



**ROYAL INSTITUTE
OF TECHNOLOGY**

Aided inertial navigation field tests using an RLG IMU

JESPER GRANDIN

Master's of Science Thesis in Geodesy
TRITA-GIT EX 07-014

School of Architecture and the Built Environment
Royal Institute of Technology (KTH)
100 44 Stockholm, Sweden

December 2007

Abstract

In this thesis a system for aided inertial navigation (aided INS or AINS) has been tested through a number of kinematic experiments. In the experiments, data was collected and post-processed by different methods. The system was built up by an IMU (inertial measurement unit) aided by a GPS receiver and an odometer. To fuse and filter the sensor data a Kalman filter from the AINS Matlab Toolbox has been used.

The following equipment was used

- iNAV-RQH; IMU from iMAR
- System 4000SSE; GPS receiver from Trimble
- Correvit L-CE; odometer from Corsys-Datron

The following methods (and combinations of these) were used on the acquired data

- INS with *Zero velocity updates* (ZUPT)
- INS under *non holonomic constraints*
- INS integrated with GPS
- INS integrated with odometer

The work resulted in a number of trajectories; some of these could be compared with a reference trajectory, in other cases this was not possible due to the nature of the experiment. Also there are no guidelines or standards on how to test a system in order to compare it with other systems.

Over all, good results were achieved with the system. The AINS toolbox worked out well and it was indicated that the odometer could be used as an important aiding source in an aided inertial navigation system.

Sammanfattning

I det här examens arbetet har ett understött system för tröghets navigering (Aided INS, AINS) testats genom ett antal kinematiska experiment. Under dessa experiment har data samlats in och efterberäknats med olika metoder. Systemet var uppbyggt av en IMU (Inertial measurement unit) med stöd ifrån en GPS mottagare och en odometer, för att sammanfoga och filtrera sensordata användes ett Kalman filter ifrån AINS Toolbox, en Matlab toolbox för understödd tröghetsnavigering.

Följande utrustning användes

- iNAV-RQH; IMU från iMAR
- System 4000SSE; GPS mottagare från Trimble
- Correvit L-CE; odometer från Corsys-Datron

Följande metoder (samt kombinationer av dessa) användes på det insamlade datat

- INS med *Zero velocity update* (ZUPT)
- INS under *non holonomic constraints*
- INS integrerad med GPS
- INS integrerad med odometer

Arbetet resulterade i en mängd trajektorer som i vissa fall kunde jämföras med en referens trajektor, i andra fall var detta ej möjligt på grund av experimentets natur. Det finns heller inga riktlinjer eller standarder på hur system skall testas för att lättare kunna jämföras sinsemellan.

Generellt sett uppnåddes goda resultat med systemet. AINS toolbox fungerade bra och det påvisades att en odometer kan användas som ett viktigt stöd till ett understött system för tröghets navigering.

Acknowledgements

All the experiments and the processing of data have been done at *The Institute for Astronomic and Physical Geodesy (IAPG) Technical University of Munich* (Technische Universität München), where I had the possibility to be a guest student for one semester. I would like to thank professor Rainer Rummel for giving me that opportunity.

IAPG also provided me with a supervisor, Dr. Raul Dorobantu, who always supported me, helped me with ideas, gave me critics, inspired me and gave me guidance not only for the thesis. I would like to express my gratitude to him.

To fuse the sensor data a Matlab Toolbox have been used, the *Aided Inertial Navigation System Toolbox* (AINS Toolbox), written and developed by Naser El-Sheimy and Eun-Whan Shin at the *Mobile Multi-Sensor Research Group, Department of Geomatics Engineering, Schulich School of Engineering, University of Calgary*

The collaboration with the *Bavarian Committee for International Geodesy BEK* (*Bayerische Kommission für die Internationale Erdmessung*) gave us both useful support and data from the German Satellite Positioning Service (SAPOS).

Thanks to professor Bertrand Merminod and Dr. Jan Skaloud at the *Laboratory of Geodetic Engineering* at the *Swiss Federal Institute of Technology in Lausanne* (Ecole polytechnique fédérale de Lausanne), we could use their odometer for our experiments. They also supported us with documentation and their experience with the unit.

I would also like to thank my examiner Professor Lars Sjöberg, my supervisor in Sweden Dr Milan Horemuz, Professor Urs Hugentoubler, Günter Dichtl, Markus Heinze and Karin Hedman for there support and help. A special thank goes to Christian Ackerman; together Ackerman, Dr. Dorobantu and myself did all practical experiments.

Contents

Contents	ix
List of symbols and abbreviations	xi
1 Introduction	1
1.1 INS and geodesy	1
1.2 Mathematical notation	1
2 What is an RLG IMU?	5
2.1 Measurements and observables of an IMU	5
3 Inertial navigation	7
3.1 Reference frames and gravity models	7
3.2 Navigation equations	9
3.3 Mechanization	12
3.4 Initialization and Alignment	13
3.5 Aidings for inertial navigation	14
3.5.1 Velocity aiding	14
3.5.2 Position aiding	15
3.5.3 Lever arm effect	16
4 Hardware	17
4.1 Accelerometers	17
4.2 Gyroscopes	18
4.3 IMU	20
4.4 Odometer	20
4.5 GPS	27
5 Extended Kalman filtering	29
5.1 Filter design	29
5.2 Pre-filtering of data	32
6 Theresienwiese, Outdoor navigation mission	35

6.1	Aim of mission	36
6.2	System setup	36
6.3	Data flow	36
6.4	Data sets	36
6.5	Data preparation	37
6.6	GPS / INS / Odometer Integration with AINS toolbox	39
6.7	Results of outdoor experiment	39
6.8	Discussion of the results	45
7	City navigation mission	49
7.1	Aim of mission	49
7.2	System setup	49
7.3	Data flow	49
7.4	Data set	49
7.5	Data preparation	50
7.6	Results of Munich city centre experiment	50
7.7	Discussion of the results	50
8	High way navigation mission	55
8.1	Aim of mission	56
8.2	System setup	56
8.3	Data flow	56
8.4	Data set	56
8.5	Data preparation	56
8.6	Results of High way experiment	57
8.7	Discussion of the results	58
9	Conclusions	59
	Bibliography	61
A	Logged data	63
B	Cookbook	65
C	Manufactures specification of hardware	71

List of symbols and abbreviations

Scalars denoted in lower case letters; s

Vectors denoted in lower case bold letters; \mathbf{v}

Matrix denoted in upp case bold letters; \mathbf{M}

AINS Aided Inertial Navigation System

b Indicates the body frame

\mathbf{b} Bias vector

\mathbf{C} Rotation matrix

\mathbf{D} Matrix containing ellipsoidal radii of curvatures

d Drift

DGPS Differential GPS

e Indicates the earth frame, ECEF-frame

ϵ Error matrix correlated to \mathbf{C}

ECEF Earth Centered Earth Fixed

EKF Extended Kalman filter

ENU East, North, Up; indicates the axis of the local level system

\mathbf{E} Error matrix

\mathbf{F} Dynamics system matrix

φ Latitude

\mathbf{f} Specific force

\mathbf{G} Shape matrix of system noise

GPS Global Positioning System

- $\bar{\mathbf{g}}$ Gravitational vector
- \mathbf{g} Gravity vector
- γ Normal gravity
- \mathbf{H} Design matrix for measurements
- h Height above the ellipsoide
- IMU** Inertial Measurement Unit
- INS** Inertial Navigation System
- \mathbf{I} Identity matrix
- i indicates the inertial frame
- IAPG** Instituts für Astronomische und Physikalische Gäodesie
- \mathbf{K} Gain factor
- l Indicates the local level frame
- ℓ Measurement
- λ Longitude
- \mathbf{M} Radius of curvature in the prime vertical
- \mathbf{N} Radius of curvature of the meridian
- NED** North, East, Down; indicates the axis of the local level system
- $\mathbf{\Omega}$ Skew-symmetric matrix of the angular turn-rate vector $\boldsymbol{\omega}$
- $\boldsymbol{\omega}$ Angular turn-rate vector
- \mathbf{P} Error covariance
- \mathbf{Q} Covariance matrix of system noise
- \mathbf{R} Covariance matrix of measurements
- RMS** Root mean square
- RLG** Ring Laser Gyroscope
- \mathbf{r} Position vector
- std** Standard deviation
- TTC** Trimble Total Control

TTL Transistor Transistor Logic

\mathbf{v} Velocity vector

WGS World Geodetic System

ZUPT Zero velocity update

\mathbf{x} State vector in Kalman filter.

\mathbf{z} Update vector in Kalan filter.

Chapter 1

Introduction

The aim of this thesis has been to practically test and evaluate an aided inertial navigation system (INS), consisting of an inertial measurement unit, a GPS receiver and an odometer. It has been of special interest to see how velocity data derived from the odometer could contribute to the final solution.

1.1 INS and geodesy

Friedrich Robert Helmert defined geodesy as *the science of the measurement and mapping of the earth's surface* (Torge 2001). In the book *Inertial Navigation Systems with geodetic Applications* (Jekeli 2001), the author use this definition to relate inertial navigation with geodesy. If the inertial navigation system is used along a trajectory on the surface of the earth, it will determine the coordinates along this trajectory, and that is to measure and map a small part of the earth's surface.

Today's geodesy is dealing with higher accuracy than normally is achieved with INS, and INS is mostly used for navigation and guidance, but geodesy is still important to provide the reference frames for both navigation and guidance.

If one uses a modern definition of geodesy the need for geodesy in INS becomes even more evident. A modern definition of geodesy also includes *the external gravity field of the earth* (Torge 2001), which is crucial for an inertial navigation. To be able to isolate the measured accelerations of dynamics from the acceleration of gravity the system need this extra information.

In situations where the position is already known (for example by using GPS), one could invert the need for the gravity field data and instead use the system to estimate the gravity instead (Schwarz 2001).

1.2 Mathematical notation

In the equations, scalars, vectors and matrixes are used. The scalars are denoted by letters, vectors by bold letters and matrixes by bold capital letters. All the

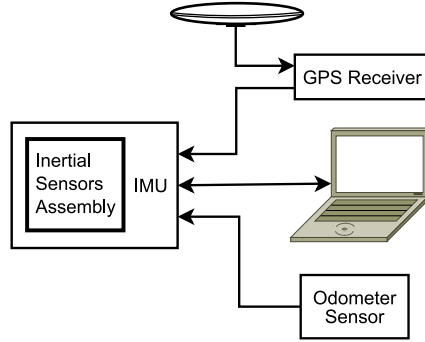


Figure 1.1. Aided inertial system.

rotation matrixes in the derivations are orthogonal, which means that the following two equations will hold

$$\mathbf{C}_b^a = (\mathbf{C}_a^b)'$$

$$\mathbf{I} = \mathbf{C}_b^a \mathbf{C}_a^b$$

where the matrix \mathbf{C} is a rotation matrix, and the rotation is done from the subscript to the superscript. The apostrophe ($'$) denotes the transpose of the matrix and \mathbf{I} denotes the identity matrix.

For the cross product of two vectors, the *skew-symmetric matrix* generally has been used. The skew-symmetric matrix is constructed from the components of a vector. If for example

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}, \quad (1.1)$$

the corresponding skew-symmetric matrix is

$$\mathbf{A} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}, \quad (1.2)$$

which could be used to get the cross product between the vectors \mathbf{a} and \mathbf{b} :

$$\mathbf{a} \times \mathbf{b} = \mathbf{A}\mathbf{b} \quad (1.3)$$

or

$$\mathbf{a} \times \mathbf{b} = -\mathbf{B}\mathbf{a}. \quad (1.4)$$

Where \mathbf{B} is the skew-symmetric matrix of vector \mathbf{b} .

The relative rotations of two reference frames are described by the angular turn rate, denoted by vector ω_{ba}^a , where the subscript denotes the turn rate from a to b . And the superscript indicates which reference frame the turn rate is given in.

The corresponding skew symmetric matrix of the turn rates is denoted Ω_{ba}^a , and the transformation between two skew-symmetric matrixes can be done by

$$\Omega_{ba}^b = C_a^b \Omega_{ba}^a C_b^a. \quad (1.5)$$

Chapter 2

What is an RLG IMU?

A ring laser gyroscope inertial measurement unit (RLG IMU) contains one triad of orthogonally assembled accelerometers and one orthogonally assembled triad of ring laser gyroscopes (RLG). Together these triads form the inertial sensor assembly (ISA). If the ISA is mounted in a housing and some pre-processing of raw data is done, this is called an inertial measurements unit (IMU). The output from an IMU can be processed and used for navigation, this is denoted inertial navigation system (INS). Although an inertial navigation system could be used stand alone, one generally aid the INS with data from other sensors, forming an aided inertial navigation system (AINS).

2.1 Measurements and observables of an IMU

In the INS one makes use of navigation equations that are solved in order to find the navigation solution, i.e. position, velocity and attitude. These equations need two kinds of observables: Specific force [m/s^2] and angular turn rate [deg/s], which are estimated from the IMU measurements.

The navigation equations will be explained in Chapter 3.

Specific force

The accelerometers sense the specific force, which is the real force acting on the unit. Such an applied force could be the force driving an object forward or the reaction of the Earth's surface acting on objects at rest (Jekeli 2001). That means that the accelerometer can not sense the gravitational field directly, a free falling unit will for example not sense any acceleration.

In scalar form specific force could be described as

$$f = \ddot{r} + \bar{g}, \tag{2.1}$$

where f is the specific force, \ddot{r} is the scalar of the second time derivative of the position vector and \bar{g} is the gravitation. Since measurements on the earth

are affected by the centripetal force, the combination of the gravitation and the centrifugal acceleration is often used in the text, that is the gravity (g).

By using equation 2.1, one could see that if the unit is at rest, i.e. if the second time derivative of the position vector is equal to zero the specific force will be equal to g (Schwarz 2001). This also implies that in order to find the second time derivative of the position vector one need to know the specific force and the gravity.

Writing the specific force equation in vectorial form, with respect to inertial frame (the reference frames will be outlined in Section 3.1) yields

$$\mathbf{f}^i = \ddot{\mathbf{r}}^i - \bar{\mathbf{g}}^i, \quad (2.2)$$

where \mathbf{f} is the specific force vector ($[f_x \ f_y \ f_z]'$), $\ddot{\mathbf{r}}$ is the second time derivative of the position vector and $\bar{\mathbf{g}}$ is the vector of gravitation.

In order to find the second time derivative of the position vector one need first to know the specific force and the gravity vector. The specific force one can measure from the accelerometers and by using a normal gravity model the gravity could be estimated (see Section 3.1). This is not so easily done. One need for example to know the attitude of the unit in order to remove the influence of gravity. Another problem is that the measurements from the accelerometers are not perfect; the outputs are biased by errors and noise. The measured data from the accelerometers could therefore be expressed as:

$$\ell_f = \mathbf{f} + \mathbf{E}\mathbf{f} + \boldsymbol{\varepsilon}_f + \mathbf{n}_f. \quad (2.3)$$

Where ℓ_f is the measurement from the accelerometers and \mathbf{f} is the specific force vector. \mathbf{E} and $\boldsymbol{\varepsilon}_f$ represent systematic errors such as non-orthogonal assembly and biased output. The vector \mathbf{n}_f is representing the sensor noise.

Angular rate

The rotation of the IMU in inertial space is represented by a vector of angular rates, but also these measures are affected by errors and noise, the measurement in vectorial form for the gyroscopes can be formed as:

$$\ell_\omega = \boldsymbol{\omega} + \mathbf{E}\boldsymbol{\omega} + \mathbf{n}_\omega + \boldsymbol{\varepsilon}_\omega. \quad (2.4)$$

Where ℓ_ω is the gyro measurements and $\boldsymbol{\omega}$ is the rotation in inertial space. \mathbf{E} and $\boldsymbol{\varepsilon}_\omega$ are systematic errors such as non-orthogonal assembly and biased output. The vector \mathbf{n}_ω is the sensor noise.

Using the measurements

It has been stated in the beginning of this section that the specific force (\mathbf{f}) and the rotation rate in inertial space ($\boldsymbol{\omega}$) are needed for the navigation equations. The relation between these true values and the measurements from the IMU have been outlined in a simplified way.

Estimates of the true values, \mathbf{f} & $\boldsymbol{\omega}$, are observable in a Kalman filter where errors and noise are modelled. The Kalman filter will be described in Chapter 5.

Chapter 3

Inertial navigation

3.1 Reference frames and gravity models

Earth frame

The earth frame (*e*-frame) is an earth centered earth fixed (ECEF) reference frame defined as

<i>Origin</i>	is in the mass centre of the earth
z^e -axis	is passing through the mean spin axis of the earth
x^e -axis	is passing through the Greenwich meridian
y^e -axis	is completing the right hand system

the positions in the *e*-frame can also be expressed in geodetic longitude (λ), latitude (φ) and height above a reference ellipsoid (h). The ellipsoid used is often the one defined by WGS-84. The relation between the cartesian and geodetic position is given by Jekeli (2001)

$$\begin{bmatrix} x^e \\ y^e \\ z^e \end{bmatrix} = \begin{bmatrix} (N + h) \cos \varphi \cos \lambda \\ (N + h) \cos \varphi \sin \lambda \\ (N(1 - \textit{eccentricity}^2) + h) \sin \varphi \end{bmatrix}. \quad (3.1)$$

Where the *eccentricity* is the eccentricity of the ellipsoid, which together with the ellipsoid major axis a , define the shape and size of the reference ellipsoid. N is the radius of curvature of the ellipsoid in prime vertical at the current latitude, which is derived from a , φ and the *eccentricity*.

The use of geodetic positions has an advantage if one also uses a local geodetic system, since that is directly associated with the geodetic position.

Inertial frame

For the practical use of INS a true inertial frame is not used, the frame used will never the less be referred as the inertial frame or (*i*-frame) since this is more convenient. The frame used in this report is defined as

Origin in the mass centre of the earth and therefore *free falling* in the universe
 z^i -axis is passing through the mean spin axis of the earth
 x^i -axis none rotating, with respect to fix stars
 y^i -axis is completing the right hand system

Local-level frame

The local-level frame (l -frame), is a local geodetic coordinate system here defined as a north-east-down (NED) system where

origin is the same as the *origin* of the IMU frame
 x^l -axis is pointing towards geodetic North
 y^l -axis is pointing towards geodetic East
 z^l -axis is pointing down, orthogonal to the reference ellipsoid

When the l -frame is used for inertial navigation one either has to neglect deviation between the plumb line and the z -axis or estimate this deviation.

There are also other definitions of the l -frames, such as east-north-up (ENU). Common to them all is that they are used to express the attitude, and to indicate the velocity of the body (Jekeli 2001).

The transformation between NED to e -frame could be done by (Schwarz 2001)

$$C_l^e = \begin{bmatrix} -\sin \varphi \cos \lambda & -\sin \lambda & -\cos \varphi \cos \lambda \\ -\sin \varphi \sin \lambda & \cos \lambda & -\cos \varphi \sin \lambda \\ \cos \varphi & 0 & -\sin \varphi \end{bmatrix}. \quad (3.2)$$

IMU frame

The IMU output is referred to this frame, defined by the manufacture. The position of all other sensors that have been rigidly connected to the IMU could be measured in this frame.

Vehicle frame

The vehicle frame used here is defined as having the origin in the origin of the IMU, the x -axis is pointing forward in the vehicles longitudinal axis, the y -axis is pointing to the right and the z -axis pointing down.

Body frame

The body frame (b -frame) is very much like the IMU frame, but is not defined by the manufacture. The *origin* is defined as the *origin* of the IMU frame, but the coordinate axis could have an other direction.

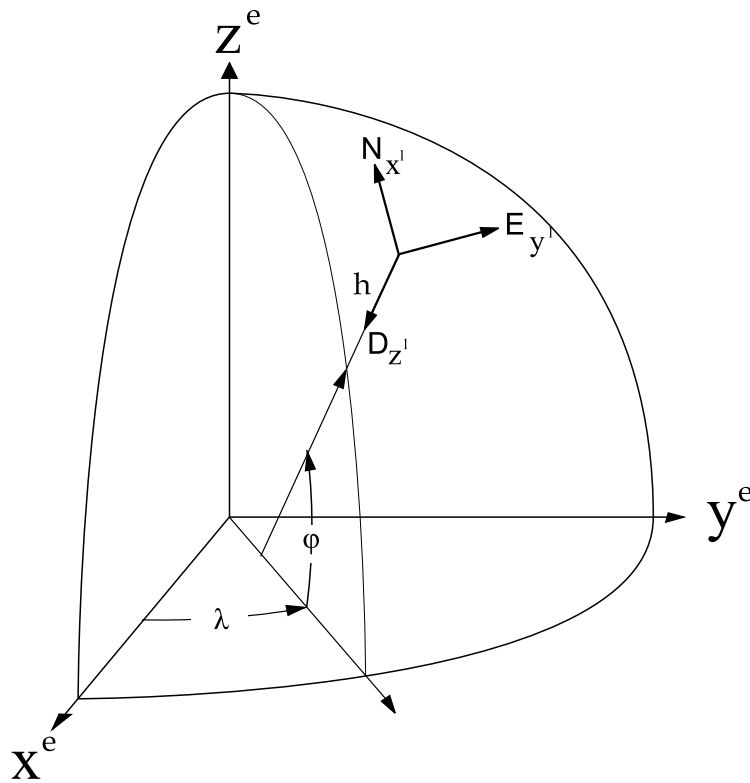


Figure 3.1. Earth frame and local level frame.

Gravity model

In order to solve the navigation equation one need to know the gravity force acting on the body at every place and time. For this a normal gravity model is used.

3.2 Navigation equations

Navigation equations in inertial system

The observables needed to form the navigation equations was explained in Chapter 2. The specific force vector was explained by Equation (2.2), were also the position vector was introduced. The formula is here changed to

$$\ddot{\mathbf{r}}^i = \mathbf{f}^i + \bar{\mathbf{g}}^i, \quad (3.3)$$

on the right hand are the *specific force vector* and the *gravitation vector*. On the left hand side is the unknown, the second time derivative of the *position vector*. First step to solve this second order differential equation is to rewrite it to a system of first order differential equations

$$\dot{\mathbf{r}}^i = \mathbf{v}^i \quad (3.4)$$

$$\dot{\mathbf{v}}^i = \mathbf{f}^i + \bar{\mathbf{g}}^i. \quad (3.5)$$

Since the specific force is observed in the body frame, and the normal gravity vector normally is given in the earth frame or local level frame (Schwarz 2001), it is necessary to use a rotation matrix between one of these frames and the inertial frame

$$\dot{\mathbf{r}}^i = \mathbf{v}^i \quad (3.6)$$

$$\dot{\mathbf{v}}^i = \mathbf{C}_b^i \mathbf{f}^b + \mathbf{C}_e^i \bar{\mathbf{g}}^e. \quad (3.7)$$

\mathbf{C}_e^i is the rotation matrix containing the earth rotation. It can be proved (Schwarz 2001) that \mathbf{C}_b^i can be obtained by solving

$$\dot{\mathbf{C}}_b^i = \mathbf{C}_b^i \boldsymbol{\Omega}_{ib}^b, \quad (3.8)$$

where $\boldsymbol{\Omega}_{ib}^b$ is the skew symmetric matrix of observed turn rates.

The Eqs. (3.6), (3.7) and (3.8) can be formed into a set of equations

$$\dot{\mathbf{x}}^i = \begin{bmatrix} \dot{\mathbf{r}}^i \\ \dot{\mathbf{v}}^i \\ \dot{\mathbf{C}}_b^i \end{bmatrix} = \begin{bmatrix} \mathbf{v}^i \\ \mathbf{C}_b^i \mathbf{f}^b + \mathbf{C}_e^i \bar{\mathbf{g}}^e \\ \mathbf{C}_b^i \boldsymbol{\Omega}_{ib}^b \end{bmatrix} \quad (3.9)$$

where the left side is the time derivative of the *navigation state vector*, and the right hand side is the first order differential navigation equations.

Navigation equations in local level system

The Equations in (3.9) have limitations since the navigation parameters are given in inertial space while we navigate on the surface of the earth. Outlined in this section are instead the more convenient navigation equations in the local level frame.

Position equations

The origin of the local level system is fixed in the IMU, but the position of this origin is described with curve linear coordinates ($\varphi \ \lambda \ h$) in the e -frame. The position equation therefore relate the derivative of the curvilinear positions with the velocity in the l -frame.

The first derivative of the latitude position in local level frame is found by dividing the north pointing velocity vector in local level frame with the radius of curvature of the meridian (M) plus the height above the ellipsoid

$$\dot{\varphi} = \frac{\mathbf{v}_{north}^l}{(M + h)} \quad (3.10)$$

and the first derivative of the longitude position in local level frame is found by dividing the eastern velocity vector with the radius in the earth x-y plane

$$\dot{\lambda} = \frac{\mathbf{v}_{east}^l}{(N + h) \cos \varphi}. \quad (3.11)$$

Where N is the radius of curvature in the prime vertical.

The first derivative of the height above the ellipsoid is found by reversing the sign of the down velocity in local level frame.

The position equation in local level frame could now be written as (Schwarz 2001)

$$\dot{\mathbf{r}}^l = \mathbf{D}^{-1} \mathbf{v}^l \quad (3.12)$$

where

$$\mathbf{D}^{-1} = \begin{bmatrix} \frac{1}{(M+h)} & 0 & 0 \\ 0 & \frac{1}{(N+h)\cos\varphi} & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (3.13)$$

and \mathbf{v}^l is the velocity vector in local level frame.

Velocity equation

The velocity equation in local level frame is here derived from the velocity equation in the earth frame, starting with the relation between the position in inertial frame and earth frame (Hofman-Wellenhof, Legat, Wieser 2006) (Whan 2001)

$$\mathbf{r}^i = \mathbf{C}_e^i \mathbf{r}^e \quad (3.14)$$

which is differentiated once to

$$\dot{\mathbf{r}}^i = \dot{\mathbf{C}}_e^i \mathbf{r}^e + \mathbf{C}_e^i \dot{\mathbf{r}}^e \quad (3.15)$$

or written as:

$$\dot{\mathbf{r}}^i = \mathbf{C}_e^i (\dot{\mathbf{r}}^e + \boldsymbol{\Omega}_{ie}^e \mathbf{r}^e). \quad (3.16)$$

Differentiating once again we get:

$$\begin{aligned} \ddot{\mathbf{r}}^i &= \mathbf{C}_e^i (\ddot{\mathbf{r}}^e + 2\boldsymbol{\Omega}_{ie}^e \dot{\mathbf{r}}^e + \dot{\boldsymbol{\Omega}}_{ie}^e \mathbf{r}^e + \boldsymbol{\Omega}_{ie}^e \boldsymbol{\Omega}_{ie}^e \mathbf{r}^e) \\ &= \mathbf{C}_e^i (\ddot{\mathbf{r}}^e + 2\boldsymbol{\Omega}_{ie}^e \dot{\mathbf{r}}^e + \boldsymbol{\Omega}_{ie}^e \boldsymbol{\Omega}_{ie}^e \mathbf{r}^e) \end{aligned} \quad (3.17)$$

The reason why $\dot{\boldsymbol{\Omega}}_{ie}^e \mathbf{r}^e$ falls out is that the earth spin rate is constant. By substitute the left hand side with Equation (3.3) and pre-multiply with \mathbf{C}_i^e one get

$$\mathbf{f}^e + \bar{\mathbf{g}}^e = \ddot{\mathbf{r}}^e + 2\boldsymbol{\Omega}_{ie}^e \dot{\mathbf{r}}^e + \boldsymbol{\Omega}_{ie}^e \boldsymbol{\Omega}_{ie}^e \mathbf{r}^e. \quad (3.18)$$

Using the relation between and gravity, gravitation and the earth rotation (Schwarz 2001);

$$\mathbf{g}^e = \bar{\mathbf{g}}^e - \boldsymbol{\Omega}_{ie}^e \boldsymbol{\Omega}_{ie}^e \mathbf{r}^e, \quad (3.19)$$

we can rewrite Equation (3.18) as:

$$\ddot{\mathbf{r}}^e = \mathbf{f}^e - 2\boldsymbol{\Omega}_{ie}^e \dot{\mathbf{r}}^e + \mathbf{g}^e \quad (3.20)$$

were $2\boldsymbol{\Omega}_{ie}^e \dot{\mathbf{r}}^e$ is the Coriolis force due to the rotation of the earth.

The relation

$$\dot{\mathbf{r}}^e = \mathbf{C}_l^e \mathbf{v}^l \quad (3.21)$$

and its time derivative

$$\ddot{\mathbf{r}}^e = \mathbf{C}_l^e (\dot{\mathbf{v}}^l + \boldsymbol{\Omega}_{el}^l \mathbf{v}^l) \quad (3.22)$$

can now be substituted into Equation (3.20) to indicate the velocity in the e-frame expressed in the l-frame.

$$\mathbf{C}_l^e (\dot{\mathbf{v}}^l + \boldsymbol{\Omega}_{el}^l \mathbf{v}^l) = \mathbf{f}^e - 2\boldsymbol{\Omega}_{ie}^e \mathbf{C}_l^e \mathbf{v}^l + \mathbf{g}^e \quad (3.23)$$

Pre-multiplying with \mathbf{C}_e^l and rearranging leads to

$$\dot{\mathbf{v}}^l = \mathbf{f}^l - (\boldsymbol{\Omega}_{el}^l + 2\mathbf{C}_e^l \boldsymbol{\Omega}_{ie}^e \mathbf{C}_l^e) \mathbf{v}^l + \mathbf{g}^l. \quad (3.24)$$

Using the relation from Equation (1.5), $\mathbf{C}_e^l \boldsymbol{\Omega}_{ie}^e \mathbf{C}_l^e$ could be written as $\boldsymbol{\Omega}_{ie}^l$, yielding

$$\dot{\mathbf{v}}^l = \mathbf{f}^l - (\boldsymbol{\Omega}_{el}^l + 2\boldsymbol{\Omega}_{ie}^l) \mathbf{v}^l + \mathbf{g}^l. \quad (3.25)$$

As mentioned before, the specific force is sensed by the IMU in the body frame, last step is therefore to use the rotation matrix between body-frame and the local level frame.

$$\dot{\mathbf{v}}^l = \mathbf{C}_b^l \mathbf{f}^b - (2\boldsymbol{\Omega}_{ie}^l + \boldsymbol{\Omega}_{el}^l) \mathbf{v}^l + \mathbf{g}^l \quad (3.26)$$

Attitude equation

$$\dot{\mathbf{C}}_b^l = \mathbf{C}_b^l \boldsymbol{\Omega}_{lb}^b = \mathbf{C}_b^l (\boldsymbol{\Omega}_{ib}^b - \boldsymbol{\Omega}_{il}^b) \quad (3.27)$$

Navigation equations

Combining Eqs (3.12), (3.26) and (3.27) into a set of equations for the time derivate of the *navigation state vector* in local level frame yields

$$\dot{\mathbf{x}}^l = \begin{bmatrix} \dot{\mathbf{r}}^l \\ \dot{\mathbf{v}}^l \\ \dot{\mathbf{C}}_b^l \end{bmatrix} = \begin{bmatrix} \mathbf{D}^{-1} \mathbf{v}^l \\ \mathbf{C}_b^l \mathbf{f}^b - (2\boldsymbol{\Omega}_{ie}^l + \boldsymbol{\Omega}_{el}^l) \mathbf{v}^l + \mathbf{g}^l \\ \mathbf{C}_b^l (\boldsymbol{\Omega}_{ib}^b - \boldsymbol{\Omega}_{il}^b) \end{bmatrix}. \quad (3.28)$$

3.3 Mechanization

In the mechanization the navigation equations are solved by using the measurements, compensated for estimated errors, the known earth rotation and the normal gravity. In Figure 3.2 a mechanization scheme for the local level frame is presented.

As with all integration methods good start values are needed. Without a good start value, no integration will give good result. In inertial navigation these start values are found by the initialization.

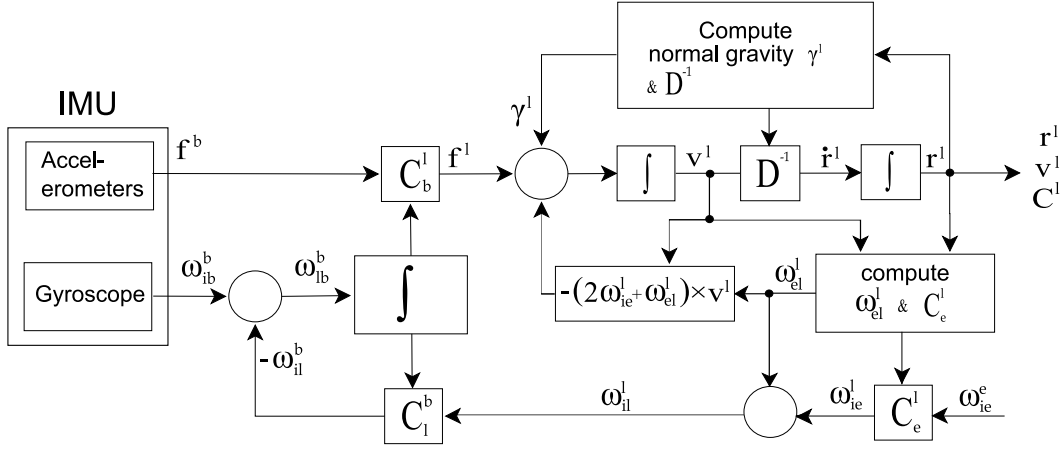


Figure 3.2. Local level frame integration scheme.

3.4 Initialization and Alignment

There are different ways of finding the initial attitude of the unit. Here is outlined one method from Britting (1971), which define the attitude without any external information except from the start latitude.

At a first stage a coarse alignment is performed. It makes use of average data from the accelerometers and gyroscopes during a static initialization epoch. From this data it is possible to analytically resolve the attitude in one step. If

$$\mathbf{g}^b = \mathbf{C}_l^b \mathbf{g}^l \quad (3.29)$$

$$\boldsymbol{\omega}_{ib}^b = \mathbf{C}_l^b \boldsymbol{\omega}_{ib}^l \quad (3.30)$$

also the following matrix relation will hold.

$$\begin{bmatrix} \mathbf{f}^{b'} \\ \boldsymbol{\omega}_{ib}^{b'} \\ (\mathbf{f}^b \times \boldsymbol{\omega}_{ib}^b)' \end{bmatrix} = \begin{bmatrix} \mathbf{f}^{l'} \\ \boldsymbol{\omega}_{ib}^{l'} \\ (\mathbf{f}^l \times \boldsymbol{\omega}_{ib}^l)' \end{bmatrix} \mathbf{C}_b^l \quad (3.31)$$

The rotation matrix could than be solved by

$$\mathbf{C}_b^l = \begin{bmatrix} \mathbf{f}^{l'} \\ \boldsymbol{\omega}_{ib}^{l'} \\ (\mathbf{f}^l \times \boldsymbol{\omega}_{ib}^l)' \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{f}^{b'} \\ \boldsymbol{\omega}_{ib}^{b'} \\ (\mathbf{f}^b \times \boldsymbol{\omega}_{ib}^b)' \end{bmatrix} \quad (3.32)$$

In the NED local level frame the gravity vector will be

$$\mathbf{f}^l = -\mathbf{g}^l = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \quad (3.33)$$

and the earth rotation vector

$$\boldsymbol{\omega}_{ib}^l = \boldsymbol{\omega}_{ie}^l = \begin{bmatrix} \omega_{ie} \cos \phi \\ 0 \\ -\omega_{ie} \sin \phi \end{bmatrix} \quad (3.34)$$

which gives

$$\begin{aligned} \mathbf{C}_b^l &= \begin{bmatrix} 0 & 0 & -g \\ -\omega_{ie} \cos \phi & 0 & -\omega_{ie} \sin \phi \\ 0 & -g\omega_{ie} \cos \phi & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{f}^{b'} \\ \boldsymbol{\omega}_{ib}^{b'} \\ (\mathbf{f}^b \times \boldsymbol{\omega}_{ib}^b)' \end{bmatrix} \\ &= \begin{bmatrix} \frac{-\tan \phi}{g} & \frac{1}{\omega_{ie} \cos \phi} & 0 \\ 0 & 0 & \frac{-1}{g\omega_{ie} \cos \phi} \\ \frac{-1}{g} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{f}^{b'} \\ \boldsymbol{\omega}_{ib}^{b'} \\ (\mathbf{f}^b \times \boldsymbol{\omega}_{ib}^b)' \end{bmatrix} \end{aligned} \quad (3.35)$$

After the course alignment a fine alignment is performed. The measurements are fed to the mechanization and from the mechanization the navigation solution is fed to the Kalman filter. The Kalman filter then estimates the errors of the navigation solution and feed them back to the mechanization. In this way the attitude is refined for each time.

The errors are estimated by making use of the fact that we know that the IMU is in static mode, and that the only observed velocity should be the one of the earth rotation. (To be able to find the heading of the IMU it is necessary that the performance of the IMU is good, If the noise is larger than the earth spin rate, this is not possible.)

3.5 Aidings for inertial navigation

In Section 3.2 the navigation equations has been formulated, a mechanization scheme to solve these equations was described in Section 3.3, and one method to find the start values was outlined in 3.4.

After the mechanization the navigation state vector is fed to the Kalman filter, were the errors are estimated with help of supplementary aiding data and aiding techniques. These techniques will here be outlined.

3.5.1 Velocity aiding

One of the error states in the Kalman filter is the velocity error. It could be seen in Equation 3.28 that the error in velocity directly affect the error in position. So if one could estimate the error in the velocity one would also correct the position state.

Zero velocity update

One way to estimate the velocity error is through a zero velocity update (ZUPT). When the unit is in rest, the only forces acting on the unit should be the one

from the gravity field of the earth and the one from earth rotation, this external information is provided to the Kalman filter, where the difference between the static model and the actual observation are used to estimate the errors.

Non holonomic constraints

The non holonomic constraints is well suited for car navigation. Since the real velocity of the car is constrained to the longitudinal axis of the car, one also have the possibility to constrain the velocity model to one axis, and to assume that all other velocities are effects of errors or noise. The constrained model would therefore be

$$\begin{aligned} v_y^v &= 0 \\ v_z^v &= 0. \end{aligned} \tag{3.36}$$

This will of course only hold as long as the car does not drift side ways or jumps vertically.

Odometer aiding

Another way to estimate the velocity error is to provide the filter with external velocity data. An odometer (See 4.4) could provide this data. This could be used together with non holonomic constraints so that

$$\begin{aligned} v_x^v &= v_{odometer} \\ v_y^v &= 0 \\ v_z^v &= 0 \end{aligned} \tag{3.37}$$

Also the odometer data contains errors and noise. And even if the odometer would deliver error free data, errors due to system assembly could cause problem. If the axis were the aided data are sensed (vehicles longitudinal axis) and the axis of the b -frame are not aligned it may cause a scale factor error, that, if it is not corrected, will cause an accumulated position error. (Also see the lever arm effect in Section 3.5.3)

GPS velocity aiding

The GPS system could provide very accurate velocity in three dimensions.

3.5.2 Position aiding

As seen in Chapter 3 the position is related to both the velocity and the attitude.

Aiding with GPS

The errors of the positions derived from GPS are likely to be of gaussian white nature, which could be used to effectively prevent the accumulating errors of INS

to grow unbounded. On the other hand are the position errors of the INS smaller than the errors for GPS for short periods of time. The challenge is to combine and benefit from these characteristics.

3.5.3 Lever arm effect

The lever arm is the vector that separates the sensors from the origin of the b -frame. Since this coincide with the origin of the IMU the inertial sensor data is automatically corrected to refer to this point by the IMU. Still it is necessary to make the lever arm correction for the external aiding sensors, such as GPS and Odometer.

The offset vectors between the sensors (I.e. GPS antenna phase centre, Odometer) and the centre of the IMU is measured, and denoted $\Delta \mathbf{r}^b$. The lever arm in the navigation frame could then be found by

$$\Delta \mathbf{r}^l = \mathbf{C}_b^l \Delta \mathbf{r}^b \quad (3.38)$$

The correction of the GPS position for the lever arm is done by (Eun-Whan 2005)

$$\mathbf{r}_{IMU}^l = \mathbf{r}_{GPS}^l - \mathbf{D}^{-1} \mathbf{C}_b^l \Delta \mathbf{r}^b \quad (3.39)$$

Where \mathbf{r}_{GPS}^l is the position of the GPS antenna phase center and \mathbf{r}_{IMU}^l is the corrected position at the center of the IMU. The matrix \mathbf{D}^{-1} have been used earlier in Section 3.2.

$$\mathbf{D}^{-1} = \begin{bmatrix} \frac{1}{(M+h)} & 0 & 0 \\ 0 & \frac{1}{(N+h)\cos\varphi} & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (3.40)$$

The lever arm correction for the GPS derived velocity is written as

$$\mathbf{v}_{imu}^l = \mathbf{v}_{gps}^l - \mathbf{C}_b^l \boldsymbol{\Omega}_{lb}^b \Delta \mathbf{r}_{gps}^b \quad (3.41)$$

and for the odometer velocity

$$\mathbf{v}_{imu}^l = \mathbf{C}_b^l \mathbf{v}_{odo} - \mathbf{C}_b^l \boldsymbol{\Omega}_{lb}^b \Delta \mathbf{r}_{odo}^b \quad (3.42)$$

Chapter 4

Hardware

4.1 Accelerometers

The principles of integrating accelerometer data to velocity and positions was described in Section 2.1 and 3.3. Here is a short outline of how the one axis force rebalancing accelerometer *QA2000-40* from Honeywell sense the specific force (Peters, Stoddard, Meridith 1998).

The accelerometer is built up around a proof-mass made of quartz, which is allowed to flex through an etched flexure. When the proof-mass is affected by acceleration it will flex and shift position, the new position of the proof-mass is determined by a capacity pickoff module. In the electro magnetic feedback system the position of the mass is used to torque the mass back to zero position (rebalancing). This closed rebalancing servo loop make sure that the amplitude of the flexing proof mass is minimal, which results in a good linear output (Dorobantu, Gerlach 2004). This also means that the current that goes through the rebalancing electro magnets is propositional to the specific force acting on the quartz proof mass.

Inside the accelerometer there is also an temperature sensor, used to model the performance parameters affected by variations in temperature.

The output from the *QA2000-40* is analog and is therefore converted into a digital signal in the IMU by a AD converter.

It should also be mentioned that the *QA2000-40* is an one axis sensor, implying that three orthogonally assembled accelerometers have to be used.



Figure 4.1. The QA2000-40 accelerometer from Honeywell. [Honeywell]

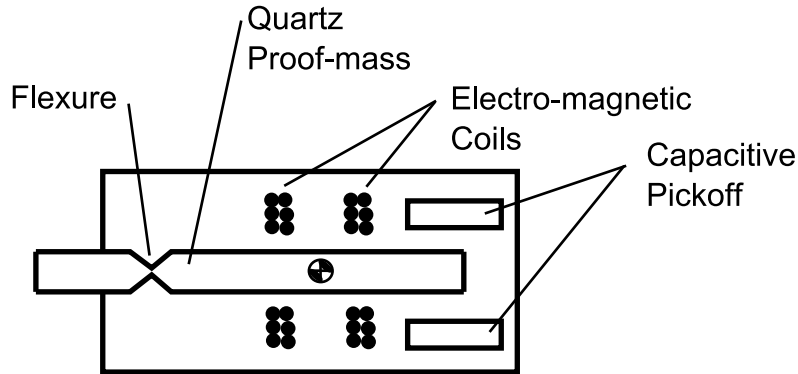


Figure 4.2. In this simple sketch one can see the proof-mass that can flex in the vertical sensible axis due to the flexure. The two pickoff plates sense the capacity which change when the proof-mass moves. This is fed to the electro magnetic coils that rebalance the proof-mass to its original position.

Accelerometer measurements

In Chapter 2 was given an simple equation (Equation 2.3) that described the relations between measurements, observables and errors. Now when the principle of an force rebalancing accelerometer have been outlined, is it also easier to understand the types and sources of the errors, therefore the measurement equation is extended

$$\ell_f = \mathbf{f} + \mathbf{b} + (\mathbf{S}_1 + \mathbf{S}_2 \mathbf{f}) \mathbf{f} + \mathbf{N} \mathbf{f} + \mathbf{n}_f \quad (4.1)$$

Where

ℓ_f is the measurement

\mathbf{f} is the specific force

\mathbf{b} is the bias of the accelerometer, an offset of the output

\mathbf{S}_1 is the linear scale factor errors between the output and the acting force

\mathbf{S}_2 is the *non* linear scale factor errors between the output and the acting force

\mathbf{N} represent the non-orthogonally of the accelerometer assembly

\mathbf{n}_f is the sensor noise

For performance data of the QA2000-40, please refer to Appendix C.

4.2 Gyroscopes

The Ring Laser Gyro from Honeywell is built of a closed triangle shaped resonant cavity with mirrors in each corner. Inside the closed cavity two laser beams are created, one is traveling clock wise (*cw*) and the other one counter clockwise (*ccw*). The beams have the same integer number of wavelengths, and when the cavity is in rest (in inertial space), they will also have the same length of traveling and the same frequency.

If the cavity is rotated *cw*, the frequency of the *cw* laser beam has to increase in order to keep the number of wavelengths constant despite the fact that the way of traveling now is shorter due to the rotation. The *ccw* laser beam will instead have to decrease in frequency (Siouris 1993).

The frequency shift that will appear between the beams when the cavity is rotated is proportional to the rotation rate and could be found by detecting the change in the fringe pattern produced when the two signals are interfered in the detector. (see Figure 4.3).

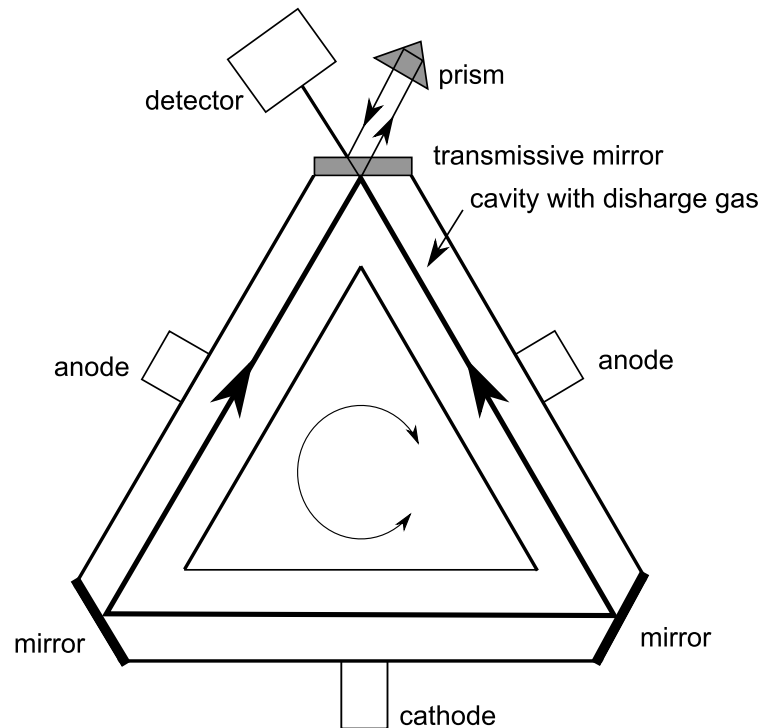


Figure 4.3. One axis ring laser gyroscope. The *cw* and *ccw* laser beams are created by the gas discharge when high voltage is applied on the anode-cathode pairs. One of the corner mirrors allows some of the light to pass through, and to reach the optic detector.

RLG sensor errors

$$\ell_\omega = \omega + d + S\omega + N\omega + n_\omega \quad (4.2)$$

ℓ_ω is the measurement

ω is the rotation in inertial space

d is the drift of the gyro, an integrated output offset.

S is the linear scale factor errors between the output and the real rotation.

N represent the non-orthogonality of the gyroscope axis, due to non orthogonal assembly

\mathbf{n}_ω is the sensor noise

For performance data of the Ring laser gyro, please refer to Appendix C.

4.3 IMU

The iNAV-RQH IMU from iMAR is the housing for the above described sensors. It also contains other hardwares such as AD converters, clock device, storing capacity and CPU. Files for sensor calibration are also stored internally and used before output.

I have been using the iNAV-RQH as an measuring device, to acquire and record inertial data for post processing. The unit could also be used as an real time inertial navigation system, using the internal CPU, algorithms and Kalman filter for real time processing.

4.4 Odometer

Odometer in general

The name odometer comes from The Greek words *hodós* meaning path, and *métron* meaning measure. One simple form of odometer is the wheel odometer where the number of rotations of a wheel is counted and the perimeter of the wheel is known. The distance traveled is the number of rotations, times the perimeter, and if the velocity is of interest one just have to divide the perimeter with the time of each rotation.

In our experiments have we not been using a wheel odometer, but an optical odometer, which could measure distances without surface contact. The optic odometer is property of *Ecole polytechnique fédérale de Lausanne*, Switzerland, and have been borrowed for our experiments. The data have been acquired by the IMU, time tagged, and written into the IMU file together with the other sensor data.

Principles of Correvit L-CE

The Correvit L-CE, from the manufacture Corrsys-Datron, is an advanced optic sensor for odometric use, which make use of optical gratings to find the velocity (Corrsys-Datron Sensorsysteme). A grating could be composed by a set of spaced parallel elements, where the space between the elements are equal to the width of the element. The characteristics of such a grid is described by the density of the elements, and denoted k .

If light reflected from the surface of the street pass this grating, and the grating moves in an direction orthogonal to the elements it will cause a modulation of the light intensity that pass the grating. This modulation is sensed by a sensor and its frequency could be found, denoted f . Fig 4.4

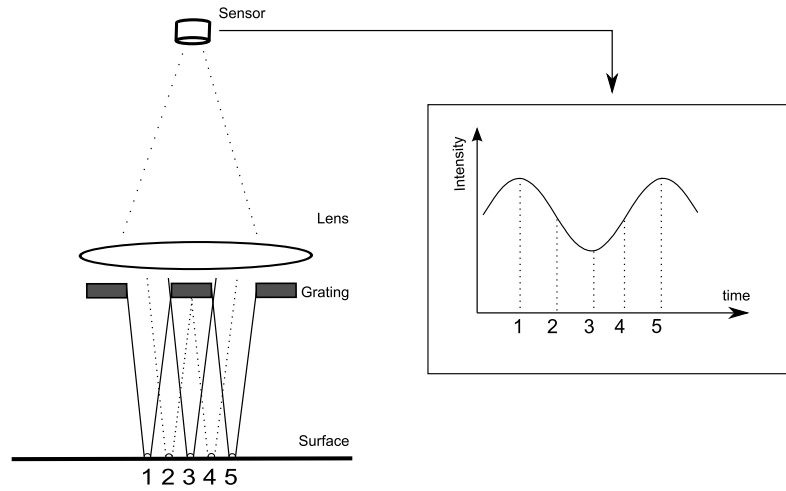


Figure 4.4. To better understand the optical odometer principle, one considers an odometer displacement in the left direction, or equivalently, a right relative displacement of the light spot 1 (reflected from the road structure), with its successive time positions 2...5. The corresponding time variation of the sensor light intensity has approximately a sine-form (the maximum light intensity corresponds to position 1, the minimum one to the position 3, the next maximum to position 5, and so on.)

The signal frequency tells how many peaks that is detected per second, and we already know the distance between each peak from k .

Because of the optic scale factor of the lens system it is also needed to multiply with a scale factor, denoted m . The observed frequency could therefore be described as

$$f = \frac{mv}{k}, \quad (4.3)$$

where v denote the velocity over the surface of the street.

In a more advanced optic odometer, advanced prism gratings are used together with *two* sensors. The two signals generated are shifted 180 degrees in phase, as seen in Figure 4.5. To remove the low-frequency superimposed common mode signal (produced, e.g. by the slow mean-intensity changes of the reflected beam - ground luminosity, dependent on the road surface structure) one difference the two signals, generating a differential-mode signal. (Figure 4.6)

When the frequency of the useful signal has been determined and the velocity derived from Equation 4.3, next step is to generate the output signal. This signal is

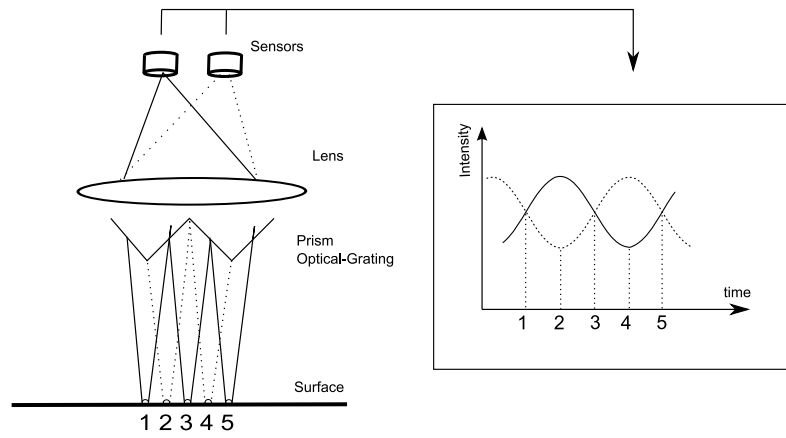


Figure 4.5. The light that reflects into the prism grating is divided so that one of the sensors only receive light from the left side of each prism, and the other sensor only will receive light from the right side of the prism. The two wave shaped modulations that occur when the odometer is moving are shifted 180 degrees.

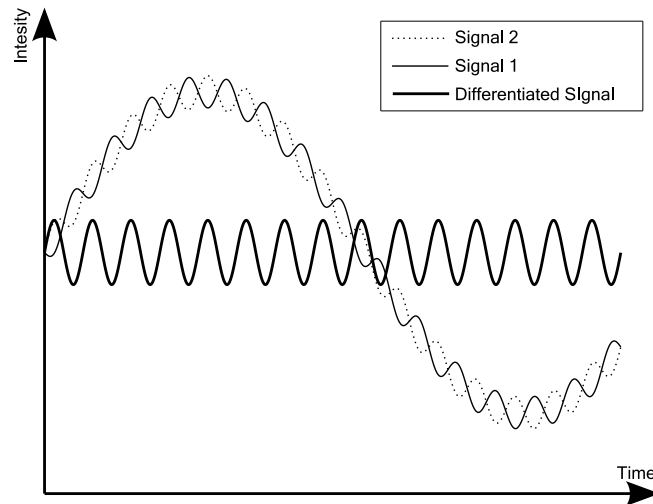


Figure 4.6. Signal 1 & 2 are shifted 180 degrees to each other, and they are both disturbed by an underlying signal of longer wave length. The effect of the disturbing signal could be eliminated by differentiate signal 1 & 2.

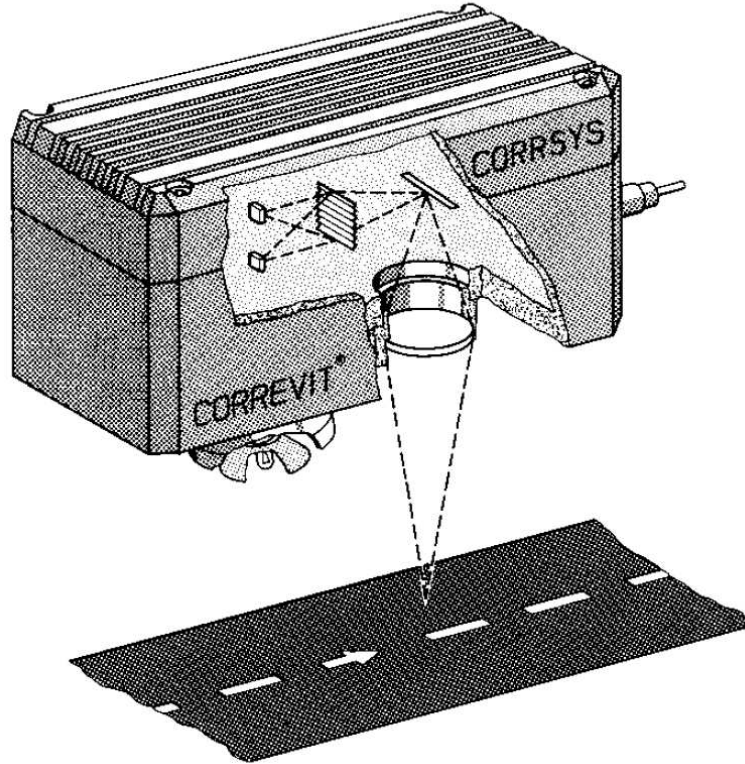


Figure 4.7. *Correvit L-CE*, below the housing to the left can one see the halogen lamp (not obvious an lamp in this sketch) that illuminates the surface. Inside: the prism grating and optic system.

of transistor-transistor logic (TTL) type, a rectangular shaped signal that theoretical is either 0 or 5 volt, down or high. The odometer is configured to send n pulses per meter, the frequency of the output signal is therefore

$$f_{signal} = n \frac{pulse}{m} \times v \frac{m}{s} \quad (4.4)$$

and the period of the pulse is

$$T = \frac{1}{f_{signal}} \quad (4.5)$$

The output with period T is shaped as one down of the length $0.25 * T$ seconds and one high of the length $0.75 * T$ seconds. (figure 5.1)

Laboratory test of *Correvit L-CE*

The performance test was carried out by registering the odometer output when used on a test surface. The test surface was a rotating turntable for which the turn rate was measured using stroboscopic light of known frequency.

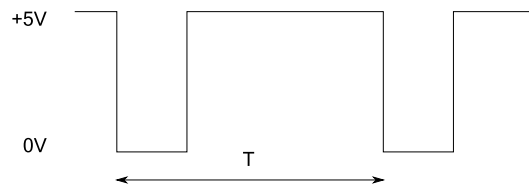


Figure 4.8. The TTL signal is used to transmit the velocity data from the odometer. The period T is equal to one pulse, and the odometer is configured to send n pulses per meter.

Method

The turn table was calibrated for the scale factor between input signal to the turntable, and the turn rate of the turntable. This was done by changing the frequency of the stroboscopic light until it coincided with the turn rate. It was very convenient that the turntable had an scale engraving in degrees on the side. The scale factor was than estimated by a least square estimation.

The scale factor between odometer measured velocity and the calibrated turntable velocity was estimated by logging data from the rotating turntable, and adjusted the data by least square estimation.

We also tested the influence of the test surface. Starting with a paper surface of fine texture and a limited light pattern, we later added black dots to get a surface with a pattern easier to process for the odometer.

We also made a registration with half of the normal sampling rate, to see if there would be any changes in the data acquisition.

It should be mentioned that the test was carried out in two steps. First the odometer was used together with the turntable and test surface. In step two the turn rate of the turntable was examined with the stroboscope.

Test setup

The turn rate of the turntable was observed by a stroboscope assembled by a LED and a signal generator. The shape of the signal was controlled by an oscilloscope.

The input signal to the turntable was changed by a stabilized power supply.

The odometer was set to send 446 pulses per meter, send by the TTL signal. This signal was send to the IMU where the signal was quantified and processed before it was written into a text file in the laptop computer. (Also see Figure 1.1)

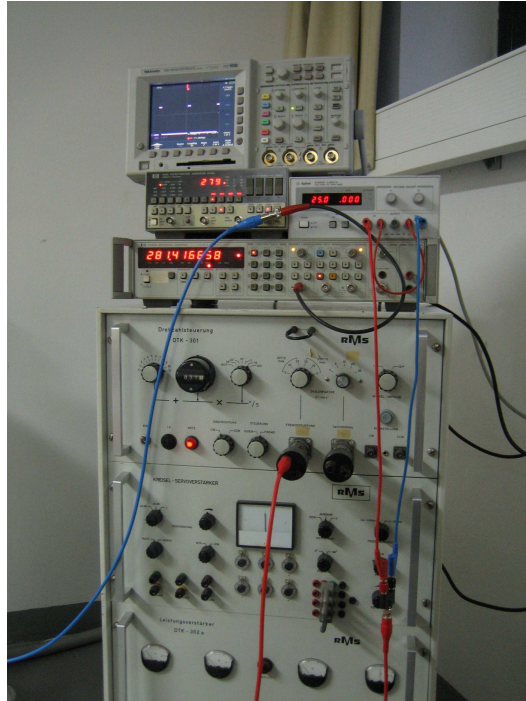


Figure 4.9. Setup. On the top is the oscilloscope, under to the right is the power supply, under the power supply, the signal generator and the larger device is the turntable control unit.

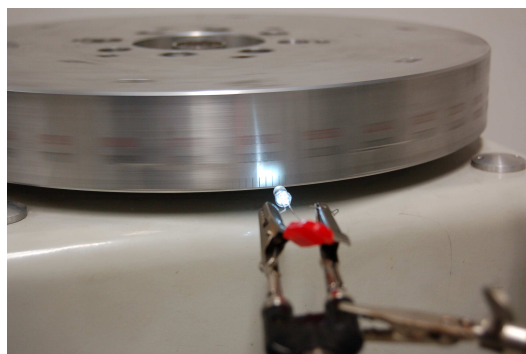


Figure 4.10. Picture of the turntable and the stroboscope. By adjusting the frequency of the stroboscope, the turn rate of the turn table could be determined.

Result of laboratory test

The scale factor between input voltage (V) and turn rate (deg/s) was found to be $11.237 deg/Vs$, and the standard error of the residuals were $2.203 deg/Vs$. This values correspond with the velocity of the maximum radii of the turn table by; $0.029 m/Vs$ and standard error $0.006 m/Vs$.

To better understand the results from the estimated scale factor for the odometer, one should first see the result of the error characteristics for the odometer.

At a velocity of 1,6 m/s, measurements had a standard error of 0.11 m/s. But in the test data, ranging between 0.16 and 1.6 m/s, there is also a linear progression of around 0.1 times the velocity for the standard error.

The test with different patterns on the test surface resulted in an decrease of the standard error. By adding black dots to the surface the standard error decreased from 0.17 m/s to 0.12 m/s at a velocity of 1.6 m/s. In Figure 4.11 the measured velocity is plotted, one could see the difference in noise and also the sampling error. One could also see that the mean velocity have dropped to 1.62 from 1.63 m/s, when the dots were added. We also found that this difference is increasing with velocity.

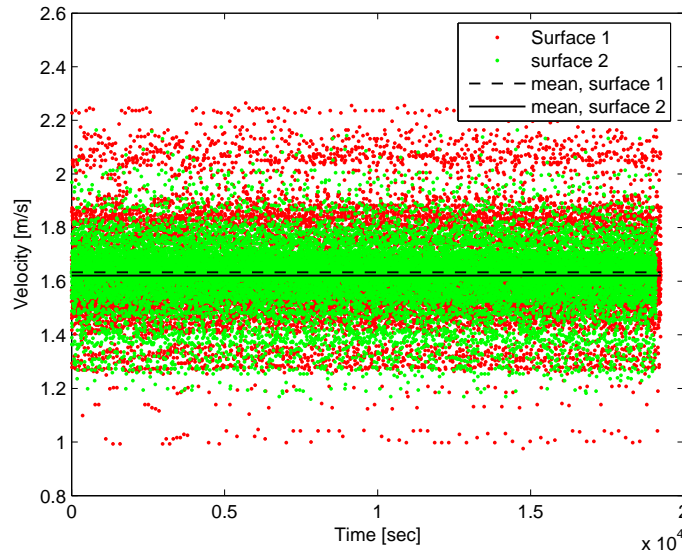


Figure 4.11. Plot of odometer derived velocities, where surface umber two is having a pattern of better contrast than surface number one.

By generating a TTL signal with a signal generator, we could see that this data, as well as the true odometer data, was affected by a sampling error when quantified with the IMU. This could be seen as bands of denser data in Figure 4.11.

An registration was also done were the TTL was reduced to 230 pulses per meter, but without any improvement.

The relation between the velocity of the turn table and the measured velocity by the odometer was measured with a test surface of the better type and was found to be 1.018.

Conclusions of laboratory test

The test showed that the absolute velocity and the linearity of the odometer was hard to determine or estimate with our methods and hardware.

The noise of the odometer had an small, positive correlation with the velocity, but more important was the relation between the velocity noise and the pattern of the measured surface. An surface with strong random dots in high contrast lowered the noise remarkably compared to a pattern with lower contrast. This change in the measurements also lowered the average velocity, which would result in non linear errors if traveling over varying surfaces.

The error caused by the IMU registration could be a part of this velocity decrease. There is a IMU sampling error and the effect is more legible with larger noise. The error in the data acquisition has not been further investigated.

Because of the dynamic surface and the higher velocities in a real navigation mission I have not found the data useful for a odometer calibration of high precision.

4.5 GPS

GPS plays an important role in my experiments. But aside from the specifications of the Trimble 4000SSE in Appendix C no further details of the hardware or the Global Positioning System (GPS) will be given in this report.

All GPS solutions are relative, differential, solutions of either pseudorange or phase shift types.

Chapter 5

Extended Kalman filtering

The nine parameter navigation state vector that was developed in Chapter 3 contains errors due to the sensors, not perfect assembly of sensors and uncertainty of the gravity field.

These errors are often modeled and estimated by a Kalman filter. Using data from the aiding sensor to estimate the errors, this results in a data fusion of IMU and aiding sensor data.

In this chapter the principles of a 15 error states extended Kalman filter (EKF) will be outlined (Schwarz 2001) (Jekeli 2001) (Shin 2001) (Welch, Bishop 2001). The 15 states are nine states for errors in the navigation states, plus six sensor error states.

5.1 Filter design

Error analysis

The dynamic error process that takes place in the INS due to sensor errors could be studied by expanding the navigation equations into a Taylor series, truncated to the terms of first order. By the truncation become the error dynamics equations linear, this assumes that the errors are "sufficiently" small (Jekeli 2001). In practice the error dynamics equations are derived by perturbing the navigation equation differentially.

It is also possible to include the perturbed sensor biases and scale factors in the error states. A system of linear differential equations for the error states referring to Equation 3.9 with augment for sensor errors could be developed as in Schwarz (2001):

$$\delta \dot{\mathbf{x}}^i = \begin{bmatrix} \delta \dot{\mathbf{r}}^i \\ \delta \dot{\mathbf{v}}^i \\ \dot{\boldsymbol{\epsilon}}^i \\ \dot{\mathbf{d}}_g \\ \dot{\mathbf{b}}_a \end{bmatrix} = \begin{bmatrix} \delta \mathbf{v}^i \\ \mathbf{f}^i \boldsymbol{\epsilon}^i + \mathbf{N}^i \delta \mathbf{r}^i \\ \mathbf{C}_b^i \mathbf{d}_g \\ -\alpha \mathbf{d}_g + \mathbf{w}_d \\ -\beta \mathbf{b}_a + \mathbf{w}_b \end{bmatrix}. \quad (5.1)$$

or

$$\delta\dot{\mathbf{x}} = \mathbf{F}\delta\mathbf{x} + \mathbf{G}\mathbf{w} \quad (5.2)$$

where

$$\delta\mathbf{x} = \left[\delta\mathbf{r}' \quad \delta\mathbf{v}' \quad \boldsymbol{\epsilon}' \quad \mathbf{d}'_g \quad \mathbf{b}'_a \right]' \quad (5.3)$$

are the first nine error states ($\delta\mathbf{r}$ $\delta\mathbf{v}$ $\boldsymbol{\epsilon}$) of the state parameters in the dynamic system (\mathbf{r} \mathbf{v} \mathbf{C}_b^l). And the next six states (\mathbf{d}_g \mathbf{b}_a) are the drift of the gyroscope and the bias of the accelerometer (see Chapter 4). \mathbf{N} is a coefficient matrix that describe the linearized normal gravity error due to position error. \mathbf{F} is the dynamic system matrix. White noise is introduced by (\mathbf{w}) and the shaping matrix \mathbf{G} .

System integration

The fusion of INS data and data from the aiding systems i.e. sensor integration, could be done in a numerous ways. One could order the ways it is done by how tight the coupling between the sensor data is in the filter. In a tight coupled system could for example every single satellite observations be directly used together with the inertial data. In such a system would observations from less than four satellites still be useful.

In a loosely coupled filter the inertial data is processed through the mechanization, the odometer velocity data is derived by a separate routine and the GPS positions and velocities are generated by separate software or routines. The EKF is than used to fuse the data together. A flow chart of a loosely coupled system be seen in Figure 5.1.

By continuously improve the error states of the INS by estimations based on both INS performance and the aiding measurements the sensor data are fused together. This estimation starts by forming the error measure $\delta\mathbf{z}$

$$\delta\mathbf{z}^l = \mathbf{x}_{ins}^l - \mathbf{x}_{aiding}^l \quad (5.4)$$

where \mathbf{x}_{ins}^l is the current navigation state from the mechanization and \mathbf{x}_{aiding}^l is one or more of the navigation states measured from an aiding sensor and corrected for eventual lever arm effect.

Error update and correction

The next step is to blend the error state ($\delta\mathbf{x}$) and the error measure ($\delta\mathbf{z}$),

$$\delta\hat{\mathbf{x}}_k = \delta\hat{\mathbf{x}}_k^- + \mathbf{K}_k(\delta\mathbf{z}_k - \mathbf{H}_k\delta\hat{\mathbf{x}}_k^-) \quad (5.5)$$

were \mathbf{K} is a blending gain factor matrix, $\hat{\mathbf{x}}^-$ is an prediction (forward projection) of the estimated error state, \mathbf{H}_k relates the error states to the measurements and $\delta\hat{\mathbf{x}}_k$ is the updated estimate of the error states at the current state k .

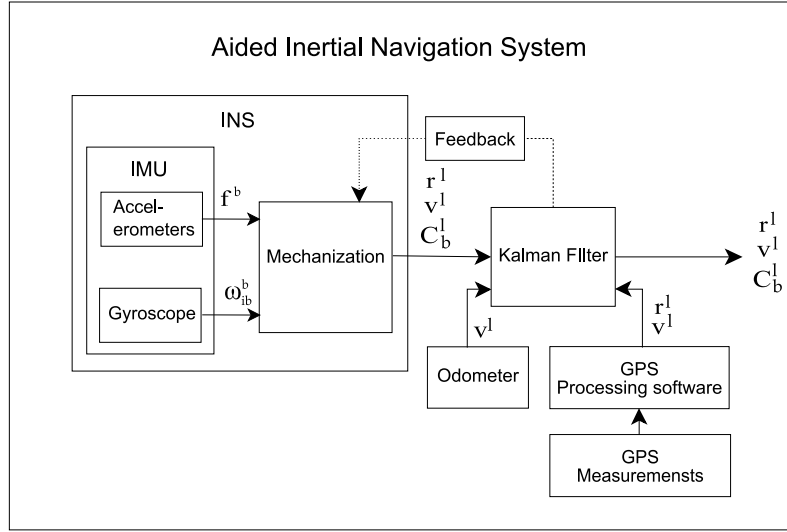


Figure 5.1. Loosely coupled aided inertial navigation system, where the INS system have been described in Figure 3.2. Note that the GPS data and the odometer data have been transformed by the use of C_e^l and C_b^l , were the later is fed from the KF output.

To relate this to what we actually want to achieve, an error correction of the INS navigation state, it is now shown that

$$\mathbf{x} = \mathbf{x}_{ins} - \delta\mathbf{x}$$

is an combined navigation solution, since $\delta\mathbf{x}$ by Formula (5.4) and (5.5) relate to the aiding state \mathbf{x}_{aiding} via the error measure $\delta\mathbf{z}$.

Finding the optimal solution

The predicted estimate of the error state $\delta\hat{\mathbf{x}}^-$, and the gain factor \mathbf{K} In Equation 5.5 are still unknown. Finding these is the core of the EKF and could be written by four equations (Jekeli 2001).

Time update (prediction) equations:

$$\delta\hat{\mathbf{x}}_k^- = \phi_{t_k, t_{k-1}} \delta\hat{\mathbf{x}}_{k-1} \quad (5.6)$$

$$\mathbf{P}_k^- = \phi_{t_k, t_{k-1}} \mathbf{P}_{k-1} \phi_{t_k, t_{k-1}}' + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k' \quad (5.7)$$

Were $\phi_{t_k, t_{k-1}}$ is the state transition matrix, used to predict the error states and error covariance at time k , under the assumption that the error dynamics are constant over the time interval Δt . \mathbf{Q}_k is the covariance matrix of the system noise process, and \mathbf{G}_k is the shaping matrix of the measurement noise process at time t_k

Measurements update equations:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k' (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k' + \mathbf{R}_k)^{-1} \quad (5.8)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)' + \mathbf{K} \mathbf{R}_k \mathbf{K}' \quad (5.9)$$

Where \mathbf{R}_k is the covariance matrix of the measurements at time k .

Feed back implementation

As have been pointed out before, the errors in the nine navigation states in INS are dependent of previous states, and therefore the errors are accumulating. The Kalman filter estimates these errors and makes corrections of the output data. But in time the output errors from the INS could grow larger than the Kalman filter can handle. The Kalman filter uses a linearization of the non linear dynamic system and hence only small errors can be accepted.

So depending on the performance of the IMU it is necessary to feed the result of the filtered states backwards, back to the INS mechanization. This is in also followed by setting the error states in the Kalman filter back to zero (Shin 2001)

Another form of feedback is the correction of sensor data. If the Kalman filter estimates the sensor biases and scale factor errors, these errors could be used to correct the sensor measurements prior the mechanization.

Extension of the Kalman filter

There are good reasons not to extend a Kalman filter with too many states, at least not without first analyze the observability of the states (Schumacher 2006). There is an increasing filter instability with the number of the state variables. If one have limited computer power it is also a good idea not to produce to big matrixes.

A state variable for the scale factor of the odometer would indeed be a good way to calibrate the odometer on the fly, making use of the long term stable GPS data. We also proved that the noise of the odometer depends on the surface measured, but in the Kalman filter used, the standard deviation of the odometer is fixed in the \mathbf{R} matrix. Therefore an improvement of the uncertainly estimation of the odometer derived velocities would probably improve the result as well.

5.2 Pre-filtering of data

I did some experiments where the accelerometer data was filtered prior the mechanization and Kalman filtering. This was done by a FIR II low pass filter of various degrees, together with the Matlab `FiltFilt` function which does not introduce any phase shifts.

The reason for pre-filter the data is that the data have been sampled at 200 Hz, while the true motion dynamics is much lower, i.e. the frequency of position,

velocity and attitude. (The relative high acquisition rate of 200 Hz was adopted to minimize the potential coning and sculling errors in the presence of vibrations.)

The low frequency noise and errors are effectively reduced by the fusion of GPS data, and while this is possible also the effect of high frequency noise is suppressed. But for the short time between GPS updates, in the presence of GPS outages or if GPS data is not available at all, high frequency noise will have a negative effect on the solution (Skaloud 1999).

If the frequency spectrum of the true motion dynamics could be found, it would mean that a simple low pass filter as the one described above, could be used before the mechanization and sensor integration to improve the solution. (See Figure 5.2)

To find the frequency spectrum of the motion dynamics I compared the power density spectrum of data from when the unit was in motion, and data from when the unit was in rest. Although this is not a very precise method, it gave a hint of the frequency spectrum of the motion dynamics.

Pre-filtering of data was not a major part of my work, but the results gave some hints of how the pre-filtering could be used in future work.

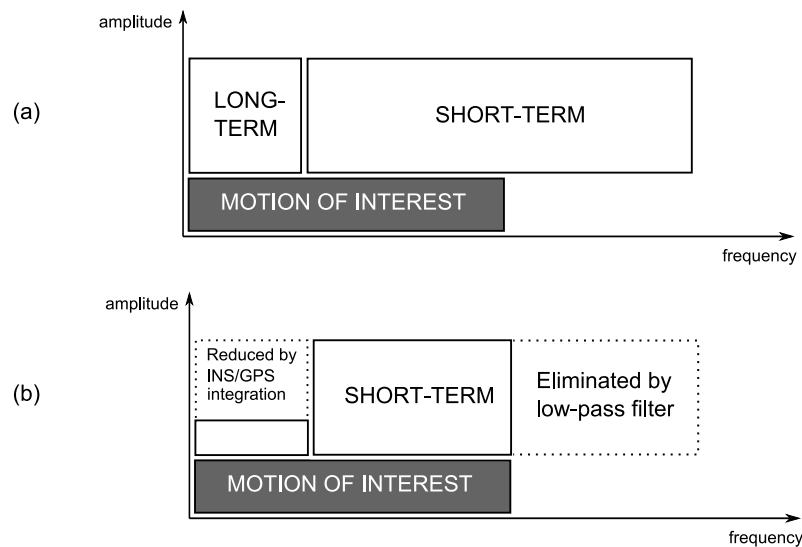


Figure 5.2. Figure after Skaloud 1999. Plot of inertial signal in the frequency domain: (a) before filtering, (b) after INS/GPS integration and low-pass filtering

Chapter 6

Theresienwiese, Outdoor navigation mission

For this mission the system was mounted on a car and the mission took place at the Theresienwiese, a big open field in Munich, Figure 6.1. Theresienwiese have been used in previous missions by IAPG. The openness and low skyline makes good conditions for receiving GPS signals, and there is a grid of paved roads running all over the field.

The performance of the optic odometer depends of the surface, (see Section 4.4) hence it was favored to do the mission on a day without a snow cover.



Figure 6.1. Theresienwiese and the system mounted on a car.

6.1 Aim of mission

The aim of this mission was to get all hardware to work together, and to see how the sensors complete each other.

6.2 System setup

The data acquisition was carried out with a system built up by the following sensors:

- iNAV-RQH from iMAR (described in 4.3)
- System 4000SSE from Trimble (described in 4.5)
- Correvit L-CE from Corsys-Datron (described in 4.4)

The IMU and GPS antenna was mounted on the roof of the car, and the odometer was mounted at the back of the car (see Photo 6.1). The GPS receiver and the operating laptop was installed inside the car. Each sensor also had its own set of accumulators for power supply inside the car. A GPS base station was put up in the center of the field, so that relative GPS position solutions could be calculated.

To be able to compute the lever arm it is needed to know the relative positions of the sensors. Therefore the positions of the IMU, odometer and GPS antenna was measured with a total station.

6.3 Data flow

The communication between the IMU and the laptop is done via Ethernet and TCP/IP protocol. During the navigation there is a constant flow of IMU data which is stored on the laptop in a text file, the *IMU file*. (See Appendix A for details about the *IMU file*). The inertial data could be stored also simultaneously on the IMU, in binary format.

Two output cables leaves the GPS receiver, one is for the NMEA signal, a navigation message standard from the *National Marine Electronics Association*, from which the GPS time could be obtained. The second cable is for the *pulse per second* (PPS) signal, which is used by the IMU for precise time synchronization with the GPS time. The raw GPS data are logged internally in the GPS receiver.

The data from the odometer is transmitted to the IMU via the digital TTL signal (see section 4.4). In the IMU the signal is quantified, pre-processed, time tagged and included in the IMU file. In our case the odometer was configured to send 446 pulses per meter.

6.4 Data sets

All the data sets started with a 15 minutes static registration, which is needed for the post-processing determination of the initial attitude (initial alignment phase). All data sets also ended with a 15 minutes static registration, which can be used for smoothing or reverse filtering of the navigation solution in post-processing.

Data set 1

The alignment phase was followed by a 7 minutes long drive in the field. One minute long ZUPTs was planned every 40-50 second. The ZUPTs was carried out with the engine turned off and the driver and operator sitting as still as possible!

Data set 2

This data set followed in much the same procedures as the first one, starting with a 15 minutes long static registration, followed with a 30 minutes long drive with one minute long ZUPT every 40-60 second.

Data set 3

The third data set was supposed to be a five minutes long drive on the Theresienwiese without any stops, I.e. without any ZUPTs. During the drive was no problem to see, but while downloading the file from the IMU an error message appeared. The file was corrupt and we were not able to restore the data.

The manufacturer have spotted the problem and the new software eliminate the bug that caused the problem.

6.5 Data preparation

IMU file

Prior the processing the sensor data is rotated to better suit the local level system used by the AINS Toolbox.

Prior the integration with other sensors it is also important to make sure that the time stamps in the measurement files coincide with the same time reference. We have used GPS time in seconds of the week (SoW).

GPS data

The AINS toolbox can be used to fuse IMU data with GPS derived positions and velocities. But there is no internal engine to process the raw GPS data in the toolbox. Therefore we have used the Trimble Total Control (TTC) software, to generate a relative pseudorange GPS solution, which could be used as input to the toolbox, and from here on simply denoted GPS solution or GPS.

TTC was also used to process a precise reference trajectory from relative phase measurements (best estimated trajectory), denoted reference.

Odometer data

The sensitive axis of the odometer was in the mission aligned with the longitudinal axis of the car, mounted 1.8 meters behind, 0.15 meters to the right and 1.46 meters below the IMU (-1.70 -0.15 -1.46, given in the body frame). The fact the odometer is

not centered at the x-axis ($y \neq 0$) means that the sensed velocity from the odometer has to be corrected with respect to the lever arm.

The fact that the sensor is placed behind and below the origin of the IMU, could be ignored, since our odometer only senses velocity in the x-axis.

The output rate from the odometer was set to 460 pulses per meter or 2.2 mm per pulse. Already in a speed of 36 km/h this will produce an output rate of 4.6 kHz.

The data is registered by the internal CPU and software in the IMU. Some problem with quantification of the odometer data have been seen, especially with higher output rates of the odometer (higher velocity).

The raw odometer data is very noisy but could be low pass filtered with good results. The data from this mission was resampled to 1 Hz, using the resample function in MatLab. The function uses an anti-aliasing filter and also compensates for time shift which arise during the filter process. The data still showed an average higher velocity than the solution from the GPS/INS integration, which might be deduced to the quantification problem. The velocity difference was corrected by a least square estimated scale factor of 0.997.

A comparison could be found in Figure (6.2), (6.3) and Table (6.1). Notice that the scale factor found by the comparison with GPS derived velocity differ from the one found during the performance test in Section 4.4.

Table 6.1.

Std error before correction	0.032 <i>m/s</i>
Std error after correction	0.030 <i>m/s</i>
Mean error before correction	0.005 <i>m/s</i>
Mean error after correction	0.000 <i>m/s</i>

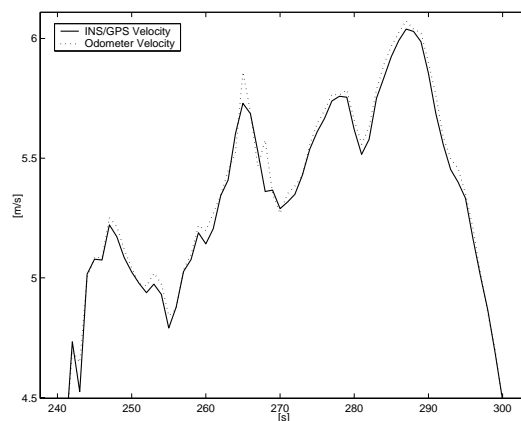


Figure 6.2. Velocity with respect to time, before correction of the resampled odometer data.

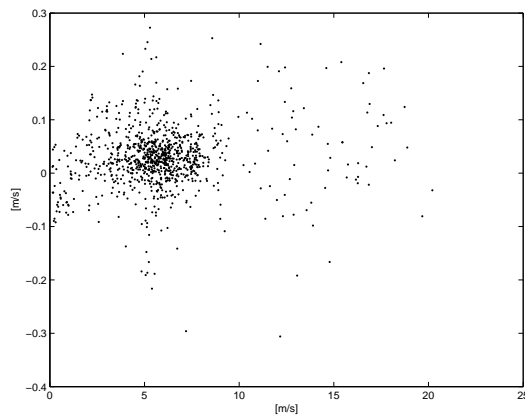


Figure 6.3. Deviation between resampled odometer velocity and GPS/INS velocity, plotted with respect to velocity.

6.6 GPS / INS / Odometer Integration with AINS toolbox

The INS data was processed and integrated with GPS and odometer data using a MatLab toolbox, the AINS toolbox. The power of the AINS Toolbox is its versatility, there are a number of tools included, and it is easy to look into the code to see how things are done, or to do some changes. The documentation is found in the manual, and in the dissertation of the toolbox author, Eun-Whan Shin (2005).

I have only been using one of two main filters in the toolbox, the extended Kalman filter (EKF). The other one is a sample based, unscented filter, that I am looking forward to test in the future.

6.7 Results of outdoor experiment

Here are the most important results from the processing of *data set 2* (see section 6.4) presented. The tuning of the Kalman filter have for each experiment been extend to an level where the results should give a understanding of the error magnitude for each aiding technique, and the relative difference between them.

As reference for the experiment a relative phase measurement GPS solution has been generated with TTC (best estimated trajectory). Deviation from this trajectory has been treated as an error.

For the numerical presentation of the trajectory error a radial root mean square (RMS) error has been used. That is the RMS of the radial error of the trajectory projected to a plane by a Gauss Krüger map projection. The RMS error is given as

$$RMS\ error = \sqrt{\frac{\sum (X - \bar{X})^2 + \sum (Y - \bar{Y})^2}{N}} \quad (6.1)$$

where X and Y denote the map coordinates of the projected test trajectory, and \bar{X}

and \bar{Y} denote the map coordinates of the projected reference trajectory. N is the number summated errors.

For the numerical representation of errors due to GPS data outages the maximal radial error of the projection during the outage has been used.

$$radial\ error = max \left(\sqrt{(X - \bar{X})^2 + (Y - \bar{Y})^2} \right) \quad (6.2)$$

For the graphical presentation is either the test trajectory presented in a map projection together with the reference trajectory, or as the error of the north, east and height above the ellipsoid components, plotted with respect to time.

INS only solution

The first experiment was to process a pure INS solution, by using IMU data only. The initial attitude of the IMU where determined using the 15 minutes long static registration (Start position was given) . (Figures 6.4 and 6.5)

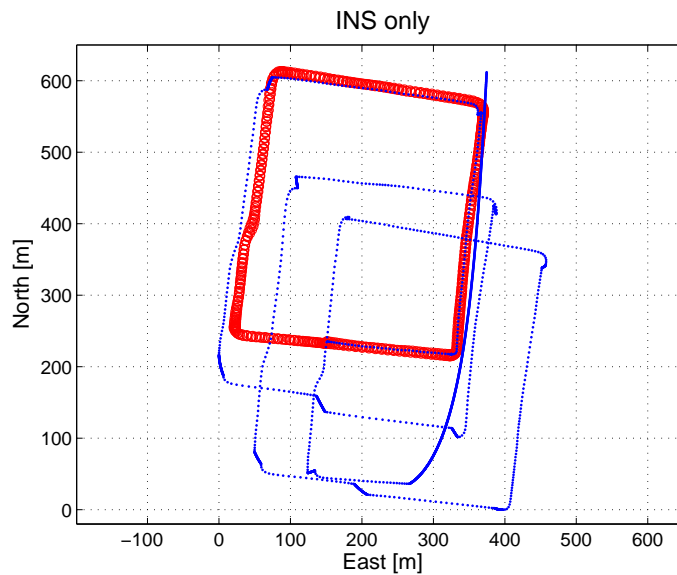


Figure 6.4. Trajectory computed with INS data only (test trajectory). The reference trajectory is seen as red circles, the test trajectory as blue dots. The 60 seconds stops could be seen from the drift in the corners.

INS only solution with pre-filtered IMU data

For this result the accelerometer data was pre-filtered for high frequency noise by using the filter described in Chapter 5.2. Still only IMU data are used. (Figure 6.6)

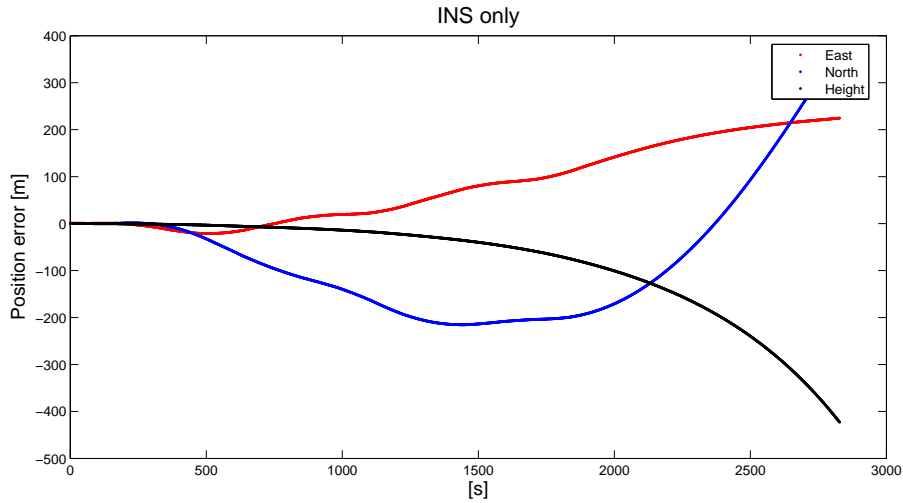


Figure 6.5. Error plot of the test trajectory in figure 6.4, timing from right after the end of the alignment phase.

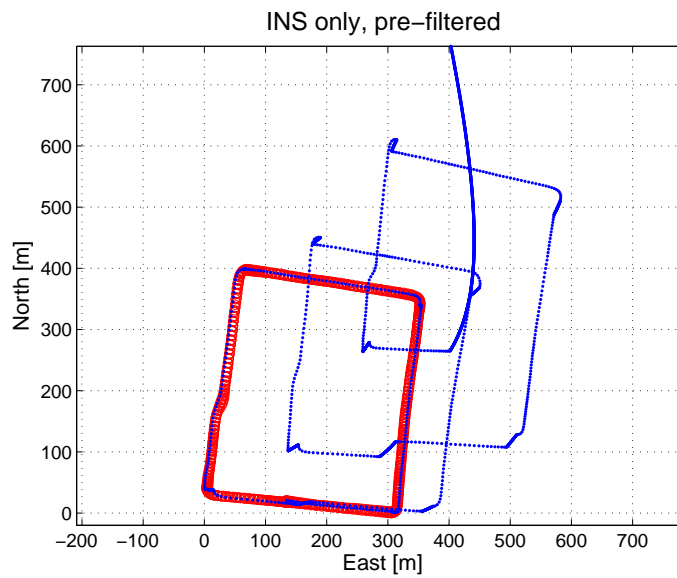


Figure 6.6. Trajectory computed with pre-filtered INS data only (test trajectory). The reference is seen as red circles, the pre-filtered test trajectory as blue dots.

INS with non holonomic constraints

During the data acquisition we were only driving forward, not jumping, nor drifting with the car. Therefore we could process the IMU data with non holonomic constraints. The RMS error was 4.52 meters. (Figure 6.7)

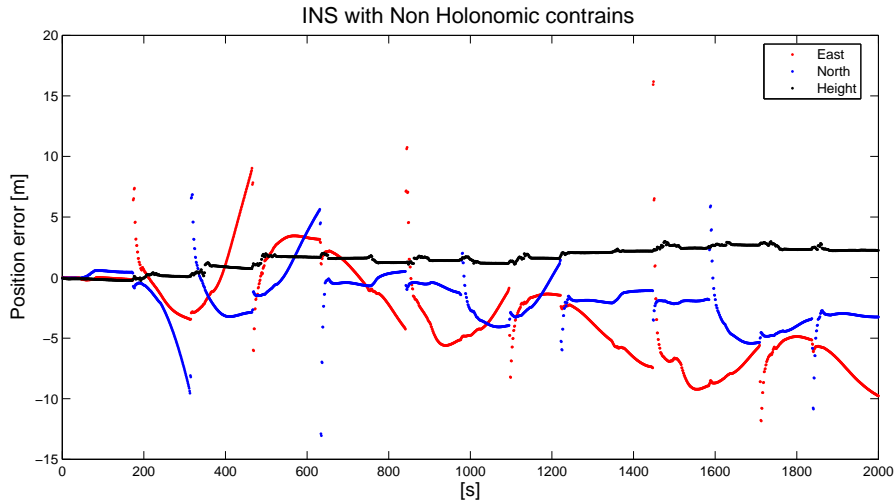


Figure 6.7. Error plot of the test trajectory, timing from right after the end of the alignment phase. The test trajectory was generated by processing IMU data under non holonomic constraints.

INS with ZUPT

During the data acquisition one minute stops were planned every 40-60 second. These stops were used to make ZUPTs during the INS processing. The RMS error was 0.47 meters. (Figure 6.8)

INS with odometer

The IMU data have here been used together with odometer derived velocity data. The AINS toolbox is original set to do a ZUPT when the odometer velocity drops below the variance that is set for the odometer. This function was suppressed in the code, velocities below the variance was instead ignored, and no ZUPT was used. The RMS error is 4.78 meters. (Figure 6.9)

INS with DGPS

When the INS data is fused with GPS data (relative pseudorange GPS) the RMS error is 0.22 meters (Figure 6.11). This should be compared with the RMS error of the pure GPS derived trajectory which was 0.39 meters (Figure 6.10).

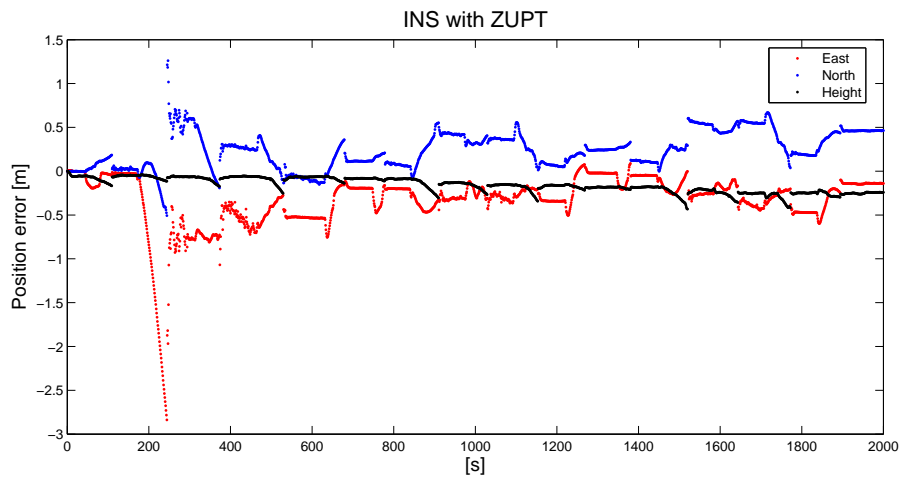


Figure 6.8. Error plot of the test trajectory, timing from right after the end of the alignment phase. The test trajectory was generated by processing IMU data with zero velocity updates.

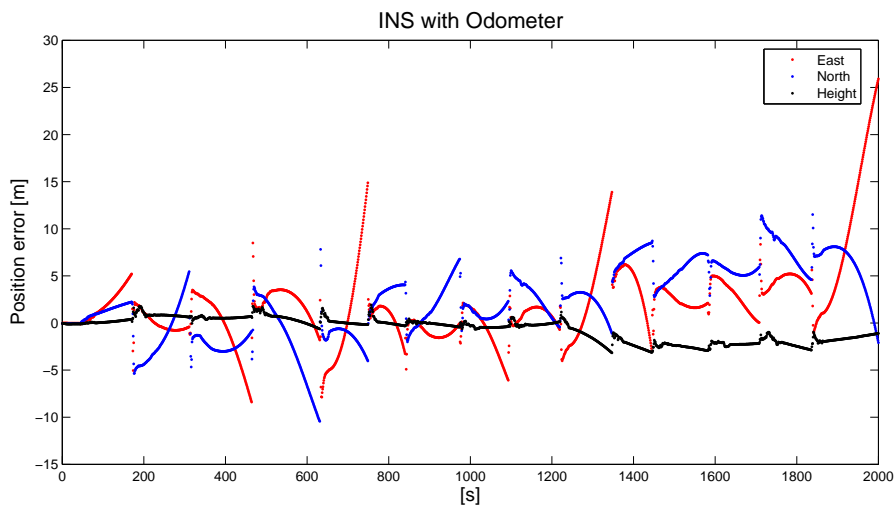


Figure 6.9. Error plot of the test trajectory, timing from right after the end of the alignment phase. The test trajectory was generated by fusing IMU and odometer data together.

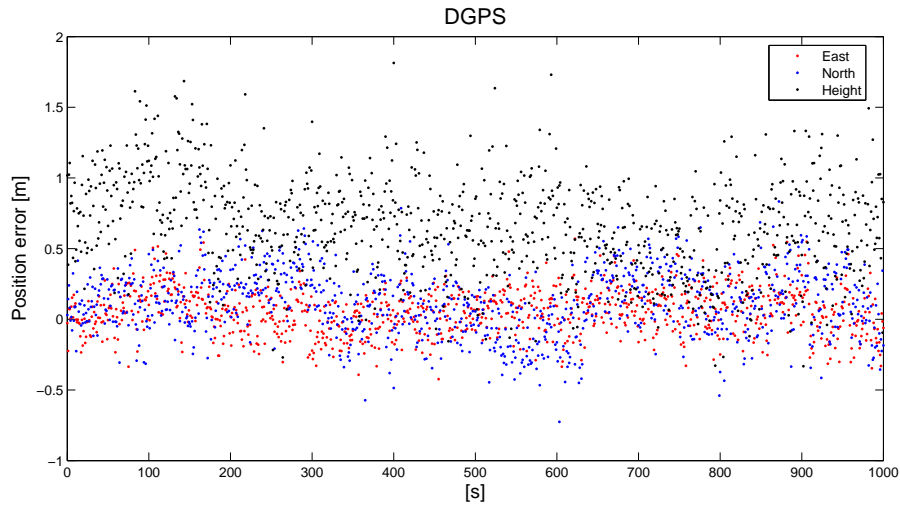


Figure 6.10. Error plot of the test trajectory, timing from right after the end of the alignment phase. The test trajectory was generated by relative pseudo range positioning using GPS.

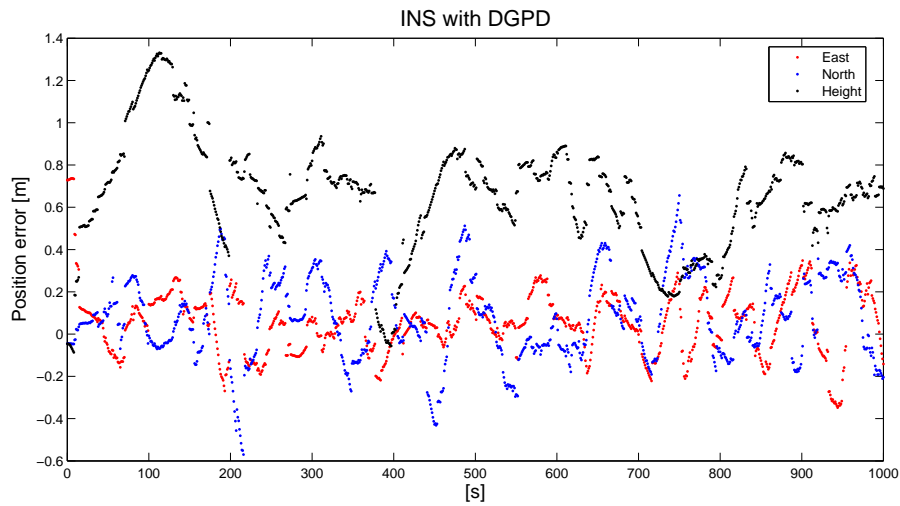


Figure 6.11. Error plot of the test trajectory, timing from right after the end of the alignment phase. The test trajectory was generated by fusing relative pseudo range positions from GPS with IMU data.

INS / DGPS with GPS data outages

The INS/GPS solution have here been processed with a 120 seconds simulated outage in the GPS data. Two results are presented, for which different configurations of the Kalman filter has been used. Figure 6.12 shows the error plot when a higher weight has been used to the GPS solution, and Figure 6.13 shows the error plot after that weight of the GPS solution have been lowered, in favor for the IMU data.

When the GPS raw data is processed in the TTC, the variances of the components in each coordinate triplet are computed. For the higher weighted GPS solution this variances are used within the AINS computation, for the lower weighted GPS solution this variances have been doubled.

(Figure 6.12)(Figure 6.13)

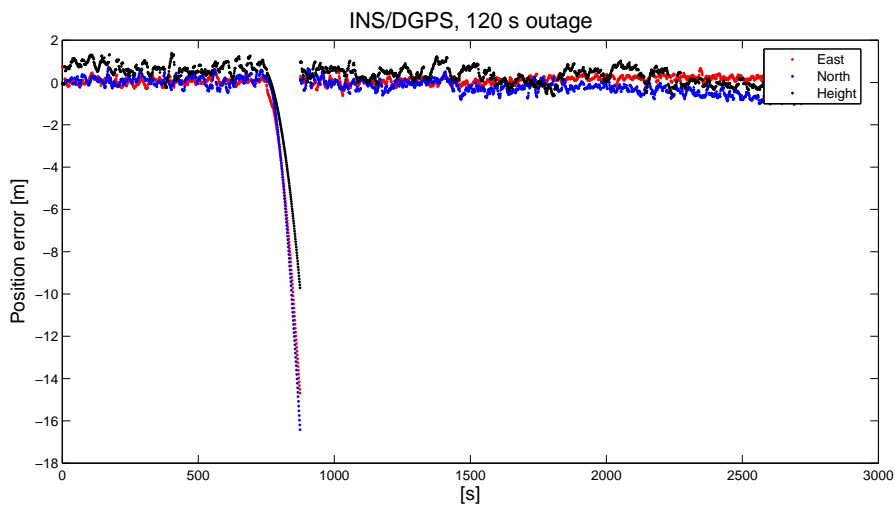


Figure 6.12. Error plot of the test trajectory, timing from right after the end of the alignment phase. The test trajectory was generated by fusing relative pseudo range positions from GPS with IMU data. In the GPS data, an simulated data outages of 120 seconds was used. In the Kalman filter, the IMU data was given a relative low weight.

6.8 Discussion of the results

The results supports in many ways the expectations one could have. The INS only solution has the characteristic unbounded accumulated errors. By pre filtering the accelerometer data one could see an improvement, the first ten minutes the radial error was not exceeding 6 meters.

In the IMU with ZUPT solution the velocity errors are bounded at the time for the ZUPT, but it is still possible to see how the position errors accumulate (clearly visible for the height component).

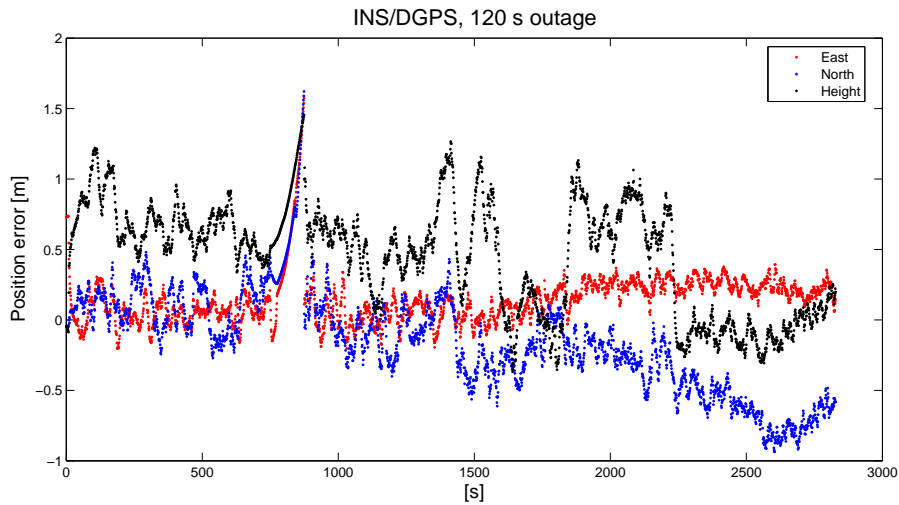


Figure 6.13. Error plot of the test trajectory, timing from right after the end of the alignment phase. The test trajectory was generated by fusing relative pseudo range positions from GPS with IMU data. In the GPS data, a simulated data outages of 120 seconds was used. In the Kalman filter, the IMU data was given a high weight.

The use of non holonomic constraints works very well together with the IMU data. Some fine tuning of the rotation matrix between the body and vehicle frame had to be done, since they were not fully aligned. An other way could be to include a state for this angles in the Kalman filter, and in that way calibrate the non holonomic constraint during good GPS conditions.

As one could expect the error of the GPS only solution from TTC have the characteristics of white noise (this was also seen in an autocorrelation of the data). This noise is reduced in the INS/GPS solution. For this sensor combination, INS/GPS, it is also of interest to see the effect of GPS data outages.

In the first configuration of the outage test (Figure 6.12), the error was growing quite rapid, much faster than after a ZUPT. This shows the importance of configuring, tuning, the Kalman filter correctly. When the weight was too high for the GPS this resulted in a bad estimation of the sensor errors. In the other figure (6.13), one could instead see how the IMU data fills the two minutes gap, just by lowering the weight of the GPS solution.

When the INS were integrated with the odometer derived velocities, the errors are notably smaller compared to the INS only solution. Still a small error accumulation is to be seen, and comparing these results with the ones from the non holonomic constraints or INS with ZUPT, the question was raised if the odometer is really needed.

Also when I processed a INS/GPS/odometer solution the result was worse com-

pared to the result without the odometer. I have not found any good configuration for this problem even if the variance of the odometer is very high, the result is bad. This is probably explained by the nature of the odometer error, it is not white noise but biased error. As seen in the performance test (4.4) the scale factor of the odometer is not too easy to estimate since it seems to vary with the measured surface. By implementing an on-line calibration of the odometer, to use during good GPS reference, one could probably reduce this error. There is also the problem with misalignment between the sensor axes, also this would be possible to calibrate on-line.

Concerning the pre-filtered data this has only been presented as an INS only solution, where it also gave good results. For more precise trajectories, where the IMU data have been processed together with GPS data, the pre-filtered IMU data have not been shown to improve the results, contrary the results have been worse. The reason could be that the high frequency really is important in the filter, due to the linearization. Or that the low pass filter has not worked too well, (time shift e.t.c.?).

Chapter 7

City navigation mission

In this mission the car mounted system was used in the city centre of Munich. The city of Munich does not house any sky scrapers but the buildings rise six, eight stories high and the streets are sometimes narrow. This does not make the best environment for receiving GPS signals.

7.1 Aim of mission

The aim of this mission was to see how the sensors could complement each other in an urban environment, also by making use of the results from the previous mission.

7.2 System setup

The Experiments was carried out with the same system as described in Chapter 6, with the difference that we did not use our own GPS base station. Thanks to the collaboration with the Bavarian Committee for International Geodesy BEK (Bayerische Kommission für die Internationale Erdmessung) we could use data from the German Satellite Positioning Service (SAPOS).

SAPOS is a net of permanent reference stations all over Germany, and we used data from a station less than 2 km away.

7.3 Data flow

The data flow was identical with the one described in Chapter 6.

7.4 Data set

The mission started at the Theresienwiese with a 15 minutes static registration. We than drove out from the field, into the city. The drive in the city lasted for 20 minutes, until we arrived at the technical university of Munich. During this time we

had some natural stops due to stop lights. The mission then ended with 15 minutes registration, which could be used in smoothing during post-processing.

7.5 Data preparation

IMU data

The IMU data was prepared as in Chapter 6.

GPS data

The GPS measurements from the mission was very poor, most of the time the number of visible satellites was below four, and there was a constant change of satellite constellations. The Trimble Total Control was used to process the data, but due to the poor number of simultaneously visible satellites, a solution could only be generated for some positions using a relative pseudorange method. The relative phase measure method generated a solution for two positions.

Both the GPS receiver and TTC is working to reduce the influence of multi path effects. In the resulting GPS positions there was still a few outliers to be seen, which I deduce to the multi path errors. For the experiment this was good, since it would be interesting to see how these errors would be handled in the EKF.

Odometer data

The Velocity data derived from the odometer was multiplied with the scale factor found in previous mission.

7.6 Results of Munich city centre experiment

No good reference is to be used this mission since no continuous GPS trajectory was generated. The best measure of a good trajectory was instead the closing error at those GPS positions we had, and by comparison to map data.

The trajectories from some of the solutions are plotted together in Figure 7.1. Figure 7.2 & 7.3 are detailed plots.

7.7 Discussion of the results

The results have shown that navigation is possible with the system, even when long GPS outages occur.

The synergy between the INS and ZUPT respective INS and GPS was easily seen. There are only fragments of a trajectory when one use GPS only, and the INS only trajectory is drifting away very early. The data fusion of INS & GPS resulted in a very good trajectory, and by doing ZUPTs one effectively bound velocity errors when the

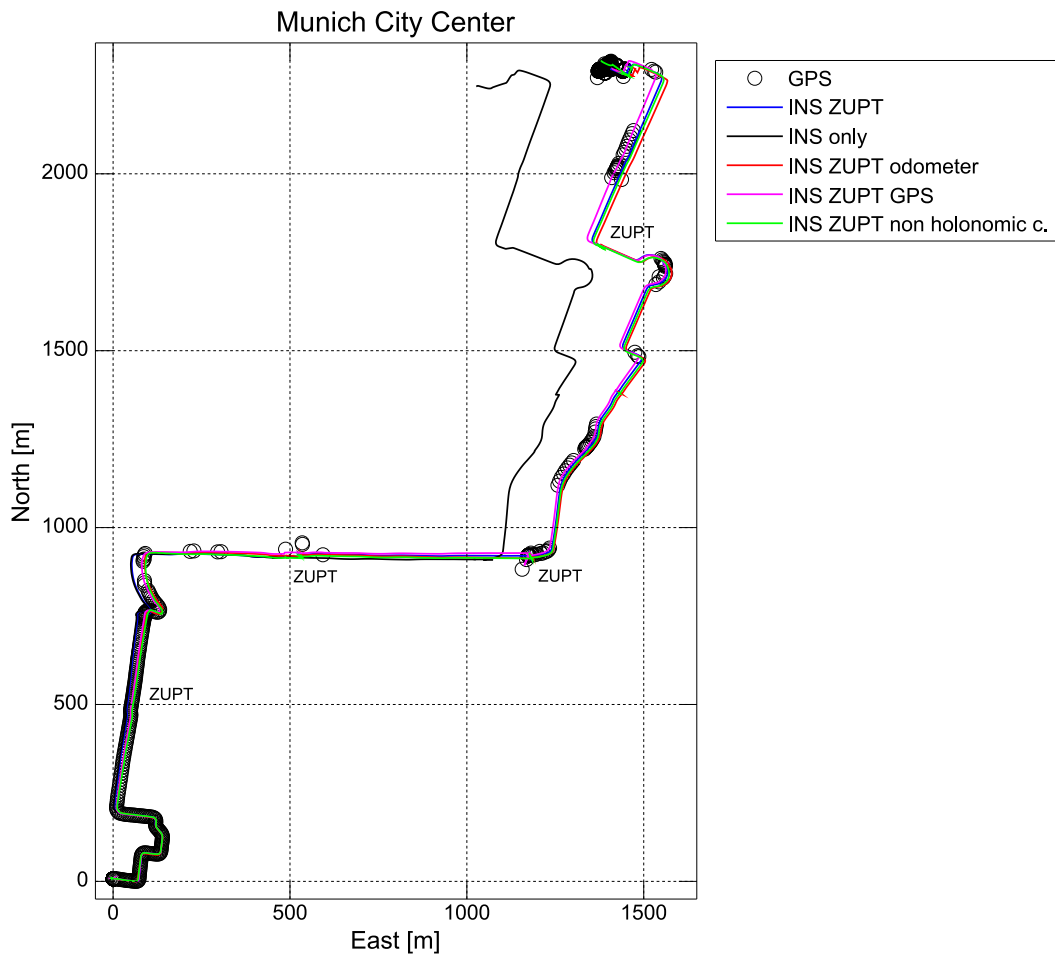


Figure 7.1. Trajectories from the Theresienwiese to the TUM. The open field of the Theresienwiese is stretching 800 m north and 500 m east.

car stops due to stop lights. The error is most probably never exceeding five meters.

The INS/ZUPT solution was very good as well. Large errors occur in the beginning of the trajectory before the first ZUPT, most probably due to error in start alignment. After the first ZUPT much of this error was corrected, but in the end the error still exceeded ten meters.

By using the INS/ZUPT solution with non holonomic constraints one reduces the error in the beginning, but there are other errors accumulating, the error at the end is few meters larger than without the non holonomic constraints.

The INS/Odometer solution was slightly worse than the INS/ZUPT solution with non holonomic constraints. The position errors of these both drifted to the east, this could be a result of a misalignment of the vehicle-IMU reference frames.

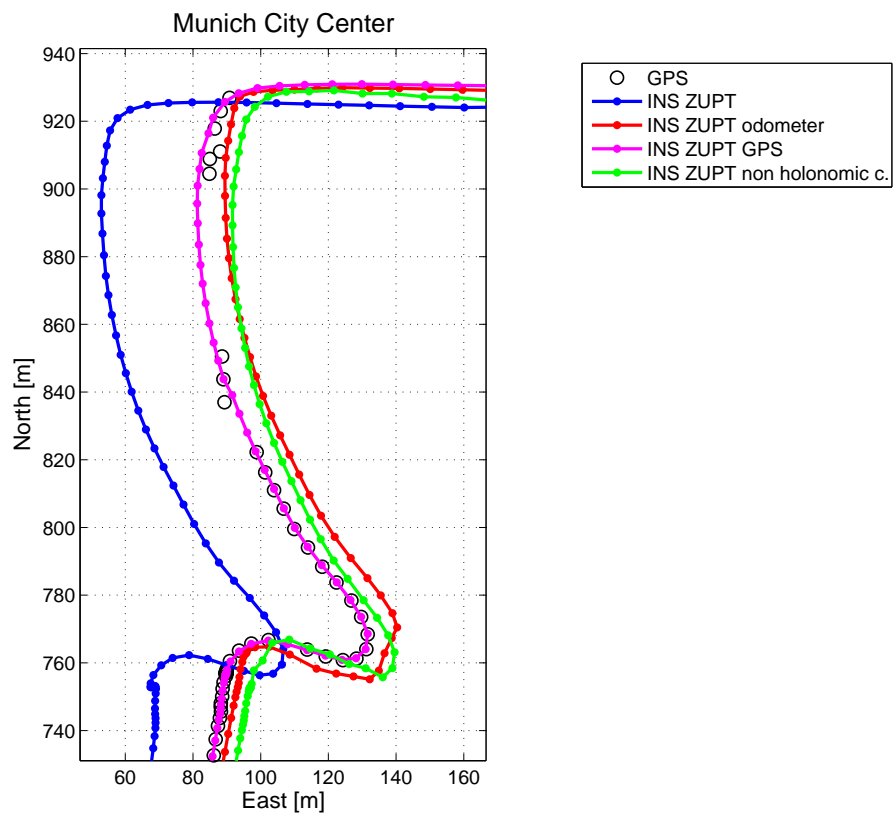


Figure 7.2. Zoom figure of figure 7.1, just when the first GPS outage occurs.

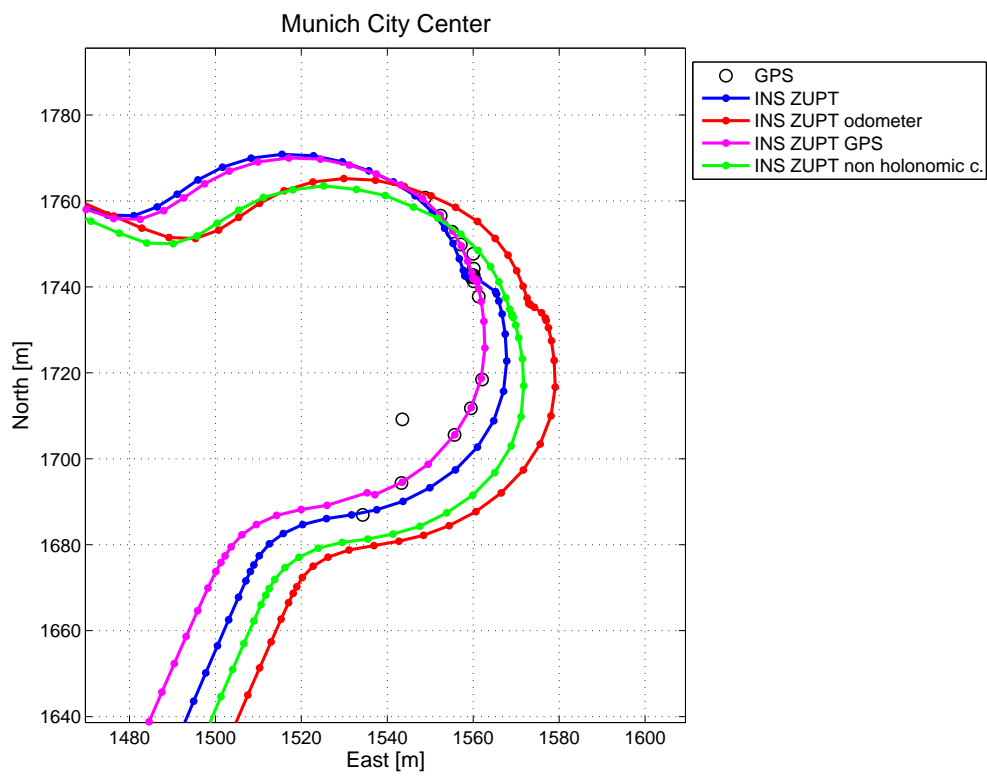


Figure 7.3. Zoom figure of the round about in the upper corner of figure 7.1.

Chapter 8

High way navigation mission

This mission took place on one of the high ways leading from out east from Munich.(Figure 8.1.) Most part of the highway is surrounded by open fields. There are some tunnels and bridges crossing the highway, but we had reason to believe that the conditions for receiving GPS signals was good.

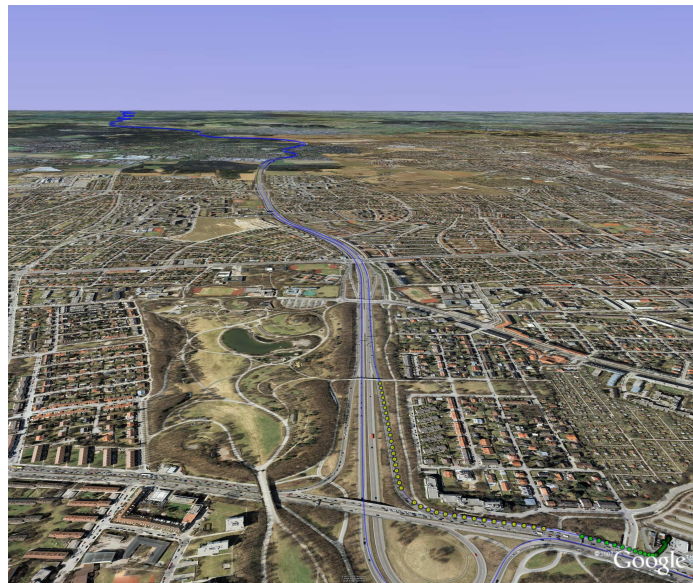


Figure 8.1. The highway leading from out east from Munich. The trajectory could be seen in blue.

8.1 Aim of mission

The previous two missions showed that processing IMU data with ZUPTs or non holonomic constraints gave better results compared to if the INS is aided with an odometer. I.e. we have not been able to show that the odometer as extra sensor is really necessary.

In this mission we wanted to see if the contribution of odometer derived velocity data would be more important if there were no regular stops in the mission.

8.2 System setup

The Experiments was carried out with the same system as described in Chapter 6. Thanks to the collaboration with the *Bavarian Committee for International Geodesy* we could once again use data from the German Satellite Positioning Service (SAPOS).

8.3 Data flow

The data flow was identical with the one described in Chapter 6.

8.4 Data set

The mission started with a 15 minutes static registration at a gas station just by the highway. One important remark here is that we were reversing the car out from the station. One has to remember this if one want to use the odometer data or to constrain the INS with non holonomic constraints.(The TTL signal from the odometer does not differ on forward and backward movement.)

After the static registration we drove on the highway leading out from the city for 40 minutes until we change the direction, heading back to Munich. The mission ended with 15 minutes static registration, which could be used for smoothing of the navigation solution in post-processing.

8.5 Data preparation

IMU data

The IMU data was prepared as in Chapter 6.

GPS data

The GPS data was processed with Trimble total control, where both relative phase measurements and pseudorange measurements were used. Although the theoretical conditions for the day where good, with a large number satellites and a low DOP

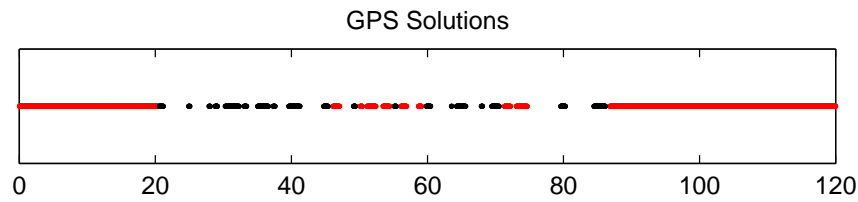


Figure 8.2. GPS solutions during the mission, with the time axis in minutes. Red denote phase measurements solution and black pseudorange measurements solution. The long periods of phase solutions are the static initialization.

values, the measurements did show something else. The recorded data was sparse and the generated trajectory showed lots of gaps.

By importing the GPS trajectory into Google Earth, one can see that the outages occur during and after tunnels, one could also see that outages occur after every passed bridge or larger obstructers. This was something that I was expecting. What I had not expected was that it would take minutes before the receiver locks on the satellites again. This problem could be more examined, but the general explanation for the problem is the low number of satellites available in the line of sight.

Odometer data

The velocity derived from the odometer was multiplied with the scale factor found in the experiment at the Theresienwiese (See Chapter 6) , and than resampled down to 1 Hz. Estimation of the scalefactor using this data set gave a similar result as previous (see Section 6.5).

8.6 Results of High way experiment

Despite the large number of outages in GPS data, the data set is good and there is lots of possibilities to process the data further. Here I present the limited tests that I have done.

INS only

After half the drive, half an hour of driving, the INS only has an radial error in the plane of 1200 meters.

INS with non holonomic constraints

With non holonomic constraints the error after half an hour was 900 meters.

INS with odometer

When the INS data is fused with odometer derived velocities, the error after half an hour was 30 meter.

INS/GPS

It is not to easy to generally describe the error during a outage, since the filter is affected by how long it have been aided with the GPS solution before the outages. I used a semi simulated outage of 120 seconds, for which there was no GPS data available for the first 90 seconds and the last 30 seconds was a simulated outage. This after a 55 seconds period of GPS aiding.

After the 120 second outage the error was 2 meters, the largest error component is in the longitudinal axis of the vehicles .

INS/GPS non holonomic constraints

The same outage as above was used, it resulted in a error of 5 meters and the largest component was a drift orthogonal to the vehicles longitudinal axis.

INS/GPS odometer

The same outage as above was used, it resulted in a error of 5 meters and the largest component was a drift orthogonal to the vehicles longitudinal axis.

8.7 Discussion of the results

The fusion of INS and GPS data showed good result, also the errors during the GPS data outages was generally lower compared to the test at the Theresienwiese (Chapter 6). This could be deduced to better performance of the IMU in higher velocities, better performance of the IMU in smoother dynamics, or better performance of the EKF during smoother dynamics. To find out which, more work have to be done concerning the error behavior.

In this mission the fusing of INS and odometer data gave good result, better than if non holonomic constraints was used.

In the last two results, the constraints and extra data from the odometer, have not improved the solution. Further processing would have to be made.

Chapter 9

Conclusions

The results have shown that the hardware used in the system could be integrated using the AINS Toolbox.

The fusion of INS and GPS data gave good results also in an urban environment with long GPS outages. The adaptive Kalman filter in the AINS Toolbox did also answered well on GPS outliers (i.e. there was a low impact on the result).

The integration of the odometer, i.e the fusion of INS data with odometer derived velocities, did produce the desired results under certain conditions. There was a problem when there was repeatedly full stops during the mission, the source for this error have not fully been investigated. (In those situations ZUPTs could be used as a powerful aiding technique). When the car and the system was on a longer mission with continuous high speed the result was better and the odometer constrained the navigation errors as desired. (Results presented in Section 8.6).

I think that the results with odometer could be improved by using an on-line calibration for the odometer scale factor as discussed in Section 5.1 (also 4.4, 6.5, 6.8 and 8.7). Also other variables such as update rate and variance of the measures have to be further examined.

The effective improvement of the navigation solution, using odometric data update, was shown best for longer missions, without undesired stops and in the presence of a poor GPS signal (i.e. in tunnels, on mountainous roads or roads lined with trees).

Bibliography

- [1] Best-Fit-Computing 2006. *COLUMBUS Version 3.6 [Software]*. USA.
- [2] Britting Keneth R. 1971. *Inertial navigation systems analysis*. Wiley-Interscience, U.S.A.
- [3] Corrsys-Datron Sensorsysteme Gmbh *Physikaisches Arbeitsprinzip der optischen CORREVIT sensoren*. Germany.
- [4] Dorobant Raul, Gerlach Christian. 2004. *Investigation of a navigation-grade RLG SIMU type iNAV-RQH*. Schriftenreihe IAPG/FESG, Technische Universität München
- [5] Hofman-Wellenhof B., Legat K., Wieser M. 2004 *Applied Mathematics In Integrated Navigation Systems* . American Institute of Aeronautics and Astronautics, USA.
- [6] Jekeli Christofer 2001. *Inertial Navigation Systems with Geodetic Applications*. Walter de Gryter, Germany.
- [7] Peters, Stoddard, Meridith 1998. *Development of a 125g Quartz Flexure Accelerometer for the RIMU Program* In: Position Location and Navigation Symposium, IEEE 1998, Palm Springs, CA, USA, pp 17-24.
- [8] Rogers Robert 200. *Applied Mathematics In Integrated Navigation Systems*. American Institute of Aeronautics and Astronautics, USA.
- [9] Schumacher Adrian 2006. *Integration of a GPS aided Strapdown Inertial Navigation System for Land Vehicles*. Royal Intitute of Technology, Stockholm Sweden
- [10] Schwarz K.P. 2001. *Lecture Notes ENGO 623,INS/GPS Integration for Geodetic Applications*. Department of Geomatics Engineering, The University of Calgary.
- [11] Schwarz , El-Sheimy 2000. *KINGSPADTM Users Manual v3.0*. Departm ent of Geomatics Engineering, The University of Calgary, Canada.
- [12] Schwarz K.P., El-Sheimy Naser 2000. *KINGSPAD Version 3.01 [software]*. The University of Calgary, Canada. ,
- [13] Shin Eun-Whan 2001. *Accurate improvement of Low Cost INS/GPS for Land Applications. UCGE Reports number 20156*. The University of Calgary, Canada.

- [14] Shin Eun-Whan 2001. *Accurate improvement of Low Cost INS/GPS for Land Applications. UCGE Reports number 20156.* The University of Calgary, Canada.
- [15] Shin Eun-Whan 2005. *AINS Toolbox for matlab [Matlab Code].* The University of Calgary, Canada.
- [16] Shin Eun-Whan 2005 *Aided Inertial Navigation System (AINS) Toolbox for MatLab Software User Manual.* The University of Calgary, Canada.
- [17] Siouris George M 1993. *Aerospace Avionics Systems A Modern Synthesis.* Academic Press, UK.
- [18] Skaloud J. 1999. *Optimizing Georeferencing of Airborne Survey Systems by INS/DGPS.* Department of Geomatics Engineering, The University of Calgary, Canada
- [19] Welch G, Bishop G 2001. *An introduction to the Kalman Filter.* University of North Carolina at Chapel Hill Department of Computer Science, USA
- [20] Wolfgang Torge 2001. *Geodesy 3rd Edition.* Walter de Gryter, Germany.

Appendix A

Logged data

For the missions at the Theresienwiese and the Munich city centre was the following data logged:

```
# Started: 01/16/2007 12:15:23 (local) 01/16/2007 11:15:23 (UTC)
# Column 1: Marker,
# Column 2: TINTRPL, TINTRPL, [ s]
# Column 3: IACC, IAcc_X, [ m/s2]
# Column 4: IACC, IAcc_Y, [ m/s2]
# Column 5: IACC, IAcc_Z, [ m/s2]
# Column 6: IOMG, IOmg_X, [ °/s]
# Column 7: IOMG, IOmg_Y, [ °/s]
# Column 8: IOMG, IOmg_Z, [ °/s]
# Column 9: ODOVEL, OdoVel_0, [ m/s]
# Column 10: ODOT, OdoT, [ s]
# Column 11: RODO, RawOdo0, [ m]
# Column 12: RVEL, RVel_0, [ m/s]
# Column 13: TIME, IMS_time, [ s]
# Column 14: ITACCF, ITAccF_0, [ μA]
# Column 15: ITACCF, ITAccF_1, [ μA]
# Column 16: ITACCF, ITAccF_2, [ μA]
# Column 17: RAWVX, RawVX, [ m/s]
# Column 18: NRPY, roll, [ °]
# Column 19: NRPY, pitch, [ °]
# Column 20: NRPY, yaw, [ °]
```

MARKER Event mark, triggered by the operator

TINTRPL Time reference constructed by the NMEA message, PPS signal and the internal time (TIME)

IACC Inertial accelerations, Incl. gravity

IOMG Inertial rotation rates, Incl. earth rotation

ODOVEL Odometer derived velocity

ODOT Odometer data

RODO Odometer data

TIME Internal time of the IMU, refer to start up.

ITACCF Current proportional with the accelerometers temperatures.

RAWVX Raw value of longitudinal velocity

NRPY The real time computed attitude by the IMU. (Using internal algorithms and CPU. This data was not used in my work)

Appendix B

Cookbook

This appendix is written to give some system specific details about the data processing. Special intended for future work with iNAV-RQH and the AINS Toolbox at IAPG.

IMU configuration file

The IMU configuration file is used to set how the IMU will handle its sensor data. For my work, I have used an earlier template wich I renamed. Thereafter it is possible to change the sampling rate and to configure and name the output file.

IMU output to AINS input

If the iMAR Navcomand have been configured as above, an ASCII output file similar to the one in Appendix A will be generated, from here on called the IMU file. To be able to use the IMU file together with GPS data and the AINS Toolbox, one have to apply the following steps.

- Change the time stamps to seconds of the week, GPS time.
- Rotate the accelerometer and gyroscope data.
- Write the above data into a binary file.
- Produce an ASCII odometer file.

How the time is changed in `txt2imu`

The matlab file `txt2imu` could be used to do the first three steps above. The file could be rewritten into a function file with some simple modifications.

To be able to read the IMU file into `txt2imu`, one first has to remove the header, and save the new file as `name.txt` (In order to follow the standard at IAPG). Then the name of the new file should be typed into the *input* line of `txt2imu`. And an output name will have to be typed into the *output* line, `outputname`

The TINTRPL time in the second column of the IMU file is given in second of the day, coordinated universal time (UTC). To have same time-reference as for the GPS measurement, one converts this time to GPS time in seconds of the week (SoW). This is done in the `txt2imu` file, and it is possible to change the leap seconds (at this time set to 14s), and to write which day of the week that the mission took place. Day 0 is Sunday and if the mission took place on a Tuesday, the seconds from two days will be added ($2*86400$ sec), plus the leap seconds.

How the rotations is done in `txt2imu`

The sensor data in the IMU file is measured along the axis of the IMU-reference frame, for the iNAV RQH this is a right handed system where the z -axis is pointing upward. For this design a East, North, Up local level reference system (ENU) is best used to express the attitude of the unit. This since, when the IMU is vertically aligned and north oriented, the axis of the IMU and the local level system will coincide. (X_{ENU}^{IMU} oriented towards East, Y_{ENU}^{IMU} oriented toward North and Z_{ENU}^{IMU} towards the up direction)

The AINS Toolbox use a North, East, Down local level system (NED). Therefore I have rotated the data from the IMU so that the rotated IMU system can coincide with the NED system.

(The positions of the IMU refer to the e -frame and will not be effected by this, only how the attitude and velocity refer to the local level system is effected)

The transformation between the two systems could be done by finding the rotation matrix R in the formulas

$$AccelerometerData_{NED} = R * AccelerometerData_{ENU}$$

and

$$GyroscopeData_{NED} = R * GyroscopeData_{ENU}$$

In `txt2imu` R is given by a direction cosine matrix (DCM), which easiest could be found through the Euler angels of rotation between the two systems. In Figure B.1 are the two system presented and one could see how two rotations differ the two systems from each other.

	Euler Angels
Following are the Euler angels used in B.1.	$roll = \pi$
	$pitch = 0$
	$yaw = -\pi/2$

Where the roll is around X-axis, pitch around the Y-axis and yaw is around the Z-axis. The direct cosine matrix could thereafter be constructed by

$$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(roll) & \sin(roll) \\ 0 & -\sin(roll) & \cos(roll) \end{bmatrix}$$

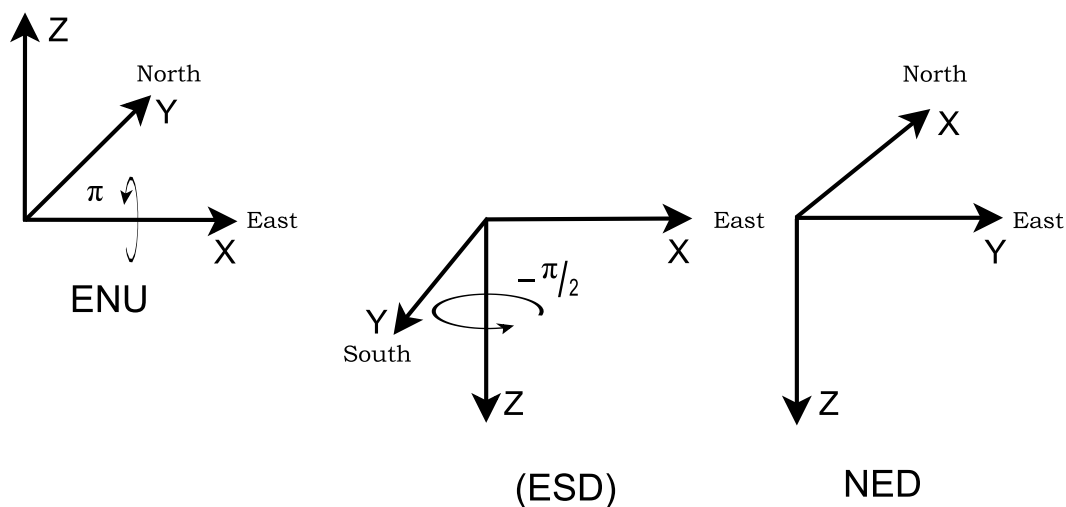


Figure B.1. To the left the local level ENU system, orientated in the earth system. Two rotations rotate the system into a NED system, shown to the right. The translation in west-east direction over the intermediate system (ESD) added only for a clearer view.

$$R_{pitch} = \begin{bmatrix} \cos(pitch) & 0 & -\sin(pitch) \\ 0 & 1 & 0 \\ \sin(pitch) & 0 & \cos(pitch) \end{bmatrix}$$

$$R_{yaw} = \begin{bmatrix} \cos(yaw) & \sin(yaw) & 0 \\ -\sin(yaw) & \cos(yaw) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$DCM = (R_{yaw} * R_{pitch}) * R_{roll}$$

which, if the values used above are used, results in

$$DCM = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

How the new file is written in txt2imu

After that the time have been changed and the rotation have been made, one also have to change the gyroscope data from degrees/second to radians/second, by multiply with $\pi/180$. The time, accelerometer and gyroscope data is then written into a binary file of the format `outputname.imu`.

How to create an ASCII odometer file with the makeOdoV file

The AINS Toolbox does not read the raw odometer data, the input should be a velocity file in ASCII format, and time tagged in the same system as the IMU and GPS data.

The velocity to use could be the odometer velocity data in the IMU output file, the ODOVEL column (See Appendix A). Or one could combine the raw odometer data (RODO) with the special odometer time column in the IMU file, the ODOT column, to derive the velocity. The later method has showed better result.

The matlab file `makeOdoV` read the `name.txt` file, and derive the velocity from RODO and ODOT. Since the velocity data is rather noisy it is filtered and resampled to 1 Hz by the matlab `resample` function. The time is not resampled, instead a new time column is created by taking the time from the first row in `name.txt` and adding one second for each new row (1 Hz).

The filtering of odometer data is something that could be examined more, by testing different filter methods, and to further examine the error between GPS and odometer velocities. (Auto correlation, time shifts?)

GPS receiver output to AINS input

Downloading GPS raw data

If the System 4000sse from Trimble has been used, the raw data could be downloaded through a serial communication, by using the *Trimble Data Transfer* software (free to download).

If the 4000sse receiver is not listed in the Device menu, it is possible to create a new device, by go to `Devices => New => GPS Receiver 4000 series=> ok`. The maximal baud rate have to be set to the same as in the receiver (I used 19200), and the parity should be set to odd.(if there is problem down loading the data, try to decrease the baud rate further, and remember to set the same baud rate in the receiver!)

GPS data processing

I have been using Trimble Total Control (TTC) for the GPS data processing. The software is easy to learn and has good help in the `Help` menu.

If TTC is used, the trajectory could be exported in the *Geodimeter Format* with the option WGS-84 (See TTC handbook or `Help` files for more information on how to export).

From the TTC generated file the time, position and standard deviation data has to be stored into a new ASCII file. For this the matlab file `TTC2AINS` could be used. In `TTC2AINS` I have used the matlab `load` command in the file, therefore the header of the TTC generated file must be removed and the letter columns changed to integers. The integers I used was

0 for ?

1 for FIXED

3 for DGPS

Create an AINS configuration file

The Manual for AINS Toolbox gives a good description of the configuration file. Here is only some complimentary information.

If one use an odometer and the IMU is mounted on the car with the IMU x-axis pointing forward, one must use a *body to vehicle* rotation of -90 degrees around the z-axis. This since the IMU data have been rotated in the `txt2imu` step, so in the body frame it is actually the y-axis that is pointing forward (longitudinal to the car). The vehicle to body rotation is easily done by the `Body to vehicle attitude` boxes in the graphic user interface. It is also possible to adjust this angle if the IMU was not fully aligned with the vehicle.

If one use non holonomic constraints, than the boxes of the `Wheel lever arm` must be set to Zero. Or one have to make a small change in the code for the `ins_ekf_psi_bs` sub routine. Otherwise the lever arm correction for the odometer will be used.

The lever arms (GPS & odometer) should be given in the body frame. I.e. they have to be rotated with R , if they are measure in the IMU-frame.

$$R = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

The values for the IMU `performance` have to be modelled and filled in. I cheated and used the values from the AINS example file, which actually is for a Litton 200.

Leave no empty boxes in the graphic user interface! If they are not needed, that means that the value should be *zero*.

AINS output to ASCII data.

The AINS output data is ether a navigation file of the form `outputname.nav` or a navigation file and a smoothed navigation file, `outputname.rts`. Both are in binary form, and the files could of course be used as they are, but it is also possible to use the `nav2txt` file to write ASCII files for further use. In ether case one must know how the output file is built up, which could be found in the AINS Manual. If a `outputname.rts` file is created one could use the matlab `flipud` function to flip the file into a file with chronological events.

Find more help

- *The AINS Toolbox manual* by the Toolbox author Eun-WhanShin
- *Estimation Techniques for Low-Cost Inertial Navigation* by Eun-WhanShin
- *Accurate improvement of Low Cost INS/GPS for Land Applications* by Eun-WhanShin
- *Investigation of a Navigation-Grade RLG SIMU type iNAV-RQH* by R. Dorobantu, C. Gerlach
- *NavCommand, Software to Operate and Configure iMAR Inertial Measuring and Surveying Systems, operation and user Instruction* from iMAR
- *iNAV-RQH Configuration and Usage* from iMAR

Appendix C

Manufactures specification of hardware

Table C.1. Principal Parameters of the Honeywell Ring Laser Gyroscope type GG1320 (Dorobantu, Gerlach 2004)

Performance	Value	Unit
Bias stability error	≤ 0.002	deg/h
Random walk noise	≤ 0.0018	deg/\sqrt{h}
Scale factor stability / nonlinearity	≤ 10	ppm
Input axes alignment error	$\leq \pm 1$	$mrad$
Maximal angular rate	± 500	deg/s
Mean time between failures	$\leq 5 \cdot 10^4$	h

Table C.2. Principal Parameters of Q-Flex accelerometers QA 2000-40 (Dorobantu, Gerlach 2004)

Performance	Value	Unit
Range	± 25	g
Bias	≤ 4000	μg
Scale factor	1.2...1.46	dB
Damping ratio	0.3...0.8	n/a
Noise power density	8	$\mu g/\sqrt{Hz}$
2 nd Order nonlinearity	15	$\mu g/g^2$
Input axes alignment error	$\leq \pm 1$	$mrad$
Maximal angular rate	± 500	deg/s
Temperature sensitivity	$\leq 3...25$	$\mu g/^\circ C$

Table C.3. Principal Parameters of iNAV-RQH, IMU from Imar (Dorobantu, Gerlach 2004)

Performance	Value	Unit
Position accuracy (unaided)	≤ 1.85	<i>km/h</i>
Sampling rate	up to 1000	<i>Hz</i>
Output rate	1...500	<i>Hz</i>
Latency	$\leq 1 \pm 0.01$	<i>ms</i>
Weight	<i>ca.13</i>	<i>kg</i>
Size	<i>ca. $\times 350 \times 213 \times 180$</i>	<i>mm</i>
Shock	60	<i>g (11ms)</i>

Table C.4. Principal Parameters of Correcit L-CE, odometer from Corsys-Datron (Corsys-Datron)

Performance	Value	Unit
Range	0.5...400	<i>km/h</i>
Resolution	2.2	<i>mm</i>
maximum deviation	$\leq \pm 0.1\%$	<i>n/a</i>

Table C.5. Principal Parameters of System 4000 SSE, GPS receiver from Trimble (Trimble)

Performance	Value	Unit
Band	<i>L1L2</i>	

Reports in Geographic Information Technology 2007-2008

The TRITA-GIT Series - ISSN 1653-5227

07-001 **Carl Schedlich**. *Turn at the roundabout: A practical assessment of spatial representations in two different GPS interfaces from a pedestrian's perspective*. Bachelor of Science thesis in geoinformatics. Supervisor: Michael Le Duc. January 2007.

07-002 **Staffan Bengtsson**. *Förändringsanalys i ortofoton*. Master of Science thesis in geoinformatics. Supervisor: Jonas Nelson and Patric Jansson. TRITA-GIT EX 07-002. March 2007.

07-003 **Joseph Addai**. *Quantification of temporal changes in metal loads - Moss data over 20 years*. Master of Science thesis in geoinformatics. Supervisor: Katrin Grunfeld. March 2007.

07-004 **Stephen Rosewarne**. *Deformation study of the Vasa Ship*. Bachelor of Science thesis in geodesy No.3097. Supervisor: Milan Horemuz. March 2007.

07-005 **Naeim Dastgir**. *Processing SAR Data Using Range Doppler and Chirp Scaling Algorithms*. Master of Science thesis in geodesy No.3096. Supervisor: Lars Sjöberg. April 2007.

07-006 **Torgny Israelsson** and **Youssef Shoumar**. *Motion Detection with GPS*. Master of Science thesis in geodesy No.3098. Supervisor: Milan Horemuz. April 2007.

07-007 **Akjol Djenaliev**. *Multicriteria decision making and GIS for railroad planning in Kyrgyzstan*. Master of Science thesis in geoinformatics. Supervisor: Hans Hauska. May 2007.

07-008 **Anna Hammar**. *Quality comparison of automatic 3D city house modelling methods from laser data*. Master of Science thesis in geodesy No.3099. Supervisor: Milan Horemuz. May 2007.

07-009 **Md Ubydul Haque**. *Mapping malaria vector habitats in the dry season in Bangladesh using Spot imagery*. Master of Science thesis in geoinformatics. Supervisor: Hans Hauska. May 2007.

07-010 **Jing Jiang**. *Analysis of the Suitable and Low-Cost Sites for Industrial Land Using Multi Criteria Evaluation: A Case of Panzhihua, China*. Master of Science thesis in geoinformatics. Supervisor: Yifang Ban. June 2007.

07-011 **Raghavendra Jayamangal**. *Quantification of coastal erosion along Spey Bay and the Spey River using photogrammetry and LiDAR imagery-derived DTMs*. Master of Science thesis in geoinformatics. Supervisor: Yifang Ban and Jim Hansom. June 2007.

07-012 **Alicia E. Porcar Lahoz**. *An analysis of how geographical factors affect real estate prices*. Master of Science thesis in geoinformatics. Supervisor: Yifang Ban. October 2007.

07-013 **Ebenezer Kwakye Bentum**. *Detection of land use change in the Accra Metropolitan Area from 1990 to 2000*. Master of Science thesis in geoinformatics. Supervisor: Hans Hauska. November 2007.

07-014 **Jesper Grandin**. *Aided inertial navigation field tests using an RLG IMU*. Master of Science thesis in geodesy No.3100. Supervisors: Milan Horemuz and Raul Dorobantu. December 2007.

TRITA-GIT EX 07-014
ISSN 1653-5227
ISRN KTH/GIT/EX-07/014-SE