JOONA PUSA

STRAPDOWN INERTIAL NAVIGATION SYSTEM AIDING WITH NONHOLONOMIC CONSTRAINTS USING INDIRECT KALMAN FILTERING

Master of Science Thesis

Examiner: Professor Robert Piché (TUT)
Examiner and topic approved in the council meeting of the Faculty of Science and Environmental Engineering on April 8, 2009
Preface

This Master of Science Thesis was written at the Department of Mathematics of Tampere University of Technology. Thesis is a result of my work as research assistant in the Personal Position Algorithms Research Group as a part of FUGAT (Future GNSS applications and techniques) project and it was written during the fall 2008 and spring 2009.

I want to thank my supervisor Prof. Robert Piché whose was the examiner of this thesis for advice and for the opportunity to work in this group and Pavel Davidson for his comments and for the introduction to the subject. I would also like to thank all my co-workers and especially M.Sc. Henri Pesonen for the valuable comments. Thanks goes also to my family, friends and everyone who has supported me during my studies.

Tampere, 15th July 2009

Joona Pusa
# Contents

Abstract 4

Tiivistelmä 5

Abbreviations and acronyms 6

Symbols 7

1 Introduction 10

2 Coordinate transformations 14
  2.1 Coordinate transformation matrix .............................. 14
  2.2 Coordinate transformation models ............................ 15
    2.2.1 Direction cosine matrix ................................. 15
    2.2.2 Euler Angles ........................................... 16
  2.3 Coordinate frames ............................................ 18

3 Inertial navigation systems 20
  3.1 Inertial navigation process ................................. 20
  3.2 Navigation equations ........................................ 22
    3.2.1 Velocity ............................................... 22
    3.2.2 Attitude ............................................... 26
    3.2.3 Position ............................................... 28

4 INS errors 30
  4.1 INS sensor errors and sources of the errors .................. 30
    4.1.1 Gyroscope errors ....................................... 31
    4.1.2 Accelerometer errors ................................. 33
  4.2 Initial alignment errors .................................... 34
  4.3 Computational errors ....................................... 35

5 Error propagation models 36
  5.1 Psi angle error model ....................................... 36
    5.1.1 Velocity error model ................................. 37
    5.1.2 Psi angle attitude error model with small angle assumption ... 39
6 Constrained Inertial navigation systems

6.1 Nonholonomic constraint

6.1.1 Indirect vs. direct approach

6.1.2 Body frame velocity error

6.1.3 Kalman filter design

6.1.4 Simulations

7 Conclusions and future work
ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Faculty of Science and Environmental Engineering, Department of Mathematics

Pusa, Joona: Starpdown inertial navigation system aiding with nonholonomic constraints using indirect Kalman filtering

Master of Science Thesis, 66 pages
Major: Mathematics
Examiner: Professor Robert Piché
July 2009

An inertial measurement unit (IMU) is not able to provide accurate navigation for a long time duration working unaided, that is without proper external aiding, especially when using low-cost inertial sensors. This could be the case e.g. during GPS outages or weak signal areas, such as urban environments, with integrated INS/GPS device. In case of poor external information, INS can be aided using land vehicle constraints. The objective of this thesis is to create a theoretical background for the use of such nonholonomic constraints in IMU error estimation.

When dealing with land vehicle navigation, we assume that the navigation unit does not slide on the ground or jump off the ground. However an IMU, when corrupted by biased sensor measurements, reports a considerable speed to the directions perpendicular to the forward motion violating the nonholonomic constraints. In this work these velocities are taken as virtual measurements of the error in vehicle’s coordinate frame. With the error propagation model and the model constructed for given measurements the position, velocity and attitude error are estimated. This is carried out with indirect Kalman filter having errors among the estimated variables. The algorithms of the filter and unaided IMU are implemented in Matlab and simulation results are provided to demonstrate the improved accuracy achieved with nonholonomically constrained IMU compared with the unaided IMU position solution.
TAMPEREEN TEKNILLINEN YLIOPISTO
Luonnontieteiden ja ympäristötekniikan tiedekunta, Matematiikan laitos

Pusa, Joona: Inertiapaikantajan avustaminen epäholonomisten rajoitteiden avulla käyttäen epäsuoraa Kalmanin suodinta
Diplomityö, 66 sivua
Pääaine: Matematiikka
Tarkastaja: Professori Robert Piché
Heinäkuu 2009


## Abbreviations and Acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLU-estimator</td>
<td>Best Linear Unbiased Estimator</td>
</tr>
<tr>
<td>DR</td>
<td>Dead Reckoning</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning Systems</td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial Measurement Unit</td>
</tr>
<tr>
<td>INS</td>
<td>Inertial Navigation System</td>
</tr>
<tr>
<td>INS/GPS</td>
<td>integrated INS and GPS device</td>
</tr>
<tr>
<td>MC</td>
<td>Monte Carlo</td>
</tr>
<tr>
<td>MEMS</td>
<td>Micromachined Electromechanical Systems</td>
</tr>
<tr>
<td>NHC</td>
<td>INS using nonholonomic constraints aiding</td>
</tr>
<tr>
<td>unaided</td>
<td>INS working without external aiding</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$\approx$</td>
<td>approximate equality</td>
</tr>
<tr>
<td>$f$</td>
<td>integral over a particular space</td>
</tr>
<tr>
<td>$\alpha_{xu}$</td>
<td>angle between x- and u-axis</td>
</tr>
<tr>
<td>$a_x$</td>
<td>acceleration to the direction of coordinate axis x</td>
</tr>
<tr>
<td>$B_x$</td>
<td>gyroscope x-axis bias</td>
</tr>
<tr>
<td>$C_{uvw}$</td>
<td>coordinate transformation matrix from xyz- to uvw- coordinate frame</td>
</tr>
<tr>
<td>$C_\psi$</td>
<td>rotation $\psi$ about defined axis</td>
</tr>
<tr>
<td>$C_b$</td>
<td>coordinate transformation matrix from body to navigation frame</td>
</tr>
<tr>
<td>$\times$</td>
<td>cross product</td>
</tr>
<tr>
<td>$d$</td>
<td>distance travelled</td>
</tr>
<tr>
<td>$\delta t$</td>
<td>time increment</td>
</tr>
<tr>
<td>$\delta f$</td>
<td>accelerometer bias</td>
</tr>
<tr>
<td>$\nabla p$</td>
<td>bias vector in platform frame</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>gyroscope bias</td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>gyroscope bias skew symmetric matrix</td>
</tr>
<tr>
<td>$\Delta f$</td>
<td>deviation of f</td>
</tr>
<tr>
<td>$F_c$</td>
<td>Coriolis force</td>
</tr>
<tr>
<td>$f$</td>
<td>specific force acceleration</td>
</tr>
<tr>
<td>$F$</td>
<td>motion model transition matrix</td>
</tr>
<tr>
<td>$g$</td>
<td>gravitation</td>
</tr>
<tr>
<td>$g_l$</td>
<td>local gravity vector</td>
</tr>
<tr>
<td>$G$</td>
<td>noise gain matrix</td>
</tr>
<tr>
<td>$h$</td>
<td>height above the Earth surface</td>
</tr>
<tr>
<td>$H_k$</td>
<td>measurement model transition matrix</td>
</tr>
<tr>
<td>$\xi$</td>
<td>meridian deflection of the local gravity vector</td>
</tr>
<tr>
<td>$\eta$</td>
<td>deflection of the local gravity vector perpendicular to the meridian</td>
</tr>
<tr>
<td>$K_k$</td>
<td>Kalman gain matrix</td>
</tr>
<tr>
<td>$\lim$</td>
<td>limit</td>
</tr>
<tr>
<td>$\psi$</td>
<td>yaw angle</td>
</tr>
<tr>
<td>$\theta$</td>
<td>pitch angle</td>
</tr>
</tbody>
</table>
\( \phi \) roll angle
\( \lambda \) local latitude
\( l \) local longitude
\( M_y, M_z \) cross coupling coefficients
\( n_x \) noise term
\( N_i \) zero mean random variable
\( \omega_x \) angular velocity through the coordinate axis x
\( \tilde{\omega}_x \) x-axis angular rate measurement exerted on the gyroscope
\( \omega_{ie} \) angular velocity between the inertial and the Earth frame in inertial frame
\( \Omega \) Earth rotation rate
\( \Omega_{ie} \) skew symmetric matrix representing the cross product \( \omega_{ie} \times \)
\( p_{xyz} \) position vector in xyz- coordinate frame
\( \tilde{P}_k \) prior covariance
\( \tilde{P}_k \) posterior covariance
\( Q_k \) process noise covariance
\( R_k \) measurement covariance
\( R \) Earth radius
\( r \) position vector
\( S_x \) scale factor
\( \sigma \) standard deviation
\( \sigma^2 \) variance
\( t \) time
\( t_k \) sampling time k
\( T_S \) period of Schuler oscillation
\( T \) small angle transformation matrix
\( u \) unit vector representing the direction "up"
\( u_k \) state noise vector
\( v \) velocity vector
\( v_e \) ground velocity defined as \( \frac{dr}{dt} \)
\( v_n \) velocity north component
\( v_e \) velocity east component
\( v_d \) velocity down component
\( v_k \) measurement noise
\( \text{Var}(f) \) variance of f
\( X_n \) navigation frame x-axis
\( \bar{x}_k \) mean value
\( x \) state vector
\( \bar{x}_k \) prior estimate
\( \hat{x}_k \) \hspace{1em} \text{posterior estimate} \\
\( Y_n \) \hspace{1em} \text{navigation frame y-axis} \\
\( z_k \) \hspace{1em} \text{measurement vector} \\
\( Z_n \) \hspace{1em} \text{navigation frame z-axis} \\

\textbf{Subscripts and superscripts} \\

\( b \) \hspace{1em} \text{body frame} \\
\( e \) \hspace{1em} \text{Earth frame} \\
\( i \) \hspace{1em} \text{inertial frame} \\
\( n \) \hspace{1em} \text{navigation frame} \\
\( p \) \hspace{1em} \text{platform frame} \\
\( t \) \hspace{1em} \text{true frame} \\
\( x \) \hspace{1em} \text{direction of coordinate axis x} \\
\( y \) \hspace{1em} \text{direction of coordinate axis y} \\
\( z \) \hspace{1em} \text{direction of coordinate axis z} \\
\( k \) \hspace{1em} \text{sampling time}
Chapter 1

Introduction

Throughout history, the science of navigation has been playing an important role. By navigation we mean the art of finding the way from one place to another. Different kinds of navigation techniques have been used for centuries. One of the earliest navigation approaches were based on dead-reckoning (deduced reckoning, DR). DR has been used in marine applications and long-distance flights for a long time and is still used by navigators today. A dead-reckoning system requires two measurements at minimum; direction measurement and speed measurement with respect to the previous knowledge of the position. Therefore it has been possible to use dead-reckoning without having any highly developed equipment. The distance traveled from one point to another is computed by multiplying the time underway by the speed of the vessel resolved considering the heading angle. Position changes summed with previous position obtains the present position, as stated in [11]. A simple dead-reckoning device includes compass heading in combination with for example odometer, which measures the distance travelled over ground. In modern approaches dead-reckoning is used with electronically measured heading and velocity and it is used in navigation applications today.

An equivalent construction can be made with inertial sensors to sense translational and rotational motion with respect to inertial frame. This approach is known as inertial navigation, which is considered as a form of dead-reckoning. In inertial navigation we use the inertial quality of an object meaning that object maintains its velocity and angular velocity if not affected by any external force. Since 1940s inertial navigation systems (INSs) have been used in many applications. High cost has limited the use of such systems to military and scientific usages. INS uses inertial sensors, as accelerometers and gyros, to detect and measure motion based on physical laws of motion. Inertial systems, and also DR, are self-contained and do not need any knowledge outside the vehicle unlike other types of navigation systems.
These dead-reckoning type of systems are accurate and effective in planar environments but do not provide accurate information when the deviation from planar motion is significant. That is why we want to use inertial measurement units (IMUs) to provide 3-D position and velocity information. IMUs are inertial navigation systems which typically contains three orthogonal rate-gyroscopes and three orthogonal accelerometers, measuring angular velocity and linear acceleration respectively. Though the use of IMU instead of dead-reckoning increases the required computation, IMUs are very robust to external interference. This is due to accurately predicted sensor performance which is not affected by the changes in external fields or vehicle parameters. In this work when talking about INS we mean the specific construction of an inertial measurement unit (IMU). The position, velocity and orientation can be tracked by processing signals from these devices as concluded in Section 3.2.

There are two main approaches to implement INS, which differ with the frame of reference in which the rate-gyroscopes and accelerometers operate. The first approach uses inertial sensors mounted on a platform which is isolated from any external rotational motion, known as stable platform type system. This means that the platform is held in an alignment with some global frame which we are navigating. The second approach is known as strapdown INS and uses inertial sensors mounted rigidly onto the device. The output is measured in the devices frame rather than in the global frame of navigation. The orientation is tracked with gyroscopes measuring the rotation rate and the inertial measurements are transformed to the navigation frame computationally. The main advantages of the use of strapdown system are the decrease in navigation system size, power and cost. In this work the strapdown approach is examined.

The sensors used in strapdown INSs can be generally divided into three groups; navigation, tactical and consumer grade sensors. Sensors of navigation grade are very expensive satisfying high-accuracy requirements. Also tactical grade sensors have accuracy comparable with navigation grade sensors but are too expensive for any consumer. Consumer grade sensors, also referred to as low-cost sensors, are significantly cheaper and possible to utilize in commercial applications. Low-cost sensors are enabling a new generation of commercial navigation applications especially when aided with other sensors.

Among the low-cost sensors recent advances in the construction of the micromachined electromechanical systems (MEMS) has made it possible to manufacture light and small navigation systems. This has increased an interest in the topic of inertial navigation and the application range to for example human and animal tracking, as discussed in [30]. MEMS sensors are built using silicon micro-maching techniques which have low part counts and they are relatively cheap to manufacture in large quantities. MEMS gyroscopes use the Coriolis effect by measuring the secondary vibration and calculating angular velocity due to the Coriolis force. MEMS accelerometers can be either mechanical or solid state sensors. The advantages of MEMS sensors examined
in this work are small size, power consumption, maintenance and price with detriment of far less accuracy compared with e.g. optical gyros.

A pure INS integrates several differential equations constructed on inertial measurements and calculates a navigation solution. As a result, small errors in measurements accumulate due to integration processes and grow into large position and velocity errors. Especially when dealing with low-cost inertial sensors having large sensor errors, the system provides very inaccurate results when navigation is performed without correction over long periods of time. Consequently the inertial navigation system must be aided with external aiding instrumentation to correct navigation errors periodically. We call these kind of systems as aided or integrated INSs. INS aiding can be carried out in numerous ways, of which the most commonly used are different kinds of radio-navigation aids and especially Global Positioning System (GPS) to carry out on-line calibration and error estimation, discussed for example in [27] and [12].

Due to the accuracy of GPS an integrated GPS/INS device can provide quite accurate navigation regardless of the inertial sensors used. However, in this work we are interested in providing improvements in navigation of self-contained INS. This may be the case for example during GPS outages or other failures in GPS navigation performed with integrated GPS/INS. When INS is used unaided, navigation can not be performed accurately over long time particularly in case of low-cost sensors. In this case so called constrained INSs are examined. INS can be constrained with different kinds of speed constraints and constraining the acceleration with physical constraints, some of which are presented in [7]. These constraints can also be considered as external aids for INS and some of the external aids can be regarded as constraints. In this work, however, we are especially interested in nonholonomic constraints which constrain the vehicle frame velocity in directions perpendicular to forward motion assumed to be zero. We will introduce a model for error propagation of INS state variables from different points of view and examine how the errors that result from violating the assumption of nonholonomic constraint are propagating. The goal is to examine whether we can improve the accuracy of INS navigation by this knowledge of error source. The most common approaches for error propagation models are presented in Sections 5.1 and 5.2 and the error propagation of stationary INS is examined in Section 5.3.

There are two primary ways to integrate INS with external aiding knowledge. The first one is the total state approach, also referred as the direct approach, where the state includes estimated variables and the sensor measurements are used directly. The other approach is called the error state or indirect approach, where the state vector includes the errors of estimated variables and the errors between INS computed variables and external aiding variables are used as measurements. The total state approach using integrated GPS/INS is introduced in [16] and using vehicle model constraints is discussed in [10]. Nevertheless, for our purposes it is advantageous to use the error state approach. It gives linear equations for angle errors and makes the system easier to modify to different kinds of measurement data contrary to the total state approach.
In Chapter 6 we will construct a Kalman filter with error state approach based on external information of nonholonomic constraints considering the INS computed body frame velocities violating these constraints as virtual measurements of the error. The goal is to apply this model for INS system to get improvements in navigation accuracy with working sensors of different grades, especially concentrating on low-cost devices.

This work is organized as follows. In Chapter 2 the basic concepts and methods of coordinate frames and coordinate transformations crucial in inertial navigation calculations are introduced and stated. Especially the Euler angle transformation is discussed. In Chapter 3 the inertial navigation system construction is introduced and navigation equations for direct integration to obtain position, velocity and attitude are presented. In Chapter 4 we discuss error sources and error modeling. Chapter 5 constructs the propagation models for the error estimates and the error propagation of stationary INS is examined based on these models. The concept of constrained INSs focusing on nonholonomic constraints and the testing part with MC simulations are included in Chapter 6. In Chapter 7 we draw conclusions and outline future research interests.
Chapter 2

Coordinate transformations

The concept of coordinate frame is very important in inertial navigation computations. A coordinate frame is an analytic abstraction, defined by three consecutively numbered unit vectors, that are mutually perpendicular to one another in the right-hand sense \[23\]. For reasonable navigation the solution has to be given in coordinate frame agreed beforehand. Coordinate frame systems can be either Cartesian or curvilinear e.g. polar coordinate system. In this work only Cartesian right handed axis systems are discussed.

In the inertial navigation computations the coordinate frame has to be transformed commonly. The sensors constitute one coordinate frame, the measurements from the sensors are defined with respect to the inertial coordinate frame and to compute the gravitation, the position in the geodetical coordinate frame has to be known. In this chapter the idea of coordinate frame transformation is presented, transformation models are discussed and different coordinate frames needed in inertial navigation are also performed.

2.1 Coordinate transformation matrix

The static orientation of one coordinate frame with respect to another can be defined with rotation. This rotation determines a coordinate transformation which is a conversion from one system to another, to describe the same space. The conversion can be established with a coordinate transformation matrix. The notation \( C_{\text{from}}^{\text{to}} \) is used to represent rotation from one coordinate frame to another coordinate frame. It means that if vector \( \mathbf{p} \) has a presentation in \( xyz \)-frame as

\[
\mathbf{p}^{xyz} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}
\]
and a presentation in $uvw$-frame as

$$
p^{uvw} = \begin{bmatrix}
  p_u \\
  p_v \\
  p_w 
\end{bmatrix}
$$

the connection between these presentations can be established with coordinate transformation matrix as

$$
p^{uvw} = C^{uvw}_{xyz} p^{xyz} \tag{2.1}
$$

The components of the vector are expressed in terms of the vector components along unit vectors parallel to the respective coordinate axes. In Figure 2.1 we can see the transformation of the position vector $P$ from $xyz$-coordinate frame to $uvw$-coordinate frame in the $xy$-plane when rotating about third coordinate axis $z$ through an angle $\alpha$.

Coordinate transformation matrices also satisfy the composition rule for coordinate frames $a$, $b$ and $c$ which is defined as

$$
C^c_b C^b_a = C^c_a
$$

Figure 2.1: The transformation of the position vector $p$ from $xyz$-coordinate frame to $uvw$-coordinate frame in the $xy$-plane when rotating about third coordinate axis $z$ through an angle $\alpha$.

## 2.2 Coordinate transformation models

### 2.2.1 Direction cosine matrix

The direction cosine matrix is a $3 \times 3$ matrix, which columns are the unit vectors in original axes projected along the new coordinate frame reference axes. The transfor-
The coordinate transformation matrix can be written with cosines of the angles between the coordinate axes of the reference frame and the transformed frame as shown in [18] given as
\[
C_{uvw}^{xyz} = \begin{bmatrix}
\cos(\theta_{zu}) & \cos(\theta_{zv}) & \cos(\theta_{zw}) \\
\cos(\theta_{yu}) & \cos(\theta_{yv}) & \cos(\theta_{yw}) \\
\cos(\theta_{zu}) & \cos(\theta_{zv}) & \cos(\theta_{zw})
\end{bmatrix}
\] (2.2)

where \(\theta_{zu}\) is the angle between x- and u-axis.

### 2.2.2 Euler Angles

The Euler angles are used to define a coordinate transformation in terms of a set of angular rotations. The coordinate transformation matrix is defined with three consecutive rotations about the coordinate axes assuming that we do not rotate consecutively about the same axis. The positive rotation directions from frame of navigation to frame of navigation unit follow the original right-hand sense in relation to the positive directions of acceleration as seen in Figure 2.2.

In our approach we examine the transformation from navigation reference axes to body axes. First we rotate about z-axis through an angle \(\psi\), then we rotate through an angle \(\theta\) about the new y-axis and finally the system is rotated about new x-axis through an angle \(\phi\). These angles are generally called as yaw, pitch and roll angles. The concept of this transformation is illustrated in Figure 2.3, where the layers of rotations and the rotated Euler angles are shown.
Figure 2.3: The Euler angle coordinate transformation. The original frame xyz-frame is transformed to x’y’z’- frame with Euler angle transformation method. First the system is rotated through an angle \(\psi\) along the red line about the z-axis, then through angle \(\theta\) along the blue line about the new temporary y-axis and finally through an angle \(\psi\) along the green line about the final x-axis.

Now the transformation from the body coordinates to navigation reference coordinates can be defined as an inverse transformation. As we have three consecutive rotations we may write three separate transformation matrices. The rotation \(\phi\) about x-axis as

\[
C_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}
\]  

(2.3)

rotation \(\theta\) about new y-axis as

\[
C_\theta = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}
\]  

(2.4)
and rotation $\psi$ about new z-axis as

$$C_\psi = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ (2.5)

which are transposes of the rotation matrices defined for direct transformation. The total transformation matrix representing all three rotations can be established as a product of matrices $C_\theta, C_\phi$ and $C_\psi$ representing these three rotations. Therefore the coordinate transformation matrix between body frame and navigation frame presented with Euler angles is defined as

$$C_n^b = C_\psi C_\theta C_\phi$$ (2.6)

For later use we also define a small angle form of the transformation matrix represented with Euler angles. It can be defined assuming for small angles $\sin(\delta \psi) \approx \delta \psi$ and $\cos(\delta \psi) \approx 1$. With this knowledge we write the transformation matrix as

$$\begin{bmatrix} 1 & -\delta \psi & 0 \\ \delta \psi & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \delta \theta \\ 1 & 0 & 0 \\ 0 & 1 & -\delta \phi \end{bmatrix} = \begin{bmatrix} 1 & -\delta \theta & \delta \psi \\ \delta \theta & 1 & -\delta \phi \\ -\delta \psi & \delta \phi & 1 \end{bmatrix}$$ (2.7)

2.3 Coordinate frames

Fundamental to the process of inertial navigation are the accurate definitions of the coordinate frames. In this work the following orthogonal right handed coordinate frames are used, some of which are also illustrated in Figure 2.4:

- **Inertial frame** (i - frame) is the nonrotating coordinate frame of inertial space used as a reference for angular rotation measurements. Newton’s laws are effective in this frame. As an approximation we can use the ECI- frame which has the origin at the Earth centre and axes are with respect to the fixed stars.

- **Earth frame** (e - frame) is the Earth fixed frame used for position location definition. Its origin is at the Earth centre, one axis is parallel to the Earth polar axis, the other points to the Greenwich meridian and the third one completes the system to a right-hand coordinate system. The abbreviation ECEF (Earth Centered Earth Fixed) is generally used in literature.
- **Body frame** (b-frame) is a frame fixed to the right-handed orthogonal sensor input axes. The accelerations and angular rates exerted from strapdown accelerometers and gyroscopes are measured in this frame.

- **Locally level coordinate frame** (l-frame) is a coordinate frame having its z-axis parallel to the upward vertical at the local Earth surface referenced position. Often used l-frames are ENU (East, North, Up) where x-axis is pointing to east and y-axis to north and NED (North, East, Down).

- **Computer frame** (c-frame) is the local level frame at the inertial navigation system computed position.

- **True frame** (t-frame) is the true local level frame at the true position.

- **Platform frame** (p-frame) is the frame in which the transformed accelerations from accelerometers and angular rates from the gyroscopes are solved to velocity and position.

- **Navigation frame** (n-frame) is here defined as the frame user chooses for navigation output. It can be any of the frames above.

![Figure 2.4: The essential coordinate frames used in inertial navigation.](image-url)
Chapter 3

Inertial navigation systems

Inertial navigation is a self-contained navigation technique in which measurements provided by accelerometers and gyroscopes are used to calculate the position and orientation of an object relative to a known initial position, velocity and orientation [30]. Inertial navigation is the only form of navigation that does not rely on external references.

The operation of inertial navigation depends on the laws of classical mechanics. By measuring the acceleration it is possible to calculate the changes in velocity and position. Acceleration measurements are used because acceleration can be measured internally whereas position and velocity measurements require an external reference.

Inertial navigation systems can be categorized generally to gimbaled and strapdown INSs. In gimbaled, also referred to as stable platform systems, the inertial sensors are mounted on a stable platform which is isolated from any external rotational motion. The navigation system is held in an alignment with the reference frame and therefore no coordinate transformation is needed.

In strapdown systems the inertial sensors are mounted rigidly onto the device and therefore accelerations and angular velocities are measured in the body frame instead of some global frame. Strapdown systems are mechanically less complex and physically smaller than gimbaled systems but require more complex computation. In this work only strapdown inertial navigation systems are discussed.

3.1 Inertial navigation process

In an inertial navigation system the acceleration is measured with accelerometers, which measure the translational motion of the vehicle. Basic form of an accelerometer is a proof mass attached via a spring to the device detecting the mass displacement.
caused by the acceleration of the device. Here we assume that the system has three accelerometers pointing in three orthogonal directions.

To get navigation information with respect to some reference frame, we have to know in which direction the accelerometers are pointing. This can be done using gyroscopes, which measure the angular rate of turn about some fixed axis. Now it is possible to measure the rotation rate of the accelerometer body and thus calculate the accelerations in desired reference frame before integration to velocity and position.

In Figure 3.1 the process of strapdown inertial navigation system is illustrated. The process has two phases: the alignment phase and the navigation phase. Inertial navigation starts with initial position, velocity and orientation or attitude and the process for determining these initial conditions is called initial alignment. In navigation phase attitude position and velocity are resolved through several integrations and one coordinate transformation.

Figure 3.1: The strapdown inertial navigation process.
3.2 Navigation equations

Inertial navigation is the process of calculating position by integration of the velocity and computing velocity by integration of total acceleration [23]. Total acceleration is the sum of gravitational and nongravitational acceleration. Nongravitational acceleration is known as specific force acceleration which is sensed on accelerometers. However, inertial navigation system needs an attitude reference for defining the angular orientation of the accelerometers which is used in the computation of the velocity and position. Therefore we have three kinds of integration processes and further three distinct differential equation describing the propagation of the desired variables; position, velocity and attitude. Angular rate is integrated to get attitude which is used to transform the accelerations into navigation coordinates and accelerations are integrated to get velocity and position. The rectangular rule is used as the integration method due to relatively short range and low accuracy applications.

Without any noise in the system the result of the integration is accurate, except the computational errors which vary mostly depending on the integration method used. In this approach it is assumed that the Earth is spherical which it really is not. In this section we first examine the equations for velocity and position from different point of views and then we discuss different models to illustrate the propagation of attitude.

3.2.1 Velocity

Inertial sensors measure acceleration and rotation rate in inertial frame. Because of the rotating reference frame, the Earth, in inertial frame we measure forces resulted from rotation in addition to the forces affecting on the navigation unit in the Earth surface. Therefore, when computing the propagation of velocity, based on inertial measurements, the Coriolis acceleration has to be taken into account. According to the theorem of Coriolis, the relation between the acceleration of the position vector \( \mathbf{r} \) in a coordinate frame fixed relative to the stars (inertial frame) and a system (the Earth frame) rotating with an angular velocity \( \mathbf{\Omega} \), is as stated in [21], defined as

\[
\left( \frac{d\mathbf{r}}{dt} \right)_i = \left( \frac{d\mathbf{r}}{dt} \right)_e + \mathbf{\omega}_{ie} \times \mathbf{r}
\]

(3.1)

where \( \mathbf{\omega}_{ie} = \begin{bmatrix} 0 & 0 & \mathbf{\Omega} \end{bmatrix}^T \) is the angular velocity between the inertial frame and the Earth frame, \( \mathbf{\Omega} \) being the Earth rotation rate. We call this equation Coriolis equation for vector \( \mathbf{r} \). Here we mark the ground speed of the navigation unit, \( \mathbf{v}_e = \left( \frac{d\mathbf{r}}{dt} \right)_e \). When Equation (3.1) is applied again to the velocity in i-frame and the expressions for
position and velocity are combined, we get the expression for the absolute acceleration $a_i$, defined as

$$a_i = a_e + 2\omega_{ie} \times v_e + \omega_{ie} \times (\omega_{ie} \times r) \quad (3.2)$$

We can see that the acceleration sensed on inertial frame is a sum of the acceleration sensed on the Earth frame, the Coriolis acceleration $2\omega_{ie} \times v_e$, which depends on the velocity, and the centrifugal acceleration $\omega_{ie} \times (\omega_{ie} \times r)$, which depends only on the position. The effects of these accelerations are illustrated in Figure 3.2.

Figure 3.2: Effects of Coriolis acceleration and centrifugal acceleration.

However, the goal is to get a differential equation for the ground velocity propagation in navigation frame, which is rotating with respect to the Earth frame. To do that
let us apply the Coriolis equation for the ground velocity with respect to navigation frame as

$$\left( \frac{d\mathbf{v}_e}{dt} \right)_i = \left( \frac{d\mathbf{v}_e}{dt} \right)_n + \omega_{in} \times \mathbf{v}_e$$ (3.3)

where the angular velocity between inertial frame and navigation frame is sum of Earth rotation rate and rotation rate between Earth frame and navigation frame, given as $\omega_{in} = \omega_{ie} + \omega_{en}$. Now we get the propagation of velocity in navigation frame as

$$\left( \frac{d\mathbf{v}_e}{dt} \right)_n = \left( \frac{d\mathbf{v}_e}{dt} \right)_i - \omega_{in} \times \mathbf{v}_e$$ (3.4)

The second term in equation (3.4), $(d\mathbf{v}_e/dt)_i$, is yet to be calculated. By differentiating the Coriolis equation submitted for vector $\mathbf{r}$ we may infer the expression for the velocity propagation in Earth frame as

$$\left( \frac{d^2\mathbf{r}}{dt^2} \right)_i = \left( \frac{d\mathbf{v}_e}{dt} \right)_i + \omega_{ie} \times \mathbf{v}_e + \omega_{ie} \times (\omega_{ie} \times \mathbf{r})$$ (3.5)

when taking into account that $\frac{d\omega_{ie}}{dt} = 0$, because the Earth rotation rate is constant. Now assuming that the acceleration in i-frame is a sum of specific force acceleration to which the navigation system is subjected and gravitation, given as $(\frac{d^2\mathbf{r}}{dt^2})_i = \mathbf{f} + \mathbf{g}$, we get

$$\left( \frac{d\mathbf{v}_e}{dt} \right)_i = \mathbf{f} - \omega_{ie} \times \mathbf{v}_e + \mathbf{g} - \omega_{ie} \times (\omega_{ie} \times \mathbf{r})$$ (3.6)

We mark the local gravity vector as $\mathbf{g}_l = \mathbf{g} - \omega_{ie} \times (\omega_{ie} \times \mathbf{r})$, which is defined as the sum of the accelerations caused by the mass attraction force and the centrifugal force expressed in inertial frame.

Now we can solve the expression for the velocity propagation in navigation frame from (3.4) using (3.6) as

$$\left( \frac{d\mathbf{v}_e}{dt} \right)_n = \left( \frac{d\mathbf{v}_e}{dt} \right)_i - \omega_{in} \times \mathbf{v}_e$$

$$= \mathbf{f} - (\omega_{ie} + \omega_{in}) \times \mathbf{v}_e + \mathbf{g}_l$$

$$= \mathbf{f} - (2\omega_{ie} + \omega_{en}) \times \mathbf{v}_e + \mathbf{g}_l$$ (3.7)

where $\omega_{in} = \omega_{ie} + \omega_{en}$. Here the Coriolis effect is a sum of the force defined according to the theory of Coriolis illustrated in Figure 3.2(a), and the force caused by the rotation between the Earth frame and the navigation frame. The equation above can be now expressed in navigation frame axes given as

$$\dot{\mathbf{v}}^n = \mathbf{f}^n - (2\omega_{ie}^n + \omega_{en}^n) \times \mathbf{v}^n + \mathbf{g}_l^n$$ (3.8)

where $\mathbf{f}^n = C_b^n \mathbf{f}^b$ and $\mathbf{f}^b$ is the acceleration exerted on accelerometers as defined above.
The angular velocities can be expressed in NED-frame as

\[
\omega_{ie}^n = \begin{bmatrix} \Omega \cos(\lambda) \\ 0 \\ -\Omega \sin(\lambda) \end{bmatrix}
\] (3.9)

and

\[
\omega_{en}^n = \begin{bmatrix} \dot{l} \cos(\lambda) \\ -\dot{\lambda} \\ -\dot{l} \sin(\lambda) \end{bmatrix}
\] (3.10)

where \( \Omega \) represents the Earth rate which is equal to 7.272205 \times 10^{-5} \text{ rad/s}, \( \lambda \) is the local latitude and \( l \) is the local longitude.

For very short time distances in inertial navigation the effects of the rotation of the Earth on the attitude computation and the Coriolis corrections in the velocity equation are no longer essential for the system. This may be due to very inaccurate inertial sensors which can not sense the Earth rotation effects. In this kind of situation we can write the propagation equation for the velocity as

\[
\dot{v}^n = C_b^n T^b + g^n
\] (3.11)

The propagation of velocity can be written in more detailed form with estimating the differential changes in longitude and latitude according to [27] given as

\[
\dot{l} = \frac{v_e}{R} \sec(\lambda)
\] (3.12)

and

\[
\dot{\lambda} = \frac{v_n}{R}
\] (3.13)

With these expressions we can write the propagation equations for the velocity components \( \mathbf{v}^n = \begin{bmatrix} v_n & v_e & v_d \end{bmatrix}^T \) (north, east, down) in navigation frame given as

\[
\dot{v}_n = f_n - v_e (2\Omega + \dot{l}) \sin(\lambda) + v_d \dot{\lambda} + \xi g
\]

\[
= f_n - 2\Omega v_e \sin(\lambda) + \frac{v_n v_d - v_e^2 \tan(\lambda)}{R} + \xi g
\] (3.14)

\[
\dot{v}_e = f_e + v_n (2\Omega + \dot{l}) \sin(\lambda) + v_d (2\Omega + \dot{l}) \cos(\lambda) - \eta g
\]

\[
= f_e + 2\Omega (v_n \sin(\lambda) + v_d \cos(\lambda)) + \frac{v_e}{R} (v_d + v_n \tan(\lambda)) - \eta g
\] (3.15)
\[ \dot{v}_d = f_d - v_e (2\Omega + \dot{l}) \cos(\lambda) - v_n \dot{\lambda} + g \]
\[ = f_d - 2\Omega v_e \cos(\lambda) - \frac{v_n^2 + v_d^2}{R} + g \]  
(3.16)

Here the force vector is again given as
\[
\begin{bmatrix} f_n \\ f_e \\ f_d \end{bmatrix} = C_n^b f^b
\]  
(3.17)

representing the true force defined as non-gravity force per unit mass exerted on the accelerometers [16]. It can be computed from the outputs of the accelerometers with the transformation matrix \( C_n^b \) between body frame and navigation frame derived from the gyroscope outputs. The factors \( \xi \) and \( \eta \) represent the angular deflections in the direction of the local gravity vector.

### 3.2.2 Attitude

Let us now examine the propagation of the transformation matrix \( C_n^b \) which represent the propagation of the attitude of inertial navigation system. We start from defining
\[
\dot{C}_n^b = \lim_{\delta t \to 0} \frac{\delta C_n^b}{\delta t} = \frac{C_n^b(t + \delta t) - C_n^b(t)}{\delta t}
\]  
(3.18)

We use a matrix product to get
\[
C_n^b(t + \delta t) = C_n^b(t) M(t)
\]  
(3.19)

Here the matrix \( M(t) \) transforms from b-frame at time \( t \) to b-frame at time \( t + \delta t \). It can be expressed for small angle rotations as presented earlier, given as
\[
M(t) = \begin{bmatrix} 1 & -\delta \psi & \delta \theta \\
\delta \psi & 1 & -\delta \phi \\
-\delta \theta & \delta \phi & 1 \end{bmatrix}
\]  
(3.20)

where \( \delta \phi \), \( \delta \theta \) and \( \delta \psi \) are the small rotation angles between differential time distances about b-frame coordinate axes defined e.g. as roll, pitch and yaw axes as presented in [13]. It can be written as
\[
M(t) = I + \delta \Psi
\]  
(3.20)
CHAPTER 3. INERTIAL NAVIGATION SYSTEMS

The small angle assumption is valid when the time difference approaches zero. Now we can modify equation (3.18) using equations (3.20) and (3.19) as follows, to get

\[ \dot{C}_b^m = \lim_{\delta t \to 0} \frac{C_b^m(t + \delta t) - C_b^m(t)}{\delta t} \]

\[ = \lim_{\delta t \to 0} \frac{C_b^m(t)(I + \delta \Psi) - C_b^m(t)}{\delta t} \]

\[ = C_b^m \lim_{\delta t \to 0} \frac{\delta \Psi}{\delta t} \]

In the limit as \( \delta t \to 0 \), the transformation matrix propagation can be written with propagation of angles \( \psi, \phi \) and \( \theta \), defined as

\[ \dot{C}_b^m = C_b^m \begin{bmatrix} 0 & -\dot{\psi} & \dot{\theta} \\ \dot{\psi} & 0 & -\dot{\phi} \\ -\dot{\theta} & \dot{\phi} & 0 \end{bmatrix} \] (3.21)

The angle propagations correspond to the angular velocity between body frame and inertial frame representing the rotation rates through roll, pitch and yaw axes, defined as \( \omega_{nb}^b = [\omega_x \omega_y \omega_z]^T \). We also notice that the matrix is skew symmetric having angular velocities as components. This kind of matrix can be written with a cross product of angular velocity vector. Here we mark the skew symmetric matrix representing the cross product of angular velocity vector as

\[ \Omega_{nb}^b = \omega_{nb}^b \times \]

Accordingly, the transformation matrix between the body frame and the navigation frame propagates in accordance with the following equation

\[ \dot{C}_b^m = C_b^m \Omega_{nb}^b \] (3.22)

However, we can not evaluate \( \omega_{nb}^b \) directly and that is why we want to convert it in more applicable form as follows. We start from equation

\[ C_b^m = C_i^m C_b^i \] (3.23)

Using (3.22) we can write

\[ \dot{C}_n^i = C_n^i (\omega_{in}^n \times) \] (3.24)

The transpose of the equation above can be given as

\[ \dot{C}_i^m = -(\omega_{in}^n \times) C_i^m \] (3.25)
By differentiating equation (3.23) with respect to time, we get
\[
\dot{C}_b^m = \dot{C}_i^b C_b^i + C_i^b \dot{C}_b^i
\] (3.26)

Using relation \( \dot{C}_b^i = C_b^i (\omega_{ib}^b \times) \) and (3.25), we end up with the attitude propagation of inertial navigation unit with respect to the navigation coordinate frame, that is
\[
\dot{C}_b^m = C_b^m (\omega_{ib}^b \times) - (\omega_{in}^n \times) C_b^m
\] (3.27)

This form can be used in inertial navigation computations. \( \omega_{ib}^b \) is the angular velocity exerted on gyroscope and \( \omega_{in}^n = \omega_{ie}^n + \omega_{en}^n \), which are known quantities, presented in equations (3.9) and (3.10).

The propagation of attitude may also be expressed with Euler angles. In the Chapter 2 the transformation matrix is presented with Euler angles. Now we want to examine how the rotation angles \( \theta, \psi \) and \( \phi \) are propagating. The rotation rates in body frame can be written with the angle propagation rates. The propagation rate is defined with respect to the new rotation axis resulted from the previous rotation. It means that we transform the angle propagations to the original coordinate frame. Therefore we have
\[
\begin{bmatrix}
\omega_x \\
\omega_y \\
\omega_z
\end{bmatrix} = \begin{bmatrix}
\dot{\phi} \\
0 \\
0
\end{bmatrix} + C_T^\phi \begin{bmatrix}
0 \\
0 \\
\dot{\psi}
\end{bmatrix} + C_T^\phi C_T^\theta \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}
\] (3.28)

By arranging the equation above we get the propagation equation for the Euler angles as presented in [27] and [20], given as
\[
\begin{bmatrix}
\dot{\phi} \\
\dot{\theta} \\
\dot{\psi}
\end{bmatrix} = \begin{bmatrix}
1 & \sin(\phi) \sin(\theta)/\cos(\theta) & \cos(\phi) \sin(\theta)/\cos(\theta) \\
0 & \cos(\phi) & -\sin(\phi) \\
0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta)
\end{bmatrix} \begin{bmatrix}
\omega_x \\
\omega_y \\
\omega_z
\end{bmatrix}
\] (3.29)

3.2.3 Position

The position propagation in navigation frame can be determined from the velocity of the navigation unit. Therefore the propagation of the position in navigation frame can simply be given as
\[
\begin{bmatrix}
\dot{r}_n \\
\dot{r}_e \\
\dot{r}_d
\end{bmatrix} = \begin{bmatrix}
v_n \\
v_e \\
v_d
\end{bmatrix}
\] (3.30)
Also often used way to express the position propagation is to determine position in the Earth-frame. The orientation between the Earth frame and the local level frame is illustrated in Figure 3.3.

![Figure 3.3: The location of the true frame with respect to the Earth frame.](image)

The position is given in terms of angular orientation of the local vertical of the navigation frame with respect to the Earth (expressed with latitude and longitude) and height above the Earth. Now we can express the propagation of position coordinates in the Earth frame with differential equations discussed in [23] defined as

\[
\dot{C}_n^e = C_n^e(\omega^e_{en} \times) \quad (3.31)
\]

and

\[
\dot{h} = \mathbf{u} \cdot \mathbf{v}^n \quad (3.32)
\]

where \( \mathbf{u} \) is an unit vector representing the direction ”up” in the local-frame and \( \omega^e_{en} = \begin{bmatrix} i \cos \lambda & -\dot{\lambda} & -i \sin \lambda \end{bmatrix}^T \).
Chapter 4

INS errors

In practice the accuracy to which the inertial navigation system is able to function is limited by the errors in the data which is given to the system and also by imperfections in the construction of the system components. In long durations the INS output is going to be strongly biased. The rate at which navigation errors grow over long distances of time is related to the accuracy of the inertial sensors as well as the accuracy of the initial alignment. Any errors in either initial alignment phase or in the navigation phase are integrated in the INS navigation algorithm. The errors will propagate over time and determine the accuracy of the inertial navigation system.

According to [27], errors can be categorized in three different sources; initial alignment errors, sensor errors and computational errors. In this chapter these three error sources are discussed.

4.1 INS sensor errors and sources of the errors

Inertial sensors are always influenced by errors that determine the accuracy of the output that can be measured. In our system we assume we have three kinds of error terms; bias, scale factor error and cross-coupling error.

Let us first examine the angular rate measurement provided by the gyroscope having $x$-axis as an input axis. In the ideal case the gyroscope does not sense any data that should be sensed by other gyros from different input axes. However here the input axis $x$ could be biased having cross-coupling error with the reference frame. The reference frame is assumed to be perfectly orthogonal. Considering the error terms introduced above we have the angular rate measurement exerted on the gyroscope given as

$$\tilde{\omega}_x = (1 + S_x)\omega_x + M_y\omega_y + M_z\omega_z + B_x + n_x$$  (4.1)
where $\omega_x$ is the turn rate of the gyroscope about x axis. $B_x$ is the bias of the gyroscope at x-axis and $n_x$ is the zero mean noise term. $S_x$ is the scale factor and $M_y$ and $M_z$ are the cross-coupling coefficients.

The model presented in (4.1) is acceptable for accelerometers as well. The same error terms appear, we just have to replace the angular velocity $\omega_x$ with acceleration $a_x$.

In general we have to realize that the error described above can include several components at the same time. Fixed terms, terms varying with temperature, switch-on to switch-on variations and in-run variations as discussed in [27] and the next subsection.

The rate which total navigation errors grow over long distances of time is strongly related to the accuracy of the inertial sensors. Moreover the accuracy of the sensors is roughly proportional to the sensor price. In Table 4.1 the approximate values of biases and scale factor errors are presented for three different grades of inertial sensors; navigation, tactical and consumer grade sensors. As can be seen there is huge difference in the sensor accuracies between distinct grades. This means that the navigation with sensors available for any user can not be performed even near as accurately as the navigation with expensive sensors. Therefore considering the consumer grade navigation we have to place the accuracy requirements in totally different time perspective. In Chapter 6 we can see simulation results which illustrate the navigation results comparing different sensor grades.

<table>
<thead>
<tr>
<th>Error</th>
<th>Navigation grade</th>
<th>Tactical grade</th>
<th>Consumer grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerometer bias</td>
<td>10 - 50 $\mu$g</td>
<td>100 - 500 $\mu$g</td>
<td>10 - 20 mg</td>
</tr>
<tr>
<td>Accelerometer scale factor</td>
<td>10 - 50 ppm</td>
<td>100 - 300 ppm</td>
<td>1 - 3 %</td>
</tr>
<tr>
<td>Gyroscope bias</td>
<td>0.005 - 0.01 deg/h</td>
<td>1 - 10 deg/h</td>
<td>$\geq$ 1000 deg/h</td>
</tr>
<tr>
<td>Gyroscope scale factor</td>
<td>10 - 50 ppm</td>
<td>100 - 300 ppm</td>
<td>1 - 3 %</td>
</tr>
<tr>
<td>Price</td>
<td>100,000€</td>
<td>25,000€</td>
<td>100€</td>
</tr>
</tbody>
</table>

### 4.1.1 Gyroscope errors

Usually the gyroscope errors define the accuracy of the inertial navigation system due to the difficulty of measuring angular rates. If the system operates unaided, gyro bias indicates the increase of angular error over time. In aided system the gyro drift is mainly affecting the heading accuracy over time. Small gyro drift indicates better angular corrections and also longer possible navigation time where the aiding information may not be present.
The bias of a rate gyro refers to the sensor output which is present even in the absence of an applied input rotation. In other words, it means the offset of the output from the true value. In general case the measurement of angular rate can include different types of bias components. The component which is present every time the sensor is switched on called turn-on bias. It is predictable and can therefore be corrected. The bias component dependent on temperature can be handled with suitable calibration. There is also the random bias which varies from gyroscope switch-on to switch-on and the in-run random bias which varies throughout the run. In our problem we assume that the turn-on bias is corrected and we have suitable calibration for temperature dependent bias. Assuming that the systematic errors are compensated, it is mainly the switch-on to switch-on and in-run variations which influence the performance of the inertial system which the sensors are installed [27].

The fixed bias, marked here as $B_x$, is the bias which should have relative long correlation time and so remain the same or nearly the same during the navigation. We want this to be true only between activations and in practice the correlation time is nearly the time between switch-ons. The size of the bias is here assumed to be independent of any motion. When integrated, a constant bias error causes an angular error, $\delta \theta$, which grows linearly with time as

$$\delta \theta(t) = B_x \cdot t$$

where $t$ is representing time. Usually bias is expressed in units of degrees per hour but in case of low cost gyroscopes we use degrees per second.

The random noise part, marked as $n_x$, can be caused for example by instabilities in the gyroscope. It is assumed to have relative short correlation times due to random movements of the rotor along the spin axis. The correlation time of the error term is significant in the INS mechanization due to many integrations. The noise is fluctuating at a rate much greater than the sampling rate of the sensor. As a result the samples obtained from the sensor are perturbed by a white noise, which means a sequence of zero-mean uncorrelated random variables, $N_i$. Now each $N_i$ is identically distributed with zero mean and has a finite variance $\sigma^2$. In integration using the rectangular rule the random noise part introduces a zero-mean random walk error into the integrated signal, which is defined as stochastic process. When white noise signal, $b_x$, is integrated over time $t = n \delta t$, where $\delta t$ is the time between samples and $n$ is the number of samples, we get

$$\int_0^t b_x(\tau) \delta \tau \approx \delta t \sum_{i=1}^n N_i$$

Now the standard deviation, as stated in [30], is defined as

$$\sigma_\theta(t)^2 = \text{Var}(\int_0^t b_x(\tau) \delta \tau) = \delta t^2 n \text{Var}(N) = \sigma^2 \delta t \cdot t$$
In real applications the separation between constant part and random part of the bias is not always apparent but it can be done for example with Allan variance method, readers interested are referred to [4] or [26].

Scale factor errors are errors in the ratio relating the change in the output signal to a change in the measured input rate. It includes the scaling and misalignment errors. If the scale factor is nonlinear, additional errors result. Scale factor errors can arise for example from temperature variations. The result in the integration process is an orientation drift growing proportional to the rate and duration of the motion.

Cross-coupling errors result from the non-orthogonality of the sensor axes as an imperfection in the construction of the sensors. It is due to the fact that it is impossible to construct a system where sensors are perfectly perpendicular with one another. That is why the data sensed on one axis is also sensed on the others. However, it is important to notice that the axes don’t need to be orthogonal to get accurate results, we just have to know the angles between the axes.

4.1.2 Accelerometer errors

Also in the case of the accelerometer, the bias term is assumed to have several components; fixed and repeatable terms, temperature induced variations, switch-on to switch-on variations and in-run variations. As with gyroscopes, the examination can be reduced to the constant part of bias referring to the error in estimating the switch-on to switch-on bias and the random part of the bias referring to the in-run bias.

The constant bias of an accelerometer is the offset of its output signal from the true value, given in m/s². A constant bias error of $B_x$, when double integrated, causes an error in position which grows quadratically with time. The accumulated error in position, $\delta r$, is given as

$$\delta r(t) = \frac{B_x t^2}{2}$$

where $t$ is the time of integration.

The outputs of accelerometers are perturbed by a white noise sequence. Now the white noise creates a velocity random walk. When the corrupted samples obtained from an accelerometer are double integrated, a second order random walk with zero mean is composed in navigation position. When integrating twice the white noise signal, we get

$$\int_0^t \int_0^t b_x(\tau) \delta \tau \delta \tau \approx \delta t \sum_{i=1}^n \delta t \sum_{j=1}^n N_j$$

(4.6)
The standard deviation, as stated in [30], can be given as

\[ \sigma_r(t) = \text{Var}(\int_0^t \int_0^t b_x(\tau) \delta \tau \delta \tau) \approx \sigma^2 \cdot t^3 \cdot \delta t/3 \]  

(4.7)

Accelerometers also have scale factor errors, which define position drift proportional to the squared rate and duration of acceleration.

### 4.2 Initial alignment errors

As seen in Figure 3.1, INS process consists of the alignment phase and the navigation phase. INS integration starts with initial position, velocity and attitude and the process of determining these initial conditions is called initial alignment. The initial alignment is required for INS system to provide accurate results and it can be done in several different ways depending on the sensors that the system contains.

In case of low-cost inertial sensors, regarded here as consumer grade sensors, INS is not able to measure the Earth rotation rate. Now the initial alignment can not be performed properly because the initial attitude is not obtained [16]. It is impossible to align a low-cost inertial measurement unit in stationary mode without augmenting the system with other sensors. One way to assist the alignment phase is to use in-motion alignment, readers interested are referred to [16] and [27].

Often term alignment error is used when discussing the error modeling. Alignment generally is the process whereby the orientation of the axes of a strapdown inertial navigation system is determined with respect to the reference axis system. The fact is that the sensors and their platform can not be aligned perfectly with their assumed directions. This means that sensor errors, computational errors and also errors in the initial estimates result in errors in the computed transformations between reference frames. When present in the navigation system a portion of the computed motion along a given axis is manifested along a different axis in the actual system.

The distinction of these errors is troublesome because there exists correlation between certain error sources. For example attitude errors and accelerometer biases are lost in a strapdown system when the vehicle changes course. When a stationary navigation system maintains the alignment attitude, there is cancellation between the alignment errors and inertial sensor biases. This means that INS must change its alignment heading and relatively large navigation error is generated. This is illustrated as the stationary model is discussed. In simulations of Chapter 6 the initial alignment is assumed to be performed accurately.
4.3 Computational errors

In the strapdown navigation system computer we have computational errors, which are inaccuracies arising as a result of several sources. First we have bandwidth limitations resulting from restricted computational frequency as described in [27]. Inaccuracies will also arise from the truncation of mathematical functions used in algorithms and limitations in numerical integration method used, as discussed in [18].

When examining an inertial system it is usually assumed and ensured that navigation errors arising from computational errors are relatively small compared to the alignment error and sensor errors. Therefore the attention can be concentrated on the latter, especially in case of consumer grade inertial sensors. Nonetheless there are some computational imperfections which are expected for example as a result of certain environmental property and should be taken into account. In the simulations carried out in this work computational errors are not taken into examination.
Chapter 5

Error propagation models

The behavior of the errors in inertial navigation system can be modeled with error propagation models. The propagation model estimates the total error in the system as a result of different kinds of errors in the sensors or the initial alignment phase. Error models are developed by perturbing the nominal differential equations whose solution yields the inertial navigation system output of velocity, position and orientation [16]. The basic differential equations can be expressed in different coordinate frames, for example in computer frame or in true frame. The choice of the coordinate frame leads to the different approaches for INS error analysis. These approaches differ with one another on the definition of the error angle between the coordinate frames examined. In this chapter two of these models, the psi angle approach and the phi angle approach are introduced.

5.1 Psi angle error model

In the psi angle error analysis it is assumed that the navigation equations are solved in the computer frame. Now the error equations are derived from the perturbation of the computer frame solution. The computer frame is defined as the local level frame located in the INS computed position. In platform frame the transformed accelerations from the accelerometers and the angular rates from the gyroscopes are solved. Angle ψ represents the angle between computer frame and platform frame. In Figure 5.1 we may see how psi angle is defined with coordinate frames used in this approach.

Here we introduce the propagation equations using psi angle approach first for velocity, then for attitude with both small angle and large angle assumption and finally for position.
5.1.1 Velocity error model

The true velocity vector in the computer frame, \( \mathbf{v}^c = \begin{bmatrix} v^c_x & v^c_y & v^c_z \end{bmatrix}^T \), can be derived according to (3.8) as
\[
\dot{\mathbf{v}}^c = \mathbf{f}^c - (2\omega^e_{ie} + \omega^e_{en}) \times \mathbf{v}^c + \mathbf{g}^c
\] (5.1)
where \( \mathbf{f}^c \) is the specific force vector resolved in computer frame. However in the real application instead of \( \mathbf{f}^c \) only \( \mathbf{f}^p + \Delta \mathbf{f}^p \) are available because the accelerometers are in platform frame having errors \( \Delta \mathbf{f}^p \). Thus, the inertial navigation system computes the following velocity vector
\[
\dot{\hat{\mathbf{v}}}^c = \mathbf{f}^p + \nabla^p - (2\omega^e_{ie} + \omega^e_{en}) \times \hat{\mathbf{v}}^c + \mathbf{\hat{g}}^c
\] (5.2)
where \( \hat{\mathbf{v}}^c \) is the velocity vector in the computer frame computed by the inertial navigation system. \( \mathbf{f}^p \) is the true specific force vector resolved in the platform frame and the errors are presented with \( \nabla^p \), which is the bias vector of the accelerometers in the platform frame. Due to the existence of the error sources of INS computed variables we then have
\[
\dot{\mathbf{v}}_t^c = \mathbf{v}_t^c + \Delta \mathbf{v}_t^c
\]
\[
\dot{\mathbf{g}}_t^c = \mathbf{g}_t^c + \Delta \mathbf{g}_t^c
\] (5.3)
\( \Delta \) representing the errors in given term. Now we can construct the psi angle error propagation model for velocity by computing the difference between the true velocity
vector in the computer frame and the inertial navigation system computed velocity vector. It can established as

\[ \Delta \dot{v}^c = \dot{\hat{v}}^c - \dot{v}^c = (f^p + \nabla^p + \hat{g}^c - (2\omega_{ie} + \omega_{en}) \times \hat{v}^c) - (f^c + g^c - (2\omega_{ic} + \omega_{en}) \times v^c) \]

\[ = (f^p - C_p^c f^p) - (2\omega_{ic} + \omega_{en}) \times (\hat{v}^e - v^e) + \nabla^p + (\hat{g}^c - g^c) \]

\[ = (I - C_p^c) f^p - (2\omega_{ie} + \omega_{en}) \times \Delta v^c + \nabla^p + \Delta g^c \] (5.4)

Equation (5.4) is the general velocity error propagation model in psi angle approach without any assumptions of the error magnitudes.

However, for the later use we want to derive a linear error model with certain assumptions. Let us first assume that the angles between computer frame and platform frame are relatively small. According to (2.7) the transformation matrix from the computer frame to the platform frame can under these circumstances be given as

\[ C_p^c = (C_e^p)^T = (I - \psi \times)^T = I + \psi \times = \begin{bmatrix} 1 & -\psi_z & \psi_y \\ \psi_z & 1 & -\psi_x \\ -\psi_y & \psi_x & 1 \end{bmatrix} \] (5.5)

Using small angle assumption the error model can be thus written as

\[ \Delta \dot{v}^c = -\psi \times f^p - (2\omega_{ie} + \omega_{en}) \times \Delta v^c + \nabla^p + \Delta g^c \] (5.6)

For the gravitation error we know from [16] that

\[ \Delta g^c = -\frac{g}{R} \begin{bmatrix} \Delta r_x^c \\ \Delta r_y^c \\ \Delta r_z^c \end{bmatrix} \] (5.7)

where \( R \) is the Earth radius. By writing the cross product in matrix form, using the relation \(-\psi \times f^p = f^p \times \psi\) and Equations (3.9), (3.10), and (5.7), the error propagation model for the velocity in computer frame with an assumption of small error angles can be given as

\[
\begin{bmatrix}
\Delta \dot{v}_x^c \\
\Delta \dot{v}_y^c \\
\Delta \dot{v}_z^c
\end{bmatrix} =
\begin{bmatrix}
0 & -f_z & f_y \\
f_z & 0 & -f_x \\
-f_y & f_x & 0
\end{bmatrix}
\begin{bmatrix}
\psi_x \\
\psi_y \\
\psi_z
\end{bmatrix}
+ \begin{bmatrix}
0 & -(2\Omega + \dot{l}) \sin(\lambda) \\
(2\Omega + \dot{l}) \sin(\lambda) & \dot{\lambda} \\
-\dot{\lambda} & -(2\Omega + \dot{l}) \cos(\lambda)
\end{bmatrix}
\begin{bmatrix}
\Delta v_x \\
\Delta v_y \\
\Delta v_z
\end{bmatrix}
+ \begin{bmatrix}
-g/R & 0 & 0 \\
0 & -g/R & 0 \\
0 & 0 & -g/R
\end{bmatrix}
\begin{bmatrix}
\Delta r_x \\
\Delta r_y \\
\Delta r_z
\end{bmatrix}
+ \begin{bmatrix}
\nabla^p_x \\
\nabla^p_y \\
\nabla^p_z
\end{bmatrix}
\] (5.8)
where $\nabla^p$ is the accelerometer error in the platform frame.

### 5.1.2 Psi angle attitude error model with small angle assumption

In the psi angle approach the navigation system assumes that the platform axes and the computer frame axes are coincident with each other. The gyros are torqued with the angular velocity $\omega_{ic}$ and control the platform along the platform axes. We can write the angular velocity of the platform frame, assuming that gyro has drift $\epsilon^p$, given as

$$\omega_{ip}^p = \omega_{ic}^p + \epsilon^p \quad (5.9)$$

This means that the angular velocity in platform frame equals to the torquing rates sensed by gyros plus the gyro drifts \cite{15}, which defines the propagation of the angle error between these frames.

By multiplying both sides of the Equation (5.9) with the small angle form of the transformation matrix $C_p^c$, as presented in Equation (5.5), we get

$$(I - \psi \times)\omega_{ip}^p = C_p^c \omega_{ic}^p + (I - \psi \times)\epsilon^p \quad (5.10)$$

Now we can solve the angular velocity of the platform frame with respect to the computer frame, that is

$$\omega_{cp}^p = \psi \times \omega_{ip}^p + \epsilon^p \quad (5.11)$$

where $\omega_{cp}^p = \omega_{ip}^p - \omega_{ic}^p$. Here the second order error product term $\psi \times \epsilon^p$ is neglected.

The angular velocity $\omega_{cp}^p$ equals the psi angle rate $\dot{\psi}^p$ with small angle assumption and we can write the common psi angle propagation model as

$$\dot{\psi}^p = -\omega_{ip}^p \times \psi^p + \epsilon^p \quad (5.12)$$

where $\omega_{ip}^p = \omega_{ie}^p + \omega_{ep}^p$. Finally we build up the matrix form of the angle error propagation model as

$$
\begin{bmatrix}
\dot{\psi}_x \\
\dot{\psi}_y \\
\dot{\psi}_z
\end{bmatrix} =
\begin{bmatrix}
0 & -(\Omega + \dot{l}) \sin(\lambda) & \dot{\lambda} \\
(\Omega + \dot{l}) \sin(\lambda) & 0 & (\Omega + \dot{l}) \cos(\lambda) \\
-\dot{\lambda} & -(\Omega + \dot{l}) \cos(\lambda) & 0
\end{bmatrix}
\begin{bmatrix}
\psi_x \\
\psi_y \\
\psi_z
\end{bmatrix} +
\begin{bmatrix}
\epsilon_x \\
\epsilon_y \\
\epsilon_z
\end{bmatrix}
\quad (5.13)
$$
5.1.3 Psi angle error model for large errors

In this subsection the purpose is to develop a psi angle error model for large angle errors.

The true transition matrix is according to Equation (3.27) given as

\[ \dot{C}_b^c = C_b^c \Omega_{ib}^b - \Omega_{ib}^c C_b^c \]  \hspace{1cm} (5.14)

As we may see the true transition matrix is resolved with the true rotation rate \( \Omega_{ib}^b \).

The inertial navigation system however provides the rotation rate

\[ \hat{\Omega}_{ib}^b = \Omega_{ib}^b + \varepsilon^b \]  \hspace{1cm} (5.15)

where \( \varepsilon \) illustrates the gyro error skew-symmetric matrix, which may be large in the case of low cost IMU. Therefore matrix \( C_p^p \) is obtained using this gyro rate as

\[ \dot{C}_b^p = C_b^p \hat{\Omega}_{ib}^b - \Omega_{ib}^c C_b^p \]  \hspace{1cm} (5.16)

Let us now define

\[ \Delta C = C_p^p - C_b^c \]  \hspace{1cm} (5.17)

\[ = C_p^p - C_b^p (I - C_p^p) C_b^p \]  \hspace{1cm} (5.18)

The purpose is to examine the derivative of this discrepancy of transition matrices \( \Delta C \). Using (5.16) we construct the derivative both from (5.17) and (5.18), as stated in [17], defined as

\[ \dot{\Delta} C = - \dot{C}_p^p C_b^p + (I - C_p^p) \dot{C}_b^p \]

\[ = - \dot{C}_p^p C_b^p + (I - C_p^p) (C_b^p \hat{\Omega}_{ib}^b - \Omega_{ib}^c C_b^p) \]

\[ = C_b^p \hat{\Omega}_{ib}^b - \Omega_{ib}^c C_b^p - \dot{C}_p^p \hat{\Omega}_{ib}^b + C_b^p \Omega_{ib}^c C_b^p - \dot{C}_p^p C_b^p \]  \hspace{1cm} (5.19)

and from (5.17)

\[ \dot{\Delta} C = \dot{C}_b^p - \dot{C}_b^c \]

\[ = C_b^p \hat{\Omega}_{ib}^b - \Omega_{ib}^c C_b^p - C_b^p \Omega_{ib}^b + \Omega_{ib}^c C_b^p \]

\[ = C_b^p \hat{\Omega}_{ib}^b - \Omega_{ib}^c C_b^p - C_b^p C_p^p \Omega_{ib}^b + \Omega_{ib}^c C_p^p C_b^p \]  \hspace{1cm} (5.20)

Equality of Equations (5.19) and (5.20) leads to

\[ \dot{C}_p^p C_b^p + C_p^p C_b^p \hat{\Omega}_{ib}^b - C_p^p \Omega_{ib}^c C_b^p - C_p^p C_b^p \hat{\Omega}_{ib}^b + \Omega_{ib}^c C_p^p C_b^p = 0 \]  \hspace{1cm} (5.21)
Next we will give the desired $\Omega^p_{cp}$, the angular velocity matrix between computer frame and platform frame. First using (5.15) and multiplying right with $C_b^p$ we get

$$\dot{C}_p^c + C_p^c C_b^p \varepsilon_b^p b - C_p^c \Omega^c_{ic} + \dot{\Omega}^c_{ic} C_p^c = 0$$  \hspace{1cm} (5.22)

Then using $\dot{C}_p^c = C_p^c \Omega^p_{cp}$

$$C_p^c \Omega^p_{cp} + C_p^c C_b^p \varepsilon_b^p b - C_p^c \Omega^c_{ic} + \dot{\Omega}^c_{ic} C_p^c = 0$$  \hspace{1cm} (5.23)

and multiplying left with $C_p^c$ we end up with

$$\Omega^p_{cp} + C_b^p \varepsilon_b^p b - \Omega^c_{ic} + C_p^c \Omega^c_{ic} C_p^c = 0$$  \hspace{1cm} (5.24)

We introduce two results to make the model more convenient, as given in [17]. For the rotation rate error we have

$$\varepsilon^p = C_b^p \varepsilon_b^p b = C_p^c \varepsilon_c^p C_p^c$$  \hspace{1cm} (5.25)

and similarly for angular velocity as

$$\Omega^p_{ic} = C_p^c \Omega^c_{ic} C_p^c$$  \hspace{1cm} (5.26)

Combining Equations (5.24) and (5.26) we get

$$\Omega^p_{cp} + \varepsilon^p - \Omega^c_{ic} + \Omega^p_{ic} = 0$$  \hspace{1cm} (5.27)

The skew symmetric matrices correspond angular velocities in the sum above and the model can be given as

$$\omega^p_{cp} + \varepsilon^p - \omega^c_{ic} + \omega^p_{ic} = 0$$  \hspace{1cm} (5.28)

where the angular velocity $\omega^p_{cp}$ can be derived using equation $\omega^p_{ic} = C_c^p \omega^c_{ic}$ as

$$\omega^p_{cp} = (I - C_c^p) \omega^c_{ic} - \varepsilon^p$$  \hspace{1cm} (5.29)

We know that the angular velocity between the computer frame and the platform frame equals the derivative of the psi angle, given as $\omega^p_{cp} = \dot{\psi}$. The angular velocity $\omega^c_{ic}$ is known without error by the definition of the computer frame, exerted on gyroscopes. The general psi angle error model for large errors can be thus defined as

$$\dot{\psi} = (I - C_c^p) \omega^c_{ic} - \varepsilon^p$$  \hspace{1cm} (5.30)

Also from here we can resolve the psi angle error model for small errors by using the transition matrix for small angles in Equation (5.5).
5.1.4 Position error model

It is assumed that we are here dealing with a terrestrial navigation problem and hence the velocity is considered to be the ground velocity which is the velocity with respect to the Earth. For the ground velocity we can write a differential equation as

\[ \mathbf{v} = \frac{d\mathbf{r}^e}{dt} \]  

(5.31)

where \( \mathbf{r}^e \) is the position from the center of the Earth. Now the errorless solution for position in the Earth frame at time \( \tau \) can be presented as

\[ \mathbf{r}^e_\tau = \int_0^\tau C^e_c \mathbf{v}^e_i dt \]  

(5.32)

Here we can notice, as presented in Equation (5.3), that the inertial navigation system can not measure the velocity vector in computer frame, \( \mathbf{v}^e \), accurately but measures \( \mathbf{v}^e + \Delta \mathbf{v}^e \). We also know that \( C^e_c \) is known without error by the definition of the computer frame. For this reason the position vector \( \mathbf{r}^e \) is not available directly but instead we have

\[ \mathbf{r}^e_\tau + \Delta \mathbf{r}^e = \int_0^\tau C^e_c (\mathbf{v}^e_i + \Delta \mathbf{v}^e) \]  

(5.33)

\( \Delta \mathbf{r}^e \) being the position error. Now the expression for the position error in the Earth frame at time \( \tau \) can be given as

\[ \Delta \mathbf{r}^e = \int_0^\tau C^e_c \Delta \mathbf{v}^e_i dt \]  

(5.34)

and for the position error in computer frame at time \( \tau \) we have

\[ \Delta \mathbf{r}^e = C^e_c \int_0^\tau C^e_c \Delta \mathbf{v}^e_i dt \]  

(5.35)

By differentiating the equation above we get the position propagation in computer frame, that is

\[ \Delta \dot{\mathbf{r}}^e = \frac{d}{d\tau} (C^e_c) \int_0^\tau C^e_c \Delta \mathbf{v}^e_i dt + C^e_c \frac{d}{d\tau} (\int_0^\tau C^e_c \Delta \mathbf{v}^e_i dt) \]

(5.36)

\[ = \dot{C}^e_c \Delta \mathbf{r}^e + C^e_c \Delta \dot{\mathbf{r}}^e \]

Using Equation (3.24) the derivative of the transformation matrix can be given as

\[ \dot{C}^e_c = C^e_c (\omega^c_c \times) \]

For the transpose we have

\[ \dot{C}^e_c = (\dot{C}^e_c)^T = (\omega^c_c \times)^T (C^e_c)^T = - (\omega^c_c \times) C^e_c \]
Now Equation (5.36) can be modified to get the psi angle position error propagation model used for small and large errors, that is

$$\Delta \dot{r}^c = -\omega_\text{ce}^c \times \Delta r^c + \Delta v^c$$ \hspace{1cm} (5.37)

This can be presented in a matrix form of the linear position error propagation using (3.10), given as

$$\begin{bmatrix}
\Delta \dot{r}_x^c \\
\Delta \dot{r}_y^c \\
\Delta \dot{r}_z^c
\end{bmatrix} =
\begin{bmatrix}
0 & -\dot{l}\sin(\lambda) & \dot{\lambda} \\
\dot{l}\sin(\lambda) & 0 & \dot{l}\cos(\lambda) \\
-\dot{\lambda} & -\dot{l}\cos(\lambda) & 0
\end{bmatrix}
\begin{bmatrix}
\Delta r_x^c \\
\Delta r_y^c \\
\Delta r_z^c
\end{bmatrix} +
\begin{bmatrix}
\Delta \dot{v}_x^c \\
\Delta \dot{v}_y^c \\
\Delta \dot{v}_z^c
\end{bmatrix}$$ \hspace{1cm} (5.38)

5.1.5 The total linear error model for small angles

Total linear model can now be written to the system of propagating errors using equations (5.8), (5.13) and (5.38). The model is presented in detail when constrained navigation is discussed in chapter 6.

5.2 Phi angle error model

In the phi angle approach the navigation equations are solved in the true frame, which is the local level frame in the true position. The angle $\phi$ is the angle between the true frame and the platform frame as illustrated in Figure 5.1. Phi angle approach is also called as perturbation error analysis and true frame approach.

In this section first the velocity error model, then attitude error model and finally the position error model with phi angle approach are presented.

5.2.1 The velocity error model

The propagation equation for the velocity in the true frame according to (3.8) can be defined as

$$\dot{v}_i^t = f^t - (2\omega_{ie}^t + \omega_{ei}^t) \times v^t + g^t$$ \hspace{1cm} (5.39)

Here $f^t$ is the specific force vector defined as the non-gravity force vector per unit mass resolved in the true frame. As we saw in the previous section only $f_p + \Delta f_p$ is available exerted on the accelerometers and inertial navigation system computes velocity vector defined as

$$\dot{v}^t = f^p + \nabla^p - (2\omega_{ie}^t + \omega_{ei}^t) \times v^t + g^t$$ \hspace{1cm} (5.40)
where $f_p$ is the true specific force resolved in the platform frame and $\gamma^p$ is the accelerometer bias in the platform frame. Due to the existence of the error sources also the angular velocities are not available unbiased and therefore we have for the INS computed variables

\[
\begin{align*}
\hat{v}^t_t &= v^t_t + \Delta v^t_t \\
\hat{g}^t_t &= g^t_t + \Delta g^t_t \\
\hat{\omega}^t_{ic} &= \omega^t_{ic} + \Delta \omega^t_{ic} \\
\hat{\omega}^t_{et} &= \omega^t_{et} + \Delta \omega^t_{et}
\end{align*}
\] (5.41)

where $\Delta$ represents the deviation from the true value. Now the phi angle error model for velocity can be constructed by computing the velocity difference between true frame velocity and the INS computed velocity defined in (5.41) as

\[
\Delta \dot{v}^t = \dot{\hat{v}}^t - \dot{v}^c
= \left( (f^p + \gamma^p) + (2\hat{\omega}^t_{ic} + \hat{\omega}^t_{et}) \times \hat{v}^t \right) - \left( f^t + g^t - (2\omega^t_{ic} + \omega^t_{et}) \times v^t \right)
= \left( I - C^t_p \right) f^p - (2\omega^t_{ic} + \omega^t_{et}) \times \Delta v^t - (2\Delta \omega^t_{ic} + \Delta \omega^t_{et}) \times v^t + \gamma^p + \Delta g^t
\] (5.42)

The second order error product terms as $\Delta \omega \times \Delta v^t$ are ignored. By using the small angle assumption we can write the transformation matrix using the error angle as in psi angle model, now for the the error angle $\phi$ represented in Figure 5.1, given as

\[
C^p_t = C^t_p)^T = (I + \phi \times)^T = I - \phi \times
\] (5.43)

Using this assumption of small angles the velocity error model in phi angle approach can be given as

\[
\Delta \dot{v}^t = -\phi \times f^p - (2\omega^t_{ic} + \omega^t_{et}) \times \Delta v^t - (2\Delta \omega^t_{ic} + \Delta \omega^t_{et}) \times v^t + \gamma^p + \Delta g^t
\] (5.44)

### 5.2.2 The attitude error model

For phi angle approach only the error model with assumption of small error angles is developed. As it is shown above the angular velocity $\omega^t_{et}$ is not available from the gyros but only $\omega^t_{et} + \Delta \omega^t$. The navigation system assumes that the platform axes are coincident with the true frame and gyros control the platform along the platform axes. The gyros have also a drift rate $\epsilon$ as a result of different kinds of errors as bias, scale factor errors etc. For the angular velocity resolved in the platform frame with respect to the inertial frame, we have

\[
\omega^p_{ip} = \omega^t_{et} + \Delta \omega^t + \epsilon^p
\] (5.45)
Again this means that the angular velocity in the platform frame is equal to the gyro torquing plus the gyro drifts. Multiplying the equation above with the transition matrix rotating from the true frame to the platform frame as presented in (5.43), we have
\[
(I - \phi \times)\omega_p^p = C_t^p \omega_t^t + C_t^p \Delta \omega^t + (I - \phi \times)e^p
\]
(5.46)

By rearranging this equation we can solve the desirable angular velocity between the platform frame and the true frame as
\[
\omega_{ip}^t = \phi \times \omega_{ip}^p + \Delta \omega^p + e^p
\]
(5.47)

where \( \omega_{ip}^t = \omega_{ip}^p - \omega_{it}^p \). Also here the second order terms are neglected. With small angle assumption we assume that the resolved angular velocity represents the propagation of the phi angle, defined as \( \dot{\phi} \). The attitude error propagation model in phi angle approach can thus be given as
\[
\dot{\phi}^p = -\omega_{ip}^p \times \phi^p + \Delta \omega^p + e^p
\]
(5.48)

5.2.3 The position error model

The position at time \( \tau \) in the Earth frame is given without error as
\[
r_e^e = \int_0^\tau C_t^e \mathbf{v}^t dt
\]
(5.49)

Again the navigation system can not compute the velocity \( \mathbf{v}^t \) but instead computes \( \mathbf{v}^t + \Delta \mathbf{v}^t \). Similarly the navigation system can not produce the transformation matrix \( C_e^e \) but computes the transition matrix between the Earth frame and the computer frame according to the definition of the computer frame. The transition matrix between the Earth frame and the true frame can thus be given as
\[
C_t^e + \Delta C_t^e = C_c^e = C_t^e C_c^e = C_t^e (I + \Delta \theta \times)
\]
(5.50)

where \( \Delta \theta \) is the angle between the true frame and the navigation frame. For these reasons the position in the Earth frame has also error term and the system generates
\[
r_e^e + \Delta r_e^e = \int_0^\tau (C_t^e + C_t^e \Delta \theta \times)(\mathbf{v}^t + \Delta \mathbf{v}^t) dt
\]
(5.51)

Next the position error in the true frame can be resolved neglecting the second order terms, defined as
\[
\Delta r_{e}^t = C_c^e \int_0^\tau (C_t^e \Delta \mathbf{v}^t + C_c^e \Delta \theta \times \mathbf{v}^t) dt
\]
(5.52)
To get the position error propagation model the equation above is differentiated with the derivation rules for multiplication as

$$\Delta \dot{r}_t = \frac{d}{d\tau} \int_0^\tau (C_t^e \Delta v_t^i + C_t^e \Delta \theta \times v^i) dt$$

$$+ C_t^e \frac{d}{d\tau} (\int_0^\tau (C_t^e \Delta v_t^i + C_t^e \Delta \theta \times v^i) dt)$$

(5.53)

Again we use the derivative of the transformation matrix as given in Equation (3.27) the propagation model for position error can be established as

$$\Delta \dot{r}_t = -(\omega_t^i \times \Delta r_t^i) + \Delta v_t^i + \Delta \theta \times v_t^i$$

(5.54)

### 5.3 The error propagation of stationary INS

In this section we examine how the error of the inertial navigation system propagates when the navigation unit is assumed to be stationary. This is done in order to illustrate the sensitivity of the INS system to different kinds of errors and sensor biases and to see the behavior of the error propagation model. Here the psi angle error model with small angle assumption is examined. First we construct a reduced error propagation model for stationary system.

We can see that the error model introduced in Section 5.1 can be reduced when the system is assumed to be stationary. The derivatives of the latitude and longitude can be set to zero and also the acceleration components can be assumed to be zero when the system is not moving. When it is assumed that INS is operating at known altitude or the system has some external information to compensate the altitude errors the vertical loops can be eliminated. Including these reductions the psi angle error model presented in Equations (5.8), (5.13) and (5.38) can be given as

$$\frac{d}{dt} \begin{bmatrix} \delta r_N \\ \delta r_E \\ \ldots \\ \delta v_N \\ \delta v_E \\ \ldots \\ \delta \psi_N \\ \delta \psi_E \\ \delta \psi_D \end{bmatrix} = \begin{bmatrix} 0_{2\times2} & I_{2\times2} & 0_{2\times3} \\ A_{21} & A_{22} & A_{23} \\ 0_{3\times2} & 0_{3\times2} & A_{33} \end{bmatrix} \begin{bmatrix} \delta r_N \\ \delta r_E \\ \ldots \\ \delta v_N \\ \delta v_E \\ \ldots \\ \delta \psi_N \\ \delta \psi_E \\ \delta \psi_D \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \ldots \\ \delta f_N \\ \delta f_E \end{bmatrix}$$

(5.55)

where $\delta r$ is the position error, $\delta v$ is the velocity error and $\delta \psi$ is the attitude angle error. $\delta f$ represents the accelerometer bias and $\epsilon$ is the gyro drift.
CHAPTER 5. ERROR PROPAGATION MODELS

The matrices in (5.55) are defined as

\[
A_{21} = \begin{bmatrix}
  -g/R & 0 \\
  0 & -g/R
\end{bmatrix}
\]

\[
A_{22} = \begin{bmatrix}
  0 & -2\Omega \sin(\lambda) \\
  2\Omega \sin \lambda & 0
\end{bmatrix}
\]

\[
A_{23} = \begin{bmatrix}
  0 & g & 0 \\
  -g & 0 & 0
\end{bmatrix}
\]

\[
A_{33} = \begin{bmatrix}
  0 & -\Omega \sin(\lambda) & 0 \\
  \Omega \sin(\lambda) & 0 & \Omega \cos(\lambda) \\
  0 & -\Omega \cos(\lambda) & 0
\end{bmatrix}
\]

where \( g \) is the gravitational acceleration, \( R \) is the Earth radius and \( \lambda \) is the latitude and \( \Omega \) is the Earth rotation rate.

5.3.1 Stationary error model analysis

In this subsection the purpose is to examine the differential equation, defined in (5.55), which is the error propagation model with psi angle approach for stationary navigation unit. We illustrate the effect of different kinds of error sources to the total error and examine the performance of the system in presence of initial errors and biases.

First the differential equation (5.55) can be analyzed by computing the eigenvalues of the system. In this case one eigenvalue is zero and the rest can be proven to be purely imaginary. Zero as eigenvalue means that we have more than one equilibrium point to which the trajectory converges. On the other hand imaginary eigenvalues refer to the fact that the solution is stable but not asymptotically stable, as stated in [8]. It means that the solution is not moving towards an equilibrium solution and not away from it and can be written as a sum of trigonometric functions. In the solutions we can recognize both the affect of zero eigenvalue and the imaginary eigenvalue predicted oscillation examined below.

Because of the fact that the Earth is not flat we need to be tilting the navigation platform to keep the axes of the true frame north and east axes horizontal as we move on the surface. To compensate for the travel over the surface of the Earth, the platform must be rotated by an amount \( d/R \), where \( d \) is the distance travelled and \( R \) is the radius of curvature of the Earth. This can be considered as an equation of motion of a simple pendulum of length \( R \). It is named as a Schuler pendulum and the oscillation
is known as Schuler oscillation \[1\]. The period of the oscillation, the Schuler period, can be computed as

$$T_S = 2\pi \sqrt{\frac{R}{g}} \approx 84.4\text{min}$$

where \(g\) is magnitude of local gravitation. Let us now examine the oscillation established in error models with couple of examples.

### 5.3.2 Simulations

In examining the error propagation we consider some error source for the system and with the error propagation model we compute the resulted total state error as a function of time. Simulations were done with stationary strapdown INS, where the errors were calculated as a result of various sources.

First we examine the case where the attitude reference of the stationary vehicle is in error by a small initial tilt angle \(\psi\). This causes a horizontal accelerometer to sense acceleration \(-g\psi\). The measured acceleration causes the inertial navigation system to think that it is moving and computes a velocity and hence a change in position. This in turn torques the gyros in a direction so that the tilt reduces but when the accelerometer is level, the system has computed a considerable speed. This is like pulling a pendulum off centre and letting it swing.

In the Figure 5.2(a) the results of simulation of stationary INS effected by initial tilt error of 100 microradians on position error are shown. We may see that the error is oscillating at Schuler frequency with a period of around 84 minutes. The peak of the position error is around 1300 m and it is reached after approximately forty minutes of navigation. In longer navigation times not only Schuler but also Foucalt oscillation and 24-hours oscillation are noticeable as seen in \[27\] and Figure 5.3.

The other case is to examine system with a gyroscope drift error. This is another type of error which excites a Schuler oscillation. The effect of a gyro drift causes a tilt to build up in the platform, resulting in an oscillatory acceleration error. The velocity and position oscillate at the Schuler frequency. However, this time the velocity does not oscillate about a zero mean, so the position error is now a Schuler oscillation superimposed on a ramp function.

In the Figure 5.2(b) the total error in position of the simulated stationary INS resulted from tilt gyro drift of 0.01 deg/h is shown. The Schuler oscillation and ramp function can be noticed. After one hour of navigation the position error is about 1300 m. This is the error when navigating with navigation grade gyros. With the error propagation model we also calculated the total position errors resulted with gyro biases, relative to two other sensor grades defined in Table 4.1, additional to navigation grade. A gyro
(a) Total position error as a result of initial tilt error (100 microradians) follows Schuler periods.

(b) Total position error as a result of gyro bias (0.01 deg/h) oscillates at Schuler frequency superimposed on a ramp function.

Figure 5.2: Different error sources excite distinct total position error propagation oscillating at Schuler frequency.

bias of tactical grade gyro provides position error which is after ten minutes navigation around 1600 m. With consumer grade gyro the position error grows dramatically and after one minute it reaches 1500 m. It is understandable that using low-cost sensors, the navigation accuracy is poor and improvements are desirable.

Next we want to compare the effects of different kinds of error sources in stationary inertial navigation system. In Figure 5.3 the total errors are shown as a result of four various error sources. In the figure we can see the influence of both initial alignment errors as initial angle error and sensor errors as accelerometer and gyroscope biases. Also the total error of system effected by all these errors is illustrated in the figure.
We may also notice that excluding initial errors, the total error in the position domain is governed by gyro biases. The gyro bias is integrated once when computing the angle for the acceleration transformation which is then double integrated to get position error. We saw that with gyro bias of magnitude 100 deg/h the position error is after one minute of navigation around 150 m. The same error magnitude results of 8 mg accelerometer bias. In general when constructing INS, it is easier/cheaper to achieve required accelerometer accuracy and the gyro accuracy is limiting the system behavior.

As we can see in the simulations above, any lack of precision in the measurement used in inertial navigation system is passed from one estimate to the next with overall uncertainty in the precision of the calculated quantity, drifting with time [27]. In practice, the errors must be periodically reset, as stated in [19], using some kind of heading information.

One way to aid the inertial navigation system and reset the errors is to use external measurements which can bound navigation errors. External measurements are for example satellite navigation aids and radio navigation aids. Another way to aid the INS is to use on-board measurements as Doppler radar and magnetic sensors. Navigation solution can also be constrained to limit the error growth as discussed in [7]. Constraints are for example velocity constraint, speed constraint and acceleration constraint. Also so called nonholonomic constraint can be used to bound errors as presented in the next chapter.
Chapter 6

Constrained Inertial navigation systems

As we have seen in the previous section the error of the inertial navigation system grows unboundedly without any assumptions of the users motion, that is using free inertial navigation. In general INS is not able to provide accurate navigation for long time durations without proper external aiding. The error growth can be nevertheless limited if the navigation solution can be constrained. Inertial navigation constraints can for example be static constraints, discussed in previous section, where the velocity is constrained to zero when the vehicle is not moving with respect to the Earth. This makes it possible to handle sensor biases in the initial alignment process. Acceleration can be constrained by physical constraints and speed can be constrained due to fixed or measured velocity. Also travel can be constrained for example with odometer measuring the distance travelled. Moreover an interesting way to constrain the INS is to use land vehicle velocity constraints, assuming that the navigation unit should not have velocity in body frame directions perpendicular to the forward motion. This is known as nonholonomic constraint and is discussed in detail in next section.

6.1 Nonholonomic constraint

In the case of a land vehicle which is moving on the surface of the Earth the motion of the vehicle can be governed by two nonholonomic constraints. Under ideal conditions when the vehicle does not jump off the ground and does not slide on the ground, the velocities of the vehicle in the plane perpendicular to the forward direction are assumed to be zero. In practical situation these constraints are quite reasonable for normal vehicles in traffic, though violated due to for example the vibrations of the vehicle engine and suspension and also by side slip during cornering. In our case we assume that these nonholonomic constraints have zero mean behavior and that is why
we can approximate the vertical and horizontal velocities in body frame as

\[
\begin{align*}
v_b^y & \approx 0 \\
v_b^x & \approx 0
\end{align*}
\] (6.1)

The purpose is to use these constraints to give measurement knowledge for the system and then filter the estimate of error propagation variables. In this section we first discuss the approaches for the use of constraint information, then we construct the error model for body frame velocity as a function of error estimates and design a Kalman filter to combine the given measurements and knowledge of the system behavior. Finally we carry out some simulations to illustrate the functionality and usefulness of the constructed approach.

### 6.1.1 Indirect vs. direct approach

In designing a filtering platform for the inertial navigation system with the given knowledge from nonholonomic constraints, we have to consider whether we use the indirect or the direct approach, also referred to as the error state and the total state approaches. In the total state formulation the measurements are INS outputs from inertial sensors and external source signals. In the error state formulation the measurements presented to the filter are the differences between the INS computed and the external source data. It means that in the direct approach the total state is among the variables of the filter whereas in the indirect approach the errors are among the estimated variables.

There are a few reasons to use indirect approach specially in this problem. First of all the filtering algorithm for error state approach can be written with linear models as seen earlier in this work. For total state approach we have to write non-linear models to get a well-described system.

If the filter fails for some reason the direct navigation algorithm will fail whereas the indirect algorithm can produce estimates of position, velocity and attitude by working as an integrator on the INS data. On the other hand the direct filter is advantageous to the fault identification.

Because the indirect filter is out of the INS loop the external data can be produced at lower frequency than the inertial sensor data. Its sampling rate can be much lower than that of the direct filter. The indirect filter can also be easily modified for the systems where different amounts of measurements are available at a given time. For these reasons the error state formulation is used in essentially all terrestrial aided inertial navigation systems.


6.1.2 Body frame velocity error

Let us now look at the velocity in body frame in order to examine the behavior of the body frame velocity error. We assume that the body velocity is calculated from the navigation frame velocity given as $\dot{v}^b = \dot{C}^b_n \dot{v}^n$. It has a velocity error due to the velocity errors in navigation frame and also to the error in the transformation matrix. Therefore the body frame velocity with its error can be defined as

$$v^b + \delta v^b = C^b_n T(v^n + \delta v^n)$$ (6.2)

Matrix $T$ represents the small angle transformations of the error angles in transformation matrix. It can be written according to (2.7) as

$$T = \begin{bmatrix} 1 & -\delta \psi_3 & \delta \psi_2 \\ \delta \psi_3 & 1 & -\delta \psi_1 \\ -\delta \psi_2 & \delta \psi_1 & 1 \end{bmatrix}$$ (6.3)

where $\delta \psi$ is the angle error. Now we notice that this may also be expressed as

$$T = I + (\Psi^n \times)$$ (6.4)

where $\Psi$ represents the residual errors in the angles defining the rotations from the navigation frame to the body frame. Now we can resolve the velocity error in the body frame from (6.2) as

$$\delta v^b = C^b_n \delta v^n + C^b_n (\Psi^n \times) v^n$$ (6.5)

This can still be modified to get the linear equation for body frame velocity errors as function of error estimates, used in filtering process as a measurement model, that is

$$\delta v^b = C^b_n \delta v^n - C^b_n (v^n \times) \Psi^n$$ (6.6)

We have to take some assumptions into account in our approach. First we have to assume that the body frame or the frame of the navigation object is aligned with the sensor frame. We also assume that the origin of the sensor frame is in the same place as the axis of rotation. In the case where this assumption is not true the system is affected by so called lever arm effect. This could be the case in the car navigation when the navigation device is located in the front part of the car and the axis of rotation is in the center. More details of lever arm effect can be found for example in [14] and [9].
6.1.3 Kalman filter design

The task here is to solve an estimate for the state error at present time. We have an error propagation model, discussed previously in this work, to describe the dynamics of the system and virtual measurements of the body frame velocity error as a result of nonholonomic constraints. These are dependent on the present state. Now we want to determine the best estimates of the system variables given measurements and the knowledge of the system behavior [25].

This is the reason to use filtering. The idea of the filtering is to exploit all the measurements, also the ones that are detected earlier. Filtering has many advantages compared to static solving. For example with filtering it is possible to use all the measurements optimally regardless the amount of the measurements at each step. This is a great advantage specially when we do not have measurements at present time. That is why filtering is useful in a case where we have an inertial navigation system with aiding measurements from different sources e.g. GPS in addition to virtual measurements of nonholonomic constraint, odometer measurements, radar etc. Now the filter can solve the state optimally in presence of all measurements even with overdetermined system and also in the case of GPS or other measurement outages.

Filtering is also attractive from the computational point of view due to its recursive nature. This means that only the estimated state from the previous time step and the current measurement are needed to compute the estimate for the current state.

Kalman filter, given in [29], solves the filtering problem in the special case where models are linear and the initial condition and errors are normally distributed. This assumption is quite limiting but on the other hand Kalman filter also gives the best linear unbiased estimator (BLU-estimator, [2]) even in the case where the distributions are not normal. Due to this fact Kalman filter can be reasonably used without knowing the actual behavior of the errors.

The Kalman filter has two distinct phases: Predict and Update. The predict phase uses the state estimate from the previous timestep to produce an estimate of the state at the current timestep. In the update phase, measurement information at the current timestep is used to refine this prediction so that it minimizes the expected value of the square of the estimation error.

The next step is to construct a Kalman filter based on the virtual observations given by the nonholonomic constraints to estimate the errors of the inertial navigation system. Here we assume that no other measurements are present to illustrate the assistance of the nonholonomic constraints for the unaided INS. This is the case for example during GPS outage in integrated INS/GPS system. The same construction can be used also when there is more measurements with linear measurement models. First
we construct the motion and measurement models and after we present the Kalman filter algorithm.

**Motion model**

The state vector of the filter includes the errors in position, velocity and attitude given as
\[
\mathbf{x} = \begin{bmatrix} \delta r_N & \delta r_E & \delta r_D & \delta v_N & \delta v_E & \delta v_D & \delta \psi_N & \delta \psi_E & \delta \psi_D \end{bmatrix}^T \tag{6.7}
\]
As a motion model of the filter we use the error propagation model with psi angle approach defined as
\[
\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u} \tag{6.8}
\]
where the \(\mathbf{F}\) is the state transition matrix and \(\mathbf{G}\) is the noise gain matrix. The state noise vector \(\mathbf{u}\) including the sensor errors is given as
\[
\mathbf{u} = \begin{bmatrix} \delta f_N & \delta f_E & \delta f_N & \epsilon_E & \epsilon_N & \epsilon_E \end{bmatrix}^T
\]
The transition matrices for the motion model are defined in equations (5.8), (5.13) and (5.38) as
\[
\begin{pmatrix}
F_{11} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\
F_{21} & F_{22} & F_{23} \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & F_{33}
\end{pmatrix}
\]
and
\[
\mathbf{G} = \begin{bmatrix} \mathbf{0}_{3\times6} \\
\mathbf{I}_{6\times6} \end{bmatrix}
\]
where the block matrices of \(\mathbf{F}\) are given as
\[
F_{11} = \begin{bmatrix} 0 & -\dot{\phi}\sin\lambda & \dot{\lambda} \\ \dot{\phi}\sin\lambda & 0 & \dot{\phi}\cos\lambda \\ -\dot{\lambda} & -\dot{\phi}\cos\lambda & 0 \end{bmatrix}
\]
\[
F_{21} = \begin{bmatrix} -g/R & 0 & 0 \\ 0 & -g/R & 0 \\ 0 & 0 & -g/R \end{bmatrix}
\]
\[
F_{22} = \begin{bmatrix} 0 & -2(\Omega + \dot{\phi})\sin\lambda & \dot{\lambda} \\ 2(\Omega + \dot{\phi})\sin\lambda & 0 & 2(\Omega + \dot{\phi})\cos\lambda \\ -\dot{\lambda} & -2(\Omega + \dot{\phi})\cos\lambda & 0 \end{bmatrix}
\]
\[
F_{23} = \begin{bmatrix} 0 & -f_D & f_E \\ f_D & 0 & -f_N \\ -f_E & f_N & 0 \end{bmatrix}
\]
\[
F_{33} = \begin{bmatrix}
0 & -\left(\Omega + \dot{\phi}\right) \sin \lambda & \dot{\lambda} \\
\left(\Omega + \dot{\phi}\right) \sin \lambda & 0 & \left(\Omega + \dot{\phi}\right) \cos \lambda \\
-\dot{\lambda} & -\left(\Omega + \dot{\phi}\right) \cos \lambda & 0 
\end{bmatrix}
\]

The model above is developed in continuous time. We convert it to a discrete motion model using first order Euler method as follows. Let \( dt \) be the interval between two sampling times \( t_k \) and \( t_{k+1} \), we have

\[
\dot{x}(t_k) \approx \frac{x(t_{k+1}) - x(t_k)}{dt} \approx F(t_k)x(t_k) + G(t_k)u(t_k)
\]

Therefore the discrete motion model equation can be given as

\[
x(t_{k+1}) = x(t_k) + dtF(t_k)x(t_k) + dtG(t_k)u(t_k)
\]

For convenience, let \( x(t_k) = x_k \) and \( x(t_k) = u_k \). The process model used in filtering is now defined as

\[
x_{k+1} = F_k x_k + G_k u_k
\]

where \( F_k = I + dtF(t_k) \) and \( G_k = dtG(t_k) \).

**Measurement model**

As measurements of the filter we use the velocity errors in z- and y-directions in the body frame

\[
z_k = \begin{bmatrix} \delta v^b_y \\ \delta v^b_z \end{bmatrix}
\]

We assumed that the true velocity in the body frame perpendicular to the forward motion is zero and we compare it with the body frame velocity computed by the inertial navigation system. That is why the velocity observed by the INS is assumed as velocity error. We get the velocity error measurements as

\[
z_k = \begin{bmatrix} v^b_{y,INS} - v^b_{y,TRUE} \\ v^b_{z,INS} - v^b_{z,TRUE} \end{bmatrix} = \begin{bmatrix} v^b_{y,INS} \\ v^b_{z,INS} \end{bmatrix}
\]

The measurement model can now be constructed from Equation (6.6), defined for velocities in y- and z-directions, given as

\[
z_k = H_k x_k + v_k
\]

where the design matrix of the measurement model is defined as

\[
H_k = \begin{bmatrix} 0_{2 \times 3} & H_{21,k} & H_{31,k} \end{bmatrix}
\]
The block matrices of $H$ are defined from (6.6) as

$$H_{21} = \begin{bmatrix} c_{12} & c_{22} & c_{32} \\ c_{13} & c_{23} & c_{33} \end{bmatrix}$$

$$H_{31} = \begin{bmatrix} -v_Dc_{22} + v_Ec_{32} & v_DC_{22} - v_NC_{32} & -v_Ec_{12} + v_NC_{22} \\ -v_Dc_{23} + v_Ec_{33} & v_DC_{23} - v_NC_{33} & v_Ec_{13} - v_NC_{23} \end{bmatrix}$$

where $c$ is the element of transformation matrix $C^b_n$

**Kalman filter algorithm**

Finally we can write the Kalman filter algorithm according to [5]. First the prior estimate is computed with the knowledge of the previous state estimate as

$$\bar{x}_k = F_{k-1}\hat{x}_{k-1}$$

The prior covariance is given as

$$\bar{P}_k = F_{k-1}\hat{P}_{k-1}F_{k-1}^T + G_kQ_kG_k^T$$

The Kalman gain matrix is defined as

$$K_k = \bar{P}_kH_k^T(H_k\bar{P}_kH_k^T + R_k)^{-1}$$

And finally the state/posterior estimate and the posterior covariance are given as

$$\hat{x}_k = \bar{x}_k + K_k(z_k - H_k\bar{x}_k)$$

$$\hat{P}_k = (I - K_kH_k)\bar{P}_k$$

### 6.1.4 Simulations

In this section we have simulated the use of nonholonomic constraints as the inertial navigation aiding information. Different grades of inertial sensors are used and compared in the simulations. Simulations were done by constructing sensor measurements and computing the true route from these measurements. The true route is computed by setting all the errors to zeros. Next we added biases, scale factor errors and stochastic noise to the measurements to get measurements applied from corrupted inertial sensors. After that we computed estimates of the state with unaided INS integrator and with nonholonomically constrained INS. The results with different routes, velocities and magnitudes of sensor errors are described in this section.
First we examine how the violations of nonholonomic constraints result in errors in position. In Figure 6.1(a) the INS calculated body frame velocities in y- and z-directions are plotted in the case of corrupted sensor measurements. We may see that INS detects velocity in these directions even when dealing with land vehicles assumed not to have side slips or jumps. In the Figure 6.1(b) we can see how the 3-D position error develops when unaided INS, that is INS integrator described in Section 3.2, is perturbed by errors which result in these velocities. INS using given body frame velocities as measurements of the error in the Kalman filter based on nonholonomic constraints, can reduce position error dramatically to fifth of the unaided one.

(a) INS integrator calculates body frame velocities violating the assumed land vehicle constraints.

(b) When the body frame velocities are used as measurements of error in the filter, total error can be reduced considerably.

Figure 6.1: INS calculated body frame velocities gives knowledge of error to reduce the total error with filtering.

The algorithm was tested further using so called Monte Carlo (MC) simulation method. This means stochastic techniques using random numbers and probability statistics to investigate the problem. The process of Monte Carlo simulation can be considered as
follows. We have a parametric model for the system and we generate a set of random inputs. In our case it means that we add normal noise to the errorless true route sensor measurements. From these inputs we evaluate the output with both the unaided INS and the nonholonomically constrained INS model. This construction is then repeated for certain amount of random input sets. The results are analyzed first by calculating mean values of the residuals as shown in the Figure 6.2. Then we define the average distances and the standard deviations at given time tags from the true route illustrated in Figure 6.3. Finally the percentages of estimates which are within given distance from the true route are calculated for consumer grade INSs.

In Figure 6.2 on the left the mean position errors between the true track and the estimated track, both for the unaided INS and the INS using nonholonomic assumption, are illustrated and compared.

Figure 6.2: On the left the mean position errors of 50 set of measurements for three sensor grades are calculated. On the right an average example of the estimated routes in relation to true route are illustrated. Though the error magnitudes between different sensor grades are distinct, the percentage improvements achieved with nonholonomic filter are roughly the same.

The given values were calculated as mean of 50 measurement sets which were corrupted with bias, scale factor error and stochastic noise. The measurements were detected
once per second. Sensor measurements were simulated according to different sensor grade accuracies having approximately the noise levels given in Table 4.1. In this simulation we traveled with constant speed along the track. Also a representative examples of the estimated routes in relation to the true route are plotted for all sensor grades.

The results illustrate the improvements achieved with nonholonomic filter using sensor measurements of different grades. In the filter we used quite small measurement variances compared to the state model error variance. Noticeable is that the percentage improvement compared with unaided and constrained INS is nearly the same between the grades, though the error magnitudes are of different size range. Roughly we can say that the navigation time can be doubled to get the same position accuracy when sensors are replaced with ones from more accurate sensor grade.

In Figure 6.3 we can see the simulated navigation unit trajectory used in following experiment. The navigation unit is traveling at speed of 10m/s for five minutes and measurements are obtained ten times per second. The estimated routes shown in the Figure are calculated with sensors simulated as tactical grade ones. We may notice that the shape of the route is followed by both INS approaches, but prominent improvements are achieved with aided INS.

![Figure 6.3: The simulated navigation trajectory and the estimated trajectories with tactical grade sensors both aided and unaided. Time tags are $t_1$, $t_{2.5}$ and $t_5$.](image-url)
In Table 6.1 the mean error distances and the standard deviations of three sensor grades at given time tags are calculated. This was done as the mean of 200 simulated corrupted sets of sensor measurements for the track shown in the Figure 6.3. It can be seen that with aided INS not only the mean error distances are remarkably smaller but also great improvements are achieved with the standard deviation levels.

Table 6.1: Mean errors and standard deviations

<table>
<thead>
<tr>
<th>Sensor grade</th>
<th>time(min)</th>
<th>( \bar{x} )(m)</th>
<th>( \sigma )(m)</th>
<th>( \bar{x} )(m)</th>
<th>( \sigma )(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5</td>
<td>197.1</td>
<td>7.4</td>
<td>63.4</td>
<td>3.7</td>
</tr>
<tr>
<td>Consumer</td>
<td>1</td>
<td>713.7</td>
<td>20.9</td>
<td>82.9</td>
<td>7.8</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4049.2</td>
<td>89.9</td>
<td>780.3</td>
<td>39.8</td>
</tr>
<tr>
<td>Tactical</td>
<td>1</td>
<td>42.9</td>
<td>15.2</td>
<td>5.8</td>
<td>2.7</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>218.3</td>
<td>63.7</td>
<td>31.2</td>
<td>4.8</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>670.2</td>
<td>177.6</td>
<td>88.3</td>
<td>9.0</td>
</tr>
<tr>
<td>Navigation</td>
<td>2.5</td>
<td>17.9</td>
<td>8.0</td>
<td>4.6</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>206.5</td>
<td>85.1</td>
<td>92.9</td>
<td>5.5</td>
</tr>
</tbody>
</table>

Table 6.2 shows the results of simulation where we traveled 30 seconds along track of two 90 degree turns. At the beginning we accelerate and at the end we brake. The simulation was done with consumer grade sensors only. In the table we can see the percentages of the estimates within given error distances at time tags \( t_{15} \) and \( t_{30} \). The portions were calculated from 500 sets of measurements. The aided INS can keep within 50 meters from the true route major portions of cases while unaided INS loses its accuracy after 15 seconds of traveling.

Table 6.2: Percentages of estimates within given distance from the true route.

<table>
<thead>
<tr>
<th>INS</th>
<th>approach error(m) ≤</th>
<th>15s</th>
<th>30s</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>50</td>
<td>38.0</td>
<td>0.4</td>
</tr>
<tr>
<td>unaided</td>
<td>80</td>
<td>92.6</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td>120</td>
<td>100.0</td>
<td>9.8</td>
</tr>
<tr>
<td>NHC</td>
<td>10</td>
<td>20.8</td>
<td>14.0</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>98.8</td>
<td>72.0</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>100.0</td>
<td>96.6</td>
</tr>
</tbody>
</table>
Chapter 7

Conclusions and future work

In this thesis we discussed self-contained inertial navigation systems needing external aiding to overcome the lack of navigation accuracy. The information of nonholonomic constraints was integrated with INS by indirect error state approach using Kalman filter and the theoretical background for the use of land vehicle constraints as external aids for INS was constructed. In addition we compared the performance of nonholonomically aided INS and unaided INS in various simulations. This was done to see if these constraints provide any advantages to use in inertial navigation. Different tracks and grades of sensors were used to see how the systems behaves on average in Monte Carlo simulations.

The tests show that when an inertial navigation system is considered using nonholonomic constraints we can improve the behavior and navigation accuracy compared to INS working unaided regardless the track or sensor grades used. We can see that in the presence of sensor errors, INS senses imaginary velocity perpendicular to forward motion in body frame. This velocity leads to error in the state estimates of attitude, velocity and finally position. When this imaginary velocity is considered as measurement of velocity error and submitted to the Kalman filter constructed with error state approach the total error in state estimates is dramatically decreased.

We also compared the behavior when sensor measurements were simulated from different accuracy grades. The three sensor grades resulted different magnitude of error but the improvement achieved with land vehicle constraints, compared with the corresponding unaided error, is more or less of the same amount between the grades. Not only the error means but also the standard deviations values were decreased using the indirect filter. Concerning navigation with low-cost sensors, e.g. car navigation, we noticed that the improvements were significant already within short travel times such as possible GPS outage times.

The promising results in simulations point out that further studies in constrained INSs should be carried out. This work provides a theoretical implementation for the
use of constraints and the model developed with indirect error state approach can be easily modified between systems using different kinds of external aids. In the future it would be interesting to examine different kinds of constraints used to aid the INS. For example the use of an odometer to constrain the distance travelled could be worth of studying. It would be interesting to compare along-track and cross-track errors in this case.

Future work should also include tests with real data. Using integrated INS/GPS the INS behavior during GPS outage or outliers could be examined. Moreover, proper examination of the effect of the trajectory and observability analysis is left for future study.
Bibliography


