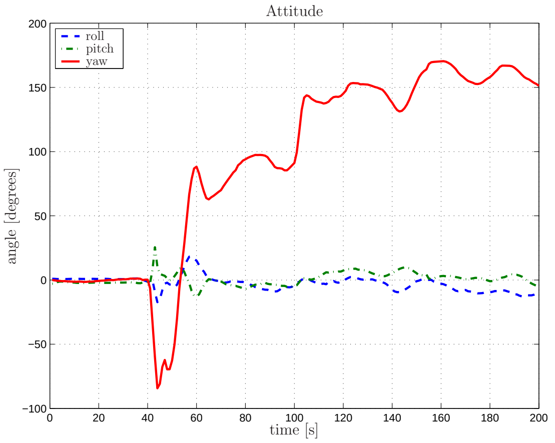


Figure 5.10: Estimated attitude from the navigation algorithm using nonholonomic constraints.

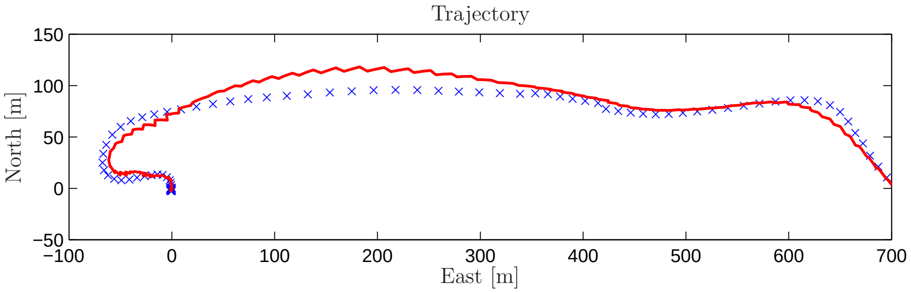


Figure 5.11: Estimated position for the first 110 s using nonholonomic constraints.

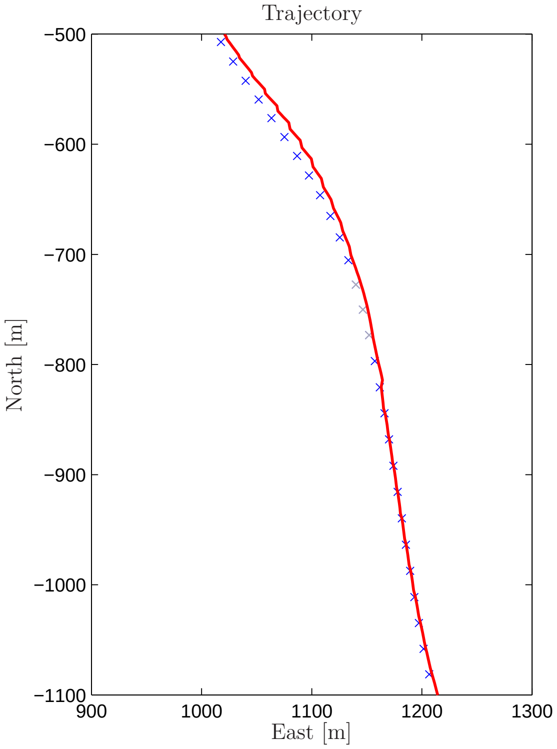


Figure 5.12: Estimated position with nonholonomic constraints between 138 s and 167 s, with a simulated GPS outage (3 light gray crosses). The axes denote the position relative to the starting point.

5.3 Error Analysis

In order to study the influence of different error sources on the output states, an analysis based on simulated data was conducted. The simulated data for the test trajectory is from [4]. The vehicle is stationary for the first 100 seconds. Then, it describes a wide turn while accelerating to 18 km/h, finally heading eastward. After two left, one right, and another left turn, the vehicle stops. All data is available error-free.

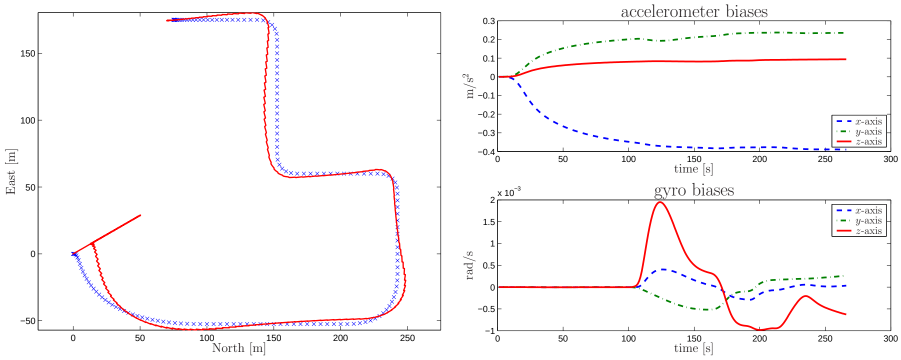
5.3.1 Accelerometers

In this section, the gyro and GPS position data is not perturbed by any errors, i.e. perfectly known. However, several errors, also in combinations, were added to the acceleration data. The analyzed errors are scaling $\mathbf{K}_a = \mathrm{diag}(k_{x_a}, k_{y_a}, k_{z_a})$, bias $\mathbf{b}_a = [b_{x_a}, b_{y_a}, b_{z_a}]^T$, misalignment $(\alpha_x, \alpha_y, \alpha_z)$, and Gaussian white noise with variance σ_a^2. Table 5.1 shows the values that were used. These values were obtained from an accelerometer calibration as described in Section 3.8.1. The standard deviation of the noise was $\sigma_a^2 = 0.03\,\mathrm{m/s^2}$, what complies with the real noise level in the sensor data. The influence of noise on the acceleration data does not make a big difference. The Kalman filter with its low-pass characteristic is able to smooth out the perturbed sensor data. Also

Table 5.1: Error parameters for the accelerations.

scaling			bias [m/s^2]			misalignment [rad/s]		
k_{x_a}	k_{y_a}	k_{z_a}	b_{x_a}	b_{y_a}	b_{z_a}	α_x	α_y	α_z
1.0013	1.0040	0.9926	-0.4420	0.2548	0.1020	0.0303	0.0120	0.0003

a scaling or misalignment of the individual sensors does not affect the position estimation. Figure 5.13 shows the effect of a sensor bias on the position estimation and the corresponding bias estimation. Even though the biases converge to the correct values, an error in the position estimation persists. Note that the gyro biases can be considered as almost zero (note the ordinate scaling 10^{-3}).



(a) Position in the tangent frame **(b)** Sensor biases

Figure 5.13: Influence of a bias in the accelerometer data.

Combinations of several different errors do not influence the output much, unless a bias is added to the sensor data.

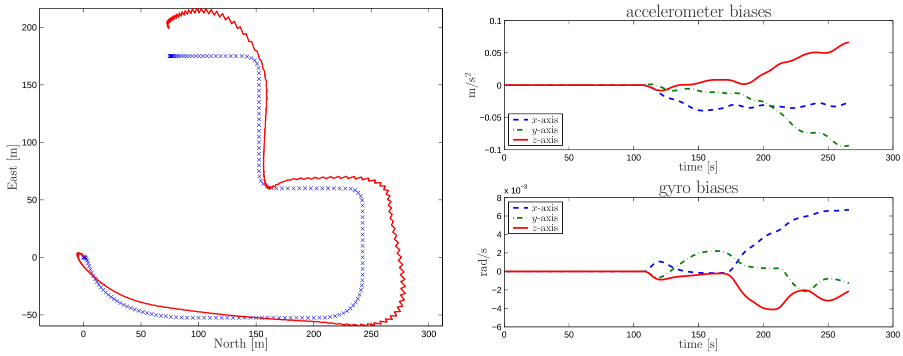
5.3.2 Gyros

For the analysis of the gyro error influence, all other sensor data were kept error free. The same error terms are used as for the accelerometer data. Namely the scaling $\mathbf{K}_g = \text{diag}(k_{x_g}, k_{y_g}, k_{z_g})$, bias $\mathbf{b}_g = [b_{x_g}, b_{y_g}, b_{z_g}]^T$, misalignment $(\gamma_x, \gamma_y, \gamma_z)$, and Gaussian white noise with variance σ_g^2. Table 5.2 shows the used values. Since the calibration of the real gyro sensors is not yet possible, arbitrary values were assumed. The standard deviation of the noise was $\sigma_g^2 = 0.004\,\text{rad/s}$, as it can be approximately measured from the real sensor data. As for the accelerations, noisy gyro data does not have much influence on the output; neither does a scaling or misalignment. However, a bias in the gyro data has big effect as it can be seen in Figure 5.14. The gyro biases do not seem to

Table 5.2: Error parameters for the angular rates.

scaling			bias [m/s^2]			misalignment [rad/s]		
k_{x_g}	k_{y_g}	k_{z_g}	b_{x_g}	b_{y_g}	b_{z_g}	γ_x	γ_y	γ_z
0.9874	1.0008	1.0628	-0.1500	0.0900	0.1200	0.0300	0.0020	0.0100

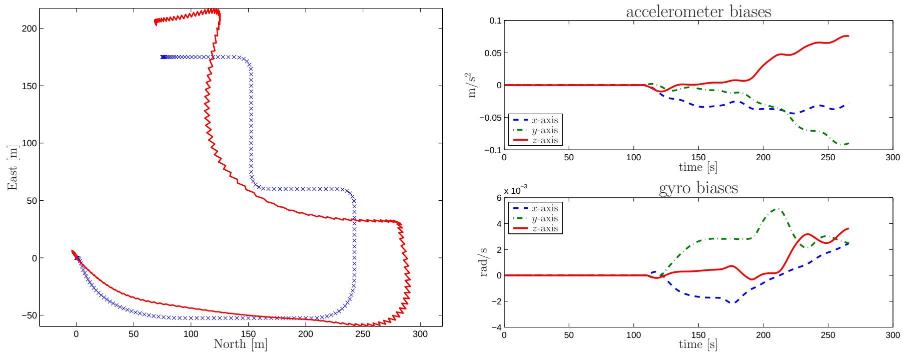
converge, instead the accelerometer biases deviate quite much from their true value which is 0.



(a) Position in the tangent frame **(b)** Sensor biases

Figure 5.14: Influence of a bias in the gyro data.

Combining further error sources with the bias, leads to large deviations in all outputs, see Figure 5.15.

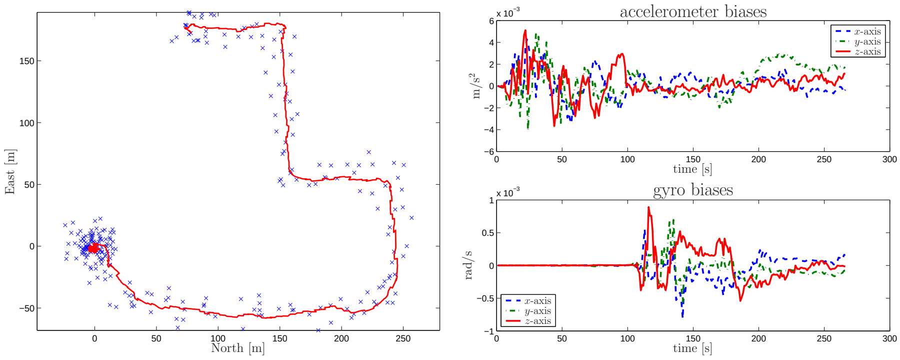


(a) Position in the tangent frame **(b)** Sensor biases

Figure 5.15: Influence of noise, bias, scaling, and misalignment in the gyro data.

5.3.3 GPS

In the last analysis, all the data is error free, except for Gaussian white noise with a standard deviation of $\sigma_{GPS} = 8\,\mathrm{m}$ that was added to the GPS positions. It should be noted that the real GPS position error is not white and could be modeled as a Markov process. Figure 5.16 shows that an error in the GPS positions causes some deviations in the navigation output, but its effect is not as serious as a bias in the IMU sensors.



(a) Position in the tangent frame **(b)** Sensor biases

Figure 5.16: Influence of Gaussian white noise in the GPS-positions.

5.3.4 Summary

The highest influence on the navigation output can be identified in the IMU sensor bias. It is important and good that these errors are modeled in the navigation algorithm. But, as long as these errors do not converge to the true value, a considerable error will result. For a good convergence of all the errors, the input signal has to excite the states sufficiently. A calibration of both, the accelerometers and gyros will contribute to the desired accuracy as well.

Chapter 6

Conclusions and Further Work

In this thesis, a GPS aided INS was integrated using an in-house constructed low-cost IMU, an off-the-shelf GPS receiver, and an extended Kalman filter. First, a software had to be developed to obtain the IMU data. Using a calibration method, some error parameters of the accelerometers could be found. After pre-processing of the data obtained from field tests, the GPS aided INS algorithm could be tested. It turned out to be rather difficult to get the system to work with real-world data. However, using nonholonomic constraints, good results in terms of the position estimation but also the attitude could be achieved compared to the case without constraints. Finally, the navigation algorithm was implemented in C++ and through a MEX wrapper also made available in MATLAB.

6.1 Conclusions

It is possible to sample IMU/GPS data using the data acquisition software on a PC, although a limitation in the flexibility has to be taken into account. Possibly, because of some different tolerances for the voltage levels in the RS232 interface chip sets of computers had the effect, that the serial interface caused errors such that the polling routine on the host PC stopped working. Interestingly, this phenomenon could only be observed and reproduced with DELL laptops of different types. The available IBM laptop did not show this effect.

The calibration algorithm works well. It could only be applied on the accelerometers though, but should in practice also work for the gyros.

The integrated GPS aided INS without further enhancements and constraints, is strongly dependent on GPS position estimates. In order to get an acceptable result, the covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ have to be set so that the system highly trusts the GPS positions. Furthermore, the convergence of the accelerometer and gyro biases is not guaranteed. Depending on the tuning of the covariance matrices, the biases can ”converge” to different values.

The use of nonholonomic constraints drastically improved the accuracy of the position estimation but also the attitude. Even a simulated GPS outage of up to 5 seconds does not influence the position estimation much. The attitude is even more stable, provided the biases have converged and the attitude is correct before the outage begins.

The available real-world data from the field test is useful since it contains some turns in the beginning and does not contain any GPS outages. However, the length of the stationary phase and the length of the non-stationary phase may be a bit short and should be increased in the future. It should also be noted that no reference trajectory is available; therefore, the position error cannot objectively be assessed.

An error analysis showed that the sensor biases have the highest influence on the position estimation, where the gyro biases contribute a bit more than the accelerometer biases. Incorrect gyro biases together with a wrong scaling or misalignment even add more error to the output. Therefore, a calibration of the gyros could further improve the accuracy.

The conclusions can be summarized to the following points:

- the data acquisition works, but is depending on the used PC,
- a method for the calibration is available that can be used for both the accelerometers and gyros,
- nonholonomic constraints significantly improve the accuracy,
- the trajectory needs to be planed and has to include left and right turns in the beginning and should last longer, i.e. at least 5 minutes.

6.2 Recommendations and Further Work

A lot of knowledge was gained throughout the work on this project. This section mentions the important aspects for the future work and lists further tasks that have to be solved in order to advance towards a good working system:

- In cases with strong vibrations, the data from the x- and y-gyro as well as the z-accelerometer contains more noise than the other sensors. This problem can be solved with a better mechanical fix of the two vertical boards that are mounted on the main circuit board.
- The currently used gyros can sense a maximal angular rate of 150°/s. In applications with a land vehicle such as a car, the sensed angular rates are usually much lower. By choosing another gyro with a lower maximal angular rate or using the reference voltage output of the sensors for the reference input at the signal amplifiers, a higher resolution could be obtained.
- A further increase in the accuracy of the sampled sensor data, a 16 bit analog to digital converter could be used.
- It may be possible to switch the output data format of the GPS receiver from NMEA 0183 to a binary protocol. The advantage will be a higher Baud-rate and if available, even the pseudorange and Doppler measurements could be obtained. This will provide the requisites for a tightly coupled GPS aided INS integration.

- To calibrate the gyros, a constant rotating platform has to be constructed or found.

- For both, the accelerometer and the gyro, nine positions are required for the calibration. With the help of a simulation, the nine best positions out of 18 possible positions have to be found.

- More sensor errors have to be modeled. The inclusion of the scale factors in the error model will provide a better accuracy. These scale factors could then also compensate a small remaining error in the misalignment of the three sensor axes. Further, a model for the drift could also be studied and included.

- A method to estimate the GPS delay is presented in [29] and should be considered for the future work.

- It could be of an advantage to also estimate the lever-arm and include an error state as described in [21].

- The paper [31] proposes the use of a Sigma-Point Kalman Filter with the variants unscented Kalman filter and central difference Kalman filter. The authors mention that they investigate the feasibility to use a low-cost IMU (<US$1000) and still be able to keep the estimation performance.

- Methods to damp or overcome GPS position errors can be studied and integrated, to improve the quality of the GPS aid.

Bibliography

[1] J. A. Farrell and M. Barth, *The Global Positioning System and Inertial Navigation.* McGraw-Hill, New York, NY, 1999.

[2] K. R. Britting, *Inertial Navigation Systems Analysis.* Wiley-Interscience, John Wiley & Sons, Inc, 1971.

[3] A. B. Chatfield, *Fundamentals of High Accuracy Inertial Navigation.* American Institute of Aeronautics and Astronautics, Reston, VA, 1997.

[4] I. Skog, "A low-cost GPS Aided Inertial Navigation System for Vehicular Applications," Master's thesis, Royal Institute of Technology, Sweden, Mar. 2005, IR-SB-EX-0506.

[5] P. Strömbäck, "Integrating GPS and INS with a Tightly Coupled Kalman Filter - Using Proximity Information for Compensating Signal Loss," Master's thesis, Royal Institute of Technology, Sweden, Mar. 2003, IR-SB-EX-0306.

[6] R. Dorobantu and C. Gerlach, "Characterisation and Evaluation of a Navigation-Grade RLG SIMU," *European Journal of Navigation*, vol. 2, no. 1, pp. 63–78, Feb. 2004.

[7] C. Altmayer, "Enhancing the Integrity of Integrated GPS/INS Systems by Cycle Slip Detection and Correction," *IEEE Intelligent Vehicles Symposium 2000*, pp. 174–179, Oct. 2000.

[8] W. J. Clinton. (2006, Jan.) GPS Selective Availability Stop. U.S. Department of Defense (Coast Guard). [Online]. Available: http://www.navcen.uscg.gov/gps/selective_availability.htm

[9] B. Hofmann-Wellenhof, K. Legat, and M. Wieser, *Navigation - Principles of Positioning and Guidance.* Springer, Wien/New York, 2003.

[10] J. Zander, B. Slimane, and L. Ahlin, *Principles of Wireless Communications.* Stockholm: Royal Institute of Technology, 2005.

[11] A. El-Rabbany, *Introduction to GPS - The Global Positioning System.* London: Artech House, 2002.

[12] USCG Navigation Center, *GPS Standard Positioning Service - Performance Standard*, USCG Navigation Center, Oct. 2001.

[13] G. Beutler, M. Rothacher, S. Schaer, T. Springer, J. Kouba, and R. Neilan, "The International GPS Service (IGS): An Interdisciplinary Service in Support of Earth Sciences," *Adv. Space Res.*, vol. 23, no. 4, pp. 631–635, 1999.

[14] USCG Navigation Center, *GPS Standard Positioning Service - Signal Specification*, USCG Navigation Center, Jun. 1995.

[15] B. W. Parkinson and J. J. Spilker, *Global Positioning System: Theory and Applications.* Progress in Astronautics and Aeronautics, 1996, vol. 163.

[16] T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation.* Prentice-Hall, Englewood Cliffs, NJ, 2000.

[17] Q. Honghui and J. Moore, "Direct Kalman filtering approach for GPS/INS integration," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 2, pp. 687–693, Apr. 2002.

[18] F. Gustafsson, *Adaptive Filtering and Change Detection.* John Wiley & Sons, New York, NY, 2000.

[19] D. Goshen-Meskin and I. Y. Bar-Itzhack, "Observability Analysis of Piece-Wise Constant Systems – Part II: Application to Inertial Navigation In-Flight Alignment," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no. 4, pp. 1068–1075, Oct. 1992.

[20] Y. F. Jiang and Y. P. Lin, "Error Estimation of INS Ground Alignment Through Observability Analysis," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no. 1, pp. 92–97, Jan. 1992.

[21] S. Hong, M. H. Lee, H.-H. Chun, S.-H. Kwon, and J. L. Speyer, "Observability of Error States in GPS/INS Integration," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 2, pp. 731–743, Mar. 2005.

[22] R. Grover, P. Brown, and Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, 3rd ed. John Wiley & Sons, Inc, 1997.

[23] S. Sukkarieh, E. M. Nebot, and H. F. Durrant-Whyte, "A High Integrity IMU/GPS Navigation Loop for Autonomous Land Vehicle Applications," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 572–578, Jun. 1999.

[24] E.-H. Shin, "Accuracy Improvement of Low Cost INS/GPS for Land Applications," Master's thesis, University of Calgary, Dec. 2001.

[25] I. Skog and P. Händel, "Calibration of a MEMS Inertial Measurement Unit," *XVII IMEKO World Congress*, Sep. 2006.

[26] G. Dissanayake, S. Sukkarieh, E. Nebot, and H. Durrant-Whyte, "The Aiding of a Low-Cost Strapdown Inertial Measurement Unit Using Vehicle Model Constraints for Land Vehicle Applications," *Robotics and Automation, IEEE Transactions on*, vol. 17, pp. 731–747, Oct. 2001.

[27] J. Wang, H. Lee, S. Hewitson, and H.-K. Lee, "Influence of Dynamics and Trajectory on Integrated GPS/INS Navigation Performance," *Journal of Global Positioning Systems*, vol. 2, no. 2, pp. 109–116, Dec. 2003.

[28] I. Skog, A. Schumacher, and P. Händel, "A Versatile PC-Based Platform For Inertial Navigation," *7th Nordic Signal Processing Symposium*, Jun. 2006, submitted.

[29] H. K. Lee, J. G. Lee, and G.-I. Jee, "Calibration of Measurement Delay in Global Positioning System/Strapdown Inertial Navigation System," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 2, pp. 240–247, Apr. 2002.

[30] I. Skog, "The Troge Software," internal document, Royal Institute of Technology, Jun. 2005.

[31] R. van der Merwe, E. A. Wan, and S. I. Julier, "Sigma-Point Kalman Filters for Nonlinear Estimation and Sensor-Fusion," *American Institute of Aeronautics and Astronautics*, Mar. 2004.

[32] I. Skog, P. Händel, and A. Schumacher, "The TROGE Project," in *KTH and FOI workshop on navigation*, Stockholm, Sweden, Apr. 9 2006, proceedings IR-EE-SB 2006:009 Royal Institute of Technology, Stockholm, Sweden.

View publication stats