A Low-Cost GPS Aided Inertial Navigation System for Vehicular Applications

ISAAC SKOG

Master of Science Thesis
Stockholm, Sweden 2005-03-09

IR-SB-EX-0506
Abstract

In this report an approach for integration between GPS and inertial navigation systems (INS) is described. The continuous-time navigation and error equations for an earth-centered earth-fixed INS system are derived. Using zero order hold sampling, the set of equations is discretized. An extended Kalman filter for closed loop integration between the GPS and INS is derived. The filter propagates and estimates the error states, which are fed back to the INS for correction of the internal navigation states. The integration algorithm is implemented on a host PC, which receives the GPS and inertial measurements via the serial port from a tailor made hardware platform, which is briefly discussed. Simulation results of the system are presented.
# Contents

1 Introduction
   1.1 Background .............................................. 4
   1.2 System Overview ................................... 5
   1.3 Thesis Outline and Contributes ..................... 6

2 Coordinate Systems ........................................... 8
   2.1 Coordinate Frame Definition ........................... 8
      2.1.1 Earth-centered earth-fixed frame (ECEF, e-frame) .... 8
      2.1.2 Local geodetic frame (t-frame) .................... 8
      2.1.3 Inertial frame (i-frame) ......................... 10
      2.1.4 Body frame (b-frame) ............................ 10
   2.2 Coordinate Frame Transformation ...................... 10
      2.2.1 Projection ...................................... 10
   2.3 Plane Rotations ....................................... 12
      2.3.1 Introduction ..................................... 12
      2.3.2 Rotation matrices ................................ 14
      2.3.3 Small angle rotation ............................. 14
   2.4 Rotating Coordinate Frame .............................. 16

3 Inertial Navigation Equation ............................... 18
   3.1 Fundamental Equations of Inertial Navigation ........... 18
      3.1.1 Gravity force .................................... 18
      3.1.2 Inertial force .................................... 19
      3.1.3 Equations of motion ............................. 19
   3.2 Navigation Equations .................................. 20
      3.2.1 General Navigation Equations .................... 20
      3.2.2 ECEF-frame navigation equations ................. 21
   3.3 ECEF Navigation Error Model .......................... 22
      3.3.1 ECEF navigation error equations ................ 22

4 Discretization ..................................................... 27
   4.1 Discretization ......................................... 27
      4.1.1 Discrete time navigation equations ............... 27
      4.1.2 Discrete time error equations .................... 29

5 Integration of GPS and INS ..................................... 30
   5.1 The Extend Kalman Filter .............................. 30
## CONTENTS

6 Hardware 33
   6.1 The Inertial Measurement Unit 33
   6.2 Data Acquisition 33

7 Simulation Results 35
   7.1 Driving Scenario 35

8 Conclusions and Further Work 39
   8.1 Conclusions 39
   8.2 Further Work 39

A The Kalman Filter 42
   A.1 Derivation of the Kalman Filter Equations 42
Chapter 1

Introduction

1.1 Background

Today many vehicles are equipped with global positioning system (GPS) receivers that constantly can provide the driver with information about the vehicle’s position with an accuracy in the order of 15-100 meters [1]. However, the GPS receiver has two major weaknesses. The slow update rate, only once a second for most receivers, and the sensitivity to blocking of the satellite signals. This can cause problems when for example plotting a vehicle’s movements on a map.

Imagine a car that is travelling at high speed and suddenly makes a fast turn around a corner of a house. If the position of the car is only determined once a second, and out of this position estimates of the car’s movement is plotted, it most likely will look as the car drove through the corner of the house. After the turn the car drives into a tunnel and all of a sudden the position of the car can’t be determined at all.

Inertial navigation systems (INS) can provide position, velocity and attitude estimates at a high rate, typically 100 times per second and are, to the opposite of the GPS receiver, self contained [2]. Instead it relies on Newton’s laws of motion, from which it can be concluded that if an object’s initial position, velocity and attitude are known all further positions, velocities and attitudes can be determined by integrations of the accelerations and angular rates of the object. However, due to the integrative nature of the INS, low frequency noise and sensor biases are amplified. The unaided INS may therefore have unbounded position, velocity and attitude errors [3]. These complementary properties make an integration of the two systems suitable. Indeed it has been shown that by combining the short time accuracy and high update rate of the INS with the long term accuracy of the GPS, it is possible to obtain a navigation system with not only high update rate but also good accuracy [2].

Until recently inertial measurement units (IMU) have been too expensive for private persons. But lately some low cost micro electro mechanical systems (MEMS) gyros and accelerometers have been introduced on the market. This has opened the door for development of low cost inertial navigation systems, suitable for integration with GPS-receivers.
1.2 System Overview

The purpose of the system described in this thesis is to combine GPS and INS data in an optimal way to obtain a navigation system with both higher update rate and smaller position error than the stand alone GPS-receiver. In Figure 1.1 a system overview with its different input and output signals is shown. A common integration method is to let the INS be the major navigation system and use the position estimates of the GPS-receiver to estimate and correct the errors in the INS. The integration between INS and GPS is refereed to as a GPS aided INS.

Before integrating the two systems the provided data must be transformed into a common coordinate system, suitable for integration. The GPS-receiver provides position estimates in the earth-centered earth-fixed (ECEF) coordinate frame. This is a cartesian coordinate system that rotates with earth and has its origin at the earth center of mass. In the case of the INS it is more complicated. The accelerations and angular rates are measured with respect to the inertial frame of reference and resolved in the instrumental frame of the accelerometers and the gyros, respectively. An inertial frame is a coordinate frame in which Newton’s laws of motion applies, that is in rest or in linear motion. From the instrumental frame the measurements must be transformed to the platform-frame, which is the coordinate system associated with the platform onto which the accelerometers and gyros are mounted. The platform is refereed to as the inertial measurement unit (IMU). Finally the angular rates, resolved in platform coordinates are used to transform the accelerations into ECEF-coordinates, where they are processed and integrated with the GPS data.

There are several approaches on how to design a GPS aided INS: tightly coupled or loosely coupled, closed or open loop, pseudorange or position aided. In this thesis a loosely coupled position aided method is proposed. Essentially this means that GPS and INS works independently and that there is no direct data exchange between them, i.e loosely coupled, and that GPS-receiver position estimates and not the estimated distances to the satellites are used (pseudoranges). The loosely coupled approach allows the use of a low cost of-the-shelf GPS receiver. The difference between the GPS and INS position estimate is used to drive an extended Kalman filter, housing a model of how the errors in the INS develops with time. The error states estimated by the filter are fed back to the INS for calibration of its internal states, resulting in a closed loop design; See Figure 1.2. A comparison between the tightly and loosely coupled approach can be found in [4].
1.3 Thesis Outline and Contributes

The thesis is divided into 8 chapters:

- Chapter 2 gives an introduction to the coordinate frames used throughout this thesis.
- Chapter 3 gives a short introduction to the basic physics behind inertial navigation. From this the general navigation equations are derived. Further the ECEF navigation dynamics are presented.
- Chapter 4 describes the discretization of the continuous-time ECEF navigation equations and the corresponding error model, using zero-order-hold sampling.
- Chapter 5 gives the mathematical background for the integration between the GPS and INS, using an extended Kalman filter.
- Chapter 6 gives a short description of the tailor made hardware that has been developed within this project.
- Chapter 7 presents simulations results of the integrated navigation system for a typical driving scenario.
- Chapter 8 presents conclusions and future work.

The main contributions in this thesis are:

- Presentation of discrete ECEF navigation equations suitable for implementation in an ECEF inertial navigation system.
- Presentation of a discrete state-space model for the error dynamics of an ECEF inertial navigation system.
- Presentation of an explicit extended Kalman filter algorithm for a GPS aided INS.
- Development of the hardware described in Chapter 6.
Within the field of integration between INS and GPS data using Kalman filter techniques a lot of work has been done and the author would like to highlight two of them, which are closely related to the content of this report. First, the conference paper [1] which describes an integration method where the Kalman filter propagates the navigation stages and not the errors. Secondly, the technical report [9] in which a closed loop, pseudo range aided approach for integration between GPS and INS data is evaluated.
Chapter 2

Coordinate Systems

In inertial navigation systems the measured quantities is obtained in different coordinate frames and need to be transformed into a coordinate frame suitable for processing. A basic inertial navigation system involves at least four different frames. The accelerometers measures the platform accelerations with respect to the inertial frame of reference. The accelerations are resolved in the instrumental frame of the accelerometers and need to be transformed into platform coordinate frame by a fixed rotation matrix. Gyros measure the platform angular rates relative to the inertial frame of reference and resolves this in the instrumental frame of the gyros, which are transformed into angular rates in the platform frame by a fixed rotation matrix. From the gyro measurements a rotation matrix is calculated which transform the accelerations in the platform-frame into the used navigation frame, where they are processed to determine the velocity and position of the navigation system.

2.1 Coordinate Frame Definition

This section defines the different coordinate frames and corresponding notation used in this thesis. All coordinate systems are orthogonal Cartesian systems if not otherwise stated.

2.1.1 Earth-centered earth-fixed frame (ECEF, $e$-frame)

This coordinate system has its origin at the center of the earth and rotates with the earth. The axes directions are defined as follows: the $x$-axis points towards the intersection between the prime meridian and the equator, the $z$-axis points in the direction of the mean polar axis and the $y$-axis completes the right hand coordinate; See Figure 2.1. The ECEF frame is denoted by the superscript $e$.

2.1.2 Local geodetic frame ($t$-frame)

This is the coordinate system often referred to in our daily life as the north, east and down direction. It will be denoted by the superscript $t$. It’s determined by fitting at tangent plane to a fixed point of the geodetic reference ellipse. This point will be the origin of the coordinate system and the $x$-axis points towards
Figure 2.1: Relations between ECEF(e-frame)-, local geodetic(t-frame)- and inertial(i-frame)-frame.
the true north, the $y$-axis to the east and the $z$-axis completes the right hand coordinate system by pointing towards the earth’s interior.

### 2.1.3 Inertial frame ($i$-frame)

An inertial frame is a coordinate frame in which Newton’s laws of motions apply. Therefore the inertial frame is not accelerating, but can be in a linear motion. The origin of the inertial coordinate system is arbitrary, and the coordinate axis may point in any three perpendicular directions. It’s often preferable to let the origin of the inertial frame consist with the earth center of mass. This frame will be referred to as the $i$-frame and will be denoted by the superscript $i$ and should not be confused with the ideal inertial frame described above since the gravity force applies. As will be shown in the sequel (Section 2.2.1) a vector $p$ in the $i$-frame is linearly related (by a rotation matrix) to a vector in the $e$-frame, and vice versa.

### 2.1.4 Body frame ($b$-frame)

The body frame is the coordinate system associated with the vehicle. The coordinate system has its origin at the center of gravity of the vehicle and the $x$-axis points in the forward direction, the $z$-axis down through the vehicle and the $y$-axis completes the right hand coordinate system. See Figure 2.2. This frame will be denoted by the superscript $b$. In some applications the instrumentation platform is not aligned with the body frame, and thus a (orthonormal) platform frame is needed as well. The platform frame is the coordinate system of the platform which the accelerometers and the gyros are mounted on. However throughout this thesis the platform is assumed to have its coordinate axes perfectly aligned with the body coordinate axes, and therefore no distinction between the two frames will be made.

The inertial sensors resolves their measurements along the sensitivity axes of the instruments. The instrument sensitivity axes of the gyros and accelerometers spans the axes of the gyro and accelerometer instrumental frames, respectively. The instrumental frames can not be assumed to be orthogonal and identification of the misalignment matrices for transformation between the instrumental and platform frame is the primary objective of the IMU alignment, see [6].

### 2.2 Coordinate Frame Transformation

In navigation systems it’s often necessary to transform a vector from one coordinate system into another. This section describes two different methods to derive a mathematical expression for the rotation matrix relating two Cartesian, orthogonal coordinate systems. Further, small angle rotations and derivatives of rotation matrices are studied.

#### 2.2.1 Projection

The first method presented on how to find a mathematically expression for the rotation matrix relating to coordinate systems is based upon projections of a vector onto the orthonormal bases of the two coordinate systems. Mathematical
Figure 2.2: Body coordinate frame.

Figure 2.3: By projection of vector $p$ onto the bases of the two orthonormal coordinate systems the directional cosine matrix relating the two coordinate systems can be found.
2.3 Plane Rotations

this can be described as follows. Let: \{\mathbf{i}_a, \mathbf{j}_a, \mathbf{k}_a\} and \{\mathbf{i}_b, \mathbf{j}_b, \mathbf{k}_b\} be the unit vectors spanning two orthonormal coordinate systems having the same origin, see Figure 2.3. Vector \mathbf{p} can then be written in terms of the unit vectors spanning coordinate system \(A\), that is

\[
\mathbf{p} = x_a \mathbf{i}_a + y_a \mathbf{j}_a + z_a \mathbf{k}_a
\]

(2.1)

Projection of \(\mathbf{p}\) onto the unit vectors spanning coordinate system \(B\) and using the fact that the bases of the coordinate system are orthogonal results in

\[
\begin{align*}
  x_b &= \langle \mathbf{i}_b, \mathbf{p} \rangle = \langle \mathbf{i}_a, \mathbf{i}_b \rangle x_a + \langle \mathbf{j}_a, \mathbf{i}_b \rangle y_a + \langle \mathbf{k}_a, \mathbf{i}_b \rangle z_a \\
  y_b &= \langle \mathbf{j}_b, \mathbf{p} \rangle = \langle \mathbf{i}_a, \mathbf{j}_b \rangle x_a + \langle \mathbf{j}_a, \mathbf{j}_b \rangle y_a + \langle \mathbf{k}_a, \mathbf{j}_b \rangle z_a \\
  z_b &= \langle \mathbf{k}_b, \mathbf{p} \rangle = \langle \mathbf{i}_a, \mathbf{k}_b \rangle x_a + \langle \mathbf{j}_a, \mathbf{k}_b \rangle y_a + \langle \mathbf{k}_a, \mathbf{k}_b \rangle z_a
\end{align*}
\]

where scalar product \(\langle \mathbf{i}_b, \mathbf{j}_b \rangle : = \cos(\alpha_{\mathbf{i}_b, \mathbf{j}_b})\). Here \(\alpha_{\mathbf{i}_b, \mathbf{j}_b}\) denotes the angle between vector \(\mathbf{i}_b\) and \(\mathbf{j}_b\).

In matrix form this can be written as

\[
\mathbf{p}^b = \mathbf{R}_a^b \mathbf{p}^a
\]

(2.2)

where

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

(2.3)

The rotation matrix \(\mathbf{R}_a^b\) is a so called directional cosine matrix. Although \(\mathbf{R}_a^b\) has nine elements, it has only three degrees of freedom and can be uniquely described by the three Euler angles, in the sequel gathered in the vector \(\theta\) [5].

The directional cosine matrix is an orthonormal matrix, that is \((\mathbf{R}_a^b)^T \mathbf{R}_a^b = \mathbf{I}\). Hence, the inverse of the rotation matrix is the same as its transpose.

2.3 Plane Rotations

2.3.1 Introduction

Plane rotations are a convenient way to mathematically express the vector transformation between two coordinate frames with common origin, where the second coordinate system can be related to the first by a rotation around a vector \(\mathbf{v}\). In the case of \(\mathbf{v}\) being one of the coordinate axes the rotation matrix takes a simplified form. The rotation matrix between two coordinate systems related by a sequences of plane rotations can be found by multiplying the plane rotation matrices in the order of the rotations.

By plane rotations the directional cosine matrix relating coordinate system \(A\) and \(B\) in Figure 2.4 can be found in the following manner. First rotating coordinate system \(A\) around its \(z\)-axis with \(\alpha_1\) radians to align the new \(x'\)-axis with the projection of \(x_b\)-axis onto the plane spanned by \(x'\)-axis and \(y'\)-axis. See Figure 2.5(a) and 2.5(d). The rotation can be described as

\[
\begin{bmatrix}
  x' \\
  y' \\
  z'
\end{bmatrix} = \begin{bmatrix}
  \cos(\alpha_1) & \sin(\alpha_1) & 0 \\
  -\sin(\alpha_1) & \cos(\alpha_1) & 0 \\
  0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  x_a \\
  y_a \\
  z_a
\end{bmatrix}
\]

(2.4)
Figure 2.4: Coordinate system $A$ and $B$ referred to in the derivation of the rotation matrix $R^b_a$ through plane rotations.
Next, rotating the new coordinate system, denoted with prime with $\alpha_2$ radians around the $y''$-axis, resulting in that the $x''$-axis becomes aligned with the $x_b$-axis. This is illustrated in Figure 2.5(b) and 2.5(e). The rotation is described by

$$
\begin{bmatrix}
x'' \\
y'' \\
z''
\end{bmatrix} =
\begin{bmatrix}
\cos(\alpha_2) & 0 & -\sin(\alpha_2) \\
0 & 1 & 0 \\
\sin(\alpha_2) & 0 & \cos(\alpha_2)
\end{bmatrix}
\begin{bmatrix}
x' \\
y' \\
z'
\end{bmatrix}
$$

(2.5)

Then rotating the new coordinate system $\alpha_3$ radians around the $x''$-axis results in that the $y''$- and $z''$-axis aligns the $y_b$- and $z_b$-axis, respectively. See Figures 2.5(c) and 2.5(f). The last plane rotation is described by

$$
\begin{bmatrix}
x_b \\
y_b \\
z_b
\end{bmatrix} =
\begin{bmatrix}
x''' \\
y''' \\
z'''
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 \\
0 & \cos(\alpha_3) & \sin(\alpha_3) \\
0 & -\sin(\alpha_3) & \cos(\alpha_3)
\end{bmatrix}
\begin{bmatrix}
x'' \\
y'' \\
z''
\end{bmatrix}
$$

(2.6)

The rotation matrix from coordinate system A to B is found by multiplying the rotation matrices in equations (2.4)–(2.6) in the order of rotations. The rotation matrix $R^b_a$ reads

$$
R^b_a =
\begin{bmatrix}
c(\alpha_1)c(\alpha_2) & s(\alpha_1)c(\alpha_2) & -s(\alpha_2) \\
-s(\alpha_1)c(\alpha_3) + c(\alpha_1)s(\alpha_2)s(\alpha_3) & c(\alpha_1)c(\alpha_3) + s(\alpha_1)s(\alpha_2)s(\alpha_3) & c(\alpha_2)s(\alpha_3) \\
s(\alpha_1)s(\alpha_3) + c(\alpha_1)s(\alpha_2)c(\alpha_3) & -c(\alpha_1)s(\alpha_3) + s(\alpha_1)s(\alpha_2)c(\alpha_3) & c(\alpha_2)c(\alpha_3)
\end{bmatrix}
$$

(2.7)

Here $s(\cdot)$ and $c(\cdot)$ denotes the sine and cosine operation, respectively. The rotation matrices used throughout this thesis can be found below.

### 2.3.2 Rotation matrices

It is clear by now that coordinate rotations are essential parts of inertial navigation. Throughout this report the most extensively used coordinate rotation matrix is the ECEF to body frame rotation matrix $R^b_e$. It is possible to find an expression for the rotation matrix $R^b_e$ directly in terms of the three Euler angles relating the ECEF and body coordinate systems. However, a more convenient method is to divide the rotation matrix $R^b_e$ into two coordinate rotations as $R^b_e = R^g_e R^b_g$. That is, firstly a coordinate rotation from the ECEF frame to the local geodetic frame and secondly a coordinate rotation from the local geodetic frame to the body frame, described by $R^g_e$ respectively $R^b_g$. The definitions of the local geodetic to body frame and the ECEF to local geodetic frame rotation matrices are found in Table (2.1) and (2.2), respectively.

### 2.3.3 Small angle rotation

When two coordinate systems differ only by small angles rotations the corresponding directional cosine matrix can be approximated being a linear function of the rotation angles. This is a very useful property in the derivation of the INS error equations and derivative calculations of the rotation matrix. Let coordinate system A and B be differently oriented by the small angle rotations $\beta_x, \beta_y$, and $\beta_z$. The corresponding rotation matrix

$$
\begin{bmatrix}
x'' \\
y'' \\
z''
\end{bmatrix} =
\begin{bmatrix}
\cos(\beta_x) & 0 & -\sin(\beta_x) \\
0 & 1 & 0 \\
\sin(\beta_x) & 0 & \cos(\beta_x)
\end{bmatrix}
\begin{bmatrix}
x' \\
y' \\
z'
\end{bmatrix}
$$

(2.8)
2.3 Plane Rotations

Figure 2.5: Plane rotations.

Table 2.1: Definition of local geodetic- to body- frame rotation matrix.
### 2.4 Rotating Coordinate Frame

As the attitude of the vehicle changes, so does the relative angle between some of the coordinate frames in the navigation system. Hence, so does the rotation matrices relating this coordinate frames. The rate of change of the rotation matrix corresponds to the first order derivative of the directional cosine matrix. Assuming that angular velocity \( \omega_{ba} \), between coordinate system A and B is constant in the interval \( t \) to \( t + \Delta t \), then using the result from small angles rotations the rotation matrix at time \( t + \Delta t \) can be written as

\[
R_b^a(t + \Delta t) = R_b^a(t) (I + \Omega_{ba}^a \Delta t)
\]

where \( \Omega_{ba}^a \) is the skew symmetric representation of the angular rates \( \omega_{ba}^a \). The matrix \( \Omega_{ba}^a \) reads

<table>
<thead>
<tr>
<th>( R_b^a )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-s(\lambda) c(\phi))</td>
</tr>
<tr>
<td>(-s(\phi))</td>
</tr>
<tr>
<td>(-c(\lambda) c(\phi))</td>
</tr>
</tbody>
</table>

where:
- \( \lambda \) - latitude [radians]
- \( \phi \) - longitude [radians]
- \( c(\cdot) \) - cosine operator
- \( s(\cdot) \) - sine operator

Table 2.2: Definition of ECEF- to local geodetic- frame rotation matrix.

and \( \beta_z \) around the axis of system A. For small \( x \) the approximations \( \cos(x) \approx 1 \) and \( \sin(x) \approx x \) are valid by aid of Taylor series expansion. Applying this to (2.7), the small angle rotation matrix can be approximated as

\[
R_b^a(\lambda, \phi) \approx \begin{bmatrix}
1 & \beta_x & -\beta_y \\
-\beta_x & 1 & \beta_y \\
\beta_y & -\beta_x & 1
\end{bmatrix} = I + \Xi_\beta
\]

(2.8)

where \( \Xi_\beta \) is the skew symmetric matrix representation of \( \beta = [\beta_x, \beta_y, \beta_z]^T \), defined such that

\[
p \times \beta = \Xi_\beta p
\]

(2.9)

Here \( p \) is an arbitrary \( 3 \times 1 \) dimensional vector. The matrix \( \Xi_\beta \) reads

\[
\Xi_\beta = \begin{bmatrix}
0 & \beta_z & -\beta_y \\
-\beta_z & 0 & \beta_x \\
\beta_y & -\beta_x & 0
\end{bmatrix}
\]

(2.10)

Using the properties of the cross product, it is straightforward to show that \( \Xi_\beta p = -\Xi_p \beta \), where \( \Xi_p \) is the skew symmetric matrix representation of the vector \( p \).

#### 2.4 Rotating Coordinate Frame

As the attitude of the vehicle changes, so does the relative angle between some of the coordinate frames in the navigation system. Hence, so does the rotation matrices relating this coordinate frames. The rate of change of the rotation matrix corresponds to the first order derivative of the directional cosine matrix. Assuming that angular velocity \( \omega_{ba} \) between coordinate system A and B is constant in the interval \( t \) to \( t + \Delta t \), then using the result from small angles rotations the rotation matrix at time \( t + \Delta t \) can be written as

\[
R_b^a(t + \Delta t) = R_b^a(t) (I + \Omega_{ba}^a \Delta t)
\]

(2.11)
\[ \boldsymbol{\Omega}_{ba} = \begin{bmatrix} 0 & -\omega_{ba_x}^{a} & \omega_{ba_y}^{a} \\ \omega_{ba_x}^{a} & 0 & -\omega_{ba_z}^{a} \\ -\omega_{ba_y}^{a} & \omega_{ba_z}^{a} & 0 \end{bmatrix} \] (2.12)

Using the definition of derivative, the rate of change in the rotation matrix becomes

\[ \dot{\mathbf{R}}^b_a(t) = \lim_{\Delta t \to 0} \frac{\mathbf{R}^b_a(t + \Delta t) - \mathbf{R}^b_a(t)}{\Delta t} \]

\[ = \lim_{\Delta t \to 0} \frac{\mathbf{R}^b_a(t) (\mathbf{I} + \mathbf{\Omega}_{ba}^{a} \Delta t) - \mathbf{R}^b_a(t)}{\Delta t} \]

\[ = \mathbf{R}^b_a(t) \mathbf{\Omega}_{ba}^{a} \] (2.13)

where (2.11) was used in the second equality. Not surprisingly the rate of change of the rotation matrix is a function of both the current orientation and the angular rate between the coordinate systems.
Chapter 3

Inertial Navigation Equation

In this chapter the fundamental equation of inertial navigation is derived. Together with the relationships for frame rotations derived in Chapter 2 the fundamental equation of inertial navigation is used to deduce the general continuous-time navigation equation. Further, the ECEF-frame navigation equations are introduced and the physical interpretation and significance of the different terms are discussed. By mechanization of the ECEF-frame navigation equations a linear model of how the sensor errors are related to the errors in the navigation output is found. The linear error model will later be used in the integration between the INS and GPS-receiver.

3.1 Fundamental Equations of Inertial Navigation

According to Newton’s first law of motion an object tends to keep its initial speed and attitude unless affected by an unbalanced force. Hence, a change in motion is a result of a force being applied to overcome the objects inertia. Two types of forces determine the motion of a vehicle: gravity and inertial force.

3.1.1 Gravity force

Newton’s universal law of gravity states that the force \( \mathbf{f}_g \), due to gravitational attraction between two masses \( m_1 \) and \( m_2 \), see Figure 3.1 is

\[
\mathbf{f}_g = \frac{G m_1 m_2}{|\mathbf{r}|^3} \mathbf{r}
\]

(3.1)

where \( G \) denotes the universal gravity constant and \( \mathbf{r} \) the position vector of mass \( m_2 \) with respect to the center of mass \( m_1 \). Let mass \( m_1 \) be equal to the earth mass \( M \) and \( m_2 \) equal to the vehicle mass \( m \). Assuming the vehicle is at a far distance from the earth, then the vehicle sensed gravity force becomes
3.1 Fundamental Equations of Inertial Navigation

\[ f_g = \frac{-GMm}{|r|^3} \cdot r \]  

(3.2)

Rewriting this as force per vehicle unit mass yields

\[ g \triangleq \frac{f_g}{m} = \frac{-GM}{|r|^3} \cdot r \]  

(3.3)

where \( g \) denotes the acceleration of gravity.

3.1.2 Inertial force

Newton’s second law states that the acceleration of an object is directly proportional to the magnitude of the net force, in the same direction as the net force and inversely proportional to the mass of the object. In the case the inertial force being the only force acting upon the vehicle

\[ f_I = m \cdot \textbf{s} \]  

(3.4)

Here \( f_I \) is the inertial force applied to the vehicle of mass \( m \) to produce the acceleration \( \textbf{s} \). The acceleration \( \textbf{s} \) will henceforth be refereed to as the specific force.

3.1.3 Equations of motion

In the previous sections the kinematic equations for an object were derived in the absent of either the inertial force \( f_I \) or the gravity force \( f_g \). In reality the motion of an object is mostly a result of both these forces being present. The total force, \( f_{tot} \) acting upon the object can be expressed as the sum of the gravitational and inertial force, that is

\[ f_{tot} = f_g + f_I \]  

(3.5)

Substituting the left hand side with the kinematic acceleration in the \( i \)-frame times the mass \( m \) of the vehicle gives

\[ m \ddot{r}^i = f_g + f_I \]  

(3.6)

Replacing the right hand side of (3.6) with (3.2) and (3.4) gives

\[ m \ddot{r}^i = -\frac{GMm}{|r|^3} \cdot r^i + m \cdot \textbf{s}^i \]  

(3.7)

Finally the fundamental equation of inertial navigation is obtained by dividing with the mass \( m \) on both sides, resulting in
3.2 Navigation Equations

\[ \ddot{\mathbf{r}}^i = \mathbf{g}^i + s^i \]  

(3.8)

The equation states that the kinematic acceleration \( \ddot{\mathbf{r}}^i \) is equal to the sum of the gravity acceleration \( \mathbf{g}^i \) and the specific force \( s^i \).

3.2 Navigation Equations

The navigation equations are a set of differential equations describing the relationship between the INS outputs (that is position, velocity and attitude) and the inputs in terms of acceleration and angular rates. Below, they are first derived for an arbitrary navigation frame and then for the ECEF-frame, used in the GPS aided INS approach.

3.2.1 General Navigation Equations

Let \( \mathbf{r}^a \) and \( \mathbf{r}^i \) be two position vectors in the arbitrary \( a \)-frame and the \( i \)-frame, respectively. Remember that the \( i \)-frame is an inertial frame with origin at the earth center of mass, so the gravity force applies. The position vectors can be related to each other through the directional cosine matrix \( \mathbf{R}_{ia}^a \) as

\[ \mathbf{r}^i = \mathbf{R}_{ia}^a \mathbf{r}^a + \mathbf{q}^i \]  

(3.9)

where \( \mathbf{q}^i \) is a vector pointing from the origin of the \( i \)-frame to the origin of the \( a \)-frame. Differentiating twice with respect to time yields

\[ \ddot{\mathbf{r}}^i = \mathbf{R}_{ia}^a \ddot{\mathbf{r}}^a + 2 \mathbf{R}_{ia}^a \dot{\mathbf{r}}^a + \mathbf{R}_{ia}^a \dot{\mathbf{r}}^a + \ddot{\mathbf{q}}^i \]  

(3.10)

The first order derivative of the directional cosine matrix was derived in Section (2.4), and reads

\[ \dot{\mathbf{R}}_{ia}^a = \mathbf{R}_{ia}^a \Omega_{ia}^a \]  

(3.11)

where \( \Omega_{ia}^a \) was derived in (2.12). Differentiating (3.11) once more yields

\[ \ddot{\mathbf{R}}_{ia}^a = \dot{\mathbf{R}}_{ia}^a \Omega_{ia}^a + \mathbf{R}_{ia}^a \dot{\Omega}_{ia}^a \]  

(3.12)

Inserting equations (3.11) and (3.12) into (3.10) results in

\[ \ddot{\mathbf{r}}^i = \mathbf{R}_{ia}^a \ddot{\mathbf{r}}^a + 2 \mathbf{R}_{ia}^a \Omega_{ia}^a \dot{\mathbf{r}}^a + \mathbf{R}_{ia}^a ( \dot{\Omega}_{ia}^a + \Omega_{ia}^a \Omega_{ia}^a \Omega_{ia}^a ) \mathbf{r}^a + \ddot{\mathbf{q}}^i \]  

(3.13)

Finally substituting \( \ddot{\mathbf{r}}^i \) with \( \mathbf{g}^i + s^i \) and multiplying both sides with rotation matrix \( \mathbf{R}_{i}^a \) yields

\[ \mathbf{g}^a + s^a = \ddot{\mathbf{r}}^a + 2 \Omega_{ia}^a \dot{\mathbf{r}}^a + ( \dot{\Omega}_{ia}^a + \Omega_{ia}^a \Omega_{ia}^a ) \mathbf{r}^a + \ddot{\mathbf{q}}^a \]  

(3.14)

where \( \mathbf{p}^a = \mathbf{R}_{i}^a \mathbf{p}^i \) for an arbitrary vector \( \mathbf{p} \) was used. Further, \( \mathbf{R}_{i}^a \mathbf{R}_{i}^a = \mathbf{I} \). This is the general differential equation for navigation in an arbitrary \( a \)-frame [5]. Worth noting is that the derived equation is exact. Note also that in many coordinate frames and navigation systems some terms are small and may be neglected, resulting in a simplified set of equations.
3.2 Navigation Equations

3.2.2 ECEF-frame navigation equations

An ECEF navigation coordinate system implementation of the navigation equations has the advantage of producing position estimates in the same coordinate system as used by the GPS system, which simplifies the integration between the two systems [1]. Substituting the arbitrary $a$-frame with the ECEF-frame, denoted by the superscript $e$, results in

$$g^e + s^e = \ddot{r}^e + 2 \Omega_{ie}^e \dot{r}^e + \dot{\Omega}_{ie}^e r^e + \Omega_{ie}^e \Omega_{ie}^e r^e + \ddot{q}^e$$

(3.15)

where $\ddot{q}^e = 0$ and the skew-symmetric matrix $\Omega_{ie}^e$ takes the simplified form

$$\Omega_{ie}^e = \begin{bmatrix} 0 & -\omega_{ie} & 0 \\ \omega_{ie} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

(3.16)

This is due to the fact that the ECEF-frame ($e$-frame) and $i$-frame only differs in a rotation around the $z$-axis of the ECEF-frame. The earth rotational rate relative to the inertial frame is approximated to $\omega_{ie} \approx 7.292115 \times 10^{-5}$ rad/s [5].

The physical interpretation and significance of the different terms in (3.15) are as follows. The first term of the right hand side is the acceleration as observed in the ECEF-frame. The second term is the Coriolis acceleration [6]. The third term is the Euler acceleration and can be omitted, since the change in the earth rotational rate is negligible [6]. The last term is the centripetal force due to the earth rotation, which can be neglected, due to low sensitivity of the used sensors. The simplified ECEF navigation equation reads

$$\ddot{r}^e = -2 \Omega_{ie}^e \dot{r}^e + g^e + s^e$$

(3.17)

The specific force $s$ is observed in the body frame and must be transformed to the the ECEF-frame by the rotation matrix $R^b_e$ before used in the navigation equation. Actually, the specific force is measured in the platform coordinates but since the coordinate axis of the IMU-platform and the body coordinate system is assumed to be perfectly aligned the body and platform coordinate frames are equivalent. Introducing the velocity vector $v^e = \dot{r}^e$ equation (3.17) can be written as two first order differential equations, which together with the relationship for rotating coordinate frames derived in Section 2.4 gives the ECEF navigation dynamics, that is

$$\begin{bmatrix} \dot{v}^e \\ \dot{\Omega}_{ib}^e \\ \dot{R}^b_e \end{bmatrix} = \begin{bmatrix} -2 \Omega_{ie}^e v^e + g^e + R^b_e \dot{s}^b \\ \omega_{ie}^e \end{bmatrix}$$

(3.18)

These are the continuous-time versions of the equations implemented in the considered ECEF inertial navigation system. According to [9], the rotation rates, $\omega_{eb}^b$, between the ECEF-frame and the body-frame can be obtained as follows

$$\omega_{eb}^b = \omega_{ib}^b - R^b_e \omega_{ie}^e$$

(3.19)

where the angular rate $\omega_{ib}^b$ between the body frame and the inertial frame is measured by the gyros, and $\omega_{ie}^e$ is the earth rotational rate relative to the inertial frame. A graphical interpretation of how the ECEF-navigation equations are used in an ECEF INS implementation is shown in Figure 3.2 [10].
3.3 ECEF Navigation Error Model

Even though the inertial instruments have been calibrated, the measured IMU signals will be erroneous, due to environmental variations and instrument degradation. As a result, there are biases in the position and velocity estimates as well as a misalignment between the estimated and true coordinate rotation matrix. In a GPS-aided INS, the position error is estimated by comparing the position calculated by the GPS-receiver with that of the INS. The estimated position error is then related to velocity, attitude, and sensor errors through a set of differential equations, referred to as the navigation error equations. The error equations are derived below.

3.3.1 ECEF navigation error equations

The ECEF navigation error equations will be derived by mechanization of the equations in (3.18). Mechanization means that all terms in the navigation equations are substituted with the corresponding estimated or measured quantities. By Taylor series expansion and neglecting all second and higher order terms a linear model of how the different errors are related is found.

Let tilde, (~) denote a measured and hat (̂) a computed value. Then the mechanized version of the velocity equation in (3.18) reads

$$\hat{\mathbf{v}} = \mathbf{R}_e^b \tilde{s}^b + \tilde{\mathbf{g}}^e - 2 \omega^e \times \hat{\mathbf{v}}$$  \hspace{1cm} (3.20)

Thus, the accelerations $s^b$ is measured by the accelerometers, and $\Omega_{e}^c$ is a known constant matrix. All other quantities are calculated ones. Introducing the vector $\epsilon$ denoting the small angles rotations that aligns the computed and true coordinate rotation matrix, the computed and true rotation matrix can be related as

$$\mathbf{R}_e^b = (\mathbf{I} - \Omega_{e}) \mathbf{R}_e^c$$  \hspace{1cm} (3.21)

where the skew-symmetric matrix $\Omega_{e}$ reads
3.3 ECEF Navigation Error Model

$$\Omega_e = \begin{bmatrix} 0 & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & 0 & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & 0 \end{bmatrix}$$ (3.22)

Define the error as the actual value minus the computed value, respectively the actual value minus the measured value. Then the mechanized equation can be written in terms of true values and errors as

$$\dot{v}^e - \delta \dot{v}^e = (I - \Omega_e) \dot{R}_b^e (s^b - \delta s^b) + g^e - \delta g^e - 2\Omega_{ie}^e (v^e - \delta v^e)$$ (3.23)

where $\delta v^e$ denotes the error in velocity, et cetera. Expansion of (3.23) gives

$$\dot{v}^e - \delta \dot{v}^e = s^e + g^e - 2\Omega_{ie}^e v^e - \Omega_e s^e - R_b^e \delta s^b + \Omega_e R_b^e \delta s^b - \delta g^e + 2\Omega_{ie}^e \delta v^e$$ (3.24)

The first three terms on the right hand side of equation (3.24) can be identified to correspond to $\dot{v}^e$ and thus cancels out. Neglecting the second order term $\Omega_e R_b^e \delta s^b$, a linear differential equation for the velocity error is obtained, that is

$$\delta \dot{v}^e = \Omega_e s^e + R_b^e \delta s^b + \delta g^e - 2\Omega_{ie}^e \delta v^e$$ (3.25)

The term $\Omega_e s^e = -S^e \epsilon$, where $S^e$ is the skew symmetric matrix representation of the specific fores $s^e$ in ECEF coordinates. Hence, the velocity error can be calculated as

$$\delta \dot{v}^e = -S^e \epsilon + R_b^e \delta s^b + \delta g^e - 2\Omega_{ie}^e \delta v^e$$ (3.26)

where

$$S^e = \begin{bmatrix} 0 & -s_z^e & s_y^e \\ s_z^e & 0 & -s_x^e \\ -s_y^e & s_x^e & 0 \end{bmatrix}$$ (3.27)

A linear model for the small angle errors $\epsilon$ in the computed coordinate rotation matrix can be found through mechanization of the ECEF attitude equation, that is the last equation in (3.18). The mechanized ECEF attitude equation reads

$$\dot{R}_b^e = \bar{R}_b^e \bar{\Omega}_{eb}^b$$ (3.28)

Differentiating equation (3.21) once with respect to time and inserting on the left hand side of the mechanized attitude equation yields

$$(I - \Omega_e) \dot{R}_b^e - \dot{\Omega}_e R_b^e = \dot{R}_b^e \bar{\Omega}_{eb}^b$$ (3.29)

Expansion of the right hand side results in

$$(I - \Omega_e) \dot{R}_b^e - \dot{\Omega}_e R_b^e = (I - \Omega_e) \frac{R_b^e \Omega_{eb}^e - (I - \Omega_e) R_b^e \delta \Omega_{eb}^e}{\approx I}$$ (3.30)
Identifying $\mathbf{R}_b^e \mathbf{\Omega}_e^b$ to be equal to $\dot{\mathbf{R}}_e^b$ and approximating $\mathbf{I} - \mathbf{\Omega}_e \approx \mathbf{I}$, results in the following equation for the attitude errors.

$$\dot{\mathbf{\Omega}}_e = \mathbf{R}_b^e \delta \mathbf{\Omega}_e^b \mathbf{R}_e^b$$

(3.31)

Using the fact that skew-symmetric matrices $\mathbf{\Omega}_e$ and $\mathbf{\Omega}_e^b$ are totally described by the vector $\epsilon$ respectively $\omega_e^b$, the attitude error equation can be written in vector form as [5]

$$\dot{\epsilon} = \mathbf{R}_e^b \delta \omega_e^b$$

(3.32)

Here the attitude error is a linear function of the error in the computed ECEF to body frame angular rate $\delta \omega_e^b$. The angular rate error can be rewritten in terms of the measurement error of the gyros $\delta \omega_{ib}^b$ and the error in the earth rotational rate, when expressed in the body coordinates $\delta \omega_{ie}^b$. The relationship reads

$$\delta \omega_{ib}^b = \delta \omega_{ib}^b - \delta \omega_{ie}^b$$

(3.33)

The earth rotational rate is assumed to be perfectly known when expressed in ECEF-coordinates, but when converted to body-coordinates errors are introduced due to the misalignment between the computed and true rotation matrix. The error is found through the following mechanization

$$\dot{\omega}_{ie}^b = \dot{\mathbf{R}}_e^b \omega_{ie}^c$$

(3.34)

Here the mechanized ECEF to body frame rotation matrix is found as

$$\dot{\mathbf{R}}_e^b = (\dot{\mathbf{R}}_b^e)^{-1}$$

$$= ((\mathbf{I} - \mathbf{\Omega}_e) \mathbf{R}_e^b)^{-1}$$

$$= \mathbf{R}_e^b (\mathbf{I} - \mathbf{\Omega}_e)^{-1}$$

(3.35)

Expanding $(\mathbf{I} - \mathbf{\Omega}_e)^{-1}$ into geometric series and neglecting second and higher order terms the mechanized directional cosine matrix can be approximated as

$$\dot{\mathbf{R}}_e^b \approx \mathbf{R}_e^b (\mathbf{I} + \mathbf{\Omega}_e)$$

(3.36)

Inserting this into (3.34) and rewriting the equation in terms of actual values and errors results in

$$\omega_{ie}^b - \delta \omega_{ie}^b = \mathbf{R}_e^b (\mathbf{I} + \mathbf{\Omega}_e) \omega_{ie}^c$$

$$\delta \omega_{ib}^b = -\mathbf{R}_e^b \mathbf{\Omega}_e \omega_{ie}^c$$

$$\delta \omega_{ie}^b = \mathbf{R}_e^b \mathbf{\Omega}_{ie}^c \epsilon$$

(3.37)

Substituting this into (3.32) yields the final attitude error equation, that is

$$\dot{\epsilon} = \mathbf{R}_e^b \delta \omega_{ib}^b - \mathbf{\Omega}_{ie}^c \epsilon$$

(3.38)

Collecting the derived relationships for the attitude and velocity error the ECEF-navigation error equations are as follows
3.3 ECEF Navigation Error Model

\[
\begin{bmatrix}
\delta \mathbf{v}^e \\
\delta \mathbf{v}^e \\
\dot{\epsilon}
\end{bmatrix} =
\begin{bmatrix}
-\mathbf{S}^e \epsilon + R^e_b \delta \mathbf{b}^b + \delta \mathbf{g}^e - 2 \Omega^e_{te} \delta \mathbf{v}^e \\
R^e_b \delta \omega^b_{ib} - \Omega^e_{te} \epsilon
\end{bmatrix}
\] (3.39)

So far the equations describing how the errors in position, velocity and attitude are related to measurement errors have been derived. Using the knowledge about how the accelerometer and gyro errors develops with time a model for the ECEF navigation error dynamics can be found. There are several approaches for how the measurement errors can be modelled, see [6]. Here the measurement errors are modelled as a random level, and white Gaussian noise, describing the bias and the measurement noise, respectively. Assuming that the IMU sensors are the only noise sources in the system the noise enters the system equations only through the attitude and velocity state, that is the two last equations in (3.18).

A convenient way to describe a dynamic model is to write it in terms of a state space model. Defining the error state vector

\[
\delta \mathbf{x}(t) = \begin{bmatrix}
\delta \mathbf{r}^e \\
\delta \mathbf{v}^e \\
\dot{\epsilon} \\
\end{bmatrix}
\]

and the measurement noise vector

\[
u_c(t) = \begin{bmatrix}
u_{acc}(t) \\
u_{gyro}(t) \end{bmatrix}
\]

where \(u_{acc}(t)\) denotes the accelerometer noise and \(u_{gyro}(t)\) the gyro noise. Neglecting gravity errors, the navigation error state model becomes

\[
\dot{\delta \mathbf{x}}(t) = \mathbf{F}(t) \delta \mathbf{x}(t) + \mathbf{G}(t) u_c(t)
\] (3.42)

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

\[
\mathbf{F}(t) =
\begin{pmatrix}
0_3 & \mathbf{I}_3 & 0_3 & 0_3 & 0_3 \\
0_3 & -2\Omega^e_{te} & -\mathbf{S}^e & R^e_b & 0_3 \\
0_3 & 0_3 & -\Omega^e_{te} & 0_3 & R^e_b \\
0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & 0_3 & 0_3
\end{pmatrix}
\] (3.43)

and \(\mathbf{G}(t)\) is of size 15 × 6

\[
\mathbf{G}(t) =
\begin{pmatrix}
0_3 & 0_3 \\
R^e_b & 0_3 \\
0_1 & R^e_b \\
0_1 & 0_3 \\
0_3 & 0_3
\end{pmatrix}
\] (3.44)

The error equations are time-varying, since \(R^e_b\) depends on the attitude and \(\mathbf{S}^e\) on the acceleration of the vehicle. Recall that \(R^e_b\) is the rotation matrix, transforming a vector from the body-frame into the ECEF-frame. Further \(\mathbf{S}^e\) and \(\Omega^e_{te}\) are the skew symmetric matrix representation of the specific force and the earth rotational rate, respectively. In (3.43) \(\mathbf{I}_3 (0_3)\) denotes the unity (zero) matrix of order 3.

The constructed IMU platform houses three separate accelerometers and gyros, therefore the sensors’ noises are assumed uncorrelated. However, the accelerometers respectively the gyros are of the same model and thus assumed...
to have the same noise characteristics. Let $\sigma^2_{\text{acc}}$ and $\sigma^2_{\text{gyro}}$ denote the variance of the accelerometer and the gyro noise, respectively. Then the covariance matrix, $Q_c(t)$ of the Gaussian measurement noise $u_c(t)$ in (3.41) is defined as

$$E\{u_c(t) u_c^*(\tau)\} = \begin{bmatrix} \sigma^2_{\text{acc}} I_3 & 0_3 \\ 0_3 & \sigma^2_{\text{gyro}} I_3 \end{bmatrix} \delta(t - \tau)$$

$$\triangleq Q_c(t - \tau) \quad (3.45)$$

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

So far the continuous-time versions of the ECEF navigation equations have been derived from Newton’s laws of gravity and motion. Through mechanization of the navigation equations a linear state space model was found, describing how the errors in position, velocity, attitude and biases are related. Next the continuous-time ECEF navigation equations and the model for the error dynamics will be discretized, making them suitable for implementation in the considered GPS aided inertial navigation system.
Chapter 4

Discretization

4.1 Discretization

In this chapter the derived navigation equations and error dynamics are sampled, making them suitable for a discrete implementation. First the zero-order-hold sampling of the navigation equations are described, where special care is taken to preserve the properties of the rotation matrix. Next the model of the navigation error dynamics is discretized.

4.1.1 Discrete time navigation equations

Introducing the state vector \( \mathbf{d}(t) \) housing the position and velocity, and the input vector \( \mathbf{n}(t) \) housing the accelerations in ECEF coordinates, that is

\[
\mathbf{d}(t) = \begin{bmatrix} \mathbf{r}^e & \mathbf{v}^e \end{bmatrix}^T
\]

and

\[
\mathbf{n}(t) = -2\mathbf{\Omega}_{se}^e \mathbf{v}^e + \mathbf{g}^e + \mathbf{R}^e_b \mathbf{s}^e
\]

Then the two first navigation equations in (3.18) can be written in state space form as

\[
\dot{\mathbf{d}}(t) = \mathbf{A} \mathbf{d}(t) + \mathbf{B}(t) \mathbf{n}(t)
\]

where \( \mathbf{A} \) is the \( 6 \times 6 \) matrix

\[
\mathbf{A} = \begin{bmatrix} 0_3 & \mathbf{I}_3 \\ 0_3 & 0_3 \end{bmatrix}
\]

and \( \mathbf{B} \) is the \( 6 \times 3 \) matrix

\[
\mathbf{B} = \begin{bmatrix} 0_3 \\ \mathbf{I}_3 \end{bmatrix}
\]

Having a continuous-time equation as in (4.3) with a known solution at time \( t_0 \), the solution at a time \( t > t_0 \) can be represented as [7, 8].

\[
\mathbf{d}(t) = \Phi(t, t_0) \mathbf{d}(t_0) + \int_{t_0}^{t} \Phi(t, \tau) \mathbf{B}(\tau) \mathbf{n}(\tau) d\tau
\]
where the state transition matrix $\Phi(t,t_0)$ is defined as the unique solution to

$$\frac{\partial \Phi(t,\tau)}{\partial t} = A \Phi(t,\tau) \quad (4.7)$$

If the state transition matrix $A$ is time invariant, the homogenous differential equation (4.7) has the solution of the matrix exponential function, that is $\Phi(t,\tau) = \exp(A \cdot (t - t_0))$ [8]. Using the power series definition of the matrix exponential, the state transition matrix between time $kT_s$ and $(k+1)T_s$ can be calculated as

$$\Phi((k+1)T_s, kT_s) = I_6 + A T_s + \frac{A^2 T_s^2}{2!} + \ldots$$

$$= I_6 + A T_s \quad (4.8)$$

where the last equality is a result of $A^n = 0_6$ for all $n > 1$. Having the solution to the state transition matrix $\Phi((k+1)T_s, kT_s)$ and assuming $n(t)$ to be constant between the sample instances an approximative solution to the integral on the right hand side in (4.6) can be found as

$$\int_{t_0}^{t} \Phi(t,\tau) B n(\tau) d\tau = \int_{kT_s}^{(k+1)T_s} (I + A T_s) B n(\tau) d\tau$$

$$\approx T_s B n(kT_s) \quad (4.9)$$

Hence, the zero-order-hold sampling of the position and velocity ECEF navigation equations becomes

$$\begin{bmatrix} r_{e,k+1}^e \\ v_{e,k+1}^e \end{bmatrix} = \begin{bmatrix} I_3 & T_s \ I_3 \\ 0_3 & I_3 \end{bmatrix} \begin{bmatrix} r_{e,k}^e \\ v_{e,k}^e \end{bmatrix} + T_s \begin{bmatrix} 0_{3 \times 1} \\ \mathbf{R}_{b,k}^e s_{b,k}^e - 2\mathbf{\Omega}_e^b v_{e,k}^e + g_{e,k}^b \end{bmatrix} \quad (4.10)$$

When discretizing the attitude equation in (3.18) care must be taken so that the orthogonality constrains of the directional cosine matrix are maintained. Let $T_s$ denote the sample time and assume that $\mathbf{\Omega}_{eb}^b$ is constant. Then the matrix taking the solution of the attitude differential equation from time $kT_s$ to $(k+1)T_s$ is $\exp(\mathbf{\Omega}_{eb}^b T_s)$. Hence, the attitude equations can be approximated by

$$\mathbf{R}_{b,k+1}^e = \mathbf{R}_{b,k}^e e^{\mathbf{\Omega}_{eb}^b T_s} \quad (4.11)$$

By expanding the matrix exponential into a $(n,n)$ Padé approximation the orthogonality constrains of the rotation matrix are preserved [1]. Using a $(2,2)$ Padé approximation the discrete attitude equation becomes

$$\mathbf{R}_{b,k+1}^e = \mathbf{R}_{b,k}^e (2\mathbf{I}_3 + \mathbf{\Omega}_{eb}^b T_s)(2\mathbf{I}_3 - \mathbf{\Omega}_{eb}^b T_s)^{-1} \quad (4.12)$$

Equations (4.10) and (4.12) are the discrete ECEF navigation equations implemented in the INS.
4.1 Discretization

4.1.2 Discrete time error equations

In the state space model for the ECEF navigation dynamics both \( F(t) \) and \( G(t) \) are time varying. If the sample rate is chosen high compared to the rate of change in \( F(t) \) and \( G(t) \), they may be approximated as the constant matrices between the samples. Applying the same procedure as for the navigation equations the discrete state space model for the error dynamics becomes

\[
\delta x_{k+1} = \Psi_k \delta x_k + u_{d,k} \tag{4.13}
\]

where the state transition matrix is approximated as

\[
\Psi_k = \Phi((k+1)T_s, kT_s) \approx e^{F(kT_s)T_s} \tag{4.14}
\]

Expanding the matrix exponential into a power series neglecting second and higher order terms yields

\[
\Psi_k \approx I + F(kT_s)T_s \tag{4.15}
\]

The discrete-time process noise, \( u_k \) is

\[
u_{d,k} = \int_{kT_s}^{(k+1)T_s} \Phi((k+1)T_s, s)G(s)u_c(s)ds \tag{4.16}
\]

Since \( u_{d,k} \) is a linear combination of Gaussian noise, it is Gaussian distributed and described by its first and second order moments. The mean of \( u_{d,k} \) is zero, since \( u_c(t) \) is assumed zero mean. Applying the definition of covariance and assuming \( T_s \) small, the covariance of the discrete-time noise \( Q_{d,k} \) can be approximated as \[5\]

\[
Q_{d,k} = \begin{bmatrix}
G(kT_s) Q_c(kT_s) G^*(kT_s) T_s \\
\text{diag}(\sigma_{acc}^2 I_3, \sigma_{gyro}^2 I_3, 0_6)
\end{bmatrix} \tag{4.17}
\]

where \( \text{diag}(\cdot) \) denotes a block diagonal matrix. The last equality is a result of orthonormality property of the rotation matrix, \( R_e \) and that \( Q_c(t) \) is a diagonal matrix.

The definition of the state observation equation is straightforward since the GPS position estimate is used, and not the pseudo ranges. Let \( \delta y \) be the difference between the GPS and INS position estimate and \( w_{d,k} \) the error in the GPS position estimates. Then the observation equation can be written as

\[
\delta y_k = H_k \delta x_k + w_{d,k} \tag{4.18}
\]

with the state observation matrix, \( H_k \) of size \( 3 \times 15 \) defined as

\[
H_k = \begin{cases}
I_3, & k = n\ell, \; n = 1, 2, 3... \\
0_{3\times15}, & \text{otherwise}
\end{cases} \tag{4.19}
\]

where \( \ell \) denotes the ratio between the INS and GPS sample frequency.
In this Chapter a method for combining GPS and INS data is derived. The proposed method utilizes the complimentary properties of the two systems and combine them, resulting in a navigation system with both higher update rate and accuracy than the stand alone GPS-receiver. Further, the integrated navigation system will be able to provide the user with position estimates even in the absences of satellite signals. In the proposed method the INS acts as the major navigation system and the position estimates from the GPS is used to estimate the errors in the INS. The integration method is referred to as a GPS aided INS.

When GPS-data is available the system compares the position estimate from the GPS-receiver with the position calculated by the INS. The estimated position errors are fed to a Kalman filter housing the error model for the ECEF navigation error dynamics derived in Chapter 3. A derivation of the general Kalman filter equations can be found in Appendix A. The filter estimates the navigation system errors, which are used to calibrate the INS. The system is shown in Figure 1.2. The described system is actually an indirect extended Kalman filter, where the navigation equations are linearized around the output of the INS.

5.1 The Extend Kalman Filter

The discrete non-linear navigation equations (4.10) and (4.12) can be written as

\[ z_{k+1} = c(z_k, a_k) + u_k \]  \hspace{1cm} (5.1)

where \( z_k \) is the navigation system outputs: position, velocity and Euler angles \( \theta_k \) defining the rotation matrix \( R_{eb}^k \), that is the 9-element vector

\[ z_k = [ r^e_k \ v^e_k \ \theta^e_k ]^* \] \hspace{1cm} (5.2)

Further, the 6-element vector \( a_k \) contains the inputs to navigation system, accelerations and angular rates, that is

\[ a_k = [ f^b_k \ \omega^b_{ib,k} ]^* \] \hspace{1cm} (5.3)
5.1 The Extend Kalman Filter

The vector $u'_k$ is the noise in measurements of the navigation input. Linearization of the navigation equations (5.1) are first done around a known nominal trajectory, resulting in a linear model for the perturbations away from the true trajectory. To the linear error equations the standard Kalman filter is applied. Then substituting the nominal trajectory with that of the INS estimated trajectory results in an extended Kalman filter. Consider the true state vector $z_k$ and the measured input $\tilde{a}_k$ to the system written as

$$
z_k = z_{k}^{nom} + \delta z_k
$$

$$
\tilde{a}_k = a_{k}^{nom} + \delta a_k
$$

where $z_{k}^{nom}$ and $a_{k}^{nom}$ are the nominal trajectory and input. The quantity, $\delta z_k$ is the perturbation away from the true trajectory and $\delta a_k$ the bias of the measurements. Assuming $\delta z_k$ and $\delta a_k$ are small and applying a first order Taylor series expansion to $c(z,a)$, equation (5.1) can be approximated as

$$
z_{k+1}^{nom} + \delta z_{k+1} \approx c(z_{k}^{nom}, a_{k}^{nom}) + C_{1,k} \delta z_k + C_{2,k} \delta a_k + u'_k
$$

(5.6)

where

$$
C_{1,k} = \left. \frac{\partial c(z,a)}{\partial z} \right|_{z=z_{k}^{nom}} \quad C_{2,k} = \left. \frac{\partial c(z,a)}{\partial a} \right|_{a=a_{k}^{nom}}
$$

(5.7)

The Jacobians' of $c(z,a)$ are updated with nominal trajectory and input for each sample. Choosing $z_{k}^{nom}$ and $a_{k}^{nom}$ to fulfill the deterministic differential equation

$$
z_{k+1} = c(z_{k}^{nom}, a_{k}^{nom})
$$

(5.8)

and substituting this into equation (5.6) results in a linear model for the error $\delta z_k$, that is

$$
\delta z_{k+1} = C_{1,k} \delta z_k + C_{2,k} \delta a_k + u'_k
$$

(5.9)

Noting that $\delta \mathbf{x}_k = \begin{bmatrix} \delta z_k & \delta a_k \end{bmatrix}^T$ it becomes clear that $C_{1,k}$ and $C_{2,k}$ corresponds to the upper part of the navigation error state transition matrix $\Psi_k$. The lower $6 \times 6$ block matrix of $\Psi_k$ is a description of how the IMU biases $\delta a_k$ develops with time. Since this is a linear model the standard Kalman filter equations can be applied to estimate $\delta \mathbf{x}_k$ [8]. The Kalman filter equations read

$$
\begin{bmatrix}
\delta \mathbf{z}_{k+1} \\
\delta \mathbf{a}_{k+1}
\end{bmatrix} = \Psi_k 
\begin{bmatrix}
\delta \mathbf{z}_k \\
\delta \mathbf{a}_k
\end{bmatrix}
$$

(5.10)

$$
\begin{bmatrix}
\delta \mathbf{z}_k \\
\delta \mathbf{a}_k
\end{bmatrix} = \mathbf{K}_{f,k} \begin{bmatrix} \mathbf{Y}_k - \mathbf{H}_k \begin{bmatrix} z_{k}^{nom} \\ a_{k}^{nom} \end{bmatrix} - \mathbf{H}_k \begin{bmatrix} \delta \mathbf{z}_k \\
\delta \mathbf{a}_k
\end{bmatrix} \end{bmatrix}
$$

(5.11)

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

(5.12)

$$
\mathbf{P}_k = \mathbf{P}_k^{-1} - \mathbf{K}_{f,k} \mathbf{H}_k \mathbf{P}_k^{-1}
$$

(5.13)

$$
\mathbf{P}_{k+1} = \Psi_k \mathbf{P}_k \Psi_k^T + \mathbf{Q}_{d,k}
$$

(5.14)
Here $\delta \hat{a}_k$ denotes the estimated biases in the measurements, \textit{et cetera}. Variables with a minus sign, $(\cdot)^-$ are predicted values. The matrix, $R_{d,k}$ is the covariance matrix of the error, $w_{d,k}$ in the GPS position estimates, $y_k$. Now adding $z_{k}^{\text{nom}}$ to both sides of equation (5.10) and substituting $z_{k}^{\text{nom}}$ with current estimate in all equations results in an extended Kalman filter, where the time and filter update for the estimates are given below

$$\hat{z}_{k+1} = c(\hat{z}_k, \hat{a}_k) \quad (5.15)$$
$$\delta \hat{a}_{k+1} = [\Psi_k]_{10:15,10:15} \delta \hat{a}_k \quad (5.16)$$
$$\begin{bmatrix} \delta \hat{z}_k \\ \delta \hat{a}_k \end{bmatrix} = \begin{bmatrix} \delta \hat{z}_k \\ \delta \hat{a}_k \end{bmatrix} + K_{f,k} \left( y_k - H_k \begin{bmatrix} \hat{z}_k \\ 0_{6 \times 1} \end{bmatrix} \right) \quad (5.17)$$

The solution to (5.15) is provided by the INS, since this corresponds to the navigation equations. The vector $\hat{a}_k$ is the estimate of the true IMU-signal obtained by subtracting the estimated bias from the measured IMU signal. The only problem is the time update of navigation state errors $\delta \hat{z}_k$. If the estimated navigation error states are fed back to the INS for correction of the INS internal stages, the corresponding error states can be set to zero [1]. Hence, $\delta \hat{z}_k = 0_{9 \times 1}$.

The final algorithm for the integration is given in Table 5.1.

<table>
<thead>
<tr>
<th>No GPS data available. ($k \neq 100, 200, \ldots$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{a}_k = \hat{a}_k + \delta \hat{a}_k$</td>
</tr>
<tr>
<td>$\hat{z}_{k+1} = c(\hat{z}_k, \hat{a}_k)$</td>
</tr>
<tr>
<td>$\delta \hat{a}<em>{k+1} = [\Psi_k]</em>{10:15,10:15} \delta \hat{a}_k$</td>
</tr>
<tr>
<td>$P_{k+1}^- = \Psi_k P_k^- \Psi_k^* + Q_{d,k}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GPS data available. ($k = 100, 200, \ldots$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{f,k} = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$</td>
</tr>
<tr>
<td>$\begin{bmatrix} \delta \hat{z}<em>k \ \delta \hat{a}<em>k \end{bmatrix} = \begin{bmatrix} 0</em>{9 \times 1} \ 0</em>{9 \times 1} \end{bmatrix} + K_{f,k} \left( y_k - H_k \begin{bmatrix} \hat{z}<em>k \ 0</em>{6 \times 1} \end{bmatrix} \right)$</td>
</tr>
<tr>
<td>$\hat{z}_k = \hat{z}_k + \delta \hat{z}_k$</td>
</tr>
<tr>
<td>$\hat{a}_k = \hat{a}_k + \delta \hat{a}_k$</td>
</tr>
<tr>
<td>$P_k = P_k^- - K_{f,k} H_k P_k^-$</td>
</tr>
<tr>
<td>$\hat{a}_k = c(\hat{z}_k, \hat{a}_k)$</td>
</tr>
<tr>
<td>$\delta \hat{a}<em>{k+1} = [\Psi_k]</em>{10:15,10:15} \delta \hat{a}_k$</td>
</tr>
<tr>
<td>$P_{k+1}^- = \Psi_k P_k^- \Psi_k^* + Q_{d,k}$</td>
</tr>
</tbody>
</table>

Table 5.1: The algorithm for integration between GPS and INS data, with a ratio between the sample rates equal to 100 times.
Chapter 6

Hardware

This chapter includes a short description of the tailor made hardware constructed for the GPS aided inertial navigation system described in this thesis. It should be noted that the hardware and proposed calibration method have at the time of writing not yet been tested and may be refined or changed. The hardware consist of four distinguishable parts: the IMU housing the tree gyros and tree accelerometers, the GPS-receiver providing position estimates according to the NMEA protocol via a RS232 serial interface, the PC onto which the proposed integration algorithm is implemented, and the micro-controller responsible for data acquisition.

6.1 The Inertial Measurement Unit

The inertial measurement unit comprises three state-of-the-art MEMS gyros and accelerometers. The sensors are mounted with their instrumental sensitivity axes pointing in three perpendicular directions, resulting in a six degree-of-freedom IMU. Even though the sensors and their sensitivity axes are mounted to point in three perpendicular directions, they will never span a truly orthogonal coordinate system. Procedures for finding this misalignment and compensation methods can be found in [6]. Before the IMU is used the sensors must be calibrated. For the accelerometers this is straightforward since the norm of the accelerometer measurement output vector, when the IMU platform is at rest should be equal to the local earth acceleration. A common method for calibration of gyros is to use the earth rotational rate as reference, but due to the lack of sensitivity in the used sensors this is not possible and other methods must be consulted.

6.2 Data Acquisition

A micro-controller housing a ten bit analog to digital converter and a six to one analog multiplexer is used to sample the IMU sensors and control the data acquisition. More, the micro-controller masters the GPS-receiver via a RS232 serial interface and synchronizes the GPS and IMU data streams. The collected data is then transmitted over a second RS232 interface to a PC, housing the proposed integration algorithm. Further, the calibration parameters are stored
in the micro-controller and sent to the host PC at start up. Hence, calibrations are not necessary, but maybe preferable when a new software is used. Using a battery operated PC the system is fully mobile and able to perform real-time signal processing. A block diagram of the hardware is shown in Figure 6.1.
Chapter 7

Simulation Results

In this chapter simulation results of the proposed algorithm for integration between GPS and INS data will be presented. To evaluate the integration algorithm a Monte Carlo simulation has been applied to simulation data corresponding to a typical driving scenario. The Monte Carlo simulation consists of twenty different GPS and IMU sensors noise realizations generated under the same conditions. To the different data realizations the integration algorithm has been applied and the outputs have been averaged. The driving scenario and the simulation results are shown and discussed below.

7.1 Driving Scenario

The driving scenario is illustrated in Figure 7.1. The car is stationary for the first 100 seconds, then it makes a wide turn and accelerates to 18 km/h which it keeps until after the last turn. Finally the car slows down and stops. In Figure 7.1 the dashed trajectory is the position estimates generated by the GPS aided INS for one typical noise realization, i.e not a Monte Carlo simulation. The shown specks are the GPS position estimates and the solid line is the true trajectory. A ratio of 100 times was used between the INS and GPS sample ratio and the GPS position estimates had a standard deviation of 13 meters. The biases of the accelerometers and gyros were in the order of 1-2 cm/s^2 respectively 5-10 °/h. Not surprisingly, the GPS aided system clearly outperforms the GPS-system.

In Figure 7.2 the averaged errors in the estimated accelerometer biases are shown. On the average the accelerometer biases have converged already before the car starts to move after 100 seconds. Since no information about the attitude of the navigation system is gained from the GPS position estimates when the car is stationary the gyro biases and attitude errors estimates don’t starts to converge until the car starts to move, which can be seen in Figure 7.3 and Figure 7.4. The average position errors in the x, y and z direction (ECEF-coordinates) are shown in Figures 7.5, 7.6 and 7.7, respectively. The maximum and root mean square position errors for the GPS aided INS respectively the stand alone GPS-receiver are presented in Table 7.1.
7.1 Driving Scenario

Figure 7.1: Estimated and true trajectory of a typical driving sequence. First the car is stationary for 100 seconds. Then it makes a wide turn and accelerates to 18 km/h, which it keeps until after the last turn. Finally the car slows down and stops.

Figure 7.2: The average errors in the accelerometer bias estimates for the driving sequence. The errors have already converged before the car starts moving after 100 seconds.
7.1 Driving Scenario

Figure 7.3: The average errors in the gyro bias estimates for driving sequence one. The errors don’t start to converge until the car begins moving and makes large turn after 100 seconds.

Figure 7.4: The average errors in the Euler angle estimates describing the body to ECEF frame rotation matrix $R_{eb}$. As for the gyro bias estimates, information is only gained when the car is turning.
7.1 Driving Scenario

Figure 7.5: The average position error in ECEF $x$-direction.

Figure 7.6: The average position error in ECEF $y$-direction.

Figure 7.7: The average position error in ECEF $z$-direction.

<table>
<thead>
<tr>
<th></th>
<th>Peak error [m]</th>
<th>RMSE [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS</td>
<td>35.7090</td>
<td>13.7987</td>
</tr>
<tr>
<td>GPS aided INS</td>
<td>17.2451</td>
<td>2.0948</td>
</tr>
</tbody>
</table>

Table 7.1: Peak and root mean square errors for the GPS aided INS and the stand alone GPS-receiver, respectively.
Chapter 8
Conclusions and Further Work

8.1 Conclusions
The proposed algorithm for integration between GPS and INS data is sensitive to its initial condition and tuning, since the extended Kalman filter linearizes the navigation equations around the inertial navigation system outputs. When properly initialized and tuned the GPS aided INS works well and clearly outperforms the stand alone GPS-receiver. At a ratio of 100 times between the INS and GPS sample ratio the integration reduces the error in the position estimates with a factor 6 compared to the stand alone GPS-receiver. Simulations show that the convergence rate of the attitude errors and gyro biases estimates depends not only on the tunable filter parameters, but also on the trajectory of the vehicle. For the attitude error and gyro bias states to be observable the IMU platform must rotate and therefore the beginning of a vehicle path should contain turns to ensure fast convergence of the filter.

Even though a very simplified model of the GPS receiver was used in the simulations, the results indicates that the proposed integration method also should be applicable to real world data.

8.2 Further Work
Before the GPS aided INS system can be taken into use, there are several problems to be solved and further work is needed in the following areas:

- Development of a method for initial calibration of the IMU sensors scale factors, biases, misalignment, et cetera.
- Investigation of methods for estimation of the delay between the actual measurement time and output time of the GPS-receivers position estimates.
- Development of a method for initial and online alignment of the IMU platform.
8.2 Further Work

- Investigation of the possibilities of using model based filtering of the IMU sensors output.
- Investigation of suitable error models for the IMU sensors and the gravity error.
- Construction and evaluation of the IMU platform and the hardware for data acquisition.
- Development of a software package for processing, analysis and presentation of the measured data.
Bibliography


Appendix A

The Kalman Filter

The Kalman filter, which was first introduced by R.E. Kalman in 1960 provides an optimal solution to the least-squares estimator problem for applications that can be represented as a linear system driven and disturbed by Gaussian noise. The Kalman filter is a recursive algorithm that solves the minimum least-squares error problem using state-space methods. The Kalman filter equations will here be derived using an innovation approach. More details about the Kalman filter can be found in [8, 11]. The derivation below follows the derivation in [8].

A.1 Derivation of the Kalman Filter Equations

Define a discrete state-space model such as

\[
\begin{align*}
  x_{k+1} & = F_k x_k + G_k w_k \\
  y_k & = H_k x_k + v_k
\end{align*}
\]  

(A.1)

(A.2)

where the \( x_k \) is the \( n \times 1 \) state vector. The processes \( v_k \) and \( w_k \) are assumed to be zero mean white noise process, with

\[
E\left[ \begin{bmatrix} v_k \\ w_k \end{bmatrix} \begin{bmatrix} v_l^* \\ w_l^* \end{bmatrix} \right] = \begin{bmatrix} Q_k & S_k \\ S_k^* & R_k \end{bmatrix} \delta(l-k)
\]  

(A.3)

Let \( \hat{y}_k^- \) denote the linear least square estimator of \( y_k \) given the observation sequence \( \{y_0, y_1, y_2, ..., y_{k-1}\} \). Define the error between the one-step ahead predicted value and true value as

\[
e_k = \hat{y}_k^- = y_k - \hat{y}_k^-
\]  

(A.4)

Here \( e_k \) can be seen as the new information brought into the system by \( y_k \), therefore referred to as the innovation. Since \( \hat{y}_k^- \) is the least square estimator, the error per definition is orthogonal to all the subspace span by \( \{y_0, y_1, y_2, ..., y_{k-1}\} \), denoted \( L(y^{k-1}) \); See Figure A.1. Since error \( e_k \) is a linear combination of \( y_k \) and \( \hat{y}_k^- \) not only will the innovations and the sequence \( \{y_0, y_1, y_2, ..., y_{k-1}\} \) span the same subspace, i.e \( L(y^{k-1}) = L(e^{k-1}) \), the innovation sequence will also be a white sequence. Now raising the problem to find the least square estimator \( \hat{y}_k^- \), which in terms of projections can be written as
A.1 Derivation of the Kalman Filter Equations

Figure A.1: The projection of a vector $y_k$ onto the subspace $L(y^{k-1})$ corresponds to the least square estimate of the vector $y_k$ given the sequence $\{y_0, y_1, ..., y_{k-1}\}$. Note that the prediction error is orthogonal to the subspace.

$$y_k^- = \text{Proj}(y_k | L(y^{k-1}))$$
$$= \text{Proj}(H_k x_k + v_k | L(y^{k-1}))$$
$$= H_k \text{Proj}(x_k | L(y^{k-1})) + \text{Proj}(v_k | L(y^{k-1}))$$
$$= H_k \hat{x}_k^-$$ (A.5)

Here Proj($\cdot$) denotes the projection operation and the last equality is a result of $\{v_k\}$ being a white sequence. Thus it turned out that finding the linear least-squares estimator of $y_k^-$ is equivalent to finding the state prediction $\hat{x}_k^-$. Since $L(y^k) = L(e^k)$ and $\{e_k\}$ is a white sequence the state prediction can be written as

$$\hat{x}_{k+1}^- = \text{Proj}(x_{k+1} | L(e^k))$$
$$= \text{Proj}(x_{k+1} | L(e^k))$$
$$= \text{Proj}(x_{k+1} | e_k) + \text{Proj}(x_{k+1} | L(e^{k-1}))$$
$$= E(x_{k+1} | e_k^e) R_{e_k e_k}^{-1} e_k + \text{Proj}(x_{k+1} | L(e^{k-1}))$$ (A.6)

where the innovation covariance matrix is defined as $R_{e_k e_k} \triangleq E(e_k e_k^T)$. This is almost the desired form and would be exactly so if the term $\text{Proj}(x_{k+1} | L(e^{k-1}))$ could be expressed in terms of $\hat{x}_k^-$ and $e_k$. Projecting $x_{x+1}$ onto the linear subspace $L(y^{k-1})$ shows that

$$\text{Proj}(x_{k+1} | L(y^{k-1})) = \text{Proj}(H_k x_k + G_k v_k | L(y^{k-1}))$$
$$= H_k \text{Proj}(x_k | y^{k-1}) + G_k \text{Proj}(v_k | L(y^{k-1}))$$
$$= H_k \hat{x}_k^-$$ (A.7)
where the last equality is a result of $E\{v_k y_l^*\} = 0$ for $l < k$. Thus a recursive algorithm for the state prediction may be written as

$$e_k = y_k - H_k \dot{x}_k^- \tag{A.8}$$

$$\dot{x}_{k+1}^- = F_k \dot{x}_k^- + K_{p,k} e_k \tag{A.9}$$

$$K_{p,k} = E\{x_{k+1} e_k^*\} R_{e,k}^{-1} \tag{A.10}$$

$$e_0 = y_0, \quad \dot{x}_0^- = 0 \tag{A.11}$$

Now confronting the problem to find a recursion for $K_{p,k}$ and $R_{e,k}$, which are non-random quantizes and determined only by our model assumptions. Introducing the a posterior error covariance matrix $P_k^-$ defined as

$$P_k^- \triangleq E\{x_k^- x_k'^{-}\} \tag{A.12}$$

where

$$\dot{x}_k^- = x_k - \dot{x}_k^- \tag{A.13}$$

The innovation covariance matrix $R_{e,k}$ can then be written in terms of $P_{k|h-1}$ as

$$R_{e,k} = E\{e_k e_k^*\} = E\{(y_k - H_k \dot{x}_k^-) (y_k - H_k \dot{x}_k^-)^*\} = E\{(H_k \dot{x}_k^- + v_k) (H_k \dot{x}_k^- + v_k)^*\} = H_k P_k^- H_k^* + R_k \tag{A.14}$$

and the prediction Kalman gain $K_{p,k}$ as

$$K_{p,k} \triangleq E\{x_{k+1} e_k^*\} R_{e,k}^{-1} = (F_k E\{x_k e_k^*\} + G_k E\{v_k e_k^*\}) R_{e,k}^{-1} \tag{A.15}$$

where

$$E\{x_k e_k^*\} = E\{x_k (H_k \dot{x}_k^- + G_k v_k)^*\} = E\{x_k \dot{x}_k'^{-}\} H_k^* + E\{x_k v_k\} G_k = P_k^- H_k^* \tag{A.16}$$

and

$$E\{v_k e_k^*\} = E\{v_k (H_k \dot{x}_k^- + w_k)^*\} = E\{w_k x_k\} H_k^* + E\{v_k w_k^*\} = S_k \tag{A.17}$$

Thus the Kalman gain is becomes

$$K_{p,k} = (F_k P_k^- H_k^* + G_k S_k) R_{e,k}^{-1} \tag{A.18}$$

What remains is to find a recursion for $P_k^-$. As soon shall be shown $P_k^-$ can be computed via the discrete-time Riccati recursion. Using the expression for the state predictor in (A.9) the prediction error may be written as
\[ x_{k+1} = x_{k+1} - F_k \hat{x}_k - K_{p,k} e_k \]
\[ = x_{k+1} - F_k \hat{x}_k - K_{p,k} (H_k \hat{x}_k + w_k) \]
\[ = F_k \hat{x}_k + G_k w_k - K_{p,k} H_k \hat{x}_k - K_{p,k} w_k \]
\[ = (F_k - K_{p,k} H_k) \hat{x}_k - [G_k - K_{p,k}] v_k \] (A.19)

Then it straightforward to show that the a posteriori error covariance matrix may be written as
\[ P_{k+1} = E \{ \hat{x}_{k+1} \hat{x}_{k+1}^* \} \]
\[ = (F_k - K_{p,k} H_k) P_k (F_k - K_{p,k} H_k)^* \]
\[ + [G_k - K_{p,k}] \begin{bmatrix} Q_k & S_k^* \\ S_k & R_k \end{bmatrix} \begin{bmatrix} G_k \\ -K_{p,k} \end{bmatrix} \] (A.20)

After some algebra this reduces to
\[ P_{k+1} = F_k P_k^* F_k^* + G_k Q_k G_k^* - K_{p,k} R_{e,k} K_{p,k}^* \] (A.21)
which indeed is the discrete-time Riccati recursion. Bring it all together the prediction form of the Kalman filter is shown in Table A.1.

<table>
<thead>
<tr>
<th>Table A.1: The Kalman filter recursions in prediction form.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( e_k = y_k - H_k \hat{x}_k )</td>
</tr>
<tr>
<td>( \hat{x}_{k+1} = F_k \hat{x}<em>k + K</em>{p,k} e_k )</td>
</tr>
<tr>
<td>( K_{p,k} = (F_k P_k^* H_k^* + G_k S_k H_k^*) R_{e,k}^{-1} )</td>
</tr>
<tr>
<td>( P_{k+1} = F_k P_k^* F_k^* + G_k Q_k G_k^* - K_{p,k} R_{e,k} K_{p,k}^* )</td>
</tr>
<tr>
<td>( P_0 = E { x_0 x_0^* } ) ( \hat{x}_0 = E { x_0 } )</td>
</tr>
</tbody>
</table>

In many applications it is of interest to obtain an estimate of \( x_k \) given the observations \( \{ y_0, y_1, y_2, ... y_k \} \), i.e the filtered estimate. The derived Kalman filter equations can easily be modified to output both the filtered and predicted state estimate using the innovation approach once more. Starting by projecting \( x_k \) onto the subspace \( L(e^k) \) the estimator may be written as
\[ \hat{x}_k = \text{Proj}(x_k | L(e^k)) \]
\[ = \hat{x}_k + E \{ x_k e_k^* \} R_{e,k}^{-1} e_k \]
\[ = \hat{x}_k + K_{f,k} e_k \] (A.22)

where the Kalman filter gain \( K_{f,k} \) is defined as
A.1 Derivation of the Kalman Filter Equations 46

\[ K_{f,k} \triangleq E\{x_k e_k^*\} R_{e,k}^{-1} \]
\[ = E\{x_k (H_k x_k^- + v_k)^*\} R_{e,k}^{-1} \]
\[ = P_k^- H_k^* R_{e,k}^{-1} \]
\[ = P_k^- H_k^* (H_k P_k^- H_k^* + R_k)^{-1} \] (A.23)

and the error covariance \( P_k \) as

\[ P_{k|k} \triangleq E\{\hat{x}_k \hat{x}_k^*\} \]
\[ = E\{x_k x_k^*\} \]
\[ = E\{(x_k - \hat{x}_k) \hat{x}_k^*\} \]
\[ = E\{x_k - \hat{x}_k^- - K_{f,k} e_k\} x_k^* \}
\[ = E\{\hat{x}_k^- x_k^*\} - K_{f,k} E\{e_k x_k^*\} \]
\[ = P_k^- - K_{f,k} H_k P_k^- \] (A.24)

This far expressions for updating the predicted estimate \( \hat{x}_k^- \) to the filtered estimate \( \hat{x}_k \) and the covariance matrix \( P_k^- \) to \( P_k \) have been derived. If a update from the filtered estimate \( \hat{x}_k \) to the predicted \( \hat{x}_{k+1}^- \), as well as for the covariance matrix \( P_k \) to the a posterior covariance matrix \( P_{k+1}^- \) could be found, then the derived Kalman filter recursions could be rewritten to output the filtered estimate as well. Assuming no correlation between the process and measurement noise, i.e \( S_k = 0 \), the time update can be found as

\[ \hat{x}_{k+1}^- = \text{Proj}(x_{k+1} | L(e^k)) \]
\[ = \text{Proj}(F_k x_k + G_k v_k | L(e^k)) \]
\[ = F_k \hat{x}_k \] (A.25)

and

\[ P_{k+1}^- = E\{\hat{x}_{k+1}^- \hat{x}_{k+1}^-^*\} \]
\[ = E\{(F_k x_k + G_k v_k)(F_k \hat{x}_k + G_k v_k)^*\} \]
\[ = F_k P_k F_k^* + G_k Q_k G_k^* \] (A.26)

If there are correlation between the measurement and process noise sources the time update becomes quit complicated and expressions for this case may be found in [8]. Collecting the derived relations ships the Kalman filter recursion can be written in terms of measurement and time updates as in Table A.2.
A.1 Derivation of the Kalman Filter Equations

\[ x_{k+1} = F_k \hat{x}_k \]
\[ y_{k+1} = H_k \hat{x}_{k+1} \]
\[ P_{k+1} = F_k P_k F_k^T + G_k Q_k G_k^T \]

**Time update.**

**Measurement update.**

\[ K_{f,k} = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1} \]
\[ \hat{x}_k = \hat{x}_k + K_{f,k} (y_k - H_k \hat{x}_k) \]
\[ P_k = P_k^T - K_{f,k} H_k P_k \]

| \textbf{Table A.2: The Kalman filter recursions in time and measurement update form.} |