UCGE Reports
Number 20219

Department of Geomatics Engineering

Estimation Techniques for Low-Cost Inertial Navigation

(URL: http://www.geomatics.ucalgary.ca/links/GradTheses.html)

by

Eun-Hwan Shin

May 2005

THE UNIVERSITY OF CALGARY

Estimation Techniques for Low-Cost Inertial Navigation

by

Eun-Hwan Shin

A DISSERTATION

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF GEOMATICS ENGINEERING

CALGARY, ALBERTA

MAY, 2005

Abstract

Low-cost inertial sensors are characterized by high noise and large uncertainties in the outputs such as bias, scale factor and non-orthogonality. Consequently, errors associated with a low-cost INS in terms of position, velocity and attitude grow rapidly in stand-alone mode. If good performance can be achieved with low-cost inertial measurement units (IMUs), cost in existing applications can be reduced and the development of new applications may be made feasible.

As most of the uncertainties exist in the sensor error behaviour, calibration would improve the accuracy significantly. Intensive calibration, however, would also increase the cost of using an IMU. Another way to improve the accuracy will be by augmenting the IMU with many other aiding sensors: for example, odometers or speedometers.

Choosing an appropriate estimation method is a key problem when developing an aided INS. Currently, there are three approaches: (i) traditional linearized Kalman filter (LKF) or the extended Kalman filter (EKF); (ii) sampling-based filtering such as the unscented Kalman filter (UKF) and particle filters; (iii) artificial intelligence

(AI)-based estimation such as artificial neural networks (ANN) and adaptive neural fuzzy information systems (ANFIS). Of these approaches,the performance of the UKF is compared to that of other existing methods. The performance of the unscented Kalman smoother (UKS) is also compared with that of the Rauch-Tung-Striebel (RTS) smoother.

Tests have been conducted using micro-electro-mechanical systems (MEMS)-based IMUs. The most remarkable advantage of the UKF over existing methods is that it can handle large and small attitude errors seamlessly. Thus, the UKF is preferred in situations such as those where the heading error approaches the limit of what an EKF/LKF can deal with efficiently. The EKF/RTS approach would still be chosen if the attitude errors can be kept small such that the error dynamics are linear. The UKF can unify INS error models, although it cannot deal with complete uncertainties in attitude Therefore, the development stage can be simplified by using the UKF.

Acknowledgments

As is always the case, researches for a Ph.D. could not be completed without support of other peoples. First of all, I would like to express my sincere gratitude to my supervisor, Dr. Naser El-Sheimy. He continuously encouraged throughout my research and provided financial support during my graduate studies. Special thanks go to the members of my supervisory committee: Dr. Bruno Scherzinger and Dr. Yang Gao. Especially, Dr. Scherzinger's lectures and comments on aided inertial navigation have directly influenced the design of the INS developed in the dissertation. I would like to acknowledge all other examining committee members for taking time to read the draft. Dr. Klaus Peter Schwarz and Dr. Scherzinger gave many constructive suggestions in restructuring the draft.

Applanix Corporation donated their systems, which have been used to generate reference datasets in the performance comparison. Waypoint Consulting Inc. provided the GrafNav$^{\mathrm{TM}}$ software that have been used to generate DGPS solutions to aid low-cost INSs.

I would like to thank Dr. Simon Julier, original developer of the UKF, for many comments and discussions on the UKF. He also introduced me his recent research activities that have inspired me. Edgar Kraft, Physikalisches Institut der Universität Bonn, Germany, is also thanked for the discussions on designing the quaternion-based UKF. Without these international discussions, I could not have succeeded in developing the UKF in time. Edgar Kraft and Lasse Klingbeil also gave me a nice tour of their institution and city of Bonn. I was greatly impressed by their developments.

Dr. Xioji Niu, Dr. Kai-Wei Chiang and Haiying Hou are thanked for the participation in the field tests. Dr. Xiaoji Niu, in particular, gave me much constructive feedback on improving the functionality of the software and assisted me during the data processing. Haiying Hou provided me with several papers on the Allan variance analysis. Dr. Michael Kern is thanked for translating the summary into German for an article published in Germany. Cameron Ellum and Thilanka Galappaththi are thanked for their effort on proof-reading. I would also like to thank those who have spent most of the time together in the office, ENF319A.

This research was supported in part by the research grants from Natural Science and Engineering Research Council of Canada (NSERC) and Geomatics for Informed Decisions (GEOIDE) Network Centers of Excellence (NCE) to Dr. Naser El-Sheimy. I would like to thank donors of the following awards: Best paper in track award in IEEE PLANS 2004, Student sponsorship in ION GNSS 2004, and Innovations in mobile mapping award 2004. Receiving these awards encouraged my research greatly.

Finally, and most importantly, I would like to thank my wife, Su Nam Lee, my son, Joonwoo Shin, and my daughter, Hain Shin, who have given delight to my life.

Contents

List of Tables

List of Figures

List of Symbols, Abbreviations, Nomenclature

The notations used in the dissertation will be introduced here.

1. Conventions

   (a) Matrices are denoted in upper case bold letters.

   (b) Vectors are denoted in lower case bold letters.

   (c) The coordinate frames that are involved in the vector transformation are denoted as subscript and superscript. For instance, $\mathbf{C}_{\mathrm{b}}^{\mathrm{n}}$ is the direction cosine matrix from the body frame (b) to the navigation frame (n). For angular rate vectors, subscripts denote the reference and target frame, and superscripts denote the projected or realized frame. For example, $\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{n}}$ represents the angular rate vector of the body frame with respect to the inertial frame projected to the navigation frame.

   (d) Operators are defined as:

$\dot{\cdot}$	time derivative
$:=$	substitution
$\lVert \cdot \rVert$	Euclidean norm
$\hat{\cdot}$	estimated or computed values
$\bar{\cdot}$	mean
$\tilde{\cdot}$	measured values
$\star$	quaternion product
δ	error of, correction to
$\delta(\cdot)$	Dirac delta function
Δ	increment (difference) of
∇	partial derivative of a function with respect to the state
$\cdot^{-1}$	matrix and quaternion inverse
$\cdot^{T}$	matrix transpose
$\times$	cross product
$(\cdot\times)$	cross-product or skew-symmetric form of a vector
$(\cdot\times)_{e+}$	extended cross-product type (+) of a vector
$\mathrm{diag}(\cdot)$	diagonal matrix form of a vector
$E[\cdot]$	expectation of
$\Xi(\cdot,\cdot)$	defined in Eq. (2.44)

2. Symbols

α	the first scaling parameter in scaled UT
β	the second scaling parameter in scaled UT
κ	Scaling parameter for the first UT
$\gamma, \boldsymbol{\gamma}$	non-orthogonality
λ	geodetic longitude
$\boldsymbol{\nu}$	innovations vector
$\boldsymbol{\omega}$	angular rate vector
φ	geodetic latitude
ϕ	roll
$\boldsymbol{\phi}$	rotation vector pertaining to the error of the attitude DCM
Φ	transition matrix
ρ	pseudo range
θ	pitch
$\mathbf{\Theta}$	orthogonal matrix
$\delta\boldsymbol{\theta}$	rotation vector pertaining to position error
ψ	heading
σ	standard deviation
$\boldsymbol{\psi}$	rotation vector pertaining to the platform tilt error
$\boldsymbol{\xi}$	rotation vector pertaining to the Earth frame rotation
$\boldsymbol{\zeta}$	rotation vector pertaining to the navigation frame rotation
$\mathbf{b}$	bias

$\mathbf{c}$	parameters for continuous-time sensor error model
$\mathbf{C}$	direction cosine matrix
$\mathbf{d}$	parameters for discrete-time sensor error model
e, $\mathbf{e}$	measurement noise
$\mathbf{f}$	specific force vector
$\mathbf{F}$	dynamics matrix
g, $\mathbf{g}$	gravity
$g(\cdot)$	general scalar nonlinear transformation
$\mathbf{G}$	system noise input mapping matrix
h	ellipsoidal height
$\mathbf{h}[\cdot]$	measurement function
$\mathbf{H}$	design matrix for measurements
$\mathbf{I}$	identity matrix
$\boldsymbol{\mathcal{J}}$	Jacobian matrix
$\mathbf{K}$	Kalman gain matrix
$\boldsymbol{\ell}$	lever arm vector
$\mathbf{m}[\cdot]$	INS mechanization function
$\mathbf{P}$	covariance matrix of state vector
$\mathbf{q}$	quaternion
$\mathbf{Q}_k$	covariance matrix of system noise sequence vector
$\mathbf{Q}(t)$	spectral density matrix

r	range
$\mathbf{r}$	position vector
$\mathbf{R}$	covariance matrix of measurement error vector
$\mathbf{s}$	scale factor
$\mathbf{v}$	velocity vector
w	scalar system noise and weight
$\mathbf{w}$	system noise vector
$\mathbf{x}$	state vector
$\boldsymbol{\mathcal{X}}$	sigma point (SP)
$\boldsymbol{\mathcal{Y}}$	transformed SP through the GPS position measurement model
$\mathbf{z}$	measurement vector
$\boldsymbol{\mathcal{Z}}$	transformed SP through the measurement function

3. Acronyms

AI	Artificial Intelligence
ANFIS	Adaptive Neural Fuzzy Information Systems
ANN	Artificial Neural Networks
ASIC	Application-Specific Integrated Circuit
BLP	Backward Linear Prediction
DCM	Direction Cosine Matrix
DGPS	Differential GPS

EKF	Extended Kalman Filter
FLP	Forward Linear Prediction
FOG	Fiber Optic Gyroscope
GC	Geographic Consistent
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LHU	Large Heading Uncertainty
LKF	Linearized Kalman Filter
MEMS	Micro-Electro-Mechanical Systems
NED	North-East-Down
PC	Platform Consistent
PDF	Probability Density Function
PSD	Power Spectral Density
RTS	Rauch-Tung-Striebel
SHU	Small Heading Uncertainty
SUT	Scaled Unscented Transformation
SV	Satellite Vehicle
SP	Sigma Point
UAV	Unmanned Aerial Vehicles
UKF	Unscented Kalman Filter

UKS	Unscented Kalman Smoother
UT	Unscented Transformation
ZIHR	Zero Integrated Heading Rate
ZUPT	Zero Velocity Update

Chapter 1

Introduction

1.1 Background and Objectives

The last decade has shown an increasing demand for small-sized and low-cost inertial navigation systems (INSs) for use in many applications such as personal navigation, car-navigation, unmanned aerial vehicles (UAVs) and general aviation. Advances in fiber optic gyroscope (FOG) and micro-electro-mechanical systems (MEMS) technologies have shown promising signs toward the development of such systems. Compared to higher-grade systems, a low-cost INS can experience large position and attitude errors over short time intervals. This is mainly due to large uncertainties in the sensor output and therefore INSs built on these sensors are vulnerable to nonlinear error behaviour, especially when the attitude errors become very large.

If the accuracy of a low-cost INS can be improved, cost can be reduced in existing applications and new applications can emerge. As most of the uncertainties exist in the sensor error behaviour, calibration would improve the accuracy significantly. However, intensive calibration would increase the cost significantly. Another way to improve the accuracy would be aiding the INS with other complementary sensors or navigation-related information. Choosing an appropriate estimation method is a key problem in developing an aided INS.

The analysis and prediction of complex dynamic and nonlinear phenomena have become very important in various fields of research (Kitagawa and Higuchi, 2001). Navigation is a typical field of nonlinear dynamic systems and in the core of navigation system development lies the problem of estimating the states of a dynamic system. When it comes to state estimation for nonlinear systems, however, there is no single solution available that clearly outperforms all other strategies (Nørgaard et al., 2000).

Currently, three approaches have been identified in the research on estimation methods for integrated navigation systems:

1. The linearized Kalman filter (LKF) or the extended Kalman filter (EKF);

2. Sampling-based filters, such as the unscented Kalman filter (UKF) (Shin and El-Sheimy, 2004; Shin, 2004) and particle filters (Bergman, 1999); and

3. Artificial intelligence (AI)-based methods, such as artificial neural networks (ANN) (Chiang and El-Sheimy, 2002) or adaptive neural fuzzy information systems (ANFIS) (El-Sheimy et al., 2004).

The LKF (or EKF) has long been used in designing navigation software; for instance see Farrell and Barth (1998), Rogers (2000) and Schwarz and Wei (2000). The EKF simply applies the Taylor series expansion for the nonlinear system along with observation equations, and takes terms to the first order, where the probability density function (PDF) is approximated by a Gaussian distribution (Gordon et al., 1993); for details see Gelb (1974), Maybeck (1994b) and Grewal and Andrews (2001). In practice, however, the EKF has shown several limitations. First, the derivation of the Jacobian matrices for both the system and observation equations is nontrivial in most applications and often leads to significant implementation difficulties (Julier et al., 1995). Use of distinct error models results in different styles of filter, each of which has its own application; for example, the phi-angle error model or the psi-angle error model (Benson Jr., 1975), and the large heading uncertainty (LHU) or small heading uncertainty (SHU) models (Scherzinger, 1996). Hence, filter designers should be extremely cautious in choosing an appropriate error model, or else critical errors can occur during the development of a navigation system. Second, only small errors are allowed to be delivered to the EKF (Sukkarieh, 2000, p. 21); otherwise, in the presence of nonlinear error behaviour, the first-order approximations can cause biased solutions (Maybeck, 1994b, p. 218) and inconsistency of the covariance update (Lerro and Bar-Shalom, 1993), which can lead to filter instability. Although second-order filters can correct the bias terms, their computational cost is high because the calculation of second order derivatives (Hessian) is required.

In sampling-based filtering methods, computation of derivatives is not required. Sequential Monte Carlo filters, also known as particle filters, have been developed for nonlinear/non-Gaussian processes based on Bayesian filtering theory. Basic sequential Monte Carlo methods were already introduced in the 1950s to solve problems in physics (Andrieu et al., 2003). These methods were overlooked and forgotten due to the lack of computing power (Doucet, 1998), and it was only after the early 1990s that these methods could be used for practical problems such as target tracking for radar, communications, econometrics and computer vision (Godsill et al., 2000). See, for instance, Gordon et al. (1993), Kitagawa (1996), Liu and Chen (1995, 1998), and Doucet (1998). Bergman (1999) used a particle filter in terrain navigation. Since particle filters do not make any assumption on the PDF, they have shown superior numerical accuracy to other filtering methods. However, particle filters are not suitable for high-dimensional systems with a high repetition rate, unless parallel processing is employed. This is mainly because enormous random particles (samples) must be generated and transformed through the nonlinear function.

The development of the UKF was pioneered by Julier and Uhlmann (1996) and Julier et al. (1995, 2000). Other researches were, in turn, stimulated by this work. Nørgaard et al. (2000) derived the second-order divided difference (DD2) filter based on the central divided difference and showed its similarity to the UKF. Van der Merwe et al. (2000) used the UKF to improve the performance of the particle filter. Lefebvre et al. (2002) argued that the UKF is a special case of the linear regression

Kalman filter. The underlying intuition of the UKF is that, with a fixed number of parameters, it should be easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function or transformation (Julier and Uhlmann, 1996). The state distribution is again represented by a Gaussian random variable, but is now specified using a minimal set of carefully chosen sample points, called the sigma points (SPs). SPs capture the true mean and covariance of the PDF and, when propagated through the true nonlinear system, capture the transformed mean and covariance accurately up to the second order for any nonlinearity (Wan and van der Merwe, 2001, p. 228). The difference between the UKF and particle filters is that the former takes samples deterministically while the latter randomly selects samples.

A fundamental difference between AI-based estimation methods and the other two types of estimation methods is that AI-based methods do not use any mathematical models in the system dynamics and measurements; for instance, see Chiang and El-Sheimy (2002) and El-Sheimy et al. (2004). Although AI-based methods are simpler in terms of design, there are also limitations to this approach. First, they do not use any statistical information as input, and they do not output the statistics associated with the solution. The statistics of the output play a significant role in post-processing applications such as surveying. In addition, as will be discussed in Chapters 3 and 4, for the LKF, EKF, and sampling-based approaches, smoothing can be applied to obtain best estimates superior in performance to the AI-based methods. Thus, AI-based methods are not suitable for post-processing applications.

Second, the mathematical models of vehicle motion and measurements are well developed and AI-based methods developed to date neglect these legacies. Thus, AI-based methods can show superior performance only when long measurement outages occur. Measurement outages, however, are not so important an issue in multi-sensor integrated navigation systems because there almost always exists at least one measurement update option. For instance, in land-vehicle navigation, outages of the global positioning system (GPS) can be relieved by measurements from odometers or speedometers. Third, if the vehicle experiences dynamics range not included in the training set, acceptable performance of the estimator can hardly be expected. Therefore, the integrity of the system may not be guaranteed. Finally, hardware cost will be increased for on-line training due to the heavy computational burden.

The topic of this dissertation is unscented estimation techniques for low-cost aided inertial navigation. AI-based estimation methods are disregarded because smoothers have not yet been developed. As existing methods, an LKF with a feedback loop, which can be considered as the EKF, and the Rauch-Tung-Striebel (RTS) smoother will be developed first. Then, a UKF and an unscented Kalman smoother (UKS) will be developed as new methods. This dissertation will, in particular, investigate the following:

1. Design and implement of the UKF for aided INS.

2. Analysis of the UKF performance against a comparable EKF.

3. Demonstration of performance advantages that the UKF has over the EKF.

Performance of the EKF and UKF will be compared experimentally using MEMS IMUs.

1.2 Dissertation Outline

In Chapter 2, the essential building blocks for an aided INS are developed. Attitude representations frequently used in an INS will be introduced. Various reference frames will be defined. Forward and backward digital INS mechanization algorithms will be presented. Mathematical models for several aiding sources will be discussed. Alignment methods will be reviewed and evaluated for their pertinence in a low-cost INS.

In Chapter 3, the EKF and the RTS smoother are developed. Various INS error models will be introduced. The traditional LKF will be discussed in detail together with its feedback structure. The relationship between the LKF and the rigorous style of EKF will be addressed. Finally, the implementation of the RTS smoother for an aided INS will be discussed.

The development of the UKF and UKS is covered in Chapter 4. Proposals for the unscented transformations (UT) will be reviewed. Then, the UT will be extended to the development of a quaternion-based forward and backward UKF. These two solutions will be combined by the UKS.

In Chapter 5, the performance of the EKF/RTS and the UKF/UKS will be compared using MEMS IMUs.

Finally, in Chapter 6, the major findings of this research will be summarized. Conclusions will be drawn and topics warranting further research will be suggested.

Chapter 2

Fundamentals of Aided Inertial Navigation

In this chapter, background information will be provided that serves as a basis for the development and analysis in later chapters. Since attitude representations play a significant role in INS development, Section 2.1 summarizes commonly used attitude parameterizations and describes conversions from one form to another. Section 2.2 defines various reference frames required in the implementation of an INS. The forward and backward INS mechanization algorithms are developed in Sections 2.3 and 2.4, respectively. The aiding sources for a low-cost INS will be described in Section 2.5. Finally, in Section 2.6, current alignment techniques are reviewed and their problems when applied to development of a low-cost INS are discussed.

2.1 Attitude Parameterizations

Numerous attitude parameterization techniques are surveyed in Shuster (1993). Among them, the direction cosine matrix (DCM), quaternion, rotation vector, and Euler angles are commonly used in inertial navigation; for example, see Savage (2000). This section discusses relationships between these parameterizations with the emphasis on quaternions. Readers interested in a more thorough treatment of rotations and quaternions are referred to Altmann (1986) and Kuipers (1999). General frames 'a,' 'b,' and 'c' will be used throughout this section. The quaternion, rotation vector, and DCM for the transformation from the 'b' frame to the 'a' frame are represented by $\mathbf{q}_{\mathrm{b}}^{\mathrm{a}}$, $\boldsymbol{\phi}$, and $\mathbf{C}_{\mathrm{b}}^{\mathrm{a}}$, respectively. The "rotation vector" defines an axis of rotation and magnitude for a rotation about the rotation vector (Savage, 2000, p. 3-21). As illustrated in Figure 2.1, the frame 'b' is initially aligned with the frame 'a,' and then is rotated about $\boldsymbol{\phi}$ to obtain the final attitude of the frame 'b,' equivalent to rotating the frame 'a' to obtain the frame 'b.'

2.1.1 Quaternion Algebra

The quaternion $\mathbf{q}_{\mathrm{b}}^{\mathrm{a}}$ is a four-dimensional vector composed of a scalar part s and a three-dimensional vector part $\mathbf{v}$:

$$\mathbf{q}_{\mathrm{b}}^{\mathrm{a}} \equiv \begin{bmatrix} s \\ \mathbf{v} \end{bmatrix} . \tag{2.1}$$

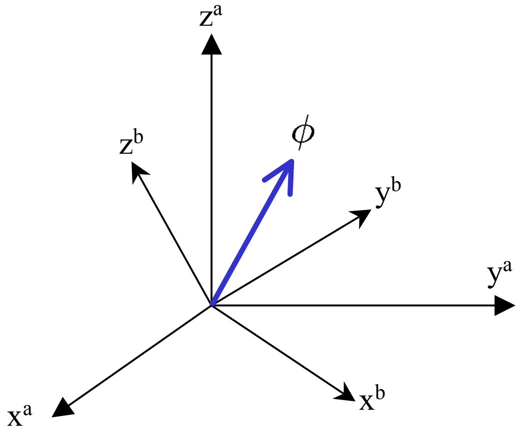

Figure 2.1: The rotation vector

We will consider only the case $s \geq 0$ because $\mathbf{q}_{\mathrm{b}}^{\mathrm{a}}$ is equivalent to $-\mathbf{q}_{\mathrm{b}}^{\mathrm{a}}$, which will be explained in Section 2.1.3. The inverse (conjugate) of the quaternion is written as

$$(\mathbf{q}_{\mathrm{b}}^{\mathrm{a}})^{-1} = \begin{bmatrix} s \\ \\ -\mathbf{v} \end{bmatrix}. \tag{2.2}$$

The product of the two quaternions

$$\mathbf{q}_{\mathrm{c}}^{\mathrm{a}} \equiv \begin{bmatrix} s_1 \\ \\ \mathbf{v}_1 \end{bmatrix}, \quad \mathbf{q}_{\mathrm{b}}^{\mathrm{c}} \equiv \begin{bmatrix} s_2 \\ \\ \mathbf{v}_2 \end{bmatrix}$$

is defined as

$$\mathbf{q}_{\mathrm{b}}^{\mathrm{a}} = \mathbf{q}_{\mathrm{c}}^{\mathrm{a}} \star \mathbf{q}_{\mathrm{b}}^{\mathrm{c}} \equiv \begin{bmatrix} s_1 s_2 - \mathbf{v}_1^T \mathbf{v}_2 \\ \\ s_1 \mathbf{v}_2 + s_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2 \end{bmatrix}, \tag{2.3}$$

where $\star$ denotes the quaternion product (Miller, 1983).

2.1.2 Rotation Vector Rate

Assume the frame 'b' is rotating with respect to the frame 'a' with the rate $\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ab}}$. The rotation vector rate can then be written as

$$\begin{aligned}\dot{\boldsymbol{\phi}} &= \boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ab}} + \frac{1}{2}\boldsymbol{\phi} \times \boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ab}} + \frac{1}{\|\phi\|^2}\left(1 - \frac{\|\phi\| \sin \|\phi\|}{2(1-\cos\|\phi\|)}\right) \boldsymbol{\phi} \times \left(\boldsymbol{\phi} \times \boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ab}}\right) \qquad (2.4)\\ &\approx \boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ab}} + \frac{1}{2}\boldsymbol{\phi} \times \boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ab}} + \frac{1}{12}\boldsymbol{\phi} \times \left(\boldsymbol{\phi} \times \boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ab}}\right),\end{aligned}$$

where $\dot{\boldsymbol{\phi}} = d\boldsymbol{\phi}/dt$. Eq. (2.4) was first derived by Bortz (1971) and, hence, referred to as the Bortz equation. The second and third terms on the right hand side of Eq. (2.4) are due to non-inertially measurable angular motion (non-commutativity rate vector) (Bortz, 1971).

2.1.3 Quaternion in Terms of Rotation Vector

The quaternion in Eq. (2.1) can be expressed in terms of the rotation vector $\boldsymbol{\phi}$ as follows (Savage, 2000, p. 3-48):

$$\begin{aligned}\mathbf{q}^{\mathrm{a}}_{\mathrm{b}} &= \begin{bmatrix} \cos\|0.5\boldsymbol{\phi}\| \\ \dfrac{\sin\|0.5\boldsymbol{\phi}\|}{\|0.5\boldsymbol{\phi}\|} 0.5\boldsymbol{\phi} \end{bmatrix}, \qquad (2.5)\\ \cos\|0.5\boldsymbol{\phi}\| &= 1 - \frac{\|0.5\boldsymbol{\phi}\|^2}{2!} + \frac{\|0.5\boldsymbol{\phi}\|^4}{4!} - \cdots,\\ \frac{\sin\|0.5\boldsymbol{\phi}\|}{\|0.5\boldsymbol{\phi}\|} &= 1 - \frac{\|0.5\boldsymbol{\phi}\|^2}{3!} + \frac{\|0.5\boldsymbol{\phi}\|^4}{5!} - \cdots,\end{aligned}$$

where $\|\cdot\|$ denotes the Euclidean norm. $-\mathbf{q}_{\mathrm{b}}^{\mathrm{a}}$ can be expressed as follows:

$$-\mathbf{q}_{\mathrm{b}}^{\mathrm{a}} = \begin{bmatrix} -\cos\|0.5\boldsymbol{\phi}\| \\ -\dfrac{\sin\|0.5\boldsymbol{\phi}\|}{\|0.5\boldsymbol{\phi}\|}0.5\boldsymbol{\phi} \end{bmatrix} = \begin{bmatrix} \cos(\pi - \|0.5\boldsymbol{\phi}\|) \\ -\dfrac{\sin(\pi - \|0.5\boldsymbol{\phi}\|)}{0.5(2\pi - \|\boldsymbol{\phi}\|)}0.5(2\pi - \|\boldsymbol{\phi}\|)\dfrac{\boldsymbol{\phi}}{\|\boldsymbol{\phi}\|} \end{bmatrix}, \qquad (2.6)$$

which states that $-\mathbf{q}_{\mathrm{b}}^{\mathrm{a}}$ represents a rotation with the magnitude of $2\pi - \|\boldsymbol{\phi}\|$ about the opposite direction of $\boldsymbol{\phi}$. Figure 2.2 illustrates that both rotating the frame ‘a’ with the magnitude $\|\boldsymbol{\phi}\|$ about one direction and rotating the frame ‘a’ with the magnitude $2\pi - \|\boldsymbol{\phi}\|$ about the other direction result in the same frame ‘b.’ Therefore, $\mathbf{q}_{\mathrm{b}}^{\mathrm{a}}$ and $-\mathbf{q}_{\mathrm{b}}^{\mathrm{a}}$ represent the same attitude.

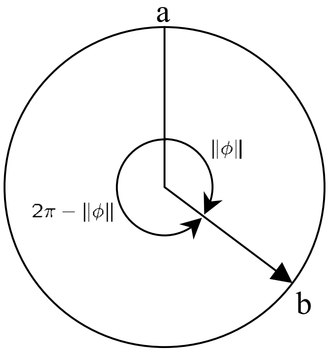

Figure 2.2: Equivalence of a quaternion to its negative

2.1.4 Rotation Vector in Terms of Quaternion

Let $\mathbf{q}_{\mathrm{b}}^{\mathrm{a}} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T$. The rotation vector can then be computed from (Savage, 2000, p. 3-49)

$$\begin{aligned} \boldsymbol{\phi} &= \frac{1}{f} \begin{bmatrix} q_2 & q_3 & q_4 \end{bmatrix}^T, \\ f &\equiv \frac{\sin \|0.5\boldsymbol{\phi}\|}{\|\boldsymbol{\phi}\|} = \frac{1}{2}\left(1 - \frac{\|0.5\boldsymbol{\phi}\|^2}{3!} + \frac{\|0.5\boldsymbol{\phi}\|^4}{5!} - \frac{\|0.5\boldsymbol{\phi}\|^6}{7!} + \cdots\right), \\ \|0.5\boldsymbol{\phi}\| &= \tan^{-1} \frac{\sin \|0.5\boldsymbol{\phi}\|}{\cos \|0.5\boldsymbol{\phi}\|} = \frac{\sqrt{q_2^2 + q_3^2 + q_4^2}}{q_1}. \end{aligned} \tag{2.7}$$

If $q_1 = 0$, then

$$\boldsymbol{\phi} = \pi \begin{bmatrix} q_2 & q_3 & q_4 \end{bmatrix}^T. \tag{2.8}$$

2.1.5 DCM in Terms of Quaternion

The DCM can be obtained from the corresponding quaternion vector as (Savage, 2000, p. 3-46)

$$\mathbf{C}_{\mathrm{b}}^{\mathrm{a}} = \begin{bmatrix} (q_1^2 + q_2^2 - q_3^2 - q_4^2) & 2(q_2q_3 - q_1q_4) & 2(q_2q_4 + q_1q_3) \\ 2(q_2q_3 + q_1q_4) & (q_1^2 - q_2^2 + q_3^2 - q_4^2) & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 - q_1q_3) & 2(q_3q_4 + q_1q_2) & (q_1^2 - q_2^2 - q_3^2 + q_4^2) \end{bmatrix}. \tag{2.9}$$

2.1.6 DCM in Terms of Rotation Vector

The DCM can be written in terms of the rotation vector as (Savage, 2000, p. 3-26)

$$\mathbf{C}_{\mathrm{b}}^{\mathrm{a}} = \mathbf{I} + \frac{\sin \|\boldsymbol{\phi}\|}{\|\boldsymbol{\phi}\|} (\boldsymbol{\phi}\times) + \frac{1 - \cos \|\boldsymbol{\phi}\|}{\|\boldsymbol{\phi}\|^2} (\boldsymbol{\phi}\times)(\boldsymbol{\phi}\times), \tag{2.10}$$

where $(\boldsymbol{\phi}\times)$ denotes the cross-product (skew-symmetric) form of $\boldsymbol{\phi} = \begin{bmatrix} \phi_\mathrm{x} & \phi_\mathrm{y} & \phi_\mathrm{z} \end{bmatrix}^T$,

$$(\boldsymbol{\phi}\times) \equiv \begin{bmatrix} 0 & -\phi_\mathrm{z} & \phi_\mathrm{y} \\ \phi_\mathrm{z} & 0 & -\phi_\mathrm{x} \\ -\phi_\mathrm{y} & \phi_\mathrm{x} & 0 \end{bmatrix}. \tag{2.11}$$

The derivation of Eq. (2.10) is also found in Bortz (1971). If $\|\boldsymbol{\phi}\|$ is very small, $\sin\|\boldsymbol{\phi}\|/\|\boldsymbol{\phi}\| \approx 1$ and the second-order term can be neglected. Therefore, Eq. (2.10) can be approximated by

$$\mathbf{C}_\mathrm{b}^\mathrm{a} \approx \mathbf{I} + (\boldsymbol{\phi}\times)\,. \tag{2.12}$$

2.1.7 Quaternion in Terms of DCM

The best method for extracting the quaternion elements from the DCM was a subject of considerable interest in the recent past and at the end of this controversy the algorithm to be introduced here was shown to be the most robust (Shuster, 1993; Savage, 2000). The algorithm works as follows. First, we compute

$$\begin{aligned} P_1 &= 1 + \mathrm{tr}(\mathbf{C}_\mathrm{b}^\mathrm{a}), & P_2 &= 1 + 2c_{11} - \mathrm{tr}(\mathbf{C}_\mathrm{b}^\mathrm{a}), \\ P_3 &= 1 + 2c_{22} - \mathrm{tr}(\mathbf{C}_\mathrm{b}^\mathrm{a}), & P_4 &= 1 + 2c_{33} - \mathrm{tr}(\mathbf{C}_\mathrm{b}^\mathrm{a}), \end{aligned} \tag{2.13}$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a matrix and c_{ij}'s, $1 \le i, j \le 3$, are ijth elements of the DCM $\mathbf{C}_\mathrm{b}^\mathrm{a}$.

If $P_1 = \max(P_1, P_2, P_3, P_4)$, then:

$$q_1 = 0.5\sqrt{P_1}, \quad q_2 = \frac{c_{32} - c_{23}}{4q_1}, \quad q_3 = \frac{c_{13} - c_{31}}{4q_1}, \quad q_4 = \frac{c_{21} - c_{12}}{4q_1}. \tag{2.14a}$$

If $P_2 = \max(P_1, P_2, P_3, P_4)$, then:

$$q_2 = 0.5\sqrt{P_2}, \quad q_3 = \frac{c_{21} + c_{12}}{4q_2}, \quad q_4 = \frac{c_{13} + c_{31}}{4q_2}, \quad q_1 = \frac{c_{32} - c_{23}}{4q_2}. \tag{2.14b}$$

If $P_3 = \max(P_1, P_2, P_3, P_4)$, then:

$$q_3 = 0.5\sqrt{P_3}, \quad q_4 = \frac{c_{32} + c_{23}}{4q_3}, \quad q_1 = \frac{c_{13} - c_{31}}{4q_3}, \quad q_2 = \frac{c_{21} + c_{12}}{4q_3}. \tag{2.14c}$$

If $P_4 = \max(P_1, P_2, P_3, P_4)$, then:

$$q_4 = 0.5\sqrt{P_4}, \quad q_1 = \frac{c_{21} - c_{12}}{4q_4}, \quad q_2 = \frac{c_{13} + c_{31}}{4q_4}, \quad q_3 = \frac{c_{32} + c_{23}}{4q_4}. \tag{2.14d}$$

Finally, if $q_1 < 0$, then $\mathbf{q} := -\mathbf{q}$.

2.1.8 DCM in Terms of Euler Angles

A classical method for describing the attitude between two coordinate frames is through an Euler angle rotation sequence (Savage, 2000, p. 3-31). The DCM can be computed from the Euler angles, roll (ϕ), pitch (θ), and heading (ψ), as follows

(Savage, 2000, p. 3-33):

$$\begin{cases} c_{11} = \cos\theta\cos\psi \\ c_{12} = -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi \\ c_{13} = \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ c_{21} = \cos\theta\sin\psi \\ c_{22} = \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi \\ c_{23} = -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ c_{31} = -\sin\theta \\ c_{32} = \sin\phi\cos\theta \\ c_{33} = \cos\phi\cos\theta \end{cases} . \tag{2.15}$$

2.1.9 Euler Angles in Terms of DCM

The pitch angle can be computed as

$$\theta = \tan^{-1}\frac{\sin\theta}{\cos\theta} = \tan^{-1}\frac{-c_{31}}{\sqrt{c_{32}^2 + c_{33}^2}}, \tag{2.16a}$$

where $|\theta| \leq \pi/2$ because the positive square root is chosen (Savage, 2000, p. 3-34). When $|\theta| \neq \pi/2$ ($|c_{31}| < 0.999$), the roll and heading angles can be computed as (Savage, 2000, p. 3-34)

$$\phi = \tan^{-1}\frac{\sin\phi}{\cos\phi} = \tan^{-1}\frac{c_{32}}{c_{33}}, \tag{2.16b}$$

$$\psi = \tan^{-1}\frac{\sin\psi}{\cos\psi} = \tan^{-1}\frac{c_{21}}{c_{11}}. \tag{2.16c}$$

If $|c_{31}| \geq 0.999$, only linear combinations of the roll and heading can be computed (Savage, 2000, p. 3-35):

$$\begin{cases} \psi - \phi = \tan^{-1} \dfrac{c_{23} - c_{12}}{c_{13} + c_{22}} & : \quad c_{31} \leq -0.999, \\ \psi + \phi = \pi + \tan^{-1} \dfrac{c_{23} + c_{12}}{c_{13} - c_{22}} & : \quad c_{31} \geq 0.999. \end{cases} \tag{2.17}$$

2.1.10 Quaternion in Terms of Euler Angles

The quaternion can be computed from the Euler angles as follows (McGreevy, 1986):

$$\mathbf{q}_{\mathrm{b}}^{\mathrm{a}} = \begin{bmatrix} \cos\frac{\phi}{2}\cos\frac{\vartheta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\sin\frac{\vartheta}{2}\cos\frac{\psi}{2} \\ \sin\frac{\phi}{2}\cos\frac{\vartheta}{2}\cos\frac{\psi}{2} - \cos\frac{\phi}{2}\sin\frac{\vartheta}{2}\sin\frac{\psi}{2} \\ \cos\frac{\phi}{2}\sin\frac{\vartheta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\cos\frac{\vartheta}{2}\sin\frac{\psi}{2} \\ \cos\frac{\phi}{2}\cos\frac{\vartheta}{2}\sin\frac{\psi}{2} - \sin\frac{\phi}{2}\sin\frac{\vartheta}{2}\cos\frac{\psi}{2} \end{bmatrix}. \tag{2.18}$$

2.2 Reference Frames and Transformations

Many reference frames are involved in INS development and analysis because a vector represented in one frame must frequently be transformed into another. For instance, the specific force and angular rate are resolved in the body frame of the IMU. To acquire the navigation states, these measurements have to be transformed into the navigation frame of interest. This section first defines various reference frames and then describes the properties pertaining to each and the relationships between them.

2.2.1 The Inertial Frame

The inertial frame (i-frame) is an ideal frame of reference in which ideal accelerometers and gyroscopes fixed to the i-frame have zero outputs. However, since it is very hard to construct a strict i-frame, a quasi-inertial frame is used in practice. This frame has its origin at the centre of the Earth and axes that are non-rotating with respect to distant galaxies. Its z-axis is parallel to the spin axis of the Earth, its x-axis points towards the mean vernal equinox, and its y-axis completes a right-handed orthogonal frame as shown in Figure 2.3. The vernal equinox is the ascending node between the celestial equator and the ecliptic. So, the right ascension system is used as the inertial frame, since it closely approximates an inertial frame (Schwarz, 1999, p. 114).

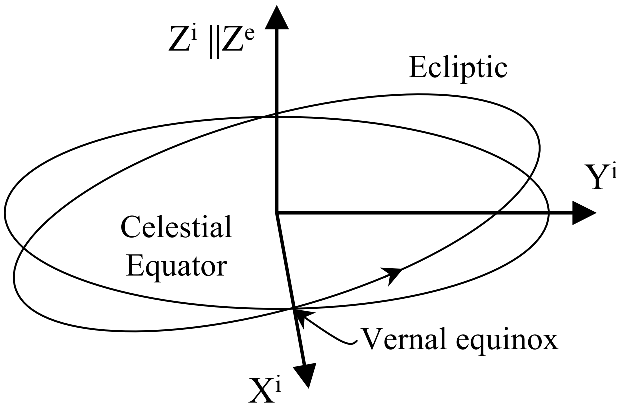

Figure 2.3: The inertial frame

2.2.2 The Earth Frame

The Earth frame (e-frame) has its origin at the centre of mass of the Earth and axes that are fixed with respect to the Earth. Its x-axis points toward the mean meridian of Greenwich, its z-axis is parallel to the mean spin axis of the Earth, and its y-axis completes a right-handed orthogonal frame.

The rotation rate vector of the e-frame with respect to the i-frame projected to the e-frame is given as

$$\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{e}} = \begin{bmatrix} 0 & 0 & \omega_e \end{bmatrix}^T , \tag{2.19}$$

where ω_e is the magnitude of the rotation rate of the Earth ($7.2921158\times10^{-5}\,{}^{\mathrm{rad}}/_{\mathrm{s}}$).

The position vector in the e-frame can be expressed in terms of the geodetic latitude (φ), longitude (λ) and height (h) as follows (Schwarz and Wei, 2000):

$$\mathbf{r}^{\mathrm{e}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \equiv \begin{bmatrix} (R_{\mathrm{N}} + h)\cos\varphi\cos\lambda \\ (R_{\mathrm{N}} + h)\cos\varphi\sin\lambda \\ (R_{\mathrm{N}}(1-e^2) + h)\sin\varphi \end{bmatrix} , \tag{2.20}$$

where e is the first eccentricity of the reference ellipsoid, and R_{N} is the radius of curvature in the prime vertical.

Geodetic coordinates can be computed from the e-frame coordinates using the geometry shown in Figures 2.4 and 2.5. The longitude can be computed simply by

$$\lambda = \tan_2^{-1}(y, x), \tag{2.21}$$

where $\tan_2^{-1}(\cdot)$ is the four-quadrant inverse tangent function. The latitude and height

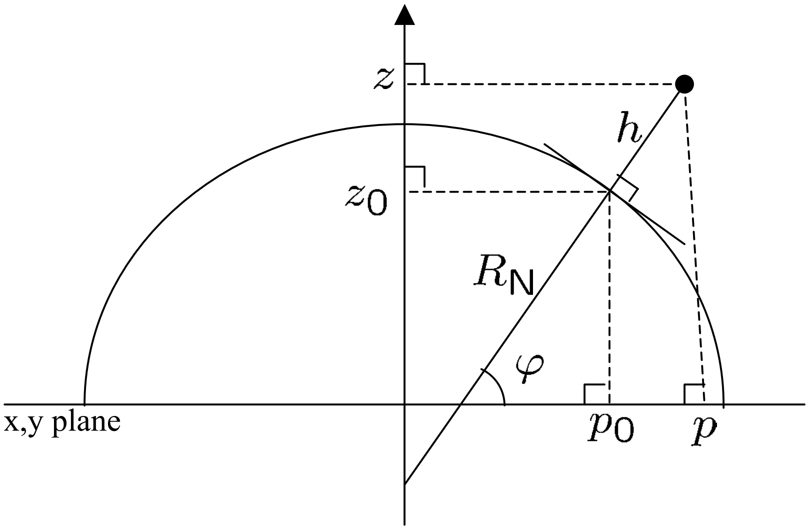

Figure 2.4: Geometry between the Earth frame and geodetic coordinates

computation is more complicated and a commonly used method will be introduced. First, the height and latitude are initialized as follows:

$$h_0 = 0 \text{ and } \varphi_0 = \tan_2^{-1}(z, (1-e^2)p), \tag{2.22}$$

and $p = \sqrt{x^2+y^2}$. Then, the following computations are repeated until the convergence is achieved:

1. Compute the radius of curvature in the prime vertical, R_N.

2. Compute the updated height:

$$h = p/\cos\varphi - R_\mathrm{N}. \tag{2.23}$$

3. Compute the updated latitude:

$$\varphi = \tan_2^{-1}\left(z, p\left(1 - \frac{e^2 R_\mathrm{N}}{R_\mathrm{N}+h}\right)\right). \tag{2.24}$$

However, Eq. (2.23) is singular in near polar areas and is also sensitive to the latitude computation error. Therefore, a non-singular computation procedure will be developed here based on the geometry shown in Figure 2.4. If p is small, then the position is in near polar areas. Hence,

$$h = \mathrm{sign}(|z| - |z_0|)\sqrt{(p - p_0)^2 + (z - z_0)^2}, \tag{2.25a}$$

where

$$\begin{aligned} p_0 &= R_{\mathrm{N}} \cos\varphi, \\ z_0 &= R_{\mathrm{N}}(1 - e^2)\sin\varphi, \end{aligned}$$

and sign($\cdot$) denotes the sign of the argument. On the other hand, if p is large,

$$h = \mathrm{sign}(p - p_0)\sqrt{(p - p_0)^2 + (z - z_0)^2}. \tag{2.25b}$$

2.2.3 The Navigation Frame

The navigation frame (n-frame) is a local geodetic frame which has its origin coinciding with that of the sensor frame, with its x-axis pointing toward geodetic north, its z-axis orthogonal to the reference ellipsoid pointing down, and its y-axis completing a right-handed orthogonal frame, i.e. the north-east-down (NED) system as shown in Figure 2.5.

As the n-frame is singular at the poles where the north direction cannot be defined, the wander frame is used in practice. Although the n-frame is used in the research

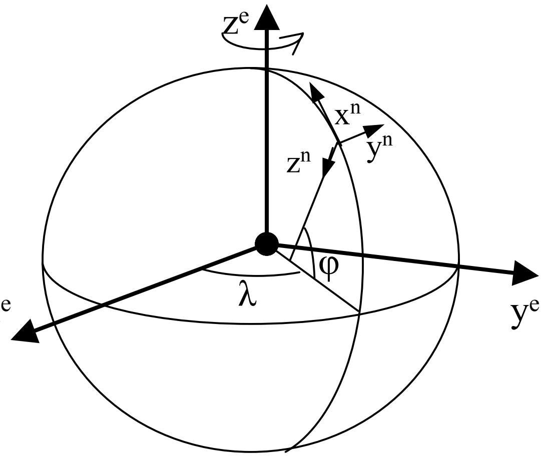

Figure 2.5: The Earth frame and the navigation frame

owing to the simplicity in visualization during the analysis, an n-frame navigator can easily be converted to a wander frame navigator.

The DCM from the n-frame to the e-frame is expressed in terms of the geodetic latitude and longitude as

$$\mathbf{C}_{\mathrm{n}}^{\mathrm{e}} = \begin{bmatrix} -\sin\varphi\cos\lambda & -\sin\lambda & -\cos\varphi\cos\lambda \\ -\sin\varphi\sin\lambda & \cos\lambda & -\cos\varphi\sin\lambda \\ \cos\varphi & 0 & -\sin\varphi \end{bmatrix}. \tag{2.26}$$

The quaternion corresponding to $\mathbf{C}_{\mathrm{n}}^{\mathrm{e}}$ is written as

$$\mathbf{q}_{\mathrm{n}}^{\mathrm{e}} = \begin{bmatrix} \cos(-\pi/4-\varphi/2)\cos(\lambda/2) \\ -\sin(-\pi/4-\varphi/2)\sin(\lambda/2) \\ \sin(-\pi/4-\varphi/2)\cos(\lambda/2) \\ \cos(-\pi/4-\varphi/2)\sin(\lambda/2) \end{bmatrix}. \tag{2.27}$$

The Earth's rotation rate can be described in the n-frame using

$$\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}} = \mathbf{C}_{\mathrm{e}}^{\mathrm{n}}\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{e}} = \begin{bmatrix} \omega_e \cos\varphi & 0 & -\omega_e \sin\varphi \end{bmatrix}^T . \tag{2.28}$$

The rotation rate of the n-frame with respect to the e-frame is called the transport rate, which can be expressed in terms of the rate of change of latitude and longitude as (Titterton and Weston, 1997, p. 52)

$$\boldsymbol{\omega}_{\mathrm{en}}^{\mathrm{n}} = \begin{bmatrix} \dot{\lambda}\cos\varphi \\ -\dot{\varphi} \\ -\dot{\lambda}\sin\varphi \end{bmatrix} . \tag{2.29}$$

Substituting $\dot{\varphi} = v_N/(R_{\mathrm{M}} + h)$ and $\dot{\lambda} = v_E/(R_{\mathrm{N}} + h)\cos\varphi$ into Eq. (2.29) yields

$$\boldsymbol{\omega}_{\mathrm{en}}^{\mathrm{n}} = \begin{bmatrix} v_{\mathrm{E}}/(R_{\mathrm{N}} + h) \\ -v_{\mathrm{N}}/(R_{\mathrm{M}} + h) \\ -v_{\mathrm{E}}\tan\varphi/(R_{\mathrm{N}} + h) \end{bmatrix} , \tag{2.30}$$

where h is height; v_N, v_E are velocities in the north and east direction, respectively; and R_{M} is the meridian radius of curvature.

2.2.4 The Computer Frame

The computer frame (c-frame) is the frame that the INS computer assumes to be the true navigation frame (Scherzinger, 1996). Hence, as shown in Figure 2.6, there exists a rotation vector, $\delta\boldsymbol{\theta}$, describing the misalignment of the c-frame with respect to the true n-frame (Benson Jr., 1975):

$$\delta\boldsymbol{\theta} = \begin{bmatrix} \delta\lambda\cos\varphi \\ -\delta\varphi \\ -\delta\lambda\sin\varphi \end{bmatrix}, \tag{2.31}$$

where $\delta\varphi$ and $\delta\lambda$ are errors in latitude and longitude, respectively. The DCM from the n-frame to the c-frame can be written as

$$\mathbf{C}_{\mathrm{n}}^{\mathrm{c}} = \mathbf{I} - (\delta\boldsymbol{\theta}\times). \tag{2.32}$$

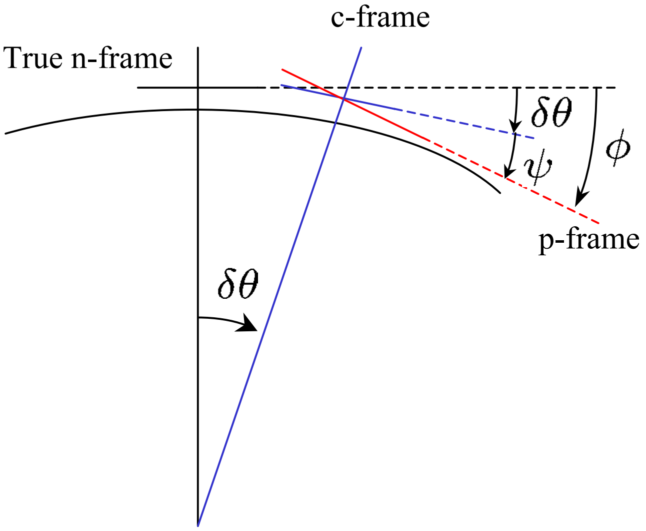

Figure 2.6: Relationship between the true navigation frame, computer frame and platform frame (Scherzinger, 2004)

The north and east position errors can be written as follows:

$$\delta r_{\mathrm{N}} = \delta\varphi\,(R_{\mathrm{M}} + h), \tag{2.33}$$

$$\delta r_{\mathrm{E}} = \delta\lambda\,(R_{\mathrm{N}} + h)\cos\varphi. \tag{2.34}$$

Substituting Eqs. (2.33) and (2.34) into Eq. (2.31) yields

$$\delta\boldsymbol{\theta} = \begin{bmatrix} \delta r_{\mathrm{E}} / \left(R_{\mathrm{N}} + h\right) \\ -\delta r_{\mathrm{N}} / \left(R_{\mathrm{M}} + h\right) \\ -\delta r_{\mathrm{E}} \tan\varphi / \left(R_{\mathrm{N}} + h\right) \end{bmatrix}. \tag{2.35}$$

2.2.5 The Body Frame

The body frame (b-frame) is the frame in which the accelerations and angular rates generated by the strapdown accelerometers and gyroscopes are resolved (Scherzinger, 1996). The b-frame axes will be the same as the IMU's body axes here.

2.2.6 The Platform Frame

The platform frame (p-frame) is the frame in which the transformed acceleration from the accelerometers and angular rates from the gyroscopes are resolved (Scherzinger, 1996). Hence, the computed DCM from the b-frame to the n-frame is the same as the DCM from the b-frame to the p-frame:

$$\hat{\mathbf{C}}_{\mathrm{b}}^{\mathrm{n}} = \mathbf{C}_{\mathrm{b}}^{\mathrm{p}} = \mathbf{C}_{\mathrm{n}}^{\mathrm{p}}\mathbf{C}_{\mathrm{b}}^{\mathrm{n}}, \tag{2.36}$$

where $\hat{\cdot}$ denotes the computed quantity. As shown in Figure 2.6, the $\boldsymbol{\psi}$-angle is the orientation difference between the p-frame and the c-frame. Hence, for small attitude errors,

$$\mathbf{C}_{\mathrm{c}}^{\mathrm{p}} = \mathbf{I} - \left(\boldsymbol{\psi}\times\right). \tag{2.37}$$

The difference in the orientation of the p-frame from the true n-frame is referred to as the $\boldsymbol{\phi}$-angle:

$$\mathbf{C}_{\mathrm{n}}^{\mathrm{p}} = \mathbf{I} - (\boldsymbol{\phi}\times)\,. \tag{2.38}$$

Since $\mathbf{C}_{\mathrm{n}}^{\mathrm{p}} = \mathbf{C}_{\mathrm{c}}^{\mathrm{p}}\mathbf{C}_{\mathrm{n}}^{\mathrm{c}}$, to the first order we can write

$$\boldsymbol{\phi} = \boldsymbol{\psi} + \delta\boldsymbol{\theta}. \tag{2.39}$$

2.2.7 The Computed Platform Frame

Scherzinger (1996) introduced the computed platform frame (c'-frame) to derive the large heading uncertainty model. The c'-frame is defined as the coordinate frame whose x-y plane is coplanar with the c-frame and whose azimuth coincides with the p-frame azimuth. If only the roll and pitch errors are very small, then Eq. (2.37) should be rewritten as

$$\mathbf{C}_{\mathrm{c}}^{\mathrm{p}} = \mathbf{C}_{\mathrm{c}'}^{\mathrm{p}}\mathbf{C}_{\mathrm{c}}^{\mathrm{c}'} = \left[\mathbf{I} - \left(\boldsymbol{\psi}_{\mathrm{xy}}\times\right)\right]\mathbf{C}_{\mathrm{c}}^{\mathrm{c}'}, \tag{2.40}$$

where

$$\boldsymbol{\psi}_{\mathrm{xy}} = \begin{bmatrix} \psi_{\mathrm{x}} \\ \psi_{\mathrm{y}} \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{C}_{\mathrm{c}}^{\mathrm{c}'} = \begin{bmatrix} \cos\psi_{\mathrm{z}} & \sin\psi_{\mathrm{z}} & 0 \\ -\sin\psi_{\mathrm{z}} & \cos\psi_{\mathrm{z}} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.41}$$

In the above equation, ψ_{x}, ψ_{y} and ψ_{z} are the elements of the $\boldsymbol{\psi}$-angle.

2.2.8 The Vehicle Frame

The vehicle frame (v-frame) is an orthogonal axis set that is aligned with the roll, pitch and heading axes of a vehicle; i.e. forward-transversal-down, as shown in Figure 2.7. This frame is required because sometimes the IMU's b-frame is not parallel to the v-frame due to space limitations within the vehicle.

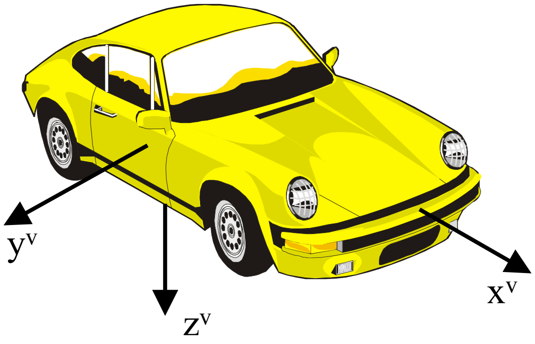

Figure 2.7: The vehicle frame

2.3 Forward INS Mechanization

Savage (2000) describes the two-speed INS mechanization algorithms that are the result of over 20 years of worldwide developments in strapdown inertial navigation algorithms. The high-speed part (1-4 kHz) computes the coning and sculling motion, and the medium-speed part (100-400 Hz) generates the navigation states; i.e. position, velocity, and attitude. However, advances in modern computer technology will eventually motivate the use of the single-speed algorithm (Savage, 2000, p. 7-4).

Development of any discrete-time INS mechanization algorithms start from the continuous-time counterpart:

$$\begin{aligned}
\dot{\mathbf{v}}^{\mathrm{n}} &= \mathbf{C}_{\mathrm{b}}^{\mathrm{n}}\mathbf{f}^{\mathrm{b}} + \mathbf{g}^{\mathrm{n}} - (2\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}} + \boldsymbol{\omega}_{\mathrm{en}}^{\mathrm{n}}) \times \mathbf{v}^{\mathrm{n}}, && (2.42a)\\
\dot{\mathbf{C}}_{\mathrm{n}}^{\mathrm{e}} &= \mathbf{C}_{\mathrm{n}}^{\mathrm{e}}\left(\boldsymbol{\omega}_{\mathrm{en}}^{\mathrm{n}}\times\right), && (2.42b)\\
\dot{h} &= -v_{\mathrm{D}}, && (2.42c)\\
\dot{\mathbf{C}}_{\mathrm{b}}^{\mathrm{n}} &= \mathbf{C}_{\mathrm{b}}^{\mathrm{n}}\left(\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}}\times\right) - \left(\boldsymbol{\omega}_{\mathrm{in}}^{\mathrm{n}}\times\right)\mathbf{C}_{\mathrm{b}}^{\mathrm{n}}, && (2.42d)
\end{aligned}$$

where v_{D} is the downward velocity. In this section, a single-speed algorithm will be developed by simplifying Savage (2000) such that the performance of the discrete-time algorithms should be as close as possible to that of the above continuous-time dynamics equations. Quaternions will be used extensively in this development.

2.3.1 Sensor Output

To achieve precise digitization of the sensed accelerations and angular rates, high-grade IMUs typically apply precise analog integration as a part of digitization. Thus, the outputs from such systems are incremental angles, $\Delta\tilde{\boldsymbol{\theta}}_k$, and incremental velocities due to the specific force, $\Delta\tilde{\mathbf{v}}_{\mathbf{f},k}^{\mathrm{b}}$, where $\tilde{}$ denotes values corrupted by sensor errors. However, most low-cost IMUs usually output the angular rate, $\tilde{\boldsymbol{\omega}}_{\mathrm{ib}}^{\mathrm{b}}$, and the specific force, $\tilde{\mathbf{f}}^{\mathrm{b}}$, directly. In these systems, therefore, the following integration procedures must be applied initially:

$$\begin{aligned} \Delta\tilde{\mathbf{v}}^{\mathrm{b}}_{\mathbf{f},k} &= \int_{t_{k-1}}^{t_k} \tilde{\mathbf{f}}^{\mathrm{b}} dt, & (2.43\mathrm{a}) \\ \Delta\tilde{\boldsymbol{\theta}}_k &= \int_{t_{k-1}}^{t_k} \tilde{\boldsymbol{\omega}}^{\mathrm{b}}_{\mathrm{ib}} dt. & (2.43\mathrm{b}) \end{aligned}$$

2.3.2 Error Compensation

Let us first define the following operator:

$$\boldsymbol{\Xi}(\mathbf{a},\mathbf{b}) = \begin{bmatrix} a_1 & b_1 & b_2 \\ b_3 & a_2 & b_4 \\ b_5 & b_6 & a_3 \end{bmatrix}, \qquad (2.44)$$

where $\mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^T$ and $\mathbf{b} = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \end{bmatrix}^T$ are arbitrary 3×1 and 6×1 vectors, respectively. Then, known sensor errors can be compensated for as follows:

$$\begin{aligned} \Delta\boldsymbol{\theta}_k &= \left[\mathbf{I} + \boldsymbol{\Xi}(\mathbf{s}^0_{\mathrm{g}}, \boldsymbol{\gamma}^0_{\mathrm{g}})\right]^{-1} \left[\Delta\tilde{\boldsymbol{\theta}}_k - \mathbf{b}^0_{\mathrm{g}}\Delta t_k\right] \\ &\approx \left[\mathbf{I} - \boldsymbol{\Xi}(\mathbf{s}^0_{\mathrm{g}}, \boldsymbol{\gamma}^0_{\mathrm{g}})\right] \left[\Delta\tilde{\boldsymbol{\theta}}_k - \mathbf{b}^0_{\mathrm{g}}\Delta t_k\right], \qquad (2.45\mathrm{a}) \end{aligned}$$

$$\begin{aligned} \Delta\mathbf{v}^{\mathrm{b}}_{\mathbf{f},k} &= \left[\mathbf{I} + \boldsymbol{\Xi}(\mathbf{s}^0_{\mathrm{a}}, \boldsymbol{\gamma}^0_{\mathrm{a}})\right]^{-1} \left[\Delta\tilde{\mathbf{v}}^{\mathrm{b}}_{\mathbf{f},k} - \mathbf{b}^0_{\mathrm{a}}\Delta t_k\right] \\ &\approx \left[\mathbf{I} - \boldsymbol{\Xi}(\mathbf{s}^0_{\mathrm{a}}, \boldsymbol{\gamma}^0_{\mathrm{a}})\right] \left[\Delta\tilde{\mathbf{v}}^{\mathrm{b}}_{\mathbf{f},k} - \mathbf{b}^0_{\mathrm{a}}\Delta t_k\right], \qquad (2.45\mathrm{b}) \end{aligned}$$

where $\Delta t_k = t_k - t_{k-1}$; $\mathbf{b}$ and $\mathbf{s}$ are 3×1 vectors of biases and scale factors, respectively; the superscript '0' denotes known quantities; the subscripts 'a' and 'g' represent the accelerometer and gyroscope triads, respectively; and the non-orthogonalities can be written as

$$\boldsymbol{\gamma} = \begin{bmatrix} \gamma_{\mathrm{xy}} & \gamma_{\mathrm{xz}} & \gamma_{\mathrm{yx}} & \gamma_{\mathrm{yz}} & \gamma_{\mathrm{zx}} & \gamma_{\mathrm{zy}} \end{bmatrix}^T . \tag{2.46}$$

2.3.3 Velocity Update

Referring to Eq. (2.42a), the digital algorithm for velocity update can be written generally as (Savage, 2000, p. 7-30)

$$\mathbf{v}_k^{\mathrm{n}} = \mathbf{v}_{k-1}^{\mathrm{n}} + \Delta\mathbf{v}_{\mathbf{f},k}^{\mathrm{n}} + \Delta\mathbf{v}_{\mathrm{g/cor},k}^{n}, \tag{2.47}$$

where $\Delta\mathbf{v}_{\mathrm{g/cor}}^{\mathrm{n}}$ is the velocity increment due to the gravity and Coriolis force; and $\Delta\mathbf{v}_{\mathbf{f},k}^{\mathrm{n}}$ is the velocity increment due to the specific force, which can be written as follows (Savage, 2000, p. 7-33,7-52):

$$\Delta\mathbf{v}_{\mathbf{f},k}^{\mathrm{n}} = \frac{1}{2}\left[\mathbf{C}_{\mathrm{n}(k-1)}^{\mathrm{n}(k)} + \mathbf{I}\right]\mathbf{C}_{\mathrm{b}(k-1)}^{\mathrm{n}(k-1)}\Delta\mathbf{v}_{\mathbf{f},k}^{\mathrm{b}(k-1)}, \tag{2.48a}$$

$$\begin{aligned} \Delta\mathbf{v}_{\mathbf{f},k}^{\mathrm{b}(k-1)} &= \int_{t_{k-1}}^{t_k} \mathbf{C}_{\mathrm{b}(t)}^{\mathrm{b}(k-1)}\mathbf{f}^{\mathrm{b}}dt \\ &\approx \Delta\mathbf{v}_{\mathbf{f},k}^{\mathrm{b}} + \frac{1}{2}\Delta\boldsymbol{\theta}_k \times \Delta\mathbf{v}_{\mathbf{f},k}^{\mathrm{b}} \\ &\quad + \frac{1}{12}\left(\Delta\boldsymbol{\theta}_{k-1} \times \Delta\mathbf{v}_{\mathbf{f},k}^{\mathrm{b}} + \Delta\mathbf{v}_{\mathbf{f},k-1}^{\mathrm{b}} \times \Delta\boldsymbol{\theta}_k\right), \end{aligned} \tag{2.48b}$$

where the superscript/subscript b(k) and n(k) denote the b-frame and n-frame at time t_k, respectively. The second and third terms on the right hand side of Eq. (2.48b) correspond to the rotational and sculling motion, respectively, and are required because we cannot integrate angular rate and linear acceleration concurrently in a digital implementation. Substituting $\mathbf{C}^{\mathrm{n}(k)}_{\mathrm{n}(k-1)} = \mathbf{I} - (\boldsymbol{\zeta}_k \times)$ into Eq. (2.48a) yields

$$\begin{aligned} \Delta\mathbf{v}^{\mathrm{n}}_{\mathbf{f},k} &= [\mathbf{I} - (0.5\boldsymbol{\zeta}_k \times)]\,\mathbf{C}^{\mathrm{n}(k-1)}_{\mathrm{b}(k-1)} \Delta\mathbf{v}^{\mathrm{b}(k-1)}_{\mathbf{f},k}, && (2.49a) \\ \boldsymbol{\zeta}_k &= [\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{ie}} + \boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{en}}]_{k-1/2}\,\Delta t_k, && (2.49b) \end{aligned}$$

where $\boldsymbol{\zeta}_k$ is the rotation vector corresponding to the n(k)-frame with respect to the attitude of the n($k-1$)-frame; the subscript '$k - 1/2$' denotes values at the midway in the interval $[t_{k-1}, t_k]$. Since $\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{ie}}$ and $\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{en}}$ are functions of the position and velocity, the midway position and velocity have to be computed first. Because the position and velocity at time t_k are not available as yet, extrapolations must be made. First, the height extrapolation can be computed as follows:

$$h_{k-1/2} = h_{k-1} - \frac{v_{\mathrm{D},k-1}\Delta t_k}{2}. \qquad (2.50)$$

The latitude and longitude can be extrapolated as follows:

$$\begin{aligned} \mathbf{q}^{\mathrm{e}(k-1)}_{\mathrm{n}(k-1/2)} &= \mathbf{q}^{\mathrm{e}(k-1)}_{\mathrm{n}(k-1)} \star \mathbf{q}^{\mathrm{n}(k-1)}_{\mathrm{n}(k-1/2)}, && (2.51a) \\ \mathbf{q}^{\mathrm{e}(k-1/2)}_{\mathrm{n}(k-1/2)} &= \mathbf{q}^{\mathrm{e}(k-1/2)}_{\mathrm{e}(k-1)} \star \mathbf{q}^{\mathrm{e}(k-1)}_{\mathrm{n}(k)}, && (2.51b) \end{aligned}$$

where

$$\begin{aligned} \mathbf{q}^{\mathrm{n}(k-1)}_{\mathrm{n}(k-1/2)} &= \begin{bmatrix} \cos\|0.5\boldsymbol{\zeta}_{k-1/2}\| \\ \dfrac{\sin\|0.5\boldsymbol{\zeta}_{k-1/2}\|}{\|0.5\boldsymbol{\zeta}_{k-1/2}\|} 0.5\boldsymbol{\zeta}_{k-1/2} \end{bmatrix}, && (2.51c) \\ \mathbf{q}^{\mathrm{e}(k-1/2)}_{\mathrm{e}(k-1)} &= \begin{bmatrix} \cos\|0.5\boldsymbol{\xi}_{k-1/2}\| \\ -\dfrac{\sin\|0.5\boldsymbol{\xi}_{k-1/2}\|}{\|0.5\boldsymbol{\xi}_{k-1/2}\|} 0.5\boldsymbol{\xi}_{k-1/2} \end{bmatrix}, && (2.51d) \end{aligned}$$

$\boldsymbol{\zeta}_{k-1/2} = \boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{in}}(t_{k-1})\Delta t_k/2$ and $\boldsymbol{\xi}_{k-1/2} = \boldsymbol{\omega}^{\mathrm{e}}_{\mathrm{ie}}\Delta t_k/2$.

The velocity extrapolation can be obtained as follows:

$$\begin{aligned} \mathbf{v}^{\mathrm{n}}_{k-1/2} &= \mathbf{v}^{\mathrm{n}}_{k-1} + \frac{1}{2}\Delta\mathbf{v}^{\mathrm{n}}_{k-1}, & (2.52a) \\ \Delta\mathbf{v}^{\mathrm{n}}_{k-1} &= \Delta\mathbf{v}^{\mathrm{n}}_{\mathbf{f},k-1} + \Delta\mathbf{v}^{n}_{\mathrm{g/cor},k-1}, & (2.52b) \end{aligned}$$

where $\Delta\mathbf{v}^{\mathrm{n}}_{k-1}$ is a value used at the previous epoch and stored. The gravity and Coriolis correction term in Eq. (2.47) can be computed as follows:

$$\Delta\mathbf{v}^{\mathrm{n}}_{\mathrm{g/cor},k} = \left[\mathbf{g}^{\mathrm{n}} - (2\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{ie}} + \boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{en}}) \times \mathbf{v}^{\mathrm{n}}\right]_{k-1/2}\Delta t_k. \qquad (2.53)$$

2.3.4 Position Update

The quaternion from the n-frame to the e-frame, $\mathbf{q}^{\mathrm{e}}_{\mathrm{n}}$, contains information about the latitude and longitude. Hence, $\mathbf{q}^{\mathrm{e}}_{\mathrm{n}}$ is updated using the quaternion product chain rule as follows:

$$\begin{aligned} \mathbf{q}^{\mathrm{e}(k-1)}_{\mathrm{n}(k)} &= \mathbf{q}^{\mathrm{e}(k-1)}_{\mathrm{n}(k-1)} \star \mathbf{q}^{\mathrm{n}(k-1)}_{\mathrm{n}(k)}, & (2.54a) \\ \mathbf{q}^{\mathrm{e}(k)}_{\mathrm{n}(k)} &= \mathbf{q}^{\mathrm{e}(k)}_{\mathrm{e}(k-1)} \star \mathbf{q}^{\mathrm{e}(k-1)}_{\mathrm{n}(k)}, & (2.54b) \end{aligned}$$

where the superscript/subscript $\mathrm{e}(k)$ represents the e-frame at time t_k. The n-frame rotation is applied in Eq. (2.54a), followed by the e-frame rotation in Eq. (2.54b).

The quaternions $\mathbf{q}_{\mathrm{n}(k)}^{\mathrm{n}(k-1)}$ and $\mathbf{q}_{\mathrm{e}(k-1)}^{\mathrm{e}(k)}$ can be written as follows:

$$\mathbf{q}_{\mathrm{n}(k)}^{\mathrm{n}(k-1)} = \begin{bmatrix} \cos\|0.5\boldsymbol{\zeta}_k\| \\ \dfrac{\sin\|0.5\boldsymbol{\zeta}_k\|}{\|0.5\boldsymbol{\zeta}_k\|}0.5\boldsymbol{\zeta}_k \end{bmatrix}, \tag{2.54c}$$

$$\mathbf{q}_{\mathrm{e}(k-1)}^{\mathrm{e}(k)} = \begin{bmatrix} \cos\|0.5\boldsymbol{\xi}_k\| \\ -\dfrac{\sin\|0.5\boldsymbol{\xi}_k\|}{\|0.5\boldsymbol{\xi}_k\|}0.5\boldsymbol{\xi}_k \end{bmatrix}, \tag{2.54d}$$

where $\boldsymbol{\xi}_k = \boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{e}}\Delta t_k$ is the rotation vector corresponding to the attitude of the e(k)-frame with respect to the e($k-1$)-frame. Since the velocity is updated already, the midway velocity can be updated by interpolation: $\mathbf{v}_{k-1/2}^{\mathrm{n}} = \frac{1}{2}(\mathbf{v}_{k-1}^{\mathrm{n}} + \mathbf{v}_k^{\mathrm{n}})$. So, $\boldsymbol{\zeta}_k$ can first be recomputed using this updated midway velocity and then used in Eq. (2.54c). The updated φ and λ can be extracted from the quaternion. The height can be updated separately as follows:

$$h_k = h_{k-1} - v_{\mathrm{D},k-1/2}\Delta t_k. \tag{2.55}$$

2.3.5 Attitude Update

The attitude quaternion $\mathbf{q}_{\mathrm{b}}^{\mathrm{n}}$ update algorithm can be described as follows:

$$\mathbf{q}_{\mathrm{b}(k)}^{\mathrm{n}(k-1)} = \mathbf{q}_{\mathrm{b}(k-1)}^{\mathrm{n}(k-1)} \star \mathbf{q}_{\mathrm{b}(k)}^{\mathrm{b}(k-1)}, \tag{2.56a}$$

$$\mathbf{q}_{\mathrm{b}(k)}^{\mathrm{n}(k)} = \mathbf{q}_{\mathrm{n}(k-1)}^{\mathrm{n}(k)} \star \mathbf{q}_{\mathrm{b}(k)}^{\mathrm{n}(k-1)}. \tag{2.56b}$$

The quaternion for the b-frame update, $\mathbf{q}_{\mathrm{b}(k)}^{\mathrm{b}(k-1)}$, can be written as

$$\mathbf{q}_{\mathrm{b}(k)}^{\mathrm{b}(k-1)} = \begin{bmatrix} \cos\|0.5\boldsymbol{\phi}_k\| \\ \dfrac{\sin\|0.5\boldsymbol{\phi}_k\|}{\|0.5\boldsymbol{\phi}_k\|}0.5\boldsymbol{\phi}_k \end{bmatrix}, \tag{2.57}$$

where $\boldsymbol{\phi}_k$ is the b-frame rotation vector. Referring to Eq. (2.4), the rate of the b-frame rotation vector can be written as (Savage, 2000, p. 7-8)

$$\begin{aligned}\dot{\boldsymbol{\phi}} &\approx \boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}} + \frac{1}{2}\boldsymbol{\phi}\times\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}} + \frac{1}{12}\boldsymbol{\phi}\times\left(\boldsymbol{\phi}\times\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}}\right)\\ &\approx \boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}} + \frac{1}{2}\Delta\boldsymbol{\theta}(t)\times\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}}, \qquad (2.58)\end{aligned}$$

where

$$\Delta\boldsymbol{\theta}(t) = \int_{t_{k-1}}^{t}\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}}d\tau. \qquad (2.59)$$

Hence, $\boldsymbol{\phi}_k$ can be obtained as (Savage, 2000, p. 7-14)

$$\begin{aligned}\boldsymbol{\phi}_k &= \int_{t_{k-1}}^{t_k}\left[\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}} + \frac{1}{2}\Delta\boldsymbol{\theta}(t)\times\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}}\right]dt\\ &\approx \Delta\boldsymbol{\theta}_k + \frac{1}{12}\Delta\boldsymbol{\theta}_{k-1}\times\Delta\boldsymbol{\theta}_k, \qquad (2.60)\end{aligned}$$

where $\frac{1}{12}\Delta\boldsymbol{\theta}_{k-1}\times\Delta\boldsymbol{\theta}_k$ is the second-order coning correction term. The quaternion for the n-frame update, $\mathbf{q}^{\mathrm{n}(k)}_{\mathrm{n}(k-1)}$, is written as

$$\mathbf{q}^{\mathrm{n}(k)}_{\mathrm{n}(k-1)} = \begin{bmatrix}\cos\|0.5\boldsymbol{\zeta}_k\|\\ -\dfrac{\sin\|0.5\boldsymbol{\zeta}_k\|}{\|0.5\boldsymbol{\zeta}_k\|}0.5\boldsymbol{\zeta}_k\end{bmatrix}. \qquad (2.61)$$

Because the position was previously updated, the midway position can be recomputed by interpolation and, therefore, the updated $\boldsymbol{\zeta}_k$ can be used in the above equation. The height interpolation can be done simply by averaging: $h_{k-1/2} = \frac{1}{2}(h_{k-1} + h_k)$. The quaternion corresponding to the position change from t_{k-1} to t_k can be obtained as follows:

$$\mathbf{q}_{\delta\boldsymbol{\theta}} = \left(\mathbf{q}^{\mathrm{e}(k-1)}_{\mathrm{n}(k-1)}\right)^{-1}\star\mathbf{q}^{\mathrm{e}(k)}_{\mathrm{n}(k)}, \qquad (2.62)$$

from which the rotation vector $\delta\boldsymbol{\theta}$ can be computed. Then the position interpolation can be done as follows:

$$\mathbf{q}_{\mathrm{n}(k-1/2)}^{\mathrm{e}(k-1/2)} = \mathbf{q}_{\mathrm{n}(k-1)}^{\mathrm{e}(k-1)} \star \mathbf{q}_{0.5\delta\boldsymbol{\theta}}, \tag{2.63}$$

where $\mathbf{q}_{0.5\delta\boldsymbol{\theta}}$ is the quaternion corresponding to $0.5\delta\boldsymbol{\theta}$.

Due to numerical errors, the computed $\mathbf{q}_{\mathrm{b}}^{\mathrm{n}}$ often violates the normalization constraint. In this case, the normalization can be applied as follows (Savage, 2000, p. 7-29):

$$\begin{aligned} \mathbf{q}_{\mathrm{b}}^{\mathrm{n}} &:= (1 - e_{\mathrm{q}})\mathbf{q}_{\mathrm{b}}^{\mathrm{n}}, & \text{(2.64a)} \\ e_{\mathrm{q}} &= \frac{1}{2}\left[(\mathbf{q}_{\mathrm{b}}^{\mathrm{n}})^T \mathbf{q}_{\mathrm{b}}^{\mathrm{n}} - 1 \right], & \text{(2.64b)} \end{aligned}$$

where e_{q} is the normality error in the quaternion.

2.4 Backward INS Mechanization

Backward INS mechanization can be applied in cases where the final navigation states are known, but the initial navigation states are not. Also, if there is an independently processed forward solution, then the combination of the two solutions would yield the smoothed solution, which will be discussed in Chapters 3 and 4. The role of backward mechanization is to determine the navigation states ($\mathbf{q}_{\mathrm{n}}^{\mathrm{e}}$, h, $\mathbf{v}^{\mathrm{n}}$, $\mathbf{q}_{\mathrm{b}}^{\mathrm{n}}$) at time t_{k-1} from the given navigation states at time t_k, given the IMU measurements ($\Delta\mathbf{v}_{k-1}$, $\Delta\boldsymbol{\theta}_{k-1}$, $\Delta\mathbf{v}_k$, $\Delta\boldsymbol{\theta}_k$).

In the forward INS mechanization, the velocity update is implemented first followed by position and attitude updates successively. Referring to Eq. (2.48a), $\mathbf{C}^{\mathrm{n}(k-1)}_{\mathrm{b}(k-1)}$ is required for the velocity update. Therefore, in backward mechanization, the attitude update will be applied first to obtain $\mathbf{C}^{\mathrm{n}(k-1)}_{\mathrm{b}(k-1)}$:

$$\mathbf{q}^{\mathrm{n}(k)}_{\mathrm{b}(k-1)} = \mathbf{q}^{\mathrm{n}(k)}_{\mathrm{b}(k)} \star \mathbf{q}^{\mathrm{b}(k)}_{\mathrm{b}(k-1)}, \tag{2.65a}$$

$$\mathbf{q}^{\mathrm{n}(k-1)}_{\mathrm{b}(k-1)} = \mathbf{q}^{\mathrm{n}(k-1)}_{\mathrm{n}(k)} \star \mathbf{q}^{\mathrm{n}(k)}_{\mathrm{b}(k-1)}, \tag{2.65b}$$

where

$$\mathbf{q}^{\mathrm{b}(k)}_{\mathrm{b}(k-1)} = \begin{bmatrix} \cos\|0.5\boldsymbol{\phi}_k\| \\ -\dfrac{\sin\|0.5\boldsymbol{\phi}_k\|}{\|0.5\boldsymbol{\phi}_k\|}0.5\boldsymbol{\phi}_k \end{bmatrix}. \tag{2.65c}$$

Then, the backward velocity and position updates can be applied successively as follows:

$$\mathbf{v}^{\mathrm{n}}_{k-1} = \mathbf{v}^{\mathrm{n}}_{k} - \Delta\mathbf{v}^{\mathrm{n}}_{\mathrm{f},k} - \Delta\mathbf{v}^{\mathrm{n}}_{\mathrm{g/cor}}, \tag{2.66a}$$

$$\mathbf{q}^{\mathrm{e}(k)}_{\mathrm{n}(k-1)} = \mathbf{q}^{\mathrm{e}(k)}_{\mathrm{n}(k)} \star \mathbf{q}^{\mathrm{n}(k)}_{\mathrm{n}(k-1)}, \tag{2.66b}$$

$$\mathbf{q}^{\mathrm{e}(k-1)}_{\mathrm{n}(k-1)} = \mathbf{q}^{\mathrm{e}(k-1)}_{\mathrm{e}(k)} \star \mathbf{q}^{\mathrm{e}(k)}_{\mathrm{n}(k-1)}, \tag{2.66c}$$

$$h_{k-1} = h_k + v_{\mathrm{D},k-1/2}\Delta t_k, \tag{2.66d}$$

where

$$\mathbf{q}^{\mathrm{e}(k-1)}_{\mathrm{e}(k)} = \begin{bmatrix} \cos\|0.5\boldsymbol{\xi}_k\| \\ \dfrac{\sin\|0.5\boldsymbol{\xi}_k\|}{\|0.5\boldsymbol{\xi}_k\|}0.5\boldsymbol{\xi}_k \end{bmatrix}. \tag{2.66e}$$

2.5 Aiding Sources

For a high-grade INS, the error behaviour is well described by the Schuler dynamics, whose period is about 84.4 minutes. For the traditional INS error analysis, refer to Appendix C. However, a low-cost INS cannot run in stand-alone mode for long periods and, in extreme cases, can even experience computational failures before some part of the Schuler period can be seen. Therefore, it has to be aided by other external navigation-related information. This section will investigate aiding options for a low-cost INS.

2.5.1 The Global Positioning System

Since the GPS antenna and the IMU cannot be installed at the same place in the host vehicle, the position of the IMU is different from that of the GPS, which is known as the lever-arm effect. Assuming that several GPS antennas are installed, the lever-arm effect can be described as

$$\mathbf{r}^{\mathrm{n}}_{\mathrm{GPS},i} = \mathbf{r}^{\mathrm{n}}_{\mathrm{IMU}} + \mathbf{D}_R^{-1}\mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i}, \tag{2.67a}$$

$$\mathbf{D}_R^{-1} = \begin{bmatrix} \dfrac{1}{R_{\mathrm{M}}+h} & 0 & 0 \\ 0 & \dfrac{1}{(R_{\mathrm{N}}+h)\cos\varphi} & 0 \\ 0 & 0 & -1 \end{bmatrix}, \tag{2.67b}$$

where $\mathbf{r}^{\mathrm{n}}_{\mathrm{GPS},i}$ and $\mathbf{r}^{\mathrm{n}}_{\mathrm{IMU}}$ are the positions of the ith GPS antenna centre and the centre of the IMU in the n-frame, respectively; and $\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i}$ is the lever-arm vector of the ith

GPS antenna in the b-frame. The lever-arm effect in the velocity measurement is written as (Shin, 2001, p. 43)

$$\mathbf{v}^{\mathrm{n}}_{\mathrm{GPS},i} = \mathbf{v}^{\mathrm{n}}_{\mathrm{IMU}} - \left(\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{ie}} \times + \boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{en}} \times\right) \mathbf{C}^{\mathrm{n}}_{\mathrm{b}} \boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i} - \mathbf{C}^{\mathrm{n}}_{\mathrm{b}} \left(\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i} \times\right) \boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}}, \tag{2.68}$$

where the second term on the right-hand side can be neglected in most cases.

As illustrated in Figure 2.5.1, the range from the the ith antenna to the jth satellite can be written as

$$r^j_i = \left\| \mathbf{r}^{\mathrm{e}}_{\mathrm{SV},j} - \mathbf{r}^{\mathrm{e}}_{\mathrm{GPS},i} \right\| = \left\| \mathbf{r}^{\mathrm{e}}_{\mathrm{SV},j} - \mathbf{r}^{\mathrm{e}}_{\mathrm{IMU}} - \mathbf{C}^{\mathrm{e}}_{\mathrm{b}} \boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i} \right\|, \tag{2.69}$$

where r^j_i is the range; and $\mathbf{r}^{\mathrm{e}}_{\mathrm{SV},j}$ is the position vector of the jth satellite in the e-frame.

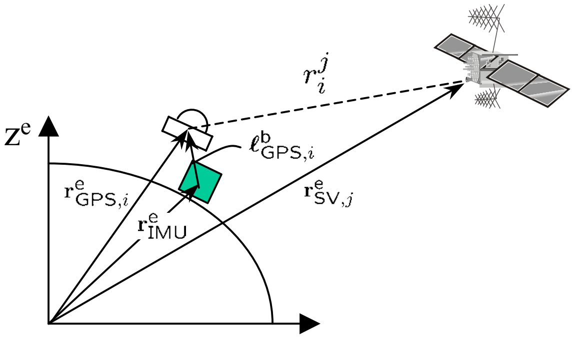

Figure 2.8: INS/GPS measurement

There are two ways to perform integration of the INS and GPS: loosely-coupled and tightly-coupled integration. In the loosely-coupled integration, the GPS-derived position and/or velocity are used to aid the INS. On the other hand, in the tightly-coupled integration scheme, the GPS observables (pseudo ranges and carrier phases) are used directly in the measurement update (Scherzinger, 2004). Therefore, Eq. (2.67a) and (2.68) can be used as the measurement model for loosely-coupled systems and Eq. (2.69) for tightly-coupled systems.

2.5.2 Vehicle Frame Measurements

The vehicle frame (v-frame) velocity can be obtained from odometers and/or speedometers. Development of a rigorous model for the v-frame velocity would require information about the steering angle, front and rear slip angle, tire pressure, angular speed of the wheels, suspension system, etc. Julier and Durrant-Whyte (2003) tried to use some of the parameters based on the two-dimensional "fundamental bicycle," which assumes that the vehicle consists of front and rear virtual wheels. Since it is very hard to acquire information about these parameters, a simplified model that has been widely accepted will be introduced here.

It is assumed that the vehicle has an along-track speed, v, and the cross-track and vertical speeds are zero, i.e. $\mathbf{v}^{\mathrm{v}} = \begin{bmatrix} v & 0 & 0 \end{bmatrix}^T$. To satisfy the condition as closely as possible, wheel sensors need to be installed on rear wheels if the front wheels are used for steering. Then, the relationship between the velocity of the vehicle at the centre of the IMU, $\mathbf{v}^{\mathrm{n}}_{\mathrm{IMU}}$, and that at the wheel, $\mathbf{v}^{\mathrm{v}}_{\mathrm{wheel}}$, can be expressed as (Scherzinger, 2004)

$$\mathbf{v}^{\mathrm{v}}_{\mathrm{wheel}} = \mathbf{C}^{\mathrm{v}}_{\mathrm{b}}\mathbf{C}^{\mathrm{b}}_{\mathrm{n}}\mathbf{v}^{\mathrm{n}}_{\mathrm{IMU}} + \mathbf{C}^{\mathrm{v}}_{\mathrm{b}}\left(\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{nb}}\times\right)\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{wheel}}, \tag{2.70}$$

where $\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{wheel}}$ is the lever-arm vector of the wheel sensor in the b-frame. Eq. (2.70) will be disturbed if significant slips exist. If the along-track speed drops below a certain predefined threshold, ZUPTs can be applied instead of the v-frame velocity measurements. The Doppler radar on an aircraft and the sonar sensors on a ship can also be classified into this category (Scherzinger, 2004).

The vehicle's heading can be measured by using a GPS compass and Eq. (2.16c) can be used as the measurement model (Shin, 2001, p. 81). If the vehicle's heading does not change while stationary, zero integrated heading rate measurements (ZIHR) can be applied to fix the heading. The ZIHR measurement model, useful on a wheeled vehicle such as a van, railway vehicle or aircraft while parked, will be developed in Section 3.3.2.

2.5.3 Other Aiding Sources

In addition to the aiding sensors mentioned previously, information from the following sources can also be used:

- Magnetic heading sensors.
- Aiding by database such as map matching or terrain navigation.
- Visual aiding with predefined targets such as the laser scanner (Hirokawa et al., 2004).

The research work in this dissertation does not depend on these sensors. Therefore, measurement models for these sensors will not be considered here.

2.6 Alignment of Low-Cost INS

Alignment is referred to as the procedure to initialize the INS: in particular, the attitude information between the b-frame and the n-frame. Several algorithms are available and can be classified in different ways. If the criterion is the amount of attitude error that the algorithm has to deal with, then these are classified as either coarse alignment or fine alignment methods. Typically, the threshold of the attitude errors between the two categories can reach a few degrees. If, on the other hand, the criterion is the dynamics of the vehicle upon initialization, then alignment methods can be classified as stationary alignment or in-motion alignment methods.

Typically, coarse alignment of an INS is done in stationary mode using leveling (by accelerometers) followed by gyro-compassing or, alternatively, an analytical method solving the two-vector measurement problem using the gravity and the Earth rotation measurements in one step as follows (Britting, 1971):

$$\mathbf{C}_{\mathrm{b}}^{\mathrm{n}} = \begin{bmatrix} \dfrac{-\tan\varphi}{g} & \dfrac{1}{\omega_{\mathrm{e}}\cos\varphi} & 0 \\ 0 & 0 & \dfrac{-1}{g\omega_{\mathrm{e}}\cos\varphi} \\ \dfrac{-1}{g} & 0 & 0 \end{bmatrix} \begin{bmatrix} \left(\mathbf{f}^{\mathrm{b}}\right)^T \\ \left(\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}}\right)^T \\ \left(\mathbf{f}^{\mathrm{b}} \times \boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}}\right)^T \end{bmatrix}, \qquad (2.71)$$

where g is gravity. However, due to their large biases and the low signal-to-noise ratio of low-cost gyroscopes, gyro-compassing and the analytical coarse alignment cannot be applied in stationary mode. Therefore, only the roll and pitch can be determined from the accelerometer measurements as follows:

$$\phi = \operatorname{sign}(f_z)\sin^{-1}(f_y/g), \tag{2.72a}$$

$$\theta = -\operatorname{sign}(f_z)\sin^{-1}(f_x/g), \tag{2.72b}$$

where $\mathbf{f}^{\mathrm{b}} = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^T$ and $\operatorname{sign}(\cdot)$ denotes the sign of a value. For the z-channel, only the sign is used because the gravity error is smaller than the bias of low-cost IMUs. The heading has to be determined from other sensors such as multi-antenna GPS or a magnetic compass.

If the IMU is installed in a consumer vehicle, we cannot expect the user to wait until the alignment is finished. Hence, in-motion alignment techniques need to be considered. The GPS-derived velocity can be used for coarse in-motion alignment if the forward axis is parallel to the velocity vector, which holds approximately for most land vehicle navigation applications. Within mechanical alignment uncertainty, the roll can be initialized to zero with a ±5° uncertainty, in most cases, on the road. The pitch and heading can be initialized as follows (see Figure 2.9):

$$\theta = \tan^{-1}\left(-v_D/\sqrt{v_{\mathrm{N}}^2 + v_{\mathrm{E}}^2}\right), \tag{2.73a}$$

$$\psi = \tan^{-1}\left(v_{\mathrm{E}}/v_{\mathrm{N}}\right), \tag{2.73b}$$

where v_{N} and v_{E} are north and east velocity, respectively.

For aircrafts or ships, however, lateral or vertical velocity components can exist

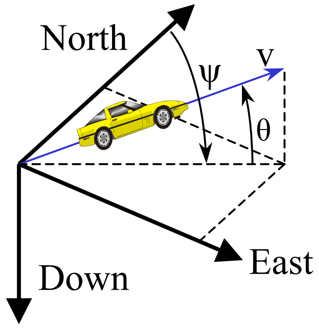

Figure 2.9: The GPS velocity and attitude

due to the wind or maneuver. Therefore, Eqs. (2.73a) and (2.73b) cannot be applied in such cases. If the heading is completely unknown, then the EKF with a large heading uncertainty (LHU) model must be used for the coarse alignment, which will be discussed in Chapter 3.

Once the coarse alignment is achieved, the fine alignment is usually applied using the EKF with a small heading uncertainty (SHU) model. Table 2.1 summarizes the alignment methods discussed in this section.

Table 2.1: Alignment methods

	Static	In-motion
Coarse	• Leveling/gyro-compassing • Analytic	• GPS velocity • EKF with LHU model
Fine	EKF with small heading uncertainty model	

Chapter 3

The Extended Kalman Filter and Smoother for Aided INS

The Kalman filter used in the field of navigation is a minimum variance estimator. Minimum variance estimation problems can be defined simply through the use of conditional expectation (Meditch, 1969, p. 162):

$$\hat{\mathbf{x}}_{k|j} = E\left[\mathbf{x}_k|\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_j\right] \tag{3.1}$$

where $E[\cdot]$ is the expectation operator; $\mathbf{x}$ is a state vector; and $\mathbf{z}$ represents measurements. If $k > j$, the estimation problem is one of prediction; if $k = j$, one of filtering; and, if $k < j$, it is one of smoothing or interpolation. In this chapter, the Kalman filter and smoother for multi-sensor integration will be described using these notations.

Traditionally, the error state Kalman filter, which can be considered as either the linearized Kalman filter (LKF) or the extended Kalman filter (EKF), has been used in the field of navigation. If an INS error control loop (feedback) exists, then the LKF can be considered as an EKF. In the development of the EKF, the system process model and the measurement model have to be linearized. Sections 3.1 and 3.2 describe the system model linearization. Following this, the EKF and a linear smoother will be developed successively.

3.1 INS Error Models

Due to the uncertainties in the sensors and the gravity field, the navigation parameters obtained from the INS mechanization equation contain errors. Many models have been developed to describe the time-dependent behaviour of these errors, the choice of which is mainly dependent on the application. This section will summarize several error models of importance and discuss implications on low-cost inertial navigation.

3.1.1 The Phi-Angle Error Model

A classical approach to INS error analysis is by perturbation analysis, where the navigation parameters are perturbed with respect to the true n-frame. The perturbation model can be obtained by perturbing all of the navigation parameters appearing in Eqs. (2.42a) to (2.42d), which is the same as applying the Taylor series expansion and retaining only the constant and linear terms (Scherzinger, 2002). Hence, the basic assumption is that all the errors are small enough, especially the heading error. The derivation of the perturbation model is described in many studies in the literature; for instance, see Britting (1971), Farrell and Barth (1998), and Schwarz and Wei (2000).

We first write the position error vector as

$$\delta\mathbf{r}^{\mathrm{n}} = \begin{bmatrix} \delta r_{\mathrm{N}} & \delta r_{\mathrm{E}} & \delta r_{\mathrm{D}} \end{bmatrix}^T \tag{3.2a}$$

where δ denotes errors and $\delta r_{\mathrm{D}} = -\delta h$ for the NED coordinate system. Perturbations on other parameters can be written as follows:

$$\hat{\mathbf{f}}^{\mathrm{b}} = \mathbf{f}^{\mathrm{b}} + \delta\mathbf{f}^{\mathrm{b}}, \tag{3.2b}$$

$$\hat{\boldsymbol{\omega}}_{\mathrm{ib}}^{\mathrm{b}} = \boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} + \delta\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}}, \tag{3.2c}$$

$$\hat{\mathbf{v}}^{\mathrm{n}} = \mathbf{v}^{\mathrm{n}} + \delta\mathbf{v}^{\mathrm{n}}, \tag{3.2d}$$

$$\hat{\mathbf{C}}_{\mathrm{b}}^{\mathrm{n}} = [\mathbf{I} - (\boldsymbol{\phi}\times)]\,\mathbf{C}_{\mathrm{b}}^{\mathrm{n}}, \tag{3.2e}$$

$$\hat{\boldsymbol{\omega}}_{\mathrm{ie}}^{\mathrm{n}} = \boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}} + \delta\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}}, \tag{3.2f}$$

$$\hat{\boldsymbol{\omega}}_{\mathrm{in}}^{\mathrm{n}} = \boldsymbol{\omega}_{\mathrm{in}}^{\mathrm{n}} + \delta\boldsymbol{\omega}_{\mathrm{in}}^{\mathrm{n}}, \tag{3.2g}$$

$$\hat{\mathbf{g}}^{\mathrm{n}} = \mathbf{g}^{\mathrm{n}} + \delta\mathbf{g}^{\mathrm{n}}. \tag{3.2h}$$

With a simplified inverse gravity model, the gravity computation error can be written as follows (Rogers, 2000, p. 70):

$$\delta\mathbf{g}^{\mathrm{n}} = \begin{bmatrix} 0 & 0 & 2g\delta r_{\mathrm{D}}/(R+h) \end{bmatrix}^T, \tag{3.3}$$

where $R = \sqrt{R_{\mathrm{M}} R_{\mathrm{N}}}$ is the Gaussian mean Earth radius of curvature. Then, the perturbation model can be written as follows (Scherzinger, 2002):

$$\delta\dot{\mathbf{r}}^{\mathrm{n}} = -\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{en}} \times \delta\mathbf{r}^{\mathrm{n}} + \delta\boldsymbol{\theta} \times \mathbf{v}^{\mathrm{n}} + \delta\mathbf{v}^{\mathrm{n}}, \tag{3.4a}$$

$$\begin{aligned}\delta\dot{\mathbf{v}}^{\mathrm{n}} = {} & \mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\delta\mathbf{f}^{\mathrm{b}} + \mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\mathbf{f}^{\mathrm{b}} \times \boldsymbol{\phi} + \delta\mathbf{g}^{\mathrm{n}} \\ & - (\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{ie}} + \boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{in}}) \times \delta\mathbf{v}^{\mathrm{n}} - (\delta\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{ie}} + \delta\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{in}}) \times \mathbf{v}^{\mathrm{n}},\end{aligned} \tag{3.4b}$$

$$\dot{\boldsymbol{\phi}} = -\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{in}} \times \boldsymbol{\phi} + \delta\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{in}} - \mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\delta\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}}. \tag{3.4c}$$

Since the attitude errors are expressed in terms of the $\boldsymbol{\phi}$-angle, defined in Eq. (2.39), this model is also called the $\boldsymbol{\phi}$-angle error model.

3.1.2 The Psi-Angle Error Model

The error analysis can also be done with respect to the c-frame. The computed navigation parameters may have one of the two representations as listed below (Scherzinger, 2002):

$$\hat{\mathbf{v}}^{\mathrm{n}} = \mathbf{v}^{\mathrm{n}} + \delta\mathbf{v}^{\mathrm{n}}_{1} = \mathbf{v}^{\mathrm{c}} + \delta\mathbf{v}^{\mathrm{c}}_{2}, \tag{3.5a}$$

$$\hat{\mathbf{g}}^{\mathrm{n}} = \mathbf{g}^{\mathrm{n}} + \delta\mathbf{g}^{\mathrm{n}}_{1} = \mathbf{g}^{\mathrm{c}} + \delta\mathbf{g}^{\mathrm{c}}_{2}, \tag{3.5b}$$

$$\hat{\boldsymbol{\omega}}^{\mathrm{n}}_{\mathrm{ie}} = \boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{ie}} + \delta\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{ie}} = \boldsymbol{\omega}^{\mathrm{c}}_{\mathrm{ie}}, \tag{3.5c}$$

$$\hat{\boldsymbol{\omega}}^{\mathrm{n}}_{\mathrm{in}} = \boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{in}} + \delta\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{in}} = \boldsymbol{\omega}^{\mathrm{c}}_{\mathrm{ic}}. \tag{3.5d}$$

In c-frame analysis, $\mathbf{C}^{\mathrm{e}}_{\mathrm{c}}$, $\boldsymbol{\omega}^{\mathrm{c}}_{\mathrm{ie}}$, and $\boldsymbol{\omega}^{\mathrm{c}}_{\mathrm{ic}}$ are known without error because we know the position and the transport rate of the c-frame from the navigation computer. Scherzinger (2002) showed that the navigation parameter errors between the two representations have the following relationships:

$$\begin{aligned} \delta\mathbf{v}_1^{\mathrm{n}} &= \delta\mathbf{v}_2^{\mathrm{c}} - \delta\boldsymbol{\theta} \times \mathbf{v}^{\mathrm{c}}, && (3.6a) \\ \delta\mathbf{g}_1^{\mathrm{n}} &= \delta\mathbf{g}_2^{\mathrm{c}} - \delta\boldsymbol{\theta} \times \mathbf{g}^{\mathrm{c}}, && (3.6b) \\ \delta\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}} &= -\delta\boldsymbol{\theta} \times \boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{c}}. && (3.6c) \end{aligned}$$

From Eqs. (3.3), (3.6b) and (2.35), we can write the gravity error in the c-frame as (Scherzinger, 1996)

$$\begin{aligned} \delta\mathbf{g}_2^{\mathrm{c}} &= \begin{bmatrix} \dfrac{-g\delta r_{\mathrm{N}}}{R_{\mathrm{M}}+h} & \dfrac{-g\delta r_{\mathrm{E}}}{R_{\mathrm{N}}+h} & \dfrac{2g\delta r_{\mathrm{D}}}{R+h} \end{bmatrix}^T \\ &\approx \begin{bmatrix} -\omega_{\mathrm{s}}^2\delta r_{\mathrm{N}} & -\omega_{\mathrm{s}}^2\delta r_{\mathrm{E}} & 2\omega_{\mathrm{s}}^2\delta r_{\mathrm{D}} \end{bmatrix}^T, \end{aligned} \tag{3.7}$$

where ω_{s} is the Schuler frequency. The c-frame analysis resulted in the following error model (Scherzinger, 2002):

$$\begin{aligned} \delta\dot{\mathbf{r}}^{\mathrm{c}} &= -\boldsymbol{\omega}_{\mathrm{ec}}^{\mathrm{c}} \times \delta\mathbf{r}^{\mathrm{c}} + \delta\mathbf{v}^{\mathrm{c}}, && (3.8a) \\ \delta\dot{\mathbf{v}}^{\mathrm{c}} &= \mathbf{f}^{\mathrm{c}} \times \boldsymbol{\psi} - (2\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{c}} + \boldsymbol{\omega}_{\mathrm{ec}}^{\mathrm{c}}) \times \delta\mathbf{v}^{\mathrm{c}} + \delta\mathbf{g}^{\mathrm{c}} + \mathbf{C}_{\mathrm{b}}^{\mathrm{p}}\delta\mathbf{f}^{\mathrm{b}}, && (3.8b) \\ \dot{\boldsymbol{\psi}} &= -(\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{c}} + \boldsymbol{\omega}_{\mathrm{ec}}^{\mathrm{c}}) \times \boldsymbol{\psi} - \mathbf{C}_{\mathrm{b}}^{\mathrm{n}}\delta\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}}. && (3.8c) \end{aligned}$$

Since the attitude errors are expressed in terms of the $\boldsymbol{\psi}$-angles, this model is called the $\boldsymbol{\psi}$-angle error model.

Benson Jr. (1975) showed the basic equivalence of the $\boldsymbol{\phi}$-angle and $\boldsymbol{\psi}$-angle error models. However, Scherzinger (2002) discussed the following distinctions in the implementation:

- The $\boldsymbol{\psi}$-angle dynamics, Eq. (3.8c), is rate-stable, independent of all other INS errors and driven only by the gyro biases.

- The position and velocity error dynamics, Eqs. (3.8a) and (3.8b), are independent of angular rate errors that are due to the Earth rate and transport rate mis-resolved by the c-frame misalignment.

- If the INS is aided by the GPS, the position errors become small, and the c-frame misalignment $\delta\boldsymbol{\theta}$ becomes very small. Hence, the $\boldsymbol{\phi}$ angle converges to the $\boldsymbol{\psi}$ angle.

- The $\boldsymbol{\psi}$-angle error model contains fewer terms and hence is more easily implemented in a Kalman filter.

3.1.3 Modified Error Models

Both the $\boldsymbol{\phi}$-angle and $\boldsymbol{\psi}$-angle error models contain mis-resolved specific force terms, $\mathbf{f}^{\mathrm{n}} \times \boldsymbol{\phi}$ and $\mathbf{f}^{\mathrm{c}} \times \boldsymbol{\psi}$, respectively. Therefore, the specific force terms appear in the transition matrix for a discrete-time Kalman filter. Consequently, a high-speed integration must be applied in the transition matrix computation if the vehicle is moving with high dynamics. For low-cost IMUs, the specific force outputs are usually corrupted by large biases and sources of noise, which can lead to composing an erroneous transition matrix and therefore distorting the estimates. Scherzinger and Reid (1994) developed modified error models to solve these problems, which cancel the specific force terms by introducing the following velocity transformations:

$$\Delta\mathbf{v}_1^{\mathrm{n}} = \delta\mathbf{v}_1^{\mathrm{n}} - \hat{\mathbf{v}}^{\mathrm{n}} \times \boldsymbol{\phi}, \tag{3.9a}$$

$$\Delta\mathbf{v}_2^{\mathrm{c}} = \delta\mathbf{v}_2^{\mathrm{c}} - \hat{\mathbf{v}}^{\mathrm{n}} \times \boldsymbol{\psi}. \tag{3.9b}$$

For brevity, the modified error models will be presented here without derivations. Refer to Scherzinger and Reid (1994) for the detailed derivation. The modified $\boldsymbol{\phi}$-angle error model is written as follows:

$$\delta\dot{\mathbf{r}}^{\mathrm{n}} = -\boldsymbol{\omega}_{\mathrm{en}}^{\mathrm{n}} \times \delta\mathbf{r}^{\mathrm{n}} + \Delta\mathbf{v}_1^{\mathrm{n}} + (\delta\boldsymbol{\theta} - \boldsymbol{\phi}) \times \hat{\mathbf{v}}^{\mathrm{n}}, \tag{3.10a}$$

$$\begin{aligned}\Delta\dot{\mathbf{v}}_1^{\mathrm{n}} = {} & \mathbf{C}_{\mathrm{b}}^{\mathrm{n}}\delta\mathbf{f}^{\mathrm{b}} + \delta\mathbf{g}_1^{\mathrm{n}} - \hat{\mathbf{g}}^{\mathrm{n}} \times \boldsymbol{\phi} - (\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{c}} + \boldsymbol{\omega}_{\mathrm{ic}}^{\mathrm{c}}) \times \Delta\mathbf{v}_1^{\mathrm{n}} \\ & -\hat{\mathbf{v}}^{\mathrm{n}} \times (\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{c}} \times \boldsymbol{\phi}) + \hat{\mathbf{v}}^{\mathrm{n}} \times \delta\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}} + \hat{\mathbf{v}}^{\mathrm{n}} \times \mathbf{C}_{\mathrm{b}}^{\mathrm{n}}\delta\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}},\end{aligned} \tag{3.10b}$$

$$\dot{\boldsymbol{\phi}} = -\boldsymbol{\omega}_{\mathrm{in}}^{\mathrm{n}} \times \boldsymbol{\phi} + \delta\boldsymbol{\omega}_{\mathrm{in}}^{\mathrm{n}} - \mathbf{C}_{\mathrm{b}}^{\mathrm{n}}\delta\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}}. \tag{3.10c}$$

The modified $\boldsymbol{\psi}$-angle error model is written as follows:

$$\delta\dot{\mathbf{r}}^{\mathrm{c}} = -\boldsymbol{\omega}_{\mathrm{ec}}^{\mathrm{c}} \times \delta\mathbf{r}^{\mathrm{c}} + \Delta\mathbf{v}^{\mathrm{c}} + \hat{\mathbf{v}}^{\mathrm{n}} \times \boldsymbol{\psi}, \tag{3.11a}$$

$$\begin{aligned}\Delta\dot{\mathbf{v}}_2^{\mathrm{c}} = {} & \mathbf{C}_{\mathrm{b}}^{\mathrm{p}}\delta\mathbf{f}^{\mathrm{b}} + \delta\mathbf{g}_2^{\mathrm{c}} - \hat{\mathbf{g}}^{\mathrm{n}} \times \boldsymbol{\psi} - (2\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{c}} + \boldsymbol{\omega}_{\mathrm{ec}}^{\mathrm{c}}) \times \Delta\mathbf{v}_2^{\mathrm{c}} \\ & -\hat{\mathbf{v}}^{\mathrm{n}} \times (\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{c}} \times \boldsymbol{\psi}) + \hat{\mathbf{v}}^{\mathrm{n}} \times \mathbf{C}_{\mathrm{b}}^{\mathrm{p}}\delta\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}},\end{aligned} \tag{3.11b}$$

$$\dot{\boldsymbol{\psi}} = -(\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{c}} + \boldsymbol{\omega}_{\mathrm{ec}}^{\mathrm{c}}) \times \boldsymbol{\psi} - \mathbf{C}_{\mathrm{b}}^{\mathrm{p}}\delta\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}}. \tag{3.11c}$$

The attitude error dynamics equations are identical in both modified error models.

3.1.4 Large Heading Uncertainty Models

The basic assumption about the error models discussed so far is that all of the attitude errors are small. However, the initial heading may have a large uncertainly or be completely unknown. For instance, if the heading is obtained from Eq. (2.73b) while the vehicle is moving backward, then the heading can have an error of ±180°. Furthermore, since helicopters can move in any direction in the air, the heading cannot be derived from the GPS velocity vector as in Eq. (2.73b). Hence, large heading uncertainty (LHU) models have been developed mainly for in-motion or in-air alignment until the heading error becomes small enough (typically a few degrees) for the fine alignment routine to be activated using a small heading uncertainty (SHU) model. If a low-cost IMU is used, the heading can be completely unknown even in stationary mode because of large gyro-compassing errors. Therefore, a LHU model can also be applied for alignment of a low-cost INS.

There are two approaches in the development of LHU models. Rogers (1997) used errors of trigonometric functions of the wander azimuth angle as a part of the state vector; similarly, Scherzinger (1996) used trigonometric functions of the heading error. In the former approach, a totally different error model is required if the heading uncertainty goes below a certain threshold. However, the latter approach, although still requiring a switch in the attitude error dynamics model, provides a continuous transition. If a low-accuracy IMU is used, the heading error can grow fast in a very short time in the absence of aiding information because uncertainties in z-gyroscope drives the heading error and the aided INS cannot estimate and regulate its heading error. This situation can also happen when the vehicle is driven with a constant speed due to the poor observability of the heading. In this case, the error model switch can be done in both directions and, therefore, the latter approach will be more appropriate. The latter approach resulted in developing the modified geographic consistent (GC) model and the modified platform consistent (PC) model. The only limitation of the latter approach is that model selection is dependent upon the type of the aiding sensors. If the GPS is used, then the modified GC model is indicated. On the other hand, if a body-referenced velocity sensor is aiding the INS, the modified PC model must be chosen. Since the modified PC model has implementation difficulties due to nonlinearity and the GPS is the the main aiding sensor, the modified GC model will be introduced here.

Scherzinger (1996) first defined the extended misalignment vector $\boldsymbol{\psi}_{\mathrm{e}}$ as follows:

$$\boldsymbol{\psi}' \equiv \begin{bmatrix} \psi_{\mathrm{x}} \\ \psi_{\mathrm{y}} \\ \sin\psi_{\mathrm{z}} \end{bmatrix}, \qquad \boldsymbol{\psi}_{\mathrm{e}} \equiv \begin{bmatrix} \boldsymbol{\psi}' \\ \cos\psi_{\mathrm{z}} - 1 \end{bmatrix}, \tag{3.12}$$

where $\sin\psi_{\mathrm{z}}$ and $\cos\psi_{\mathrm{z}} - 1$ are treated as random constants when the heading error is large (Scherzinger, 2004). For small heading errors $\boldsymbol{\psi}'$ converges to $\boldsymbol{\psi}$. The matrix representation of the “extended cross-product type (+)” operator was defined as follows:

$$(\mathbf{a}\times)_{\mathrm{e}+} \equiv \left[\begin{array}{c|c} & a_{\mathrm{x}} \\ (\mathbf{a}\times) & a_{\mathrm{y}} \\ & 0 \end{array}\right], \tag{3.13}$$

where $\mathbf{a}$ is an arbitrary 3×1 vector: $\mathbf{a} = \begin{bmatrix} a_{\mathrm{x}} & a_{\mathrm{y}} & a_{\mathrm{z}} \end{bmatrix}^T$. Then, Scherzinger (1996) derived the GC LHU model based on the ψ-angle error model:

$$\delta\dot{\mathbf{r}}^{\mathrm{c}} = -\boldsymbol{\omega}^{\mathrm{c}}_{\mathrm{ec}} \times \delta\mathbf{r}^{\mathrm{c}} + \delta\mathbf{v}^{\mathrm{c}}, \tag{3.14a}$$

$$\delta\dot{\mathbf{v}}^{\mathrm{c}} = (\mathbf{f}^{\mathrm{c}}\times)_{e+}\,\boldsymbol{\psi}_e + \mathbf{f}_{\mathrm{z}}\left(\mathbf{f}^{\mathrm{c}}, \boldsymbol{\psi}\right) - (2\boldsymbol{\omega}^{\mathrm{c}}_{\mathrm{ie}} + \boldsymbol{\omega}^{\mathrm{c}}_{\mathrm{ec}}) \times \delta\mathbf{v}^{\mathrm{c}} + \delta\mathbf{g}^{\mathrm{c}} + \mathbf{C}^{\mathrm{p}}_{\mathrm{b}}\delta\mathbf{f}^{\mathrm{b}}, \tag{3.14b}$$

$$\dot{\boldsymbol{\psi}}_{\mathrm{e}} = \begin{bmatrix} -(\boldsymbol{\omega}^{\mathrm{c}}_{\mathrm{ic}}\times)_{\mathrm{e}+} \\ \mathbf{0}_{1\times4} \end{bmatrix} \boldsymbol{\psi}_e + \begin{bmatrix} \mathbf{0}_{2\times1} \\ \omega_{\mathrm{x}}\psi_{\mathrm{y}} - \omega_{\mathrm{y}}\psi_{\mathrm{x}} \\ 0 \end{bmatrix} - \begin{bmatrix} \mathbf{C}^{\mathrm{p}}_{\mathrm{b}} \\ \mathbf{0}_{1\times3} \end{bmatrix} \delta\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}}, \tag{3.14c}$$

where $\boldsymbol{\omega}^{\mathrm{c}}_{\mathrm{ic}} = \begin{bmatrix} \omega_{\mathrm{x}} & \omega_{\mathrm{y}} & \omega_{\mathrm{z}} \end{bmatrix}^T$. With $\mathbf{f}^{\mathrm{c}} = \begin{bmatrix} f_{\mathrm{x}} & f_{\mathrm{y}} & f_{\mathrm{z}} \end{bmatrix}^T$, $\mathbf{f}_{\mathrm{z}}\left(\mathbf{f}^{\mathrm{c}}, \boldsymbol{\psi}\right)$ in Eq. (3.14b) is defined as

$$\mathbf{f}_{\mathrm{z}}\left(\mathbf{f}^{\mathrm{c}}, \boldsymbol{\psi}\right) \equiv \begin{bmatrix} 0 \\ 0 \\ f_{\mathrm{x}}(\psi_{\mathrm{x}}\sin\psi_{\mathrm{z}} + \psi_{\mathrm{y}}(\cos\psi_{\mathrm{z}} - 1)) + f_{\mathrm{y}}(\psi_{\mathrm{y}}\sin\psi_{\mathrm{z}} - \psi_{\mathrm{x}}(\cos\psi_{\mathrm{z}} - 1)) \end{bmatrix} \tag{3.15}$$

and contains nonlinear terms in the vertical channel, which can be considered as either negligible or approximately random with respect to larger long-term vertical acceleration errors (Scherzinger, 1996). If the heading uncertainty becomes small, Eq.

(3.14c) can be replaced by

$$\dot{\boldsymbol{\psi}}_{\mathrm{e}} = \begin{bmatrix} -(\boldsymbol{\omega}^{\mathrm{c}}_{\mathrm{ic}}\times)_{\mathrm{e+}} \\ \mathbf{0}_{1\times 4} \end{bmatrix} \boldsymbol{\psi}_{e} - \begin{bmatrix} \mathbf{C}^{\mathrm{p}}_{\mathrm{b}} \\ \mathbf{0}_{1\times 3} \end{bmatrix} \delta\boldsymbol{\omega}^{\mathrm{b}}, \tag{3.16}$$

which is equivalent to the attitude dynamics of the SHU $\boldsymbol{\psi}$-angle model, Eq. (3.8c).

Finally, with the velocity transformation

$$\Delta\mathbf{v} = \delta\mathbf{v}^{\mathrm{c}} - (\mathbf{v}^{\mathrm{c}}\times)_{\mathrm{e+}}\,\boldsymbol{\psi}_{\mathrm{e}}, \tag{3.17}$$

Scherzinger (1996) derived the modified GC LHU model as follows:

$$\delta\dot{\mathbf{r}}^{\mathrm{c}} = -\boldsymbol{\omega}^{\mathrm{c}}_{\mathrm{ec}} \times \delta\mathbf{r}^{\mathrm{c}} + \Delta\mathbf{v}^{\mathrm{c}} + (\mathbf{v}^{\mathrm{c}}\times)_{\mathrm{e+}}\,\boldsymbol{\psi}_{\mathrm{e}}, \tag{3.18a}$$

$$\begin{aligned} \Delta\dot{\mathbf{v}}^{\mathrm{c}} = {} & -\mathbf{g}^{\mathrm{c}} \times \boldsymbol{\psi}' - (2\boldsymbol{\omega}^{\mathrm{c}}_{\mathrm{ie}} + \boldsymbol{\omega}^{\mathrm{c}}_{\mathrm{ec}}) \times \Delta\mathbf{v}^{\mathrm{c}} + \delta\mathbf{g}^{\mathrm{c}} + \mathbf{C}^{\mathrm{p}}_{\mathrm{b}}\delta\mathbf{f}^{\mathrm{b}} \\ & -\mathbf{v}^{\mathrm{c}} \times (\boldsymbol{\omega}^{\mathrm{c}}_{\mathrm{ie}} \times \boldsymbol{\psi}') + (\mathbf{v}^{\mathrm{c}}\times)\mathbf{C}^{\mathrm{p}}_{\mathrm{b}}\delta\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}} + \mathbf{f}_{\mathrm{z}}(\mathbf{f}^{\mathrm{c}}, \boldsymbol{\psi}) + \mathbf{f}_{\mathrm{c}}(\boldsymbol{\psi}', \mathbf{v}^{\mathrm{c}}) \\ & + \begin{bmatrix} \Omega^{\mathrm{c}}_{\mathrm{E}} v^{\mathrm{c}}_{\mathrm{D}} \\ -\Omega^{\mathrm{c}}_{\mathrm{N}} v^{\mathrm{c}}_{\mathrm{D}} \\ -(3\Omega^{\mathrm{c}}_{\mathrm{N}} + 2\rho^{\mathrm{c}}_{\mathrm{N}})v^{\mathrm{c}}_{\mathrm{E}} + (3\Omega^{\mathrm{c}}_{\mathrm{E}} + 2\rho^{\mathrm{c}}_{\mathrm{E}})v^{\mathrm{c}}_{\mathrm{N}} \end{bmatrix} (\cos\psi_{\mathrm{z}} - 1), \end{aligned} \tag{3.18b}$$

$$\dot{\boldsymbol{\psi}}_{\mathrm{e}} = \begin{bmatrix} -(\boldsymbol{\omega}^{\mathrm{c}}_{\mathrm{ic}}\times)_{\mathrm{e+}} \\ \mathbf{0}_{1\times 4} \end{bmatrix} \boldsymbol{\psi}_{\mathrm{e}} + \begin{bmatrix} \mathbf{0}_{2\times 1} \\ \omega_{\mathrm{x}}\psi_{\mathrm{y}} - \omega_{\mathrm{y}}\psi_{\mathrm{x}} \\ 0 \end{bmatrix} - \begin{bmatrix} \mathbf{C}^{\mathrm{p}}_{\mathrm{b}} \\ \mathbf{0}_{1\times 3} \end{bmatrix} \delta\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}}, \tag{3.18c}$$

where $\mathbf{v}^{\mathrm{c}} = \begin{bmatrix} v^{\mathrm{c}}_{\mathrm{N}} & v^{\mathrm{c}}_{\mathrm{E}} & v^{\mathrm{c}}_{\mathrm{D}} \end{bmatrix}^{T}$, $\boldsymbol{\omega}^{\mathrm{c}}_{\mathrm{ie}} = \begin{bmatrix} \Omega^{\mathrm{c}}_{\mathrm{N}} & \Omega^{\mathrm{c}}_{\mathrm{E}} & \Omega^{\mathrm{c}}_{\mathrm{D}} \end{bmatrix}^{T}$, and $\boldsymbol{\omega}^{\mathrm{c}}_{\mathrm{ec}} = \begin{bmatrix} \rho^{\mathrm{c}}_{\mathrm{N}} & \rho^{\mathrm{c}}_{\mathrm{E}} & \rho^{\mathrm{c}}_{\mathrm{D}} \end{bmatrix}^{T}$. In Eq. (3.18b) $\mathbf{f}_{\mathrm{c}}(\boldsymbol{\psi}, \mathbf{v}^{\mathrm{c}})$ is a correction function to account for the choice of misalignment error model, Eqs. (3.18c) or (3.16) (Scherzinger, 1996). When the SHU misalignment dynamics, Eq. (3.16), is used $\mathbf{f}_{\mathrm{c}}\left(\boldsymbol{\psi}, \mathbf{v}^{\mathrm{c}}\right) = \mathbf{0}$. When the LHU misalignment dynamics, Eq. (3.18c), is used, $\mathbf{f}_{\mathrm{c}}\left(\boldsymbol{\psi}', \mathbf{v}^{\mathrm{c}}\right)$ is given as

$$\mathbf{f}_{\mathrm{c}}\left(\boldsymbol{\psi}', \mathbf{v}^{\mathrm{c}}\right) = \begin{bmatrix} \left(\Omega_{\mathrm{E}}^{\mathrm{c}} + \rho_{\mathrm{E}}^{\mathrm{c}}\right) v_{\mathrm{E}}^{\mathrm{c}} & -\left(\Omega_{\mathrm{N}}^{\mathrm{c}} + \rho_{\mathrm{N}}^{\mathrm{c}}\right) v_{\mathrm{E}}^{\mathrm{c}} & 0 \\ -\left(\Omega_{\mathrm{E}}^{\mathrm{c}} + \rho_{\mathrm{E}}^{\mathrm{c}}\right) v_{\mathrm{N}}^{\mathrm{c}} & \left(\Omega_{\mathrm{N}}^{\mathrm{c}} + \rho_{\mathrm{N}}^{\mathrm{c}}\right) v_{\mathrm{N}}^{\mathrm{c}} & 0 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{\psi}'. \tag{3.19}$$

3.2 Residual Sensor Error Models

Quite large uncertainties exist in the outputs of low-cost inertial sensors. The best way to handle these uncertainties would be to determine the sensor errors as much as possible through calibrations. Then, for instance, Eqs. (2.45a) and (2.45b) in Section 2.3 could be used for the compensation of the biases, scale factors, and non-orthogonalities. However, since calibration significantly increases the manufacturing cost, low-cost inertial sensors are rarely calibrated by the manufacturer. Even after the calibration, there still exist switch-on to switch-on error variations. Therefore, sensor errors need to be modeled in the state vector of a navigation filter in addition to the position, velocity and attitude errors. However, it should be remembered that although a navigation filter can estimate sensor errors, compensation would yield superior results because the navigation filter can concentrate on estimating small residual errors and the filter's actual behaviour is dependent on the dynamics of the vehicle.

In Section 3.2.1, typical error terms in outputs of a low-cost IMU will be discussed.

Stochastic process models useful for describing the behaviour of sensor errors will be defined in Section 3.2.2. For backward filters, backward sensor error models need to be developed. Hence, in Section 3.2.3, the relationship between the forward and backward sensor error models will be discussed.

3.2.1 Sensor Error Terms

If only the biases are considered, then the uncertainty of the sensors can be expressed as

$$\delta\mathbf{f}^{\mathrm{b}} = \mathbf{b}_{\mathrm{a}}, \quad \delta\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} = \mathbf{b}_{\mathrm{g}}, \tag{3.20}$$

where $\mathbf{b}_{\mathrm{a}}$ and $\mathbf{b}_{\mathrm{g}}$ are residual biases of the accelerometers and gyroscopes, respectively. If the scale factors are considered at the same time, then

$$\delta\mathbf{f}^{\mathrm{b}} = \mathbf{b}_{\mathrm{a}} + \mathrm{diag}(\mathbf{f}^{\mathrm{b}})\mathbf{s}_{\mathrm{a}}, \quad \delta\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} = \mathbf{b}_{\mathrm{g}} + \mathrm{diag}(\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}})\mathbf{s}_{\mathrm{g}}, \tag{3.21}$$

where diag($\mathbf{a}$) denotes the diagonal matrix form of a vector $\mathbf{a} = \begin{bmatrix} a_{\mathrm{x}} & a_{\mathrm{y}} & a_{\mathrm{z}} \end{bmatrix}^T$:

$$\mathrm{diag}(\mathbf{a}) \equiv \begin{bmatrix} a_{\mathrm{x}} & 0 & 0 \\ 0 & a_{\mathrm{y}} & 0 \\ 0 & 0 & a_{\mathrm{z}} \end{bmatrix}; \tag{3.22}$$

and $\mathbf{s}_{\mathrm{a}}$ and $\mathbf{s}_{\mathrm{g}}$ are residual scale factors of the accelerometers and gyroscopes, respectively. If the non-orthogonalities of the accelerometer triad, $\boldsymbol{\gamma}_{\mathrm{a}}$, are considered, the specific force measurement error can be described by

$$\begin{aligned}
\delta \mathbf{f}^{\mathrm{b}} &= \mathbf{b}_{\mathrm{a}} + \operatorname{diag}(\mathbf{f}^{\mathrm{b}})\mathbf{s}_{\mathrm{a}} + \mathbf{\Gamma}_{\mathrm{a}}\boldsymbol{\gamma}_{\mathrm{a}}, && (3.23a)\\
\boldsymbol{\gamma}_{\mathrm{a}} &= \begin{bmatrix} \gamma_{\mathrm{a,xy}} & \gamma_{\mathrm{a,xz}} & \gamma_{\mathrm{a,yx}} & \gamma_{\mathrm{a,yz}} & \gamma_{\mathrm{a,zx}} & \gamma_{\mathrm{a,zy}} \end{bmatrix}^T, && (3.23b)\\
\mathbf{\Gamma}_{\mathrm{a}} &= \begin{bmatrix} f_{\mathrm{y}} & f_{\mathrm{z}} & 0 & 0 & 0 & 0 \\ 0 & 0 & f_{\mathrm{x}} & f_{\mathrm{z}} & 0 & 0 \\ 0 & 0 & 0 & 0 & f_{\mathrm{x}} & f_{\mathrm{y}}, \end{bmatrix}, && (3.23c)
\end{aligned}$$

where $\mathbf{f}^{\mathrm{b}} = \begin{bmatrix} f_{\mathrm{x}} & f_{\mathrm{y}} & f_{\mathrm{z}} \end{bmatrix}^T$. Similarly, for the gyroscope triad,

$$\begin{aligned}
\delta \boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} &= \mathbf{b}_{\mathrm{g}} + \operatorname{diag}(\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}})\mathbf{s}_{\mathrm{g}} + \mathbf{\Gamma}_{\mathrm{g}}\boldsymbol{\gamma}_{\mathrm{g}}, && (3.23d)\\
\boldsymbol{\gamma}_{\mathrm{g}} &= \begin{bmatrix} \gamma_{\mathrm{g,xy}} & \gamma_{\mathrm{g,xz}} & \gamma_{\mathrm{g,yx}} & \gamma_{\mathrm{g,yz}} & \gamma_{\mathrm{g,zx}} & \gamma_{\mathrm{g,zy}} \end{bmatrix}^T, && (3.23e)\\
\mathbf{\Gamma}_{\mathrm{g}} &= \begin{bmatrix} \omega_{\mathrm{y}} & \omega_{\mathrm{z}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega_{\mathrm{x}} & \omega_{\mathrm{z}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \omega_{\mathrm{x}} & \omega_{\mathrm{y}}, \end{bmatrix}, && (3.23f)
\end{aligned}$$

where $\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} = \begin{bmatrix} \omega_{\mathrm{x}} & \omega_{\mathrm{y}} & \omega_{\mathrm{z}} \end{bmatrix}^T$ and $\boldsymbol{\gamma}_{\mathrm{g}}$ describes the non-orthogonalities of the gyroscope triad.

The evolution of the sensor errors can be expressed as follows:

$$\begin{aligned}
\dot{\mathbf{b}}_{\mathrm{a}} &= \operatorname{diag}(\mathbf{c}_{\mathrm{ab}})\mathbf{b}_{\mathrm{a}} + \mathbf{w}_{\mathrm{ab}}, && (3.24a)\\
\dot{\mathbf{b}}_{\mathrm{g}} &= \operatorname{diag}(\mathbf{c}_{\mathrm{gb}})\mathbf{b}_{\mathrm{g}} + \mathbf{w}_{\mathrm{gb}}, && (3.24b)\\
\dot{\mathbf{s}}_{\mathrm{a}} &= \operatorname{diag}(\mathbf{c}_{\mathrm{as}})\mathbf{s}_{\mathrm{a}} + \mathbf{w}_{\mathrm{as}}, && (3.24c)\\
\dot{\mathbf{s}}_{\mathrm{g}} &= \operatorname{diag}(\mathbf{c}_{\mathrm{gs}})\mathbf{s}_{\mathrm{g}} + \mathbf{w}_{\mathrm{gs}}, && (3.24d)
\end{aligned}$$

$$\dot{\boldsymbol{\gamma}}_{\mathrm{a}} = \mathrm{diag}(\mathbf{c}_{\mathrm{a}\gamma})\boldsymbol{\gamma}_{\mathrm{a}} + \mathbf{w}_{\mathrm{a}\gamma}, \tag{3.24e}$$

$$\dot{\boldsymbol{\gamma}}_{\mathrm{g}} = \mathrm{diag}(\mathbf{c}_{\mathrm{g}\gamma})\boldsymbol{\gamma}_{\mathrm{g}} + \mathbf{w}_{\mathrm{g}\gamma}, \tag{3.24f}$$

where $\mathbf{c}_{\mathrm{gb}}$, $\mathbf{c}_{\mathrm{gs}}$, $\mathbf{c}_{\mathrm{ab}}$, $\mathbf{c}_{\mathrm{as}}$, $\mathbf{c}_{\mathrm{g}\gamma}$, $\mathbf{c}_{\mathrm{a}\gamma}$, $\mathbf{w}_{\mathrm{gb}}$, $\mathbf{w}_{\mathrm{gs}}$, $\mathbf{w}_{\mathrm{ab}}$, $\mathbf{w}_{\mathrm{as}}$, $\mathbf{w}_{\mathrm{g}\gamma}$, and $\mathbf{w}_{\mathrm{a}\gamma}$ are continuous-time sensor error model parameters that can represent the random walk, the random constant and the first-order Gauss-Markov processes.

3.2.2 Stochastic Processes

To be included in the state vector, the sensor errors must be modeled. The behaviour of the errors under the operational scenario of the given application need to be investigated beforehand. The choice of the model is dependent on the operation time, sensor performance, and working environment. If the operation time is very short, then the errors can be treated practically as constants. If the IMU should run for a very long time, then the behaviour of the sensor errors must be carefully investigated.

White Noise

White noise is defined as a stationary random process having a constant spectral density function (Brown and Hwang, 1997, p. 92), which implies that power is distributed uniformly over all frequency components in the full infinite range and thus it is not physically realizable (Gelb, 1974, p. 42). White noise is, however, a very useful approximation to situations in which a disturbing noise is wideband compared with the bandwidth of the system (Gelb, 1974, p. 42) and it can be used to describe the noise in an inertial sensor. A number of random processes can be generated by passing white noise through a suitable filter (Gelb, 1974, p. 42).

Random Walk

The random walk process can be obtained when a white noise process is integrated as shown in Figure 3.1. The continuous-time state equation for the random walk is given by

$$\dot{x} = w, \tag{3.25a}$$

where $E[w(t)w(\tau)] = q(t)\delta(t-\tau)$ (Gelb, 1974, p. 79). The corresponding discrete-time precess is

$$x_{k+1} = x_k + w_k, \tag{3.25b}$$

where the noise covariance is $q_k = q(t)\Delta t_{k+1}$ (Gelb, 1974, p. 79). The state uncertainty of the random walk process increases with time (Gelb, 1974, p. 80):

$$E\left[x^2\right] = q(t)t. \tag{3.26}$$

Therefore, the random walk process is non-stationary.

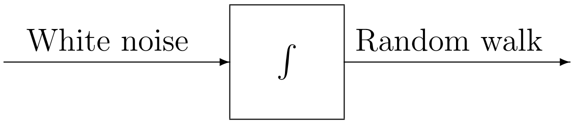

Figure 3.1: White noise and random walk

Since an INS integrates signals from accelerometers and gyroscopes, the white noise components are integrated and this will increase the uncertainty of the velocity and attitude. Velocity random walk (VRW) and angular random walk (ARW) are the terms used to describe these effects. Values of the VRW and the ARW are usually determined through the Allan (cluster) variance analysis. The Allan variance analysis for inertial sensor error modeling is well described in Tehrani (1983) and Hou and El-Sheimy (2003).

Random Constant

The random constant is a non-dynamic quantity with a fixed, albeit random, amplitude (Gelb, 1974, p. 79). The continuous and discrete processes are described in Eqs. (3.27a) and (3.27b), respectively:

$$\dot{x} = 0, \tag{3.27a}$$

$$x_{k+1} = x_k. \tag{3.27b}$$

Typical usage of the random constant will be the initial states and the extended heading error states ($\sin\psi_{\mathrm{z}}$ and $\cos\psi_{\mathrm{z}} - 1$) of the large heading uncertainty models described in Section 3.1. Non-orthogonalities of sensor triads can be dealt with by random constants during calibrations. Biases can also be considered as random constants, if the operation time is very short. If the operation time is very long, even if the state is constant, it will be preferable to add noise intentionally to prevent the state covariance from becoming non-positive definite, which results in the random walk process.

Gauss-Markov Process

A stationary Gaussian process that has an exponentially decaying autocorrelation is called a first-order Gauss-Markov process (Brown and Hwang, 1997, p. 94). For a process with correlation time T and mean squared value σ^2 (and zero mean) the model is described by (Maybeck, 1994a, p. 185)

$$\dot{x} = -\frac{1}{T}x + w, \tag{3.28a}$$

for which $q = 2\sigma^2/T$. The discrete-time model is written as

$$x_{k+1} = e^{-\Delta t_{k+1}/T} x_k + w_k, \tag{3.28b}$$

for which $q_k = \sigma^2 \left(1 - e^{-2\Delta t_{k+1}/T}\right)$. Therefore, two parameters (T and σ^2) are required to describe a Gauss-Markov process. The autocorrelation function of the Gauss-Markov model is (Maybeck, 1994a, p. 185)

$$r(\tau) = E\left[x(t)x(t+\tau)\right] = \sigma^2 e^{-|\tau|/T}, \tag{3.28c}$$

and is depicted in Figure 3.2.

The Gauss-Markov process is important in applied work because it seems to fit a large number of physical processes with reasonable accuracy, and it has a relatively simple mathematical description (Brown and Hwang, 1997, p. 95). The most outstanding characteristic of the Gauss-Markov process is that it can represent bounded

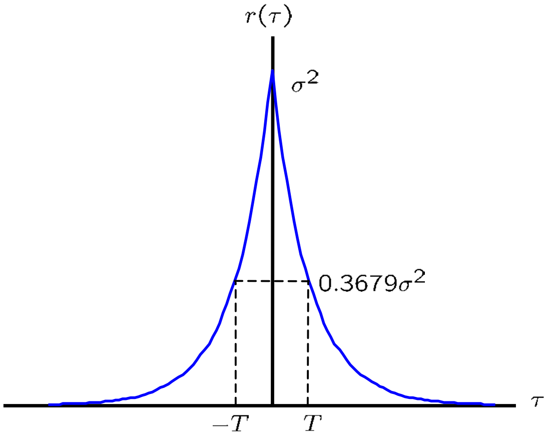

Figure 3.2: Autocorrelation function of the Gauss-Markov process

uncertainty. Hence, the Gauss-Markov process is used in INS filters to model slowly varying sensor errors such as biases and scale factors.

Let us examine the behaviour of the Gauss-Markov process using a single-state model:

$$\begin{aligned} x_k &= e^{-0.01/T} x_{k-1} + w_k, \\ z_k &= x_k + e_k, \end{aligned}$$

where $\sigma^2 = 10^2$, T=10 minutes and $E[e_k^2] = 50^2$. Figure 3.3 shows the simulation result using the Kalman filter. Measurements were available until up to 5 minutes and then the filter worked in prediction mode. The state uncertainty decreased from the designed value and converged to about 1.7 by incorporating measurements. During the measurement outage, the state will be forgotten (converge to zero) while the uncertainty will increase and eventually converge to its designed value. If $T \to \infty$, then the Gauss-Markov model becomes the random constant model. On the other

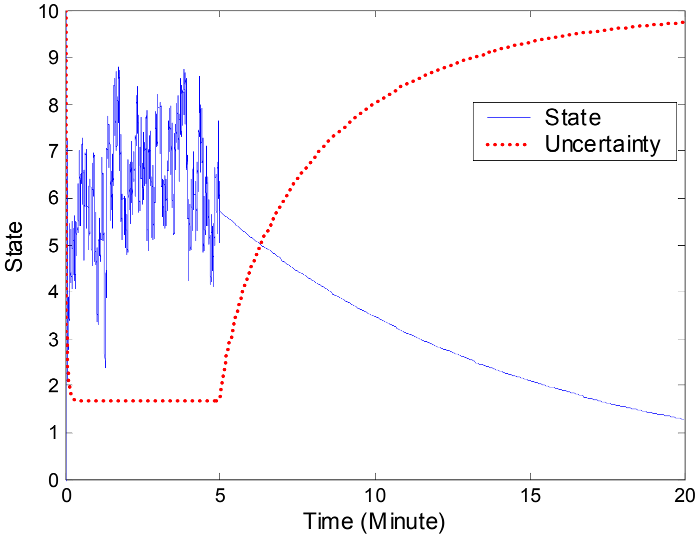

Figure 3.3: A Gauss-Markov process in free inertial navigation

hand, if $T \rightarrow 0$, then it approximates white noise.

The parameters of a Gauss-Markov process can be determined through the autocorrelation function as shown in Figure 3.2. However, as explained by Eq. (3.29a), reliable determination of autocorrelation requires considerably long-term measurements (Brown and Hwang, 1997, p. 107):

$$\frac{\mathrm{Var}[V_x(\tau)]}{\sigma^4} \leq \frac{2T}{t_c}, \tag{3.29a}$$

$$V_x(\tau) = \frac{1}{t_c - \tau}\int_0^{t_c-\tau} x(t)x(t+\tau)dt, \tag{3.29b}$$

where t_c is the total data collection time. For example, a data collection of 200 times the time constant of the process can yield merely 10 % accuracy in determining the autocorrelation function (Brown and Hwang, 1997, p. 108). Hence, if the sensor's correlation time is 4 hours, then 800 hours of continuous data collection are required to achieve 10 % accuracy. Therefore, the parameters are empirically chosen based upon the designer's understanding of the sensor behaviour under the given operational senario.

In summary, all of the random processes discussed here can be expressed by the following general model in discrete form:

$$x_{k+1} = a x_k + w_k, \tag{3.30}$$

where a is a model parameter and w_k is the driving noise. Table 3.1 lists the values of the model parameters for each of the random processes.

Table 3.1: Stochastic processes

	a	$q_k = E[w_k^2]$
Random walk	1	> 0
Random constant	1	0
Gauss-Markov	$e^{-\Delta t_{k+1}/T}$	$\sigma^2 \left(1 - e^{-2\Delta t_{k+1}/T}\right)$

3.2.3 Parameters for Backward Filters

Once the parameters of the sensor error models for forward filters are determined, those for backward filters also need to be specified. In this section, the relationship between the parameters of the forward filters and those of backward filters will be discussed based upon the linear prediction theory.

Eq. (3.30) can also be considered as a forward linear prediction (FLP) filter. The FLP error f_k can be written as

$$f_k = x_k - ax_{k-1}. \tag{3.31}$$

Hence, the mean-squared error of the FLP can be written as

$$\begin{aligned} P(a) &= E\left[f_k^2\right] \\ &= E\left[(x_k - ax_{k-1})^2\right] \\ &= E\left[x_k^2 - 2ax_kx_{k-1} + a^2x_{k-1}^2\right] \\ &= r(0) - 2ar(1) + a^2r(0), \end{aligned} \tag{3.32}$$

where $r(k)$ is the autocorrelation of a k-sample lag. Taking partial derivatives of $P(a)$ with respect to a gives the following condition:

$$r(0)a = r(1), \tag{3.33}$$

which is called the Wiener-Hopf equation and is the same as the Yule-Walker equation in autoregressive (AR) modeling (Haykin, 1996, p. 245). Hence, the filter tap-weight a can be computed from the ensemble-averaged autocorrelation function. Substituting Eq. (3.33) into Eq. (3.32) yields

$$P(a) = r(0) - ar(1). \tag{3.34}$$

To apply backward filtering, the backward sensor error model has to be developed, which can generally be written as

$$x_k = a_{\mathrm{b}}x_{k+1} + w_{k+1} \tag{3.35}$$

and can be considered as a backward linear prediction (BLP) filter. The BLP error b_k can be written as

$$b_k = x_k - a_\mathrm{b} x_{k+1}. \tag{3.36}$$

The mean-squared error of the BLP filter can be written as

$$\begin{aligned} P(a_\mathrm{b}) &= E\left[b_k^2\right] \\ &= E\left[x_k^2 - 2a_\mathrm{b} x_k x_{k+1} + a_\mathrm{b}^2 x_{k+1}^2\right] \\ &= r(0) - 2a_\mathrm{b} r(1) + a_\mathrm{b}^2 r(0). \end{aligned} \tag{3.37}$$

Taking the partial derivatives of $P(a_\mathrm{b})$ with respect to a_b yields the Wiener-Hopf equation for the BLP:

$$r(0)a_\mathrm{b} = r(1). \tag{3.38}$$

A comparison of Eqs. (3.33) and (3.38) shows that the parameters of the FLP filter and the BLP filter are the same: $a = a_\mathrm{b}$ and $P(a) = P(a_\mathrm{b})$. Therefore, a Gauss-Markov process was used to model sensor errors in forward filters, the same model can be used for the backward filters.

3.3 The Extended Kalman Filter Design

In this section, the discrete-time extended Kalman filter will be developed. The discrete-time system model will be developed first. Then, linearized measurement models for the error state will be described for the aiding sensors discussed in Section

2.5. The implementation of the error state Kalman filter is described in Section 3.3.3, and the INS error control using the estimates from the filter is discussed in Section 3.3.4.

3.3.1 Discrete-Time System

Augmentation of an INS error model with the sensor error models can be expressed in the following linear continuous-time system form:

$$\delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\delta\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t), \tag{3.39}$$

where $\mathbf{F}$ is the dynamics matrix; $\delta\mathbf{x}$ is the state vector; $\mathbf{G}$ is a noise-input mapping matrix; and $\mathbf{w}$ is a noise vector. The structure of these matrices is given in Appendix A for the $\boldsymbol{\psi}$-angle error model with the bias and scale factor error models. The elements of $\mathbf{w}(t)$ are white noises whose covariance matrix is given by

$$E[\mathbf{w}(t)\mathbf{w}(\tau)^T] = \mathbf{Q}(t)\delta(t-\tau), \tag{3.40}$$

where the operator $\delta(\cdot)$ denotes the Dirac delta function whose unit is 1/time (Gelb, 1974, p. 74–75). $\mathbf{Q}(t)$ is called the spectral density matrix.

Because strapdown inertial systems are usually implemented with high-rate sampled data, the continuous-time system equations are to be transformed to their corresponding discrete-time form:

$$\delta\mathbf{x}(t_{k+1}) = \mathbf{\Phi}(t_{k+1}, t_k)\delta\mathbf{x}(t_k) + \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(t_{k+1}, \tau)\mathbf{G}(\tau)\mathbf{w}(\tau)d\tau, \tag{3.41}$$

or in abbreviated notation

$$\delta \mathbf{x}_{k+1} = \boldsymbol{\Phi}_k \delta \mathbf{x}_k + \mathbf{w}_k, \tag{3.42}$$

where $\boldsymbol{\Phi}_k$ is the state transition matrix, and $\mathbf{w}_k$ is the driven response at t_{k+1} due to the presence of the input white noise during the time interval (t_k, t_{k+1}) (Brown and Hwang, 1992, p. 220).

If $\Delta t_{k+1} = t_{k+1} - t_k$ is very small or $\mathbf{F}(t)$ is approximately constant over Δt_{k+1}, the following numerical approximation can be applied to calculate the transition matrix:

$$\boldsymbol{\Phi}_k = \exp\left(\mathbf{F}(t_k)\Delta t_{k+1}\right) \approx \mathbf{I} + \mathbf{F}(t_k)\Delta t_{k+1}. \tag{3.43}$$

Because a white sequence is a sequence of zero-mean random variable that is uncorrelated time-wise, the covariance matrix associated with $\mathbf{w}_k$ is given by (Brown and Hwang, 1992, p. 219)

$$E[\mathbf{w}_k \mathbf{w}_i^T] = \begin{cases} \mathbf{Q}_k, & i = k \\ \mathbf{0}, & i \neq k \end{cases}. \tag{3.44}$$

$\mathbf{Q}_k$ can be expressed as (Brown and Hwang, 1992, p. 220)

$$\begin{aligned} \mathbf{Q}_k &= E\left[\mathbf{w}_k \mathbf{w}_k^T\right] \\ &= E\left\{ \left[\int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1}, \xi)\mathbf{G}(\xi)\mathbf{w}(\xi)d\xi \right] \left[\int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1}, \eta)\mathbf{G}(\eta)\mathbf{w}(\eta)d\eta \right]^T \right\} \\ &= \int_{t_k}^{t_{k+1}} \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1}, \xi)\mathbf{G}(\xi)E\left[\mathbf{w}(\xi)\mathbf{w}^T(\eta)\right]\mathbf{G}^T(\eta)\boldsymbol{\Phi}^T(t_{k+1}, \eta)d\xi d\eta. \end{aligned} \tag{3.45}$$

A trapezoidal integration of the above equation yields (Maybeck, 1994a, p. 358)

$$\mathbf{Q}_k \approx \frac{1}{2}\left[\boldsymbol{\Phi}_k \mathbf{G}(t_k)\mathbf{Q}(t_k)\mathbf{G}^T(t_k)\boldsymbol{\Phi}_k^T + \mathbf{G}(t_{k+1})\mathbf{Q}(t_{k+1})\mathbf{G}^T(t_{k+1})\right]\Delta t_{k+1}. \tag{3.46}$$

If $\mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}^T(t)$ can be treated as a constant for the time interval (t_k, t_{k+1}), an approximate solution can be obtained by substituting Eq. (3.43) into the above equation (Maybeck, 1994a, p. 358):

$$\mathbf{Q}_k \approx \frac{1}{2}\left[\mathbf{\Phi}_k\mathbf{G}(t_k)\mathbf{Q}(t_k)\mathbf{G}^T(t_k) + \mathbf{G}(t_k)\mathbf{Q}(t_k)\mathbf{G}^T(t_k)\mathbf{\Phi}_k^T\right]\Delta t_{k+1}. \tag{3.47}$$

3.3.2 Linearized Measurement Models

The derivation of the Kalman filter – a recursive, unbiased and minimum-variance estimator – starts from the random process model (i.e. Eq. (3.42)) and the following observation equations:

$$\delta\mathbf{z}_k = \mathbf{H}_k\delta\mathbf{x}_k + \mathbf{e}_k, \tag{3.48}$$

which expresses the vector measurement, $\delta\mathbf{z}_k$, at time t_k as a linear combination of the state vector, $\delta\mathbf{x}_k$, plus a random measurement error, $\mathbf{e}_k$ (Gelb, 1974; Brown and Hwang, 1992). The design matrix $\mathbf{H}_k$ is defined as

$$\mathbf{H}_k = \left.\frac{\partial\mathbf{h}[\mathbf{x}]}{\partial\mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}}, \tag{3.49}$$

where $\mathbf{h}[\mathbf{x}]$ is the nonlinear vector measurement function of the states. The measurement covariance matrix is written as

$$E\left[\mathbf{e}_k\mathbf{e}_i^T\right] = \begin{cases} \mathbf{R}_k, & i = k \\ 0, & i \neq k \end{cases}. \tag{3.50}$$

The system noise $\mathbf{w}_k$ and measurement noise $\mathbf{e}_k$ are assumed to be uncorrelated: $E\left[\mathbf{w}_k\mathbf{e}_i^T\right] = 0$ for all i, k. The rest of this section is devoted to deriving linearized measurement models for the various aiding sources presented in Section 2.5.

Linearized GPS Measurement Models

Using the truth model in Eq. (2.67a), the computed position at the centre of the ith GPS antenna can be written as follows:

$$\begin{aligned}
\hat{\mathbf{r}}^{\mathrm{n}}_{\mathrm{GPS},i} &= \hat{\mathbf{r}}^{\mathrm{n}}_{\mathrm{IMU}} + \mathbf{D}_R^{-1}\hat{\mathbf{C}}^{\mathrm{n}}_{\mathrm{b}}\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i} \\
&= \mathbf{r}^{\mathrm{n}}_{\mathrm{IMU}} + \mathbf{D}_R^{-1}\delta\mathbf{r}^{\mathrm{n}}_{\mathrm{IMU}} + \mathbf{D}_R^{-1}\left[\mathbf{I} - (\boldsymbol{\phi}\times)\right]\mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i} \\
&= \mathbf{r}^{\mathrm{n}}_{\mathrm{GPS},i} + \mathbf{D}_R^{-1}\delta\mathbf{r}^{\mathrm{n}}_{\mathrm{IMU}} - \mathbf{D}_R^{-1}\left(\boldsymbol{\phi}\times\right)\mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i} \\
&= \mathbf{r}^{\mathrm{n}}_{\mathrm{GPS},i} + \mathbf{D}_R^{-1}\delta\mathbf{r}^{\mathrm{n}}_{\mathrm{IMU}} + \mathbf{D}_R^{-1}\left(\mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i}\times\right)\boldsymbol{\phi}. \qquad (3.51)
\end{aligned}$$

The measured GPS position can be written as

$$\tilde{\mathbf{r}}^{\mathrm{n}}_{\mathrm{GPS},i} = \mathbf{r}^{\mathrm{n}}_{\mathrm{GPS},i} + \mathbf{D}_R^{-1}\mathbf{e}_{\mathbf{r}}, \qquad (3.52)$$

where $\mathbf{e}_{\mathbf{r}}$ represents the GPS position error in metres. Using Eqs. (3.51) and (3.52), the following measurement equations can be constructed for the $\boldsymbol{\phi}$-angle error model:

$$\delta\mathbf{z}_{\mathbf{r},i} = \mathbf{D}_R\left(\hat{\mathbf{r}}^{\mathrm{n}}_{\mathrm{GPS},i} - \tilde{\mathbf{r}}^{\mathrm{n}}_{\mathrm{GPS},i}\right) = \delta\mathbf{r}^{\mathrm{n}}_{\mathrm{IMU}} + \left(\mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i}\times\right)\boldsymbol{\phi} - \mathbf{e}_{\mathbf{r}}. \qquad (3.53)$$

Substituting $\boldsymbol{\phi} = \boldsymbol{\psi} + \delta\boldsymbol{\theta}$ into the above equation yields a measurement model for the $\boldsymbol{\psi}$-angle error model:

$$\begin{aligned}
\delta\mathbf{z}_{\mathbf{r},i} &= \mathbf{D}_R\left(\hat{\mathbf{r}}^{\mathrm{n}}_{\mathrm{GPS},i} - \tilde{\mathbf{r}}^{\mathrm{n}}_{\mathrm{GPS},i}\right) = \mathbf{C}^{\mathrm{n}}_{\mathrm{c}}\delta\mathbf{r}^{\mathrm{c}}_{\mathrm{IMU}} + \left(\mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i}\times\right)(\boldsymbol{\psi} + \delta\boldsymbol{\theta}) - \mathbf{e}_{\mathbf{r}} \\
&\approx \delta\mathbf{r}^{\mathrm{c}}_{\mathrm{IMU}} + \left(\mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i}\times\right)\boldsymbol{\psi} - \mathbf{e}_{\mathbf{r}}, \qquad (3.54)
\end{aligned}$$

where the term $\left(\mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i}\times\right)\delta\boldsymbol{\theta}$ is very small and therefore is neglected.

Strictly speaking, the position differencing, $\hat{\mathbf{r}}^{\mathrm{n}}_{\mathrm{GPS},i} - \tilde{\mathbf{r}}^{\mathrm{n}}_{\mathrm{GPS},i}$, in Eqs. (3.53) and (3.54) is problematic when the vehicle is near the poles or is crossing the line $\lambda = \pm 180°$. Direct differencing would yield disastrous results in the situation that the longitude outputs of the INS and the GPS are, for example, 179.999° and −179.999°, respectively. It is ambiguous whether the position error is in the latitude or in the longitude direction near the poles. A more stable means of computing the position difference would be through differencing the e-frame position vectors and transforming the position error into the navigation frame of interest:

$$\delta\mathbf{z}_{\mathbf{r},i} = \mathbf{C}^{\mathrm{n}}_{\mathrm{e}}\left(\hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{GPS},i} - \tilde{\mathbf{r}}^{\mathrm{e}}_{\mathrm{GPS},i}\right) = \mathbf{C}^{\mathrm{n}}_{\mathrm{e}}\left(\hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{IMU}} - \tilde{\mathbf{r}}^{\mathrm{e}}_{\mathrm{GPS},i}\right) + \mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i}. \tag{3.55}$$

Referring to Eq. (2.68), the computed velocity of the ith GPS antenna can be written as

$$\begin{aligned}
\hat{\mathbf{v}}^{\mathrm{n}}_{\mathrm{GPS},i} &= \hat{\mathbf{v}}^{\mathrm{n}}_{\mathrm{IMU}} - \left(\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{ie}} \times + \hat{\boldsymbol{\omega}}^{\mathrm{n}}_{\mathrm{en}}\times\right)\hat{\mathbf{C}}^{\mathrm{n}}_{\mathrm{b}}\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i} - \hat{\mathbf{C}}^{\mathrm{n}}_{\mathrm{b}}\left(\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i}\times\right)\hat{\boldsymbol{\omega}}^{\mathrm{b}}_{\mathrm{ib}} \\
&\approx \mathbf{v}^{\mathrm{n}}_{\mathrm{IMU}} + \delta\mathbf{v}^{\mathrm{n}}_{\mathrm{IMU}} - \left(\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{in}}\times\right)\mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i} + \left(\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{in}}\times\right)\left(\boldsymbol{\phi}\times\right)\mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i} \\
&\quad -\mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\left(\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i}\times\right)\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}} + \left(\boldsymbol{\phi}\times\right)\mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\left(\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i}\times\right)\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}} - \mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\left(\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i}\times\right)\delta\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}} \\
&= \mathbf{v}^{\mathrm{n}}_{\mathrm{GPS},i} + \delta\mathbf{v}^{\mathrm{n}}_{\mathrm{IMU}} - \left(\hat{\boldsymbol{\omega}}^{\mathrm{n}}_{\mathrm{in}}\times\right)\mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\left(\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i}\times\right)\boldsymbol{\phi} \\
&\quad -\mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\left(\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i} \times \boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}}\right) \times \boldsymbol{\phi} - \mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\left(\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i}\times\right)\delta\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}}. \qquad (3.56)
\end{aligned}$$

The measured GPS velocity can be written as

$$\tilde{\mathbf{v}}^{\mathrm{n}}_{\mathrm{GPS},i} = \mathbf{v}^{\mathrm{n}}_{\mathrm{GPS},i} + \mathbf{e}_{\mathbf{v}}, \tag{3.57}$$

where $\mathbf{e}_{\mathbf{v}}$ is the velocity error. Hence, the velocity error measurement equation can be written as

$$
\begin{aligned}
\delta \mathbf{z}_{\mathbf{v},i} &= \hat{\mathbf{v}}^{\mathrm{n}}_{\mathrm{GPS},i} - \tilde{\mathbf{v}}^{\mathrm{n}}_{\mathrm{GPS},i} \\
&= \delta \mathbf{v}^{\mathrm{n}}_{\mathrm{IMU}} - \left(\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{in}}\times\right) \mathbf{C}^{\mathrm{n}}_{\mathrm{b}} \left(\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i}\times\right) \boldsymbol{\phi} - \mathbf{C}^{\mathrm{n}}_{\mathrm{b}} \left(\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i} \times \boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}}\right) \times \boldsymbol{\phi} \\
&\quad + \mathbf{C}^{\mathrm{n}}_{\mathrm{b}} \left(\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i}\times\right) \delta \boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}} - \mathbf{e}_{\mathbf{v}}. \qquad (3.58)
\end{aligned}
$$

Using Eq. (2.69), the computed range between the GPS antenna and the satellite can be linearized as follows:

$$
\begin{aligned}
\hat{r}^j_i &= \left\|\hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{SV},j} - \hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{GPS},i}\right\| \\
&\approx r^j_i + \frac{\left(\hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{SV},j} - \hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{GPS},i}\right)^T}{\left\|\hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{SV},j} - \hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{GPS},i}\right\|} \left(\delta \mathbf{r}^{\mathrm{e}}_{\mathrm{SV},j} - \delta \mathbf{r}^{\mathrm{e}}_{\mathrm{GPS},i}\right) \\
&= r^j_i + \frac{\left(\hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{SV},j} - \hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{GPS},i}\right)^T}{\left\|\hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{SV},j} - \hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{GPS},i}\right\|} \delta \mathbf{r}^{\mathrm{e}}_{\mathrm{SV},j} - \frac{\left(\hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{SV},j} - \hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{GPS},i}\right)^T \mathbf{C}^{\mathrm{e}}_{\mathrm{n}}}{\left\|\hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{SV},j} - \hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{GPS},i}\right\|} \delta \mathbf{r}^{\mathrm{n}}_{\mathrm{GPS},i} \\
&= r^j_i + \frac{\left(\hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{SV},j} - \hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{GPS},i}\right)^T}{\left\|\hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{SV},j} - \hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{GPS},i}\right\|} \delta \mathbf{r}^{\mathrm{e}}_{\mathrm{SV},j} \\
&\quad - \frac{\left[\mathbf{C}^{\mathrm{n}}_{\mathrm{e}} \left(\hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{SV},j} - \hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{GPS},i}\right)\right]^T}{\left\|\hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{SV},j} - \hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{GPS},i}\right\|} \left[\delta \mathbf{r}^{\mathrm{n}}_{\mathrm{IMU}} + \left(\mathbf{C}^{\mathrm{n}}_{\mathrm{b}} \boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i}\times\right) \boldsymbol{\phi}\right]. \qquad (3.59)
\end{aligned}
$$

The term $\left(\hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{SV},j} - \hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{GPS},i}\right)^T \delta \mathbf{r}^{\mathrm{e}}_{\mathrm{SV},j} / \left\|\hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{SV},j} - \hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{GPS},i}\right\|$ is the projection of the satellite position error onto the line of sight and can be treated as the orbital error $\delta r_{\mathrm{orbit},j}$ of the jth satellite. Hence, Eq. (3.59) can be rewritten as

$$
\hat{r}^j_i = r^j_i - \frac{\left[\mathbf{C}^{\mathrm{n}}_{\mathrm{e}} \left(\hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{SV},j} - \hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{GPS},i}\right)\right]^T}{\left\|\hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{SV},j} - \hat{\mathbf{r}}^{\mathrm{e}}_{\mathrm{GPS},i}\right\|} \left[\delta \mathbf{r}^{\mathrm{n}}_{\mathrm{IMU}} + \left(\mathbf{C}^{\mathrm{n}}_{\mathrm{b}} \boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{GPS},i}\times\right) \boldsymbol{\phi}\right] + \delta r_{\mathrm{orbit},j}. \qquad (3.60)
$$

Therefore, the errors in the computed ranges are due to the orbital errors of the satellites and computed position errors of the GPS antenna.

The range measurement error in a GPS receiver originates basically in the timing error. Consequently, the pseudo-range measurement can be written as

$$\rho_i^j = r_i^j + c\left(\delta T_i - \delta t_j\right) + \delta r_{\text{iono}} + \delta r_{\text{tropo}} + \delta r_{\text{multi}} + e_\rho, \tag{3.61}$$

where c is the speed of light; δT_i is the receiver clock offset; δt_j is the satellite clock offset; δr_{iono} is the ionospheric delay; δr_{tropo} is the tropospheric delay; δr_{multi} is the multipath error; and e_ρ is the range measurement noise. Several error terms can be combined into one range bias error term, $b_{\rho,j}$. Therefore, subtracting Eq. (3.61) from Eq. (3.60) yields the following range error measurement equation:

$$\begin{aligned}\delta z_\rho &= \hat{r}_i^j - \rho_i^j \\ &= -\frac{\left[\mathbf{C}_{\text{e}}^{\text{n}}\left(\mathbf{r}_{\text{SV},j}^{\text{e}} - \hat{\mathbf{r}}_{\text{GPS},i}^{\text{e}}\right)\right]^T}{\left\|\mathbf{r}_{\text{SV},j}^{\text{e}} - \hat{\mathbf{r}}_{\text{GPS},i}^{\text{e}}\right\|}\left[\delta\mathbf{r}_{\text{IMU}}^{\text{n}} + \left(\mathbf{C}_{\text{b}}^{\text{n}}\boldsymbol{\ell}_{\text{GPS},i}^{\text{b}}\times\right)\boldsymbol{\phi}\right] \\ &\quad -c\delta T_i - b_{\rho,j} - e_\rho.\end{aligned} \tag{3.62}$$

Linearized Vehicle-Frame Velocity Measurement Model

Referring to Eq. (2.70), the computed v-frame velocity at the wheel can be written as

$$\begin{aligned}\hat{\mathbf{v}}_{\text{wheel}}^{\text{v}} &= \mathbf{C}_{\text{b}}^{\text{v}}\hat{\mathbf{C}}_{\text{n}}^{\text{b}}\hat{\mathbf{v}}_{\text{IMU}}^{\text{n}} + \mathbf{C}_{\text{b}}^{\text{v}}\left(\hat{\boldsymbol{\omega}}_{\text{nb}}^{\text{b}}\times\right)\boldsymbol{\ell}_{\text{wheel}}^{\text{b}} \\ &\approx \mathbf{C}_{\text{b}}^{\text{v}}\mathbf{C}_{\text{n}}^{\text{b}}\left[\mathbf{I} + (\boldsymbol{\phi}\times)\right]\left(\mathbf{v}_{\text{IMU}}^{\text{n}} + \delta\mathbf{v}_{\text{IMU}}^{\text{n}}\right) + \mathbf{C}_{\text{b}}^{\text{v}}\left(\boldsymbol{\omega}_{\text{nb}}^{\text{b}}\times\right)\boldsymbol{\ell}_{\text{wheel}}^{\text{b}} + \mathbf{C}_{\text{b}}^{\text{v}}\left(\delta\boldsymbol{\omega}_{\text{ib}}^{\text{b}}\times\right)\boldsymbol{\ell}_{\text{wheel}}^{\text{b}} \\ &\approx \mathbf{v}_{\text{wheel}}^{\text{v}} + \mathbf{C}_{\text{b}}^{\text{v}}\mathbf{C}_{\text{n}}^{\text{b}}\delta\mathbf{v}_{\text{IMU}}^{\text{n}} - \mathbf{C}_{\text{b}}^{\text{v}}\mathbf{C}_{\text{n}}^{\text{b}}\left(\mathbf{v}_{\text{IMU}}^{\text{n}}\times\right)\boldsymbol{\phi} - \mathbf{C}_{\text{b}}^{\text{v}}\left(\boldsymbol{\ell}_{\text{wheel}}^{\text{b}}\times\right)\delta\boldsymbol{\omega}_{\text{ib}}^{\text{b}}.\end{aligned} \tag{3.63}$$

The v-frame velocity measurement can also be expressed as

$$\tilde{\mathbf{v}}_{\text{wheel}}^{\text{v}} = \mathbf{v}_{\text{wheel}}^{\text{v}} + \mathbf{e}_{\mathbf{v}}, \tag{3.64}$$

where $\mathbf{e_v}$ is the velocity measurement noise. Therefore, the v-frame velocity error measurement equation can be expressed as

$$\begin{aligned}\delta\mathbf{z_v} &= \hat{\mathbf{v}}^{\mathrm{v}}_{\mathrm{wheel}} - \tilde{\mathbf{v}}^{\mathrm{v}}_{\mathrm{wheel}} \\ &= \mathbf{C}^{\mathrm{v}}_{\mathrm{b}}\mathbf{C}^{\mathrm{b}}_{\mathrm{n}}\delta\mathbf{v}^{\mathrm{n}}_{\mathrm{IMU}} - \mathbf{C}^{\mathrm{v}}_{\mathrm{b}}\mathbf{C}^{\mathrm{b}}_{\mathrm{n}}\left(\mathbf{v}^{\mathrm{n}}_{\mathrm{IMU}}\times\right)\boldsymbol{\phi} - \mathbf{C}^{\mathrm{v}}_{\mathrm{b}}\left(\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{wheel}}\times\right)\delta\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}} - \mathbf{e_v}, \qquad (3.65)\end{aligned}$$

where the second and third elements inside the vector measurement equation constitute the non-holonomic constraints which state that the lateral and vertical velocities are zero if the vehicle is not skidding. Substituting $\boldsymbol{\phi} = \boldsymbol{\psi} + \delta\boldsymbol{\theta}$ into the above equation and approximating $\mathbf{v}^{\mathrm{n}}_{\mathrm{IMU}}$ with $\mathbf{v}^{\mathrm{c}}_{\mathrm{IMU}}$ yield

$$\begin{aligned}\delta\mathbf{z_v} &= \mathbf{C}^{\mathrm{v}}_{\mathrm{b}}\mathbf{C}^{\mathrm{b}}_{\mathrm{n}}\delta\mathbf{v}^{\mathrm{n}}_{\mathrm{IMU}} - \mathbf{C}^{\mathrm{v}}_{\mathrm{b}}\mathbf{C}^{\mathrm{b}}_{\mathrm{n}}\left(\mathbf{v}^{\mathrm{c}}_{\mathrm{IMU}}\times\right)\left(\boldsymbol{\psi}+\delta\boldsymbol{\theta}\right) - \mathbf{C}^{\mathrm{v}}_{\mathrm{b}}\left(\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{wheel}}\times\right)\delta\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}} - \mathbf{e_v} \\ &= \mathbf{C}^{\mathrm{v}}_{\mathrm{b}}\mathbf{C}^{\mathrm{b}}_{\mathrm{n}}\left(\delta\mathbf{v}^{\mathrm{n}}_{\mathrm{IMU}} - \mathbf{v}^{\mathrm{c}}_{\mathrm{IMU}}\times\delta\boldsymbol{\theta}\right) - \mathbf{C}^{\mathrm{v}}_{\mathrm{b}}\mathbf{C}^{\mathrm{b}}_{\mathrm{n}}\left(\mathbf{v}^{\mathrm{c}}_{\mathrm{IMU}}\times\right)\boldsymbol{\psi} - \mathbf{C}^{\mathrm{v}}_{\mathrm{b}}\left(\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{wheel}}\times\right)\delta\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}} - \mathbf{e_v} \\ &= \mathbf{C}^{\mathrm{b}}_{\mathrm{n}}\delta\mathbf{v}^{\mathrm{c}}_{\mathrm{IMU}} - \mathbf{C}^{\mathrm{v}}_{\mathrm{b}}\mathbf{C}^{\mathrm{b}}_{\mathrm{n}}\left(\mathbf{v}^{\mathrm{c}}_{\mathrm{IMU}}\times\right)\boldsymbol{\psi} - \mathbf{C}^{\mathrm{v}}_{\mathrm{b}}\left(\boldsymbol{\ell}^{\mathrm{b}}_{\mathrm{wheel}}\times\right)\delta\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib}} - \mathbf{e_v}. \qquad (3.66)\end{aligned}$$

Linearized Heading Measurement Model

The heading of the vehicle is computed from the elements of the following DCM:

$$\hat{\mathbf{C}}^{\mathrm{n}}_{\mathrm{v}} = \hat{\mathbf{C}}^{\mathrm{n}}_{\mathrm{b}}\left(\mathbf{C}^{\mathrm{v}}_{\mathrm{b}}\right)^{T} = \left[\mathbf{I} - \left(\boldsymbol{\phi}\times\right)\right]\mathbf{C}^{\mathrm{n}}_{\mathrm{b}}\left(\mathbf{C}^{\mathrm{v}}_{\mathrm{b}}\right)^{T}. \qquad (3.67)$$

Let $\hat{a}_{ij}$, b_{ij}, and c_{ij} represent the ijth elements of $\hat{\mathbf{C}}^{\mathrm{n}}_{\mathrm{v}}$, $\mathbf{C}^{\mathrm{v}}_{\mathrm{b}}$, and $\mathbf{C}^{\mathrm{n}}_{\mathrm{b}}$, respectively. Then, referring to Eqs. (2.16c), the computed heading can be written as follows:

$$\hat{\psi} = \tan^{-1}\left(\hat{a}_{21}/\hat{a}_{11}\right), \qquad (3.68)$$

where

$$\begin{aligned}
\hat{a}_{11} &= b_{11}(c_{11} + c_{21}\phi_{\mathrm{z}} - c_{31}\phi_{\mathrm{y}}) + b_{12}(c_{12} + c_{22}\phi_{\mathrm{z}} - c_{32}\phi_{\mathrm{y}}) \\
&\quad + b_{13}(c_{13} + c_{23}\phi_{\mathrm{z}} - c_{33}\phi_{\mathrm{y}}), \\
\hat{a}_{21} &= b_{11}(c_{21} + c_{31}\phi_{\mathrm{x}} - c_{11}\phi_{\mathrm{z}}) + b_{12}(c_{22} + c_{32}\phi_{\mathrm{x}} - c_{12}\phi_{\mathrm{z}}) \\
&\quad + b_{13}(c_{23} + c_{33}\phi_{\mathrm{x}} - c_{13}\phi_{\mathrm{z}}).
\end{aligned}$$

Therefore, the heading error measurement equation can be written as

$$\delta z_{\psi} = \begin{bmatrix} \dfrac{\partial\hat{\psi}}{\partial\phi_{\mathrm{x}}} & \dfrac{\partial\hat{\psi}}{\partial\phi_{\mathrm{y}}} & \dfrac{\partial\hat{\psi}}{\partial\phi_{\mathrm{z}}} \end{bmatrix} \boldsymbol{\phi} + e_{\psi}, \tag{3.69}$$

where

$$\begin{aligned}
\frac{\partial\hat{\psi}}{\partial\phi_{\mathrm{x}}} &= \frac{\dfrac{\partial\hat{a}_{21}}{\partial\phi_{\mathrm{x}}}\hat{a}_{11} - \hat{a}_{21}\dfrac{\partial\hat{a}_{11}}{\partial\phi_{\mathrm{x}}}}{\hat{a}_{11}^2 + \hat{a}_{21}^2} \approx \frac{\hat{a}_{11}(b_{11}\hat{c}_{31} + b_{12}\hat{c}_{32} + b_{13}\hat{c}_{33})}{\hat{a}_{11}^2 + \hat{a}_{21}^2}, \\
\frac{\partial\hat{\psi}}{\partial\phi_{\mathrm{y}}} &= \frac{\dfrac{\partial\hat{a}_{21}}{\partial\phi_{\mathrm{y}}}\hat{a}_{11} - \hat{a}_{21}\dfrac{\partial\hat{a}_{11}}{\partial\phi_{\mathrm{y}}}}{\hat{a}_{11}^2 + \hat{a}_{21}^2} \approx \frac{\hat{a}_{21}(b_{11}\hat{c}_{31} + b_{12}\hat{c}_{32} + b_{13}\hat{c}_{33})}{\hat{a}_{11}^2 + \hat{a}_{21}^2}, \\
\frac{\partial\hat{\psi}}{\partial\phi_{\mathrm{z}}} &= \frac{\dfrac{\partial\hat{a}_{21}}{\partial\phi_{\mathrm{z}}}\hat{a}_{11} - \hat{a}_{21}\dfrac{\partial\hat{a}_{11}}{\partial\phi_{\mathrm{z}}}}{\hat{a}_{11}^2 + \hat{a}_{21}^2} \\
&\approx \frac{-\hat{a}_{11}(b_{11}\hat{c}_{11} + b_{12}\hat{c}_{12} + b_{13}\hat{c}_{13}) - \hat{a}_{21}(b_{11}\hat{c}_{21} + b_{12}\hat{c}_{22} + b_{13}\hat{c}_{33})}{\hat{a}_{11}^2 + \hat{a}_{21}^2},
\end{aligned}$$

and e_{ψ} is the heading measurement error.

Zero Integrated Heading Rate Measurement

When the vehicle's velocity is zero, although the roll and pitch errors can be controlled by applying ZUPTs, the heading error of a low-cost INS can grow rapidly due to the poor observability in heading. In this case, zero integrated heading rate (ZIHR) measurements can be used to control the heading drift (Scherzinger, 2005; Private communication).

The true heading rate is written as (Savage, 2000, p. 3-57)

$$\dot{\psi} = \begin{bmatrix} 0 & \sec\theta\sin\phi & \sec\theta\cos\phi \end{bmatrix} \boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} - \begin{bmatrix} \tan\theta\cos\psi & \tan\theta\sin\psi & 1 \end{bmatrix} \boldsymbol{\omega}_{\mathrm{in}}^{\mathrm{n}}. \tag{3.70}$$

The computed heading rate is

$$\dot{\hat{\psi}} = \begin{bmatrix} 0 & \sec\hat{\theta}\sin\hat{\phi} & \sec\hat{\theta}\cos\hat{\phi} \end{bmatrix} \tilde{\boldsymbol{\omega}}_{\mathrm{ib}}^{\mathrm{b}} - \begin{bmatrix} \tan\hat{\theta}\cos\hat{\psi} & \tan\hat{\theta}\sin\hat{\psi} & 1 \end{bmatrix} \boldsymbol{\omega}_{\mathrm{in}}^{\mathrm{n}}. \tag{3.71}$$

Because the vehicle is stationary, the transport rate is zero and the remaining is the Earth's rotation rate: $\boldsymbol{\omega}_{\mathrm{in}}^{\mathrm{n}} = \boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}}$. The sensed gyro output can be written as

$$\tilde{\boldsymbol{\omega}}_{\mathrm{ib}}^{\mathrm{b}} = \boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} + \mathbf{b}_{\mathrm{g}} + \mathbf{n}_{\mathrm{g}}, \tag{3.72}$$

where $\mathbf{b}_{\mathrm{g}}$ and $\mathbf{n}_{\mathrm{g}}$ are gyro bias and noise, respectively. We can assume that the roll and pitch errors are negligibly small during ZUPTs. However, the same cannot be said for the heading error. Hence, the heading rate error is

$$\begin{aligned}
\dot{\hat{\psi}} - \dot{\psi} &= \delta\dot{\psi} \\
&\approx \begin{bmatrix} 0 & \sec\theta\sin\phi & \sec\theta\cos\phi \end{bmatrix} \left(\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} + \mathbf{b}_{\mathrm{g}} + \mathbf{n}_{\mathrm{g}}\right) \\
&\quad - \begin{bmatrix} \tan\theta\cos(\psi+\delta\psi) & \tan\theta\sin(\psi+\delta\psi) & 1 \end{bmatrix} \boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}} \\
&\quad - \begin{bmatrix} 0 & \sec\theta\sin\phi & \sec\theta\cos\phi \end{bmatrix} \boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} \\
&\quad + \begin{bmatrix} \tan\theta\cos\psi & \tan\theta\sin\psi & 1 \end{bmatrix} \boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}} \\
&\approx \begin{bmatrix} 0 & \sec\theta\sin\phi & \sec\theta\cos\phi \end{bmatrix} \left(\mathbf{b}_{\mathrm{g}} + \mathbf{n}_{\mathrm{g}}\right) \\
&\quad - \delta\psi \begin{bmatrix} -\tan\theta\sin\psi & \tan\theta\cos\psi & 0 \end{bmatrix} \boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}} \\
&= \Omega_{\mathrm{N}} \tan\theta\sin\psi\delta\psi + b_{\mathrm{z}}^{\mathrm{n}} + \mu_{\mathrm{g}}, \qquad (3.73)
\end{aligned}$$

where Ω_{N} is the north component of the Earth's rotation rate. This is a differential equation in $\delta\psi$ whose dynamics are $\Omega_{\mathrm{N}} \tan\theta\sin\psi\delta\psi$ and whose inputs are the gyro bias and noise projected on the the vertical:

$$b_{\mathrm{z}}^{\mathrm{n}} + \mu_{\mathrm{g}} = \begin{bmatrix} 0 & \sec\theta\sin\phi & \sec\theta\cos\phi \end{bmatrix} \left(\mathbf{b}_{\mathrm{g}} + \mathbf{n}_{\mathrm{g}}\right). \qquad (3.74)$$

At this point we can compare the misresolved rate of the Earth, $\Omega_{\mathrm{N}} \tan\theta\sin\psi\delta\psi$, with the gyro errors and decide to simplify the equation by dropping the misresolution of the Earth's rate. This allows a straightforward integration as follows:

$$\begin{aligned}
\hat{\psi}_k - \hat{\psi}_{k-1} &= \delta\psi_k - \delta\psi_{k-1} \\
&\approx \begin{bmatrix} 0 & \sec\theta\sin\phi & \sec\theta\cos\phi \end{bmatrix} \Delta t_k \mathbf{b}_{\mathrm{g}} + e, \qquad (3.75a) \\
e &= \begin{bmatrix} 0 & \sec\theta\sin\phi & \sec\theta\cos\phi \end{bmatrix} \int_{t_{k-1}}^{t_k} \mathbf{n}_g dt, \qquad (3.75b)
\end{aligned}$$

where the roll and pitch are considered as constants over the time interval Δt_k. Due to the term $\sec\theta$, this type of measurement cannot be applied when the pitch is close to ±90°. The ZIHR measurement model, Eq. (3.75a), can also be used as a measurement model in the UKF.

3.3.3 Linearized Kalman Filter

The implementation of the Kalman filter can be divided into three stages: initialization, measurement update and prediction. The error state vector and its covariance are initialized as follows:

$$\delta\hat{\mathbf{x}}_{0|0} = \mathbf{0}, \quad \mathbf{P}_{0|0} = E\left[\delta\hat{\mathbf{x}}_{0|0}\delta\hat{\mathbf{x}}_{0|0}^T\right]. \tag{3.76}$$

In the prediction stage, also called the 'time update,' the estimate and its error covariance are projected ahead as follows (Brown and Hwang, 1997, p. 219):

$$\delta\hat{\mathbf{x}}_{k|k-1} = \mathbf{\Phi}_{k-1}\delta\hat{\mathbf{x}}_{k-1|k-1}, \tag{3.77a}$$

$$\mathbf{P}_{k|k-1} = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{\Phi}_{k-1}^T + \mathbf{Q}_{k-1}. \tag{3.77b}$$

In the measurement update, the Kalman gain, $\mathbf{K}_k$, is computed first, then the state and the error covariance are updated using the predicted estimate, $\delta\hat{\mathbf{x}}_{k|k-1}$, and its covariance, $\mathbf{P}_{k|k-1}$:

$$\begin{aligned}
\mathbf{P}_{\boldsymbol{\nu}\boldsymbol{\nu},k} &= \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k, && (3.78a)\\
\mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{P}_{\boldsymbol{\nu}\boldsymbol{\nu},k}^{-1}, && (3.78b)\\
\delta\hat{\mathbf{x}}_{k|k} &= \delta\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\delta\mathbf{z}_k - \mathbf{H}_k\delta\hat{\mathbf{x}}_{k|k-1}\right), && (3.78c)\\
\mathbf{P}_{k|k} &= \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_{k|k-1}\left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T. && (3.78d)
\end{aligned}$$

Eq. (3.78d) is called the "Joseph form" of the covariance update equation (Grewal and Andrews, 1993, p. 111), which yields a more stable solution than the conventional one owing to the guaranteed symmetry and reduced vulnerability to numerical errors (Maybeck, 1994a, p. 237).

Square-root filters can further improve the covariance update by propagating the square-root of the state error covariance, $\mathbf{S}$, such that $\mathbf{P} = \mathbf{S}\mathbf{S}^T$. The UD-factorization algorithm was introduced due to the heavy computational burden of the square-root implementation; it factors the state error covariance $\mathbf{P}$ into an upper-triangular matrix $\mathbf{U}$ with 1's along its main diagonal and a diagonal matrix $\mathbf{D}$: $\mathbf{P} = \mathbf{U}\mathbf{D}\mathbf{U}^T$ (Haykin, 1996, p. 327). However, with advances in modern computer technology, computational cost is not as serious a factor as it used to be. Furthermore, a Kalman filter using the UD-factorization may suffer from overflow/underflow problems (Stewart and Chapman, 1990). Therefore, the square-root implementation will be preferred for utmost quality in the covariance update (Haykin, 1996, p. 328). The square-root implementations of the Kalman covariance prediction and update are described in

Eqs. (3.79a) and (3.79b), respectively (Haykin, 1996, p. 594):

$$\begin{bmatrix} \mathbf{\Phi}_k \mathbf{S}_{k|k} & \mathbf{Q}_k^{1/2} \end{bmatrix} \mathbf{\Theta}_{1,k} = \begin{bmatrix} \mathbf{S}_{k+1|k} & \mathbf{0} \end{bmatrix}, \tag{3.79a}$$

$$\begin{bmatrix} \mathbf{R}_k^{1/2} & \mathbf{H}_k \mathbf{S}_{k|k-1} \\ \mathbf{0} & \mathbf{S}_{k|k-1} \end{bmatrix} \mathbf{\Theta}_{2,k} = \begin{bmatrix} \mathbf{P}_{\boldsymbol{\nu\nu},k}^{1/2} & \mathbf{0} \\ \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{P}_{\boldsymbol{\nu},k}^{-T/2} & \mathbf{S}_{k|k} \end{bmatrix}, \tag{3.79b}$$

where $\mathbf{Q}_k^{1/2} \mathbf{Q}_k^{T/2} = \mathbf{Q}_k$ and $\mathbf{P}_{\boldsymbol{\nu\nu},k}^{1/2} \mathbf{P}_{\boldsymbol{\nu\nu},k}^{T/2} = \mathbf{P}_{\boldsymbol{\nu\nu},k}$; $\mathbf{\Theta}_{1,k}$ and $\mathbf{\Theta}_{2,k}$ are orthogonal matrices that annihilate the right and upper-right block of the pre-array, respectively, which can be obtained through Givens rotations. For further discussion of the square-root and UD filters, the readers are advised to consult Grewal and Andrews (1993), Maybeck (1994a) and Haykin (1996).

3.3.4 INS Error Control

INS errors have to be controlled frequently in a low-cost INS. Otherwise, a low-cost INS would generate enormously large errors in an extended stand-alone operation, which could eventually lead to the violation of the small error assumption of the EKF. For horizontal position error control, $\hat{\mathbf{q}}_\mathrm{n}^\mathrm{e} = \mathbf{q}_\mathrm{c}^\mathrm{e}$ has to be corrected as follows:

$$\mathbf{q}_\mathrm{n}^\mathrm{e} = \mathbf{q}_\mathrm{c}^\mathrm{e} \star \mathbf{q}_\mathrm{n}^\mathrm{c}, \tag{3.80a}$$

$$\mathbf{q}_\mathrm{n}^\mathrm{c} = \begin{bmatrix} \cos \|0.5\delta\boldsymbol{\theta}\| \\ -\dfrac{\sin \|0.5\delta\boldsymbol{\theta}\|}{\|0.5\delta\boldsymbol{\theta}\|} 0.5\delta\boldsymbol{\theta} \end{bmatrix}, \tag{3.80b}$$

where $\delta\boldsymbol{\theta}$ can be obtained from the estimated north and east position errors using Eq. (2.35). The height is to be corrected separately as follows:

$$h = \hat{h} + \delta\hat{r}_{\mathrm{D}}. \tag{3.80c}$$

The velocity correction for the $\boldsymbol{\phi}$-angle error model can be done simply as

$$\mathbf{v}^{\mathrm{n}} = \hat{\mathbf{v}}^{\mathrm{n}} - \delta\hat{\mathbf{v}}^{\mathrm{n}}. \tag{3.81a}$$

Substituting Eq. (3.6a) into the above yields the velocity correction equation for the $\boldsymbol{\psi}$-angle error model:

$$\mathbf{v}^{\mathrm{n}} = \hat{\mathbf{v}}^{\mathrm{n}} - \delta\hat{\mathbf{v}}^{\mathrm{c}} + \delta\boldsymbol{\theta} \times \hat{\mathbf{v}}^{\mathrm{n}}. \tag{3.81b}$$

Using Eqs. (3.9a) and (3.9b), the velocity control equations for the modified error models can be written as follows:

$$\begin{aligned} \mathbf{v}^{\mathrm{n}} &= \hat{\mathbf{v}}^{\mathrm{n}} - \Delta\hat{\mathbf{v}}_1^{\mathrm{n}} - \hat{\mathbf{v}}^{\mathrm{n}} \times \hat{\boldsymbol{\phi}}, && (3.81\mathrm{c}) \\ &= \hat{\mathbf{v}}^{\mathrm{n}} - \Delta\hat{\mathbf{v}}_2^{\mathrm{c}} - \hat{\mathbf{v}}^{\mathrm{n}} \times \hat{\boldsymbol{\psi}} + \delta\boldsymbol{\theta} \times \hat{\mathbf{v}}^{\mathrm{n}}. && (3.81\mathrm{d}) \end{aligned}$$

Since $\hat{\mathbf{q}}_{\mathrm{b}}^{\mathrm{n}} = \mathbf{q}_{\mathrm{b}}^{\mathrm{p}}$, attitude correction can be written as follows:

$$\begin{aligned} \mathbf{q}_{\mathrm{b}}^{\mathrm{n}} &= \mathbf{q}_{\mathrm{p}}^{\mathrm{n}} \star \mathbf{q}_{\mathrm{b}}^{\mathrm{p}}, && (3.82\mathrm{a}) \\ \mathbf{q}_{\mathrm{p}}^{\mathrm{n}} &= \begin{bmatrix} \cos \|0.5\hat{\boldsymbol{\phi}}\| \\ \dfrac{\sin \|0.5\hat{\boldsymbol{\phi}}\|}{\|0.5\hat{\boldsymbol{\phi}}\|} 0.5\hat{\boldsymbol{\phi}} \end{bmatrix}. && (3.82\mathrm{b}) \end{aligned}$$

For the $\boldsymbol{\psi}$-angle error model, $\hat{\boldsymbol{\phi}} = \hat{\boldsymbol{\psi}} + \delta\hat{\boldsymbol{\theta}}$ has to be computed first and then Eqs. (3.82a) and (3.82b) can be applied.

3.4 Optimal Smoothing

The purpose of smoothing is to find an optimal estimate utilizing all past, current and future measurements. Meditch (1969, p. 204) classified smoothing problems into three categories, i.e. fixed-point, fixed-lag, and fixed-interval. Fixed-point smoothing is used when we are interested in the states at specific points such as orbit injection time of a satellite or initial condition of the reaction substance in a chemical process (Meditch, 1969, p. 207). Fixed-lag smoothing can be used when the existence of a fixed lag of the estimate does not impose intractable problems. Thus, the method appears attractive primarily in communication and telemetry data reduction problems (Meditch, 1969, p. 208). Fixed-interval smoothing can be used in most surveying applications, because surveying is typically amenable to post-processing where best position information is pursued for all measured points. Hence, only fixed-interval smoothing will be discussed in more detail.

In general, smoothing can be performed by combining the forward and backward filter solutions as follows:

$$\mathbf{P}_{\mathrm{sm}} = \left(\mathbf{P}_{\mathrm{f}}^{-1} + \mathbf{P}_{\mathrm{b}}^{-1}\right)^{-1}, \tag{3.83a}$$

$$\begin{aligned}\hat{\mathbf{x}}_{\mathrm{sm}} &= \mathbf{P}_{\mathrm{sm}}\left(\mathbf{P}_{\mathrm{f}}^{-1}\hat{\mathbf{x}}_{\mathrm{f}} + \mathbf{P}_{\mathrm{b}}^{-1}\hat{\mathbf{x}}_{\mathrm{b}}\right) \\ &= \mathbf{P}_{\mathrm{sm}}\left(\mathbf{P}_{\mathrm{f}}^{-1}\hat{\mathbf{x}}_{\mathrm{f}} + \mathbf{P}_{\mathrm{b}}^{-1}\hat{\mathbf{x}}_{\mathrm{f}} - \mathbf{P}_{\mathrm{b}}^{-1}\hat{\mathbf{x}}_{\mathrm{f}} + \mathbf{P}_{\mathrm{b}}^{-1}\hat{\mathbf{x}}_{\mathrm{b}}\right) \\ &= \hat{\mathbf{x}}_{\mathrm{f}} + \mathbf{P}_{\mathrm{sm}}\mathbf{P}_{\mathrm{b}}^{-1}\left(\hat{\mathbf{x}}_{\mathrm{b}} - \hat{\mathbf{x}}_{\mathrm{f}}\right),\end{aligned} \tag{3.83b}$$

where subscripts ‘f,’ ‘b,’ and ‘sm’ denote the forward, backward, and smoothed solutions, respectively. Eqs. (3.83a) and (3.83b) can be used not only for smoothing but also for combining information coming from a multi-sensor network. The implementation of the forward and backward filtering approach will be discussed in detail in Chapter 4. The inconvenience of the forward and backward filtering approach is that the full-scale backward filter has to be developed and applied separately.

The Rauch-Tung-Striebel (RTS) algorithm is a well-known fixed-interval smoother for linear filters. The RTS algorithm does not require the application of the full-scale backward filter, although it is equivalent to combining the forward and backward solutions. The algorithm is written as follows (Brown and Hwang, 1992, p. 334):

$$\delta\hat{\mathbf{x}}_{k|N} = \widehat{\delta\mathbf{x}}_{k|k} + \mathbf{A}_k\left(\widehat{\delta\mathbf{x}}_{k+1|N} - \delta\hat{\mathbf{x}}_{k+1|k}\right), \tag{3.84a}$$

$$\mathbf{P}_{k|N} = \mathbf{P}_{k|k} + \mathbf{A}_k\left(\mathbf{P}_{k+1|N} - \mathbf{P}_{k+1|k}\right)\mathbf{A}_k^T, \tag{3.84b}$$

where the smoothing gain $\mathbf{A}_k$ is given as

$$\mathbf{A}_k = \mathbf{P}_{k|k}\mathbf{\Phi}_k^T\mathbf{P}_{k+1|k}^{-1}, \tag{3.84c}$$

for $k = N-1, N-2, \ldots, 0$, and N is the total number of measurements. Figure 3.4 illustrates the computation procedure.

Usually, GPS measurement rate is much slower than INS data rate. Further, GPS measurements are sometimes not available due to signal blockages. The Kalman filter works in prediction mode between the periodic acquisition of GPS measurements and therefore the updated solutions ($\delta\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$) are not always generated during the forward filtering computation sweep. In this case, the predicted states and covariance

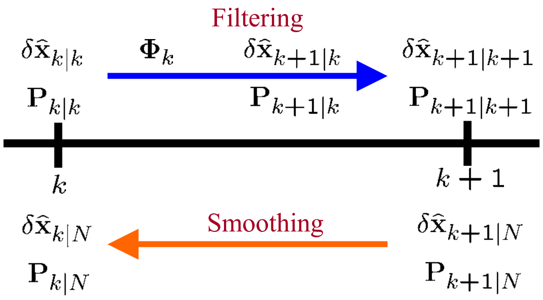
Filtering
$\delta\hat{\mathbf{x}}_{k|k}$
$\mathbf{\Phi}_k$
$\delta\hat{\mathbf{x}}_{k+1|k}$
$\delta\hat{\mathbf{x}}_{k+1|k+1}$
$\mathbf{P}_{k|k}$
$\mathbf{P}_{k+1|k}$
$\mathbf{P}_{k+1|k+1}$
k
$k+1$
$\delta\hat{\mathbf{x}}_{k|N}$
Smoothing
$\delta\hat{\mathbf{x}}_{k+1|N}$
$\mathbf{P}_{k|N}$
$\mathbf{P}_{k+1|N}$

Figure 3.4: Fixed-interval smoothing computation

are interpreted as the updated state and covariance, respectively. Once the smoothed error states are computed, the INS error control equations, discussed in Section 3.3.4, need to be applied to obtain smoothed position, velocity, and attitude. One inconvenience in the RTS algorithm is that all information of the forward filter must be saved at every processing epoch, therefore greatly increasing the storage requirement.

Chapter 4

The Unscented Kalman Filter and Smoother for Aided INS

The basic concept of the UKF is illustrated in Figure 4.1. The EKF simply transformes the mean through the given nonlinear function and considers it as the transformed mean, which is valid if the given transformation is linear. The UKF, however, samples several points from the given mean and covariance of the PDF and transforms all of the points through the given nonlinear transformation. The transformed mean and covariance are constructed from the transformed points.

Shin and El-Sheimy (2004) first applied the UKF to INS/GPS integration and demonstrated the UKF's capability of dealing with large and small attitude errors seamlessly. Unfortunately, it was possible for singularities to occur, since attitude was expressed in terms of the Euler angles, and the metric required to measure the

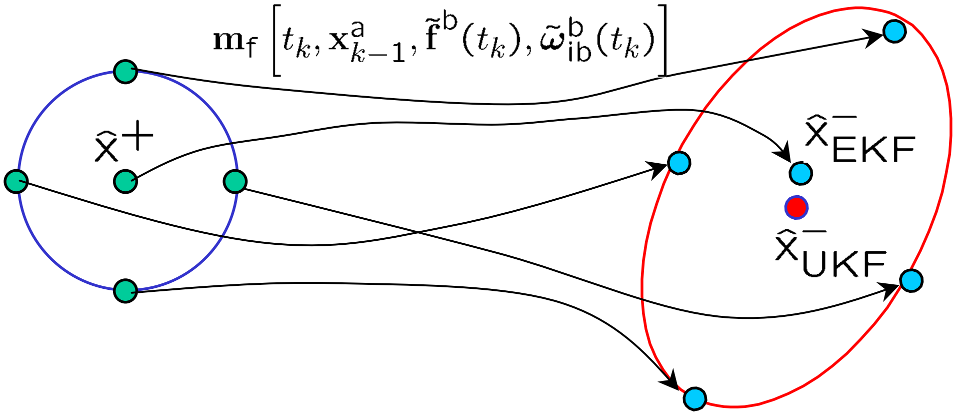
$\mathbf{m}_{\mathsf{f}}\left[t_k, \mathbf{x}^{\mathsf{a}}_{k-1}, \tilde{\mathbf{f}}^{\mathsf{b}}(t_k), \tilde{\boldsymbol{\omega}}^{\mathsf{b}}_{\mathsf{ib}}(t_k)\right]$
$\hat{x}^+$
$\hat{x}^-_{EKF}$
$\hat{x}^-_{UKF}$

Figure 4.1: The concept of the UKF

attitude difference was incomplete.

With the use of quaternion attitude representation, singularities can be resolved. Quaternion-based UKFs for attitude determination can be found in Kraft (2003) and Crassidis and Markley (2003). Kraft (2003) used rotation vectors (singular at ±180°) to express attitude covariance while Crassidis and Markley (2003) used a generalized representation, where the singularity can be placed anywhere from 180° to 360°. Shin (2004) developed a quaternion-based UKF for the integration of GPS and INS following the former approach. The structure of Shin (2004) will be used here as rotation vectors have been used extensively in Chapter 2 to support this approach.

Section 4.1 will review proposals for the unscented transformation (UT). Then, in Section 4.3, a UKF will be developed by extending the UT to the recursive estimation problem. Finally, Section 4.4 develops a smoother by combining the forward and backward UKF solutions, which will be referred to as the unscented Kalman smoother

(UKS).

4.1 Unscented Transformations

The UT refers to the procedure for obtaining a set of weights, w_i's, and sigma points, $\boldsymbol{\mathcal{X}}_i$'s, from the given mean, $\bar{\mathbf{x}}$, and covariance, $\mathbf{P}$, satisfying the following conditions:

$$\begin{cases} \sum_{i=0}^{p-1} w_i = 1 \\ \sum_{i=0}^{p-1} w_i \boldsymbol{\mathcal{X}}_i = \bar{\mathbf{x}} \\ \sum_{i=0}^{p-1} w_i (\boldsymbol{\mathcal{X}}_i - \bar{\mathbf{x}})(\boldsymbol{\mathcal{X}}_i - \bar{\mathbf{x}})^T = \mathbf{P} \end{cases}, \tag{4.1}$$

where p is the number of sigma points. This section contains a review of sigma point generation schemes and an analysis of how the UT handles nonlinearities.

4.1.1 Proposals for the UT

The first proposal for the UT, introduced in Julier and Uhlmann (1996), generates $p = 2n + 1$ sigma points, where n is the number of the states. The sigma points and weights are obtained as follows:

$$\boldsymbol{\mathcal{X}}_i = \begin{cases} \bar{\mathbf{x}} & i = 0 \\ \bar{\mathbf{x}} + \sqrt{n+\kappa}\boldsymbol{\sigma}_i & i = 1, \cdots, n \\ \bar{\mathbf{x}} - \sqrt{n+\kappa}\boldsymbol{\sigma}_{i-n} & i = n+1, \cdots, 2n \end{cases}, \tag{4.2a}$$

$$w_i = \begin{cases} \dfrac{\kappa}{n+\kappa} & i = 0 \\ \dfrac{1}{2(n+\kappa)} & i = 1, \cdots, 2n \end{cases}, \tag{4.2b}$$

where κ is a scaling parameter to adjust the effect of the fourth and higher moments of the probability distribution during given nonlinear transformations (system process and measurements); and $\boldsymbol{\sigma}_i$ is the ith column of the square-root matrix of the covariance, $\mathbf{S}$, which can be computed through Cholesky factorization:

$$\mathbf{S}\mathbf{S}^T = \mathbf{P}. \tag{4.3}$$

It was discussed in Julier and Uhlmann (1996) that $\kappa = 3 - n$ is optimal for a single-state Gaussian distribution and, for multi-dimensional systems, if $0 < n+\kappa < 3$, then the absolute error in the predicted mean is smaller than that with linearization. The characteristics in Eq. (4.1) can easily be verified using the following properties:

- The sigma points in Eq. (4.2a) are symmetric about the mean, as there exist an addition and a subtraction of the same point in Eq. (4.2a).
- $\mathbf{P} = \mathbf{S}\mathbf{S}^T = \begin{bmatrix} \boldsymbol{\sigma}_1 & \boldsymbol{\sigma}_2 & \cdots & \boldsymbol{\sigma}_n \end{bmatrix} \begin{bmatrix} \boldsymbol{\sigma}_1^T \\ \boldsymbol{\sigma}_2^T \\ \vdots \\ \boldsymbol{\sigma}_n^T \end{bmatrix} = \sum_{i=1}^{n} \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T.$

For systems with a very high sampling rate, the large number of sigma points may entail a significant computational burden. Julier and Uhlmann (2002a) reduced the number of sigma points to $n + 2$ by choosing points matching the first two moments and minimizing the third order moments (skew). Hence, the resulting sigma points are referred to as the minimal skew simplex points. The radius of the bounding hypersphere[1] of the minimal skew SPs increases rapidly as the number of states increases: $2^{n/2}$ (Julier, 2003), which causes a problem for multi-dimensional systems because the SPs can be sampled in a region far from the uncertainty level. To overcome this problem, the spherical simplex SPs were developed by Julier (2003) through imposing the following constraints on Eq. (4.1):

- The 0th weight is a free parameter that can be chosen by the designer and all other weights have the same value.
- The 0th point is the same as the mean and all other points lie on a hypersphere centred at the mean.

In the following, a brief derivation of the spherical simplex SPs will be presented for a zero-mean and unit-covariance case (i.e., $\bar{\mathbf{x}} = \mathbf{0}$ and $\mathbf{P} = \mathbf{I}$) based on Julier (2003). Then, SPs for an arbitrary mean and covariance cases will be computed from the those of zero mean and unit covariance.

Let $\boldsymbol{\mathcal{X}}^{j}_{\mathrm{u},i}$ be the ith spherical simplex SP in a j-dimensional space. Let us first find three points and weights capturing the mean and variance in one dimension. Using

[1] A hypersphere is a set of n-dimensional points whose Euclidean norm has the same value: $\{\mathbf{x} \mid \|\mathbf{x}\| = r\}$, where r is the radius of the hypersphere. For $n = 2$ and $n = 3$, it is the same a circle and a sphere, respectively.

the aforementioned constraints, the weights and SPs can be written as follows:

$$\begin{cases} w_1 = w_2 = (1 - w_0)/2 \\ \{\boldsymbol{\mathcal{X}}^1_{\mathrm{u},0}, \boldsymbol{\mathcal{X}}^1_{\mathrm{u},1}, \boldsymbol{\mathcal{X}}^1_{\mathrm{u},2}\} = \{0, -x, x\} \end{cases}, \tag{4.4}$$

where x is a value to be determined. Substituting Eq. (4.4) into the covariance condition in Eq. (4.1) yields $2w_1x^2 = 1$. Therefore, the distance from $\boldsymbol{\mathcal{X}}^1_{\mathrm{u},0}$ to $\boldsymbol{\mathcal{X}}^1_{\mathrm{u},1}$ and $\boldsymbol{\mathcal{X}}^1_{\mathrm{u},2}$ is $1/\sqrt{2w_1} = 1/\sqrt{1 - w_0}$ and the spherical simplex SPs are symmetric for one-dimensional systems:

$$\{\boldsymbol{\mathcal{X}}^1_{\mathrm{u},0}, \boldsymbol{\mathcal{X}}^1_{\mathrm{u},1}, \boldsymbol{\mathcal{X}}^1_{\mathrm{u},2}\} = \{0, -1/\sqrt{2w_1}, 1/\sqrt{2w_1}\}.$$

Let us extend the situation to the two-dimensional case. The weights can be computed as $w_1 = w_2 = w_3 = (1 - w_0)/3$. Then, as illustrated in Figure 4.2, $\boldsymbol{\mathcal{X}}^1_{\mathrm{u},1}$ and $\boldsymbol{\mathcal{X}}^1_{\mathrm{u},2}$ are shifted downward by x_1 and become $\boldsymbol{\mathcal{X}}^2_{\mathrm{u},1}$ and $\boldsymbol{\mathcal{X}}^2_{\mathrm{u},2}$, respectively. A new point is added on the vertical axis and let the coordinate be ax_1. The mean and covariance condition in Eq. (4.1) can now be written for the vertical axis as follows:

$$\begin{cases} -2w_1x_1 + w_1ax_1 = 0 \\ 2w_1x_1^2 + w_1a^2x_1^2 = 1 \end{cases}. \tag{4.5}$$

Hence, $a = 2$, $x_1 = 1/\sqrt{6w_1}$, and

$$\{\boldsymbol{\mathcal{X}}^2_{\mathrm{u},0}, \boldsymbol{\mathcal{X}}^2_{\mathrm{u},1}, \boldsymbol{\mathcal{X}}^2_{\mathrm{u},2}, \boldsymbol{\mathcal{X}}^2_{\mathrm{u},3}\} = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -1/\sqrt{2w_1} \\ -1/\sqrt{6w_1} \end{bmatrix}, \begin{bmatrix} 1/\sqrt{2w_1} \\ -1/\sqrt{6w_1} \end{bmatrix}, \begin{bmatrix} 0 \\ 2/\sqrt{6w_1} \end{bmatrix} \right\}. \tag{4.6}$$

All of the SPs are contained in a circle of radius $2/\sqrt{6w_1} = \sqrt{2/(1 - w_0)}$. However, the sum of the cubes of the coordinates in the vertical axis is not vanishing. Therefore, the spherical simplex SPs are not symmetric for multi-dimensional systems.

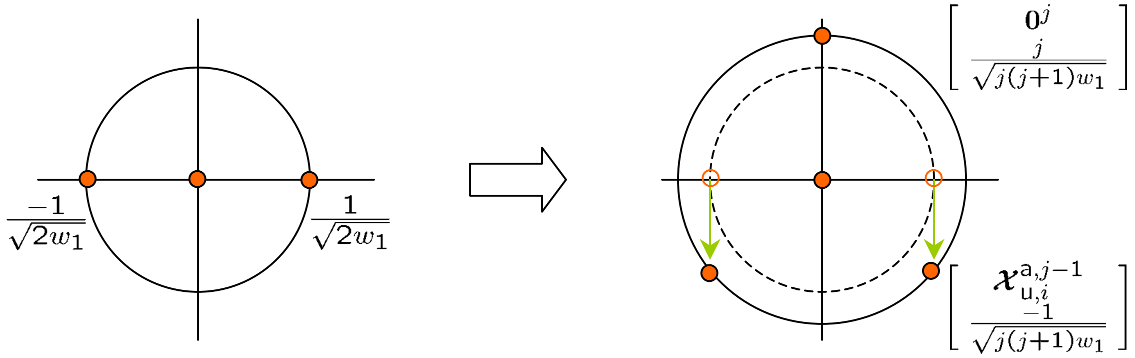
$\frac{-1}{\sqrt{2w_1}}$
$\frac{1}{\sqrt{2w_1}}$
$\begin{bmatrix} \mathbf{0}^j \\ \frac{j}{\sqrt{j(j+1)w_1}} \end{bmatrix}$
$\begin{bmatrix} \boldsymbol{\mathcal{X}}^{\mathrm{a},j-1}_{\mathrm{u},i} \\ \frac{-1}{\sqrt{j(j+1)w_1}} \end{bmatrix}$

Figure 4.2: The spherical simplex sigma point generation

If the same procedures are repeated, eventually Algorithm 1, generating $n+2$ SPs, can be obtained. For the three-dimensional case, the SPs will be contained in a sphere of radius $\sqrt{3/(1-w_0)}$. If the dimension is higher than three, the radius of the bounding hypersphere is $\sqrt{n/(1-w_0)}$.

Algorithm 1 (The Spherical Simplex UT)

1. Choose $0 \leq w_0 \leq 1$.

2. Compute weight sequence as:

$$w_i = (1-w_0)/(n+1) \text{ for } i = 1, \ldots, n+1.$$

3. Initialize the vector sequence of the sigma points as follows:

$$\boldsymbol{\mathcal{X}}^1_{\mathrm{u},0} = [0], \ \boldsymbol{\mathcal{X}}^1_{\mathrm{u},1} = \left[\frac{-1}{\sqrt{2w_1}}\right] \text{ and } \boldsymbol{\mathcal{X}}^1_{\mathrm{u},2} = \left[\frac{1}{\sqrt{2w_1}}\right]$$

4. Expand the vector sequence of the sigma points for $j = 2, \ldots, n$:

$$\boldsymbol{\mathcal{X}}_{\mathrm{u},i}^{j} = \begin{cases} \begin{bmatrix} \boldsymbol{\mathcal{X}}_{\mathrm{u},0}^{j-1} \\ \\ 0 \end{bmatrix} & \text{for } i = 0 \\ \begin{bmatrix} \boldsymbol{\mathcal{X}}_{\mathrm{u},i}^{j-1} \\ \\ -1/\sqrt{j(j+1)w_1} \end{bmatrix} & \text{for } i = 1, \ldots, j \\ \begin{bmatrix} \mathbf{0}^{j-1} \\ \\ j/\sqrt{j(j+1)w_1} \end{bmatrix} & \text{for } i = j+1 \end{cases}$$

where the superscript j denotes the dimension of the vector and the subscript i indicates the sequence of the sigma points.

The sigma points for an arbitrary mean and covariance can be obtained as follows:

$$\boldsymbol{\mathcal{X}}_i = \bar{\mathbf{x}} + \mathbf{S}\boldsymbol{\mathcal{X}}_{\mathrm{u},i}, \quad i = 0, \ldots, n+1. \tag{4.7}$$

4.1.2 The Scaled UT

Although the radius of the bounding hypersphere of the spherical simplex SPs is much smaller than that of the minimal skew simplex SPs, the radius expansion nevertheless results in a problem for high dimensional systems involving attitude estimation. For instance, if $n = 9$ and $w_0 = 0.5$, the radius becomes $\sqrt{n/(1-w_0)} \approx 4.24$; in this case, if the heading uncertainty is about 30°, then the heading can be sampled in the range of ±120°. Thus, nonlinearities outside of the uncertainty region can affect the solution. Furthermore, attitude angles do not belong to a vector space (Pennec, 1998) because they repeat themselves as the values of the angles grow. Hence, for large attitude uncertainties, some angles can be sampled twice. Therefore, the sigma points must be scaled using the method introduced in Julier and Uhlmann (2002b); this scaling scheme is reproduced in Algorithm 2.

Algorithm 2 (The Scaled UT)

1. Sigma points from the first proposal for the UT and the spherical simplex UT can be scaled by using Eqs. (4.8a) and (4.8b), respectively:

$$\boldsymbol{\mathcal{X}}'_i = \begin{cases} \bar{\mathbf{x}} & i = 0 \\ \bar{\mathbf{x}} + \alpha\sqrt{n+\kappa}\boldsymbol{\sigma}_i & i = 1, \cdots, n \\ \bar{\mathbf{x}} - \alpha\sqrt{n+\kappa}\boldsymbol{\sigma}_{i-n} & i = n+1, \cdots, 2n \end{cases}, \tag{4.8a}$$

$$\boldsymbol{\mathcal{X}}'_i = \bar{\mathbf{x}} + \alpha \mathbf{S} \boldsymbol{\mathcal{X}}_{\mathrm{u},i}, \quad i = 0, \ldots, n+1, \tag{4.8b}$$

where α is the scaling parameter, a small positive number (e.g., $10^{-4} \leq \alpha \leq 1$) (Wan and van der Merwe, 2001, p. 229).

2. Adjust the weights for the mean:

$$w_i^{\mathrm{m}} = \begin{cases} (w_0 - 1)/\alpha^2 + 1 & i = 0 \\ w_i/\alpha^2 & i \neq 0 \end{cases}. \tag{4.8c}$$

3. Adjust the weights for the covariance:

$$w_i^{\mathrm{c}} = \begin{cases} (w_0-1)/\alpha^2 + 2 + \beta - \alpha^2 & i = 0 \\ \\ w_i/\alpha^2 & i \neq 0 \end{cases}, \tag{4.8d}$$

   where β is a parameter to reduce higher order effects; $\beta = 2$ is optimal for Gaussian distributions (Julier and Uhlmann, 2002b) and will be discussed in Section 4.1.3.

Comparing the 0th weight in Eqs. (4.8c) and (4.8d) yields

$$w_0^{\mathrm{c}} = w_0^{\mathrm{m}} + 1 + \beta - \alpha^2. \tag{4.9}$$

If scaling is to be applied for the spherical simplex SPs, then the reciprocal of the radius of the bounding hypersphere can be used as the scaling parameter such that the SPs can be sampled within a range of ±1σ:

$$\alpha = \sqrt{(1-w_0)/n}. \tag{4.10}$$

The characteristics in Eq. (4.1) hold regardless of the choice of α; for instance, for the spherical simplex SPs,

$$\sum_{i=0}^{p-1} w_i^{\mathrm{m}} \boldsymbol{\mathcal{X}}_i' = \bar{\mathbf{x}} + \frac{1}{\alpha}\mathbf{S}\sum_{i=0}^{p-1} w_i \boldsymbol{\mathcal{X}}_{\mathrm{u},i} + \left(1 - \frac{1}{\alpha^2}\right)\alpha\mathbf{S}\boldsymbol{\mathcal{X}}_{\mathrm{u},0} = \bar{\mathbf{x}}, \tag{4.11a}$$

$$\begin{aligned} \sum_{i=0}^{p-1} w_i^{\mathrm{c}} \left(\boldsymbol{\mathcal{X}}_i' - \bar{\mathbf{x}}\right)\left(\boldsymbol{\mathcal{X}}_i' - \bar{\mathbf{x}}\right)^T &= \sum_{i=0}^{p-1} w_i^{\mathrm{c}} \alpha^2 \mathbf{S}\boldsymbol{\mathcal{X}}_{\mathrm{u},i}\left(\boldsymbol{\mathcal{X}}_{\mathrm{u},i}\right)^T \mathbf{S}^T \\ &= \sum_{i=0}^{p-1} w_i \mathbf{S}\boldsymbol{\mathcal{X}}_{\mathrm{u},i}\left(\boldsymbol{\mathcal{X}}_{\mathrm{u},i}\right)^T \mathbf{S}^T + \left(2 + \beta - \alpha^2 - \frac{1}{\alpha^2}\right)\alpha^2\mathbf{S}\boldsymbol{\mathcal{X}}_{\mathrm{u},0}\left(\boldsymbol{\mathcal{X}}_{\mathrm{u},0}\right)^T\mathbf{S}^T \\ &= \mathbf{P}, \end{aligned} \tag{4.11b}$$

where $\boldsymbol{\mathcal{X}}_{\mathrm{u},0} = \mathbf{0}$ is used.

4.1.3 Nonlinearities and the UT

In this section, the handling of nonlinearities by the UT will be investigated using a one-dimensional random variable based on Julier and Uhlmann (1996). Let x be a normally distributed random variable with mean and variance given as $\bar{x}$ and σ_x^2, respectively. Then, the true value, x_{T}, can be written as

$$x_{\mathrm{T}} = \bar{x} + \delta x, \tag{4.12}$$

where δx is an error and is assumed to be of zero mean: $E[\delta x] = 0$.

Let y be another random variable related to x with the following general nonlinear transformation:

$$y = g(x), \tag{4.13}$$

where $g(\cdot)$ is a continuous function. The transformed value can be written as follows:

$$\begin{aligned} y_{\mathrm{T}} &= g(\bar{x} + \delta x) \\ &= g(\bar{x}) + \nabla_g \delta x + \frac{\nabla_g^2 (\delta x)^2}{2!} + \frac{\nabla_g^3 (\delta x)^3}{3!} + \frac{\nabla_g^4 (\delta x)^4}{4!} + \cdots , \end{aligned} \tag{4.14}$$

where

$$\nabla_g^i = \left. \frac{\partial^i g(x)}{\partial x^i} \right|_{x=\bar{x}} . \tag{4.15}$$

Since the expected values of all odd-order terms are zero, the true transformed mean can be written as follows:

$$\bar{y}_{\mathrm{T}} = E[y_{\mathrm{T}}] = g(\bar{x}) + \frac{\nabla_g^2 \sigma_x^2}{2!} + \frac{\nabla_g^4 E[(\delta x)^4]}{4!} + \cdots . \tag{4.16}$$

In a Gaussian distribution, the fourth-order moment (kurtosis) is equal to three times the variance squared. Hence, substituting $E[(\delta x)^4] = 3\sigma_x^4$ into Eq. (4.16) yields

$$\bar{y}_{\mathrm{T}} = g(\bar{x}) + \frac{\nabla_g^2 \sigma_x^2}{2!} + \frac{3\nabla_g^4 \sigma_x^4}{4!} + \cdots . \tag{4.17a}$$

The error of the transformed variable can be written as

$$y_{\mathrm{T}} - \bar{y}_{\mathrm{T}} = \nabla_g \delta x + \frac{\nabla_g^2 (\delta x)^2}{2!} + \frac{\nabla_g^3 (\delta x)^3}{3!} + \frac{\nabla_g^4 (\delta x)^4}{4!} + \cdots - \frac{\nabla_g^2 \sigma_x^2}{2!} - \frac{3\nabla_g^4 \sigma_x^4}{4!} - \cdots .$$

Therefore, the transformed variance can be written as follows:

$$\begin{aligned}
(\sigma_y^2)_{\mathrm{T}} &= E[(y_{\mathrm{T}} - \bar{y}_{\mathrm{T}})^2] \\
&= \nabla_g \sigma_x^2 \nabla_g + \frac{\nabla_g E[(\delta x)^4] \nabla_g^3}{3!} + \frac{\nabla_g^2 E[(\delta x)^4] \nabla_g^2}{2! \cdot 2!} + \frac{\nabla_g^3 E[(\delta x)^4] \nabla_g}{3!} \\
&\quad - \frac{\nabla_g^2 \sigma_x^4 \nabla_g^2}{2! \cdot 2!} + \cdots \\
&= \nabla_g \sigma_x^2 \nabla_g + \frac{3\nabla_g \sigma_x^4 \nabla_g^3}{3!} + \frac{2\nabla_g^2 \sigma_x^4 \nabla_g^2}{2! \cdot 2!} + \frac{3\nabla_g^3 \sigma_x^4 \nabla_g}{3!} + \cdots .
\end{aligned} \tag{4.17b}$$

Linearization yields the following transformation:

$$\bar{y}_{\mathrm{LIN}} = g(\bar{x}), \tag{4.18a}$$

$$(\sigma_y^2)_{\mathrm{LIN}} = \nabla_g \sigma_x^2 \nabla_g, \tag{4.18b}$$

accurate up to the first and third order for the mean and covariance, respectively. As ∇_g is the same as the Jacobian in multidimensional systems, Eqs. (4.18a) and (4.18b) also explain the mean and covariance transformation in the EKF if there is no system precess noise.

Let $\{w_i, \mathcal{X}_i\}$ be a set of weights and SPs satisfying the conditions in Eq. (4.1). There exist three SPs for a one-dimensional system in both the first proposal for the UT, Eq. (4.2a), and the spherical simplex UT. The SPs can be interpreted as perturbed points around the mean and therefore can be written as

$$\mathcal{X}_i = \bar{x} + \sigma_i, \quad i = 0, 1, 2. \tag{4.19}$$

As the 0th point is equal to the mean, $\sigma_0 = 0$. For the first proposal,

$$\begin{aligned} \{w_0, w_1, w_2\} &= \left\{ \frac{\kappa}{1+\kappa}, \frac{1}{2(1+\kappa)}, \frac{1}{2(1+\kappa)} \right\} \text{ and} \\ \{\sigma_0, \sigma_1, \sigma_2\} &= \{0, \sqrt{1+\kappa}\sigma_x, -\sqrt{1+\kappa}\sigma_x\}. \end{aligned} \tag{4.20}$$

For the spherical simplex UT, w_0 can be chosen arbitrarily, $w_1 = w_2 = (1 - w_0)/2$ and

$$\{\sigma_0, \sigma_1, \sigma_2\} = \{0, -\sigma_x/\sqrt{2w_1}, \sigma_x/\sqrt{2w_1}\}. \tag{4.21}$$

It can easily be verified that all of these SPs satisfy

$$\sum_{i=0}^{2} w_i \sigma_i = 0 \text{ and } \sum_{i=0}^{2} w_i \sigma_i^2 = \sigma_x^2. \tag{4.22}$$

The transformed SPs can be written as

$$\mathcal{Y}_i = g(\bar{x} + \sigma_i) \tag{4.23}$$

$$= g(\bar{x}) + \nabla_g \sigma_i + \frac{\nabla_g^2 \sigma_i^2}{2!} + \frac{\nabla_g^3 \sigma_i^3}{3!} + \frac{\nabla_g^4 \sigma_i^4}{4!} + \cdots . \tag{4.24}$$

Hence, the transformed mean through the UT can be written as

$$\begin{aligned} \bar{y}_{\text{UT}} &= \sum_{i=0}^{2} w_i \mathcal{Y}_i \\ &= g(\bar{x}) + \frac{\nabla_g^2 \sigma_x^2}{2!} + \sum_{i=0}^{2} w_i \left[\frac{\nabla_g^3 \sigma_i^3}{3!} + \frac{\nabla_g^4 \sigma_i^4}{4!} + \cdots \right]. \end{aligned} \tag{4.25a}$$

For one-dimensional systems, the generated SPs are symmetric for both the first proposal and the spherical simplex UT because $\sigma_0 = 0$ and $\sigma_1 = -\sigma_2$. Furthermore, as $w_1 = w_2$, the third order term vanishes ($\sum_{i=0}^{2} w_i \sigma_i^3 = 0$) and therefore

$$\bar{y}_{\mathrm{UT}} = g(\bar{x}) + \frac{\nabla_g^2 \sigma_x^2}{2!} + \sum_{i=0}^{2} w_i \left[\frac{\nabla_g^4 \sigma_i^4}{4!} + \cdots \right]. \tag{4.25b}$$

Compared with the true mean, Eq. (4.17a), the transformed mean through the UT is accurate up to the third order for one-dimensional systems. Since

$$\begin{aligned} \mathcal{Y}_i - \bar{y}_{\mathrm{UT}} &= \nabla_g \sigma_i + \frac{\nabla_g^2 \sigma_i^2}{2!} + \frac{\nabla_g^3 \sigma_i^3}{3!} + \frac{\nabla_g^4 \sigma_i^4}{4!} + \cdots \\ &\quad - \frac{\nabla_g^2 \sigma_x^2}{2!} - \sum_{i=0}^{2} w_i \left[\frac{\nabla_g^3 \sigma_i^3}{3!} + \frac{\nabla_g^4 \sigma_i^4}{4!} + \cdots \right], \end{aligned}$$

the transformed variance can be written as

$$\begin{aligned} \left(\sigma_y^2\right)_{\mathrm{UT}} &= \sum_{i=0}^{2} w_i \left(\mathcal{Y}_i - \bar{y}_{\mathrm{UT}}\right)^2 \\ &= \nabla_g \sigma_x^2 \nabla_g + \sum_{i=0}^{2} w_i \left[\frac{\nabla_g \sigma_i^3 \nabla_g^2}{2!} + \frac{\nabla_g \sigma_i^4 \nabla_g^3}{3!} + \frac{\nabla_g^2 \sigma_i^3 \nabla_g}{2!} \right. \\ &\quad \left. + \frac{\nabla_g^2 \sigma_i^4 \nabla_g^2}{2! \cdot 2!} + \frac{\nabla_g^3 \sigma_i^4 \nabla_g}{3!} + \cdots \right] - \left[\frac{\nabla_g^2 \sigma_x^2}{2!} \right]^2. \end{aligned} \tag{4.25c}$$

Again, owing to the symmetry of the SPs, the third order term vanishes. Hence, comparing Eq. (4.25c) with Eq. (4.17b) shows that the transformed variance through the UT will be accurate up to the third order.

For multi-dimensional systems, however, as only the first proposal for the UT generates symmetric points, the third order term will not, in general, be canceled. Therefore, the transformed mean and covariance will be accurate up the second order. Table 4.1 summarizes the accuracy of mean and covariance transformation.

Table 4.1: Accuracy of mean and covariance transformation

	Mean	Covariance
Linearization	First order	Third order
First UT	Third order	Third order
Spherical Simplex UT	Second order	Second order

The scaled UT generates the following SPs:

$$\mathcal{X}_i' = \bar{x} + \alpha\sigma_i, \quad i = 0, 1, 2. \tag{4.26}$$

Based on the definitions of w_i^{m} and w_i^{c} in Section 4.1.2, the following characteristics hold:

$$\sum_{i=0}^{2} w_i^{\mathrm{m}} \alpha\sigma_i = \frac{1}{\alpha}\sum_{i=0}^{2} w_i\sigma_i + (1 - \frac{1}{\alpha^2})\alpha\sigma_0 = 0, \tag{4.27a}$$

$$\sum_{i=0}^{2} w_i^{\mathrm{m}} (\alpha\sigma_i)^2 = \sum_{i=0}^{2} w_i\sigma_i^2 + (\alpha^2 - 1)\sigma_0^2 = \sigma_x^2, \tag{4.27b}$$

$$\sum_{i=0}^{2} w_i^{\mathrm{c}} (\alpha\sigma_i)^2 = \sum_{i=0}^{2} w_i\sigma_i^2 + \left(2 + \beta - \alpha^2 - \frac{1}{\alpha^2}\right)\alpha^2\sigma_0^2 = \sigma_x^2, \tag{4.27c}$$

where $\sigma_0 = 0$ is used.

Transforming each of the scaled SPs yields

$$\begin{aligned} \mathcal{Y}_i' &= g(\bar{x} + \alpha\sigma_i) & (4.28) \\ &= g(\bar{x}) + \nabla_g \alpha\sigma_i + \frac{\nabla_g^2(\alpha\sigma_i)^2}{2!} + \frac{\nabla_g^3(\alpha\sigma_i)^3}{3!} + \frac{\nabla_g^4(\alpha\sigma_i)^4}{4!} + \cdots . & (4.29) \end{aligned}$$

Hence, the transformed mean through the scaled UT can be written as

$$\begin{aligned}\bar{y}_{\mathrm{SUT}} &= \sum_{i=0}^{2} w_i^{\mathrm{m}} \mathcal{Y}_i' \\ &= g(\bar{x}) + \frac{\nabla_g^2 \sigma_x^2}{2!} + \sum_{i=0}^{2} w_i^{\mathrm{m}} \left[\frac{\nabla_g^3 (\alpha\sigma_i)^3}{3!} + \frac{\nabla_g^4 (\alpha\sigma_i)^4}{4!} + \cdots \right]. \end{aligned} \tag{4.30}$$

The transformed mean is accurate up to the second order regardless of the choice of the value of α. With

$$\begin{aligned}\mathcal{Y}_i' - \bar{y}_{\mathrm{SUT}} &= \nabla_g \alpha\sigma_i + \frac{\nabla_g^2 (\alpha\sigma_i)^2}{2!} + \frac{\nabla_g^3 (\alpha\sigma_i)^3}{3!} + \frac{\nabla_g^4 (\alpha\sigma_i)^4}{4!} + \cdots \\ &\quad - \frac{\nabla_g^2 \sigma_x^2}{2!} - \sum_{i=0}^{2} w_i^{\mathrm{m}} \left[\frac{\nabla_g^3 (\alpha\sigma_i)^3}{3!} + \frac{\nabla_g^4 (\alpha\sigma_i)^4}{4!} + \cdots \right], \end{aligned} \tag{4.31}$$

the transformed variance can be written as

$$\begin{aligned}\left(\sigma_y^2\right)_{\mathrm{SUT}} &= \sum_{i=0}^{2} w_i^{\mathrm{c}} \left(\mathcal{Y}_i' - \bar{y}_{\mathrm{SUT}}\right)^2 \\ &= \nabla_g \sigma_x^2 \nabla_g + \sum_{i=0}^{2} w_i^{\mathrm{c}} \left[\frac{\nabla_g (\alpha\sigma_i)^3 \nabla_g^2}{2!} + \frac{\nabla_g (\alpha\sigma_i)^4 \nabla_g^3}{3!} + \frac{\nabla_g^2 (\alpha\sigma_i)^3 \nabla_g}{2!} \right. \\ &\quad \left. + \frac{\nabla_g^2 (\alpha\sigma_i)^4 \nabla_g^2}{2! \cdot 2!} + \frac{\nabla_g^3 (\alpha\sigma_i)^4 \nabla_g}{3!} + \cdots \right] - \left[\frac{\nabla_g^2 \sigma_x^2}{2!} \right]^2 . \end{aligned} \tag{4.32}$$

Hence, the transformed variance is also accurate up to the second order regardless of the choice of the value α. As the third order term vanishes, the transformed mean and variance are accurate up to the third order. For multi-dimensional systems, if the spherical simplex UT is applied, then it is accurate up to the second order.

Substituting $i = 0$ into Eq. (4.31) yields

$$\mathcal{Y}_0' - \bar{y}_{\mathrm{SUT}} = -\frac{\nabla_g^2 \sigma_x^2}{2!}. \tag{4.33}$$

Since $w_0^{\mathrm{c}} = w_0^{\mathrm{m}} + 1 + \beta - \alpha^2$, Eq. (4.32) can be rewritten as

$$\left(\sigma_y^2\right)_{\mathrm{SUT}} = \sum_{i=0}^{2} w_i^{\mathrm{m}} \left(\mathcal{Y}_i' - \bar{y}_{\mathrm{SUT}}\right)^2 + (1 + \beta - \alpha^2) \left[\frac{\nabla_g^2 \sigma_x^2}{2!}\right]^2 . \tag{4.34}$$

Thus, the additional parameter β can be used to adjust the effect of the fourth-order terms. Substituting

$$\sum_{i=0}^{2} w_i^{\mathrm{m}} \left(\mathcal{Y}_0' - \bar{y}_{\mathrm{SUT}}\right)^2 = \nabla_g \sigma_x^2 \nabla_g + \sum_{i=0}^{2} w_i^{\mathrm{m}} \left[\frac{\nabla_g (\alpha\sigma_i)^3 \nabla_g^2}{2!} + \frac{\nabla_g (\alpha\sigma_i)^4 \nabla_g^3}{3!} \right. \\ \left. + \frac{\nabla_g^2 (\alpha\sigma_i)^3 \nabla_g}{2!} + \frac{\nabla_g^2 (\alpha\sigma_i)^4 \nabla_g^2}{2! \cdot 2!} + \frac{\nabla_g^3 (\alpha\sigma_i)^4 \nabla_g}{3!} + \cdots \right] - \left[\frac{\nabla_g^2 \sigma_x^2}{2!}\right]^2 \tag{4.35}$$

into Eq. (4.34) yields

$$\left(\sigma_y^2\right)_{\mathrm{SUT}} = \nabla_g \sigma_x^2 \nabla_g + \sum_{i=0}^{2} w_i^{\mathrm{m}} \left[\frac{\nabla_g (\alpha\sigma_i)^3 \nabla_g^2}{2!} + \frac{\nabla_g (\alpha\sigma_i)^4 \nabla_g^3}{3!} + \frac{\nabla_g^2 (\alpha\sigma_i)^3 \nabla_g}{2!} \right. \\ \left. + \frac{\nabla_g^2 (\alpha\sigma_i)^4 \nabla_g^2}{2! \cdot 2!} + \frac{\nabla_g^3 (\alpha\sigma_i)^4 \nabla_g}{3!} + \cdots \right] + (\beta - \alpha^2) \left[\frac{\nabla_g^2 \sigma_x^2}{2!}\right]^2 . \tag{4.36}$$

Hence, subtracting the true variance, Eq. (4.17b), from the above equation yields the variance error:

$$\left(\delta\sigma_y^2\right)_{\mathrm{SUT}} = \sum_{i=0}^{2} w_i^{\mathrm{m}} \left[\frac{\nabla_g (\alpha\sigma_i)^3 \nabla_g^2}{2!} + \frac{\nabla_g (\alpha\sigma_i)^4 \nabla_g^3}{3!} + \frac{\nabla_g^2 (\alpha\sigma_i)^3 \nabla_g}{2!} + \frac{\nabla_g^2 (\alpha\sigma_i)^4 \nabla_g^2}{2! \cdot 2!} \right. \\ \left. + \frac{\nabla_g^3 (\alpha\sigma_i)^4 \nabla_g}{3!} + \cdots \right] - \frac{3\nabla_g \sigma_x^4 \nabla_g^3}{3!} - \frac{3\nabla_g^3 \sigma_x^4 \nabla_g}{3!} + (\beta - 2 - \alpha^2) \left[\frac{\nabla_g^2 \sigma_x^2}{2!}\right]^2 \tag{4.37}$$

Therefore, $\beta = 2 + \alpha^2$ will cancel the effect of the fourth-order term $(\nabla_g^2 \sigma_x^2/2!)^2$. However, other fourth-order terms also exist and the probability distribution needs to be investigated to find an optimal value of β.

Let us consider a simple example: $g(x) = x^2$. Substituting $\nabla_g = 2\bar{x}$, $\nabla_g^2 = 2$, and $\nabla_g^i = 0$ for $i > 2$ into Eqs. (4.17a) and (4.17b) yields the true values for the transformed mean and variance as follows:

$$\begin{aligned} \bar{y}_{\mathrm{T}} &= \bar{x}^2 + \sigma_x^2, && (4.38a) \\ \left(\sigma_y^2\right)_{\mathrm{T}} &= 4\bar{x}^2\sigma_x^2 + 2\sigma_x^4. && (4.38b) \end{aligned}$$

However, algorithms based on linearization transform the mean and variance as follows:

$$\begin{aligned} \bar{y}_{\mathrm{LIN}} &= \bar{x}^2, && (4.39a) \\ \left(\sigma_y^2\right)_{\mathrm{LIN}} &= 4\bar{x}^2\sigma_x^2. && (4.39b) \end{aligned}$$

Thus, the transformed mean is biased and the variance is distorted. Using Eqs. (4.25b) and (4.25c), we can write the transformed mean and variance through the UT as follows:

$$\begin{aligned} \bar{y}_{\mathrm{UT}} &= \bar{x}^2 + \sigma_x^2, && (4.40a) \\ \left(\sigma_y^2\right)_{\mathrm{UT}} &= 4\bar{x}^2\sigma_x^2 + w_1\sigma_1^4 + w_2\sigma_2^4 - \sigma_x^4. && (4.40b) \end{aligned}$$

The transformed mean is equal to the true value. If SPs from the first UT written in Eq. (4.20) are substituted into Eq. (4.40b), then

$$\left(\sigma_y^2\right)_{\mathrm{UT}} = 4\bar{x}^2\sigma_x^2 + \kappa\sigma_x^4. \tag{4.41}$$

Hence, if $\kappa = 2$ is used, the transformed variance is equal to the true variance. If the spherical simplex SPs written in Eq. (4.21) are substituted into Eq. (4.40b), then

$$\left(\sigma_y^2\right)_{\mathrm{UT}} = 4\bar{x}^2\sigma_x^2 + \left(\frac{1}{1-w_0} - 1\right)\sigma_x^4. \tag{4.42}$$

Therefore, if $w_0 = 2/3$, the transformed variance is equal to the true variance.

4.2 Generic UKF

The UKF is a straightforward extension of the UT to recursive estimation problems (Wan and van der Merwe, 2001). In the UKF, if the system model is a nonlinear function of the state and noise vector (e.g., $\mathbf{g}[\mathbf{x}, \mathbf{w}]$, an arbitrary nonlinear vector function of the state vector, $\mathbf{x}$, and the system noise vector, $\mathbf{w}$), then the system noise is generated and has to go through the system process model. Hence, the system state and noise vector are augmented as follows:

$$\mathbf{x}^{\mathrm{a}} = \begin{bmatrix} \mathbf{x} \\ \\ \mathbf{w} \end{bmatrix}, \tag{4.43}$$

where the superscript 'a' denotes the augmented states. By augmenting the state and noise vector, the effect of the noise on the covariance propagation can be described naturally. As the result, the integration in Eq. (3.45), required for the EKF, is not necessary. On the other hand, for a system model with an additive noise vector (e.g., $\mathbf{g}[\mathbf{x}] + \mathbf{w}$), there is no need to augment the state and noise vector at all.

In this section, the implementation of the UKF for system models with non-additive noise will be discussed first. Then, as a special case, that for system models with additive noise will be discussed. Throughout this section, it is assumed that the system and measurement noise are uncorrelated and the following nonlinear measurement model is given:

$$\mathbf{z}_k = \mathbf{h}\left[\mathbf{x}_k\right] + \mathbf{e}_k, \quad \mathbf{R}_k = E\left[\mathbf{e}_k \mathbf{e}_k^T\right]. \tag{4.44}$$

The spherical simplex SPs will be used in the description.

4.2.1 UKF for Systems with Non-Additive Noise

Assume that the following system model is given:

$$\mathbf{x}_{k+1} = \mathbf{g}\left[\mathbf{x}_k^{\mathrm{a}}\right] \equiv \mathbf{g}\left[\mathbf{x}_k, \mathbf{w}_k\right], \quad \mathbf{Q}_k = E\left[\mathbf{w}_k \mathbf{w}_k^T\right]. \tag{4.45}$$

Note that the left-hand side of Eq. (4.45) is not an augmented state vector as apposed to the right-hand side.

Initialization

1. Initialize the augmented state vector and its covariance:

$$\hat{\mathbf{x}}_{0|0}^{\mathrm{a}} = \begin{bmatrix} \hat{\mathbf{x}}_{0|0} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{P}_{0|0}^{\mathrm{a}} = \begin{bmatrix} \mathbf{P}_{0|0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_0 \end{bmatrix}, \tag{4.46}$$

   where $\mathbf{P}_{0|0} = E\left[(\mathbf{x}_{0|0} - \hat{\mathbf{x}}_{0|0})(\mathbf{x}_{0|0} - \hat{\mathbf{x}}_{0|0})^T\right]$ and $\mathbf{Q}_0 = E\left[\mathbf{w}_0 \mathbf{w}_0^T\right]$.

2. Generate the weights and sigma points for the zero mean and unit variance $\{w_i, \boldsymbol{\mathcal{X}}_{\mathrm{u},i}^{\mathrm{a}}\}$ using Algorithm 1. Then, compute the weights for the mean, w_i^{m}, and covariance, w_i^{c}, using Eqs. (4.8c) and (4.8d), respectively.

3. Apply Cholesky factorization to obtain the square root matrix $\mathbf{S}_{0|0}^{\mathrm{a}}$ such that $\mathbf{S}_{0|0}^{\mathrm{a}}\left(\mathbf{S}_{0|0}^{\mathrm{a}}\right)^T = \mathbf{P}_{0|0}^{\mathrm{a}}$. Then, the scaled SPs are computed as follows:

$$\boldsymbol{\mathcal{X}}_{i,0|0}^{\mathrm{a}\prime} = \hat{\mathbf{x}}_{0|0}^{\mathrm{a}} + \alpha \mathbf{S}_{0|0}^{\mathrm{a}} \boldsymbol{\mathcal{X}}_{\mathrm{u},i}^{\mathrm{a}}. \tag{4.47}$$

Prediction

1. Transform the SPs through the system process model:

$$\boldsymbol{\mathcal{X}}'_{i,k|k-1} = \mathbf{g}\left[\boldsymbol{\mathcal{X}}^{\mathrm{a}\prime}_{i,k-1|k-1}\right]. \tag{4.48a}$$

2. Compute the mean and covariance from the transformed SPs:

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{p-1} w_i^{\mathrm{m}} \boldsymbol{\mathcal{X}}'_{i,k|k-1}, \tag{4.48b}$$

$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{p-1} w_i^{\mathrm{c}} \left(\boldsymbol{\mathcal{X}}'_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right)\left(\boldsymbol{\mathcal{X}}'_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right)^T. \tag{4.48c}$$

3. Compose the augmented state vector and covariance:

$$\hat{\mathbf{x}}^{\mathrm{a}}_{k|k-1} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k-1} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{P}^{\mathrm{a}}_{k|k-1} = \begin{bmatrix} \mathbf{P}_{k|k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix}. \tag{4.48d}$$

4. Apply Cholesky factorization to obtain the square root matrix $\mathbf{S}^{\mathrm{a}}_{k|k-1}$ such that $\mathbf{S}^{\mathrm{a}}_{k|k-1}\left(\mathbf{S}^{\mathrm{a}}_{k|k-1}\right)^T = \mathbf{P}^{\mathrm{a}}_{k|k-1}$. Then, the scaled SPs are computed as follows:

$$\boldsymbol{\mathcal{X}}^{\mathrm{a}\prime}_{i,k|k-1} = \hat{\mathbf{x}}^{\mathrm{a}}_{k|k-1} + \alpha \mathbf{S}^{\mathrm{a}}_{k|k-1} \boldsymbol{\mathcal{X}}^{\mathrm{a}}_{\mathrm{u},i}. \tag{4.48e}$$

Measurement Update

1. Transform SPs through the measurement model:

$$\boldsymbol{\mathcal{Z}}_{i,k|k-1} = \mathbf{h}\left[\boldsymbol{\mathcal{X}}'_{i,k|k-1}\right]. \tag{4.49a}$$

2. Compute the predicted measurements from the transformed SPs:

$$\hat{\mathbf{z}}_{k|k-1} = \sum_{i=0}^{p-1} w_i^{\mathrm{m}} \boldsymbol{\mathcal{Z}}_{i,k|k-1}. \tag{4.49b}$$

3. Compute the covariance between the states and the measurements:

$$\mathbf{P}_{\mathbf{xz},k} = \sum_{i=0}^{p-1} w_i^{\mathrm{c}} \left(\boldsymbol{\mathcal{X}}'_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right) \left(\boldsymbol{\mathcal{Z}}_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1}\right)^T, \tag{4.49c}$$

which can be interpreted as $\mathbf{P}_{k|k-1}\mathbf{H}_k^T$ in the EKF described in the previous chapter.

4. Compute covariance of the innovation sequence:

$$\mathbf{P}_{\boldsymbol{\nu\nu},k} = \sum_{i=0}^{p-1} w_i^{\mathrm{c}} \left(\boldsymbol{\mathcal{Z}}_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1}\right) \left(\boldsymbol{\mathcal{Z}}_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1}\right)^T + \mathbf{R}_k, \tag{4.49d}$$

where the first term on the right-hand side is equivalent to $\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T$ in the EKF.

5. The rest of the update equations are written as follows:

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{xz},k}\mathbf{P}_{\boldsymbol{\nu\nu},k}^{-1}, \tag{4.49e}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}), \tag{4.49f}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{P}_{\boldsymbol{\nu\nu},k}\mathbf{K}_k^T, \tag{4.49g}$$

where $\mathbf{K}_k$ is the Kalman gain matrix.

6. Compose the augmented state vector and covariance:

$$\hat{\mathbf{x}}_{k|k}^{\mathrm{a}} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{P}_{k|k}^{\mathrm{a}} = \begin{bmatrix} \mathbf{P}_{k|k} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix}. \tag{4.49h}$$

7. Apply Cholesky factorization to obtain the square root matrix $\mathbf{S}_{k|k}^{\mathrm{a}}$ such that $\mathbf{S}_{k|k}^{\mathrm{a}}\left(\mathbf{S}_{k|k}^{\mathrm{a}}\right)^T = \mathbf{P}_{k|k}^{\mathrm{a}}$. Then, the scaled SPs are computed as follows:

$$\boldsymbol{\mathcal{X}}_{i,k|k}^{\mathrm{a}\prime} = \hat{\mathbf{x}}_{k|k}^{\mathrm{a}} + \alpha\mathbf{S}_{k|k}^{\mathrm{a}}\boldsymbol{\mathcal{X}}_{\mathrm{u},i}^{\mathrm{a}}. \tag{4.49i}$$

Equivalence of Eq. (4.49g) to the conventional Kalman covariance update can be shown as follows:

$$\begin{aligned}\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{P}_{\boldsymbol{\nu\nu},k} \mathbf{K}_k^T \\ &= \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{P}_{\boldsymbol{\nu\nu},k} \mathbf{P}_{\boldsymbol{\nu\nu},k}^{-1} \mathbf{H}_k \mathbf{P}_{k|k-1} \\ &= \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k\right) \mathbf{P}_{k|k-1}. \end{aligned} \tag{4.50}$$

4.2.2 UKF for Systems with Additive Noise

Assume that the following nonlinear system process model is given:

$$\mathbf{x}_{k+1} = \mathbf{g}\left[\mathbf{x}_k\right] + \mathbf{w}_k, \quad \mathbf{Q}_k = E\left[\mathbf{w}_k \mathbf{w}_k^T\right]. \tag{4.51}$$

Initialization

1. Initialize the state vector, $\hat{\mathbf{x}}_{0|0}$, and its covariance,

$$\mathbf{P}_{0|0} = E\left[(\mathbf{x}_{0|0} - \hat{\mathbf{x}}_{0|0})(\mathbf{x}_{0|0} - \hat{\mathbf{x}}_{0|0})^T\right].$$

2. Generate the weights and sigma points for the zero mean and unit variance $\{w_i, \boldsymbol{\mathcal{X}}_{\mathrm{u},i}\}$ using Algorithm 1. Then, the weights for the mean, w_i^{m}, and covariance, w_i^{c}, are computed using Eqs. (4.8c) and (4.8d), respectively.

3. Apply Cholesky factorization to obtain the square root matrix $\mathbf{S}_{0|0}$ such that $\mathbf{S}_{0|0}\mathbf{S}_{0|0}^T = \mathbf{P}_{0|0}$. Then, the scaled SPs are computed as follows:

$$\boldsymbol{\mathcal{X}}'_{i,0|0} = \hat{\mathbf{x}}_{0|0} + \alpha \mathbf{S}_{0|0} \boldsymbol{\mathcal{X}}_{\mathrm{u},i}. \tag{4.52}$$

Prediction

1. Transform the SPs through the system process model:

$$\boldsymbol{\mathcal{X}}'_{i,k|k-1} = \mathbf{g}\left[\boldsymbol{\mathcal{X}}'_{i,k-1|k-1}\right]. \tag{4.53a}$$

2. Compute the mean and covariance from the transformed SPs:

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{p-1} w_i^{\mathrm{m}} \boldsymbol{\mathcal{X}}'_{i,k|k-1}, \tag{4.53b}$$

$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{p-1} w_i^{\mathrm{c}} \left(\boldsymbol{\mathcal{X}}'_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right)\left(\boldsymbol{\mathcal{X}}'_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right)^T + \mathbf{Q}_{k-1} \tag{4.53c}$$

3. Apply Cholesky factorization to obtain the square root matrix $\mathbf{S}_{k|k-1}$ such that $\mathbf{S}_{k|k-1}\mathbf{S}^T_{k|k-1} = \mathbf{P}_{k|k-1}$. Then, the scaled SPs are computed as follows:

$$\boldsymbol{\mathcal{X}}'_{i,k|k-1} = \hat{\mathbf{x}}_{k|k-1} + \alpha \mathbf{S}_{k|k-1} \boldsymbol{\mathcal{X}}_{\mathrm{u},i}. \tag{4.53d}$$

Measurement Update

1. Transform SPs through the measurement model:

$$\boldsymbol{\mathcal{Z}}_{i,k|k-1} = \mathbf{h}\left[\boldsymbol{\mathcal{X}}'_{i,k|k-1}\right]. \tag{4.54a}$$

2. Compute the predicted measurements from the transformed SPs:

$$\hat{\mathbf{z}}_{k|k-1} = \sum_{i=0}^{p-1} w_i^{\mathrm{m}} \boldsymbol{\mathcal{Z}}_{i,k|k-1}. \tag{4.54b}$$

3. Compute the covariance between the states and the measurements:

$$\mathbf{P}_{\mathbf{xz},k} = \sum_{i=0}^{p-1} w_i^{\mathrm{c}} \left(\boldsymbol{\mathcal{X}}'_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right)\left(\boldsymbol{\mathcal{Z}}_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1}\right)^T. \tag{4.54c}$$

4. Compute covariance of the innovation sequence:

$$\mathbf{P}_{\boldsymbol{\nu\nu},k} = \sum_{i=0}^{p-1} w_i^{\mathrm{c}} \left(\boldsymbol{\mathcal{Z}}_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1}\right)\left(\boldsymbol{\mathcal{Z}}_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1}\right)^T + \mathbf{R}_k. \tag{4.54d}$$

5. The rest of the update equations are written as follows:

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{xz},k}\mathbf{P}_{\boldsymbol{\nu\nu},k}^{-1}, \tag{4.54e}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}), \tag{4.54f}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{P}_{\boldsymbol{\nu\nu},k}\mathbf{K}_k^T. \tag{4.54g}$$

6. Apply Cholesky factorization to obtain the square root matrix $\mathbf{S}_{k|k}$ such that $\mathbf{S}_{k|k}\mathbf{S}_{k|k}^T = \mathbf{P}_{k|k}$. Then, the scaled SPs are computed as follows:

$$\boldsymbol{\mathcal{X}}'_{i,k|k} = \hat{\mathbf{x}}_{k|k} + \alpha\mathbf{S}_{k|k}\boldsymbol{\mathcal{X}}_{\mathrm{u},i}. \tag{4.54h}$$

4.3 UKF for Aided INS

In this section, the UKF for aided INS will be developed. INS mechanization is used as the basic system model in aided INS; however, it will be modified in Section 4.3.1 to reflect the fact that system noise is generated in the UKF and has to go through the system process model. The nonlinear measurement functions discussed in Section 2.5 can be used directly in the measurement update stage; hence, there is no need to linearize measurement equations as in the EKF. Section 4.3.2 discusses the implementation of the UKF for aided INS in detail.

4.3.1 System Process Model for Aided INS

Similar to the EKF design in the previous chapter, biases, scale factors and nonorthogonalities are included in the the state vector of the UKF as listed in Table 4.2. Because the system process model should describe the effect of the sensor error terms on the navigation terms (position, velocity and attitude), it comprises both the INS mechanization and sensor error models. Note that the quaternion $\mathbf{q}_{\mathrm{b}}^{\mathrm{n}}$ is used in the state vector to express the attitude without singularity.

Sensor Output

Like in Section 2.3, angle and velocity increments are given as follows, respectively:

$$\Delta\tilde{\boldsymbol{\theta}}_k = \int_{t_{k-1}}^{t_k} \tilde{\boldsymbol{\omega}}_{\mathrm{ib}}^{\mathrm{b}} dt, \qquad (4.55a)$$

$$\Delta\tilde{\mathbf{v}}_{\mathbf{f},k}^{\mathrm{b}} = \int_{t_{k-1}}^{t_k} \tilde{\mathbf{f}}^{\mathrm{b}} dt, \qquad (4.55b)$$

where $\tilde{\cdot}$ denotes quantities corrupted by sensor errors.

Sensor Error Prediction

The discrete-time sensor error prediction can be applied as follows:

$$\mathbf{b}_{\mathrm{g},k} = \mathrm{diag}(\mathbf{d}_{\mathrm{gb}})\mathbf{b}_{\mathrm{g},k-1} + \mathbf{w}_{\mathrm{gb},k-1}, \qquad (4.56a)$$

$$\mathbf{b}_{\mathrm{a},k} = \mathrm{diag}(\mathbf{d}_{\mathrm{ab}})\mathbf{b}_{\mathrm{a},k-1} + \mathbf{w}_{\mathrm{ab},k-1}, \qquad (4.56b)$$

Table 4.2: Augmented state vector design

$\mathbf{x}$	$\mathbf{r}^{\mathrm{n}}$	3	Positions (φ, λ, h)
	$\mathbf{v}^{\mathrm{n}}$	3	Velocities $(v_{\mathrm{N}}, v_{\mathrm{E}}, v_{\mathrm{D}})$
	$\mathbf{q}_{\mathrm{b}}^{\mathrm{n}}$	4	Attitude quaternion
	$\mathbf{b}_{\mathrm{g}}$	3	Gyroscope biases
	$\mathbf{b}_{\mathrm{a}}$	3	Accelerometer biases
	$\mathbf{s}_{\mathrm{g}}$	3	Gyroscope scale factors
	$\mathbf{s}_{\mathrm{a}}$	3	Accelerometer scale factors
	$\boldsymbol{\gamma}_{\mathrm{g}}$	6	Gyro-triad nonorthogonalities
	$\boldsymbol{\gamma}_{\mathrm{a}}$	6	Accelerometer-triad nonorthogonalities
$\mathbf{w}$	$\mathbf{w}_{\mathrm{v}}$	3	Velocity noise
	$\mathbf{w}_{\phi}$	3	Attitude noise
	$\mathbf{w}_{\mathrm{gb}}$	3	Gyro bias noise
	$\mathbf{w}_{\mathrm{ab}}$	3	Accelerometer bias noise
	$\mathbf{w}_{\mathrm{gs}}$	3	Gyro scale factor noise
	$\mathbf{w}_{\mathrm{as}}$	3	Accelerometer scale factor noise
	$\mathbf{w}_{\mathrm{g}\gamma}$	6	Gyro non-orthogonality noise
	$\mathbf{w}_{\mathrm{a}\gamma}$	6	Accelerometer non-orthogonality noise

$$\mathbf{s}_{\mathrm{g},k} = \mathrm{diag}(\mathbf{d}_{\mathrm{gs}})\mathbf{s}_{\mathrm{g},k-1} + \mathbf{w}_{\mathrm{gs},k-1}, \tag{4.56c}$$

$$\mathbf{s}_{\mathrm{a},k} = \mathrm{diag}(\mathbf{d}_{\mathrm{as}})\mathbf{s}_{\mathrm{a},k-1} + \mathbf{w}_{\mathrm{as},k-1}, \tag{4.56d}$$

$$\boldsymbol{\gamma}_{\mathrm{g},k} = \mathrm{diag}(\mathbf{d}_{\mathrm{g}\gamma})\boldsymbol{\gamma}_{\mathrm{g},k-1} + \mathbf{w}_{\mathrm{g}\gamma,k-1}, \tag{4.56e}$$

$$\boldsymbol{\gamma}_{\mathrm{a},k} = \mathrm{diag}(\mathbf{d}_{\mathrm{a}\gamma})\boldsymbol{\gamma}_{\mathrm{a},k-1} + \mathbf{w}_{\mathrm{a}\gamma,k-1}, \tag{4.56f}$$

where $\mathbf{d}_{\mathrm{gb}}$, $\mathbf{d}_{\mathrm{ab}}$, $\mathbf{d}_{\mathrm{gs}}$, $\mathbf{d}_{\mathrm{as}}$, $\mathbf{d}_{\mathrm{g}\gamma}$, $\mathbf{d}_{\mathrm{a}\gamma}$, $\mathbf{w}_{\mathrm{gb},k-1}$, $\mathbf{w}_{\mathrm{ab},k-1}$, $\mathbf{w}_{\mathrm{gs},k-1}$, $\mathbf{w}_{\mathrm{as},k-1}$, $\mathbf{w}_{\mathrm{g}\gamma,k-1}$, and $\mathbf{w}_{\mathrm{a}\gamma,k-1}$ are discrete-time sensor error model parameters that can represent the random walk, the random constant and the first-order Gauss-Markov processes.

Error Compensation

The sensor measurement errors are compensated for using the following equations:

$$\begin{aligned}\Delta\boldsymbol{\theta}_k &= \left[\mathbf{I} + \Xi(\mathbf{s}_{\mathrm{g}}^0, \boldsymbol{\gamma}_{\mathrm{g}}^0) + \Xi(\mathbf{s}_{\mathrm{g},k}, \boldsymbol{\gamma}_{\mathrm{g},k})\right]^{-1} \left[\Delta\tilde{\boldsymbol{\theta}}_k - (\mathbf{b}_{\mathrm{g}}^0 + \mathbf{b}_{\mathrm{g},k})\Delta t_k - \mathbf{w}_{\phi,k-1}\right] \\ &\approx \left[\mathbf{I} - \Xi(\mathbf{s}_{\mathrm{g}}^0, \boldsymbol{\gamma}_{\mathrm{g}}^0) - \Xi(\mathbf{s}_{\mathrm{g},k}, \boldsymbol{\gamma}_{\mathrm{g},k})\right] \left[\Delta\tilde{\boldsymbol{\theta}}_k - (\mathbf{b}_{\mathrm{g}}^0 + \mathbf{b}_{\mathrm{g},k})\Delta t_k - \mathbf{w}_{\phi,k-1}\right],\end{aligned} \tag{4.57a}$$

$$\begin{aligned}\Delta\mathbf{v}_{\mathrm{f},k}^{\mathrm{b}} &= \left[\mathbf{I} + \Xi(\mathbf{s}_{\mathrm{a}}^0, \boldsymbol{\gamma}_{\mathrm{a}}^0) + \Xi(\mathbf{s}_{\mathrm{a},k}, \boldsymbol{\gamma}_{\mathrm{a},k})\right]^{-1} \left[\Delta\tilde{\mathbf{v}}_{\mathrm{f},k}^{\mathrm{b}} - (\mathbf{b}_{\mathrm{a}}^0 + \mathbf{b}_{\mathrm{a},k})\Delta t_k - \mathbf{w}_{\mathrm{v},k-1}\right] \\ &\approx \left[\mathbf{I} - \Xi(\mathbf{s}_{\mathrm{a}}^0, \boldsymbol{\gamma}_{\mathrm{a}}^0) - \Xi(\mathbf{s}_{\mathrm{a},k}, \boldsymbol{\gamma}_{\mathrm{a},k})\right] \left[\Delta\tilde{\mathbf{v}}_{\mathrm{f},k}^{\mathrm{b}} - (\mathbf{b}_{\mathrm{a}}^0 + \mathbf{b}_{\mathrm{a},k})\Delta t_k - \mathbf{w}_{\mathrm{v},k-1}\right],\end{aligned} \tag{4.57b}$$

where the superscript '0' represents known quantities and the operator $\Xi(\cdot,\cdot)$ was defined in Eq. (2.44). $\Delta\boldsymbol{\theta}_{k-1}$ and $\Delta\mathbf{v}_{\mathbf{f},k-1}^{\mathrm{b}}$ are computed and stored in the previous cycle.

Velocity Update

The time update of the velocity can be written as

$$\mathbf{v}_k^{\mathrm{n}} = \mathbf{v}_{k-1}^{\mathrm{n}} + \Delta\mathbf{v}_{\mathbf{f},k}^{\mathrm{n}} + \Delta\mathbf{v}_{\mathrm{g/cor},k}^{n}. \tag{4.58a}$$

The velocity increment due to the specific force can be computed as follows:

$$\begin{aligned}
\Delta\mathbf{v}_{\mathbf{f},k}^{\mathrm{n}} &= \left[\mathbf{I} - (0.5\boldsymbol{\zeta}_k\times)\right]\mathbf{C}_{\mathrm{b}(k-1)}^{\mathrm{n}(k-1)}\Delta\mathbf{v}_{\mathbf{f},k}^{\mathrm{b}(k-1)}, && (4.58\mathrm{b})\\
\boldsymbol{\zeta}_k &= \left[\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}} + \boldsymbol{\omega}_{\mathrm{en}}^{\mathrm{n}}\right]_{k-1/2}\Delta t_k, && (4.58\mathrm{c})\\
\Delta\mathbf{v}_{\mathbf{f},k}^{\mathrm{b}(k-1)} &\approx \Delta\mathbf{v}_{\mathbf{f},k}^{\mathrm{b}} + \frac{1}{2}\Delta\boldsymbol{\theta}_k \times \Delta\mathbf{v}_{\mathbf{f},k}^{\mathrm{b}}\\
&\quad + \frac{1}{12}\left(\Delta\boldsymbol{\theta}_{k-1} \times \Delta\mathbf{v}_{\mathbf{f},k}^{\mathrm{b}} + \Delta\mathbf{v}_{\mathbf{f},k-1}^{\mathrm{b}} \times \Delta\boldsymbol{\theta}_k\right), && (4.58\mathrm{d})
\end{aligned}$$

where the DCM $\mathbf{C}_{\mathrm{b}(k-1)}^{\mathrm{n}(k-1)}$ is obtained from the corresponding quaternion $(\mathbf{q}_{\mathrm{b}}^{\mathrm{n}})_{k-1} \equiv \mathbf{q}_{\mathrm{b}(k-1)}^{\mathrm{n}(k-1)}$ and the subscript '$k-1/2$' denote quantities at the midway. The gravity and Coriolis correction term can be computed as follows:

$$\Delta\mathbf{v}_{\mathrm{g/cor},k}^{\mathrm{n}} = \left[\mathbf{g}^{\mathrm{n}} - (2\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}} + \boldsymbol{\omega}_{\mathrm{en}}^{\mathrm{n}}) \times \mathbf{v}^{\mathrm{n}}\right]_{k-1/2}\Delta t_k. \tag{4.58e}$$

Position Update

The quaternion $\mathbf{q}_{\mathrm{n}(k-1)}^{\mathrm{e}(k-1)}$ is computed from the geodetic coordinates (φ_{k-1} and λ_{k-1}). Then, it will be updated as follows:

$$\begin{aligned}
\mathbf{q}_{\mathrm{n}(k)}^{\mathrm{e}(k-1)} &= \mathbf{q}_{\mathrm{n}(k-1)}^{\mathrm{e}(k-1)} \star \mathbf{q}_{\mathrm{n}(k)}^{\mathrm{n}(k-1)}, && (4.59\mathrm{a})\\
\mathbf{q}_{\mathrm{n}(k)}^{\mathrm{e}(k)} &= \mathbf{q}_{\mathrm{e}(k-1)}^{\mathrm{e}(k)} \star \mathbf{q}_{\mathrm{n}(k)}^{\mathrm{e}(k-1)}, && (4.59\mathrm{b})
\end{aligned}$$

where

$$
\begin{aligned}
\mathbf{q}_{\mathrm{n}(k)}^{\mathrm{n}(k-1)} &= \begin{bmatrix} \cos\|0.5\boldsymbol{\zeta}_k\| \\ \dfrac{\sin\|0.5\boldsymbol{\zeta}_k\|}{\|0.5\boldsymbol{\zeta}_k\|}0.5\boldsymbol{\zeta}_k \end{bmatrix}, && (4.59\mathrm{c}) \\
\mathbf{q}_{\mathrm{e}(k-1)}^{\mathrm{e}(k)} &= \begin{bmatrix} \cos\|0.5\boldsymbol{\xi}_k\| \\ -\dfrac{\sin\|0.5\boldsymbol{\xi}_k\|}{\|0.5\boldsymbol{\xi}_k\|}0.5\boldsymbol{\xi}_k \end{bmatrix}, && (4.59\mathrm{d}) \\
\boldsymbol{\xi}_k &= \boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{e}}\Delta t_k. && (4.59\mathrm{e})
\end{aligned}
$$

The updated geodetic coordinates (φ_k and λ_k) are computed from the updated quaternion $\mathbf{q}_{\mathrm{n}(k)}^{\mathrm{e}(k)}$. The height can be updated separately as follows:

$$
h_k = h_{k-1} - v_{\mathrm{D},k-1/2}\Delta t_k, \qquad (4.59\mathrm{f})
$$

where $v_{\mathrm{D},k-1/2}$ is the downward velocity at the midway.

Attitude Update

The attitude quaternion can be updated as follows:

$$
\begin{aligned}
\mathbf{q}_{\mathrm{b}(k)}^{\mathrm{n}(k-1)} &= \mathbf{q}_{\mathrm{b}(k-1)}^{\mathrm{n}(k-1)} \star \mathbf{q}_{\mathrm{b}(k)}^{\mathrm{b}(k-1)}, && (4.60\mathrm{a}) \\
\mathbf{q}_{\mathrm{b}(k)}^{\mathrm{n}(k)} &= \mathbf{q}_{\mathrm{n}(k-1)}^{\mathrm{n}(k)} \star \mathbf{q}_{\mathrm{b}(k)}^{\mathrm{n}(k-1)}, && (4.60\mathrm{b})
\end{aligned}
$$

where $\mathbf{q}_{\mathrm{b}(k)}^{\mathrm{b}(k-1)}$ is a quaternion for the b-frame update:

$$
\begin{aligned}
\mathbf{q}_{\mathrm{b}(k)}^{\mathrm{b}(k-1)} &= \begin{bmatrix} \cos\|0.5\boldsymbol{\phi}_k\| \\ \dfrac{\sin\|0.5\boldsymbol{\phi}_k\|}{\|0.5\boldsymbol{\phi}_k\|}0.5\boldsymbol{\phi}_k \end{bmatrix}, && (4.60\mathrm{c}) \\
\boldsymbol{\phi}_k &\approx \Delta\boldsymbol{\theta}_k + \frac{1}{12}\Delta\boldsymbol{\theta}_{k-1} \times \Delta\boldsymbol{\theta}_k. && (4.60\mathrm{d})
\end{aligned}
$$

Finally, the updated quaternion can be normalized as follows:

$$\begin{aligned}\mathbf{q}_{\mathrm{b}(k)}^{\mathrm{n}(k)} &:= (1-e_{\mathrm{q}})\mathbf{q}_{\mathrm{b}(k)}^{\mathrm{n}(k)}, &(4.60\mathrm{e})\\ e_{\mathrm{q}} &= \frac{1}{2}\left[\left(\mathbf{q}_{\mathrm{b}(k)}^{\mathrm{n}(k)}\right)^T \mathbf{q}_{\mathrm{b}(k)}^{\mathrm{n}(k)} - 1\right], &(4.60\mathrm{f})\end{aligned}$$

The overall system process model will be referred to as

$$\mathbf{x}_k = \mathbf{m}_{\mathrm{f}}\left[t_k, \mathbf{x}_{k-1}^{\mathrm{a}}, \tilde{\mathbf{f}}^{\mathrm{b}}(t_k), \tilde{\boldsymbol{\omega}}_{\mathrm{ib}}^{\mathrm{b}}(t_k)\right]. \qquad (4.61)$$

Note that the left-hand side of the above equation is not the augmented state vector, as opposed to the right-hand side.

4.3.2 Implementation of the UKF for Aided INS

Because quaternions are included, the dimension of the state vector in Table 4.2 is 34×1 excluding the noise part. As there exists a normalization constraint, the number of degrees of freedom of a quaternion is three. Hence, the dimension of the covariance matrix for the states will be 33×33. Implementation of the UKF will be discussed considering this dimensional mismatch. The spherical simplex sigma points will be used throughout this section.

Initialization

Let $\delta\mathbf{x}$ denote the augmentation of the errors in the navigation states and sensor errors terms:

$$\delta\hat{\mathbf{x}} = \begin{bmatrix} (\delta\mathbf{r}^{\mathrm{n}})^T & (\delta\mathbf{v}^{\mathrm{n}})^T & \boldsymbol{\phi}^T & \delta\mathbf{b}_{\mathrm{g}}^T & \delta\mathbf{b}_{\mathrm{a}}^T & \delta\mathbf{s}_{\mathrm{g}}^T & \delta\mathbf{s}_{\mathrm{a}}^T & \delta\boldsymbol{\gamma}_{\mathrm{g}}^T & \delta\boldsymbol{\gamma}_{\mathrm{a}}^T \end{bmatrix}^T, \tag{4.62}$$

where $\delta\mathbf{r}^{\mathrm{n}} = \hat{\mathbf{r}}^{\mathrm{n}} - \mathbf{r}^{\mathrm{n}}$, $\delta\mathbf{v}^{\mathrm{n}} = \hat{\mathbf{v}}^{\mathrm{n}} - \mathbf{v}^{\mathrm{n}}$, $\boldsymbol{\phi}$ is the rotation vector corresponding to the quaternion $\hat{\mathbf{q}}_{\mathrm{b}}^{\mathrm{n}} \star (\mathbf{q}_{\mathrm{b}}^{\mathrm{n}})^{-1}$, $\delta\mathbf{b}_{\mathrm{g}} = \hat{\mathbf{b}}_{\mathrm{g}} - \mathbf{b}_{\mathrm{g}}$, $\delta\mathbf{b}_{\mathrm{a}} = \hat{\mathbf{b}}_{\mathrm{a}} - \mathbf{b}_{\mathrm{a}}$, $\delta\mathbf{s}_{\mathrm{g}} = \hat{\mathbf{s}}_{\mathrm{g}} - \mathbf{s}_{\mathrm{g}}$, $\delta\mathbf{s}_{\mathrm{a}} = \hat{\mathbf{s}}_{\mathrm{a}} - \mathbf{s}_{\mathrm{a}}$, $\delta\boldsymbol{\gamma}_{\mathrm{g}} = \hat{\boldsymbol{\gamma}}_{\mathrm{g}} - \boldsymbol{\gamma}_{\mathrm{g}}$, and $\delta\boldsymbol{\gamma}_{\mathrm{a}} = \hat{\boldsymbol{\gamma}}_{\mathrm{a}} - \boldsymbol{\gamma}_{\mathrm{a}}$. Then, the augmented state vector and its covariance matrix can be initialized as follows:

$$\hat{\mathbf{x}}_{0|0}^{\mathrm{a}} = \begin{bmatrix} \hat{\mathbf{x}}_{0|0} \\ \mathbf{0} \end{bmatrix} \text{ and } \mathbf{P}_{0|0}^{\mathrm{a}} = \begin{bmatrix} \mathbf{P}_{0|0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_0 \end{bmatrix}, \tag{4.63}$$

where $\mathbf{P}_{0|0} = E[\delta\hat{\mathbf{x}}_{0|0}\delta\hat{\mathbf{x}}_{0|0}^T]$ and $\mathbf{Q}_0 = E[\mathbf{w}_0\mathbf{w}_0^T]$. Algorithm 1 needs to be applied to generate sigma points for zero mean and unit covariance: $\{\boldsymbol{\mathcal{X}}_{\mathrm{u},i}^{\mathrm{a}}\}$. Then, the weights for the mean, w_i^{m}, and covariance, w_i^{c}, are computed using Eqs. (4.8c) and (4.8d), respectively. The square-root matrix $\mathbf{S}_{0|0}^{\mathrm{a}}$ can be obtained such that $\mathbf{S}_{0|0}^{\mathrm{a}}(\mathbf{S}_{0|0}^{\mathrm{a}})^T = \mathbf{p}_{0|0}^{\mathrm{a}}$.

Finally, the scaled sigma points can be computed as follows:

$$\begin{bmatrix} \Delta\mathbf{r}^{\mathrm{n}}_{i,0} \\ \Delta\mathbf{v}^{\mathrm{n}}_{i,0} \\ \boldsymbol{\phi}_{i,0} \\ \Delta(\mathbf{b}_{\mathrm{g}})_{i,0} \\ \Delta(\mathbf{b}_{\mathrm{a}})_{i,0} \\ \Delta(\mathbf{s}_{\mathrm{g}})_{i,0} \\ \Delta(\mathbf{s}_{\mathrm{a}})_{i,0} \\ \Delta(\boldsymbol{\gamma}_{\mathrm{g}})_{i,0} \\ \Delta(\boldsymbol{\gamma}_{\mathrm{a}})_{i,0} \\ \mathbf{w}_{i,0} \end{bmatrix} = \alpha\mathbf{S}^{\mathrm{a}}_{0|0}\boldsymbol{\mathcal{X}}^{\mathrm{a}}_{\mathrm{u},i} \text{ and } \boldsymbol{\mathcal{X}}^{\mathrm{a}\prime}_{i,0|0} = \begin{bmatrix} \hat{\mathbf{r}}^{\mathrm{n}}_{0|0} + \Delta\mathbf{r}^{\mathrm{n}}_{i,0} \\ \hat{\mathbf{v}}^{\mathrm{n}}_{0|0} + \Delta\mathbf{v}^{\mathrm{n}}_{i,0} \\ (\mathbf{q}_{\phi})_{i,0} \star (\hat{\mathbf{q}}^{\mathrm{n}}_{\mathrm{b}})_{0|0} \\ \hat{\mathbf{b}}_{\mathrm{g},0|0} + \Delta(\mathbf{b}_{\mathrm{g}})_{i,0} \\ \hat{\mathbf{b}}_{\mathrm{a},0|0} + \Delta(\mathbf{b}_{\mathrm{a}})_{i,0} \\ \hat{\mathbf{s}}_{\mathrm{g},0|0} + \Delta(\mathbf{s}_{\mathrm{g}})_{i,0} \\ \hat{\mathbf{s}}_{\mathrm{a},0|0} + \Delta(\mathbf{s}_{\mathrm{a}})_{i,0} \\ \hat{\boldsymbol{\gamma}}_{\mathrm{g},0|0} + \Delta(\boldsymbol{\gamma}_{\mathrm{g}})_{i,0} \\ \hat{\boldsymbol{\gamma}}_{\mathrm{a},0|0} + \Delta(\boldsymbol{\gamma}_{\mathrm{a}})_{i,0} \\ \mathbf{w}_{i,0} \end{bmatrix}, \tag{4.64}$$

where

$$\mathbf{w}_{i,k} = \begin{bmatrix} (\mathbf{w}^T_{\mathrm{v}})_{i,k} & (\mathbf{w}^T_{\phi})_{i,k} & (\mathbf{w}^T_{\mathrm{gb}})_{i,k} & (\mathbf{w}^T_{\mathrm{ab}})_{i,k} & (\mathbf{w}^T_{\mathrm{gs}})_{i,k} & (\mathbf{w}^T_{\mathrm{as}})_{i,k} & (\mathbf{w}^T_{\mathrm{g}\gamma})_{i,k} & (\mathbf{w}^T_{\mathrm{a}\gamma})_{i,k} \end{bmatrix}^T$$

and

$$(\mathbf{q}_{\phi})_{i,0} = \begin{bmatrix} \cos(0.5\|\boldsymbol{\phi}_{i,0}\|) \\ \dfrac{\sin(0.5\|\boldsymbol{\phi}_{i,0}\|)}{0.5\|\boldsymbol{\phi}_{i,0}\|}0.5\boldsymbol{\phi}_{i,0} \end{bmatrix}.$$

Prediction

In the prediction stage, each of the sigma points goes through the system process model

$$\boldsymbol{\mathcal{X}}'_{i,k|k-1} = \mathbf{m}_{\mathrm{f}}\left[t_k, \boldsymbol{\mathcal{X}}^{\mathrm{a}\prime}_{i,k-1|k-1}, \tilde{\mathbf{f}}^{\mathrm{b}}(t_k), \tilde{\boldsymbol{\omega}}^{\mathrm{b}}_{\mathrm{ib}}(t_k)\right]. \tag{4.65}$$

In the generic UKF, Eqs. (4.48b) and (4.48c) are used to compute the predicted mean and covariance from the transformed sigma points, respectively:

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{p-1} w_i^{\mathrm{m}} \boldsymbol{\mathcal{X}}'_{i,k|k-1}, \tag{4.66}$$

$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{p-1} w_i^{\mathrm{c}} \left(\boldsymbol{\mathcal{X}}'_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right)\left(\boldsymbol{\mathcal{X}}'_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right)^T. \tag{4.67}$$

However, special treatments are required in averaging positions and attitudes because they do not belong to a vector space. Therefore, these equations will be modified here.

The position in each of the transformed sigma points, $\hat{\mathbf{r}}^{\mathrm{n}}_{i,k|k-1}$, can be transformed to the e-frame and the averaging can be done in the e-frame as follows:

$$\hat{\mathbf{r}}^{\mathrm{e}}_{k|k-1} = \sum_{i=0}^{p-1} w_i^{\mathrm{m}} \hat{\mathbf{r}}^{\mathrm{e}}_{i,k|k-1}, \tag{4.68}$$

which can then be transformed to the geodetic coordinates using the algorithms introduced in Section 2.2.2.

Averaging the attitude quaternions is more complicated. For instance, direct averaging DCMs, quaternions and rotation vectors does not yield the same attitude. Further, the result of direct averaging quaternions is not even a quaternion. Hence, intrinsic characteristics of rotations need to be considered using Riemannian geometry (Pennec, 1998). The intrinsic gradient descent algorithm (Kraft, 2003), explained in Algorithm 3, uses the fact that quaternion algebra provides a unique definition of the distance between two attitudes (invariant Riemannian distance). Therefore, this algorithm will be used to compute the weighted mean attitude quaternion. The number of iterations required is usually very small (in most cases, one).

Algorithm 3 (Weighted Mean Quaternion)

Given quaternions $\mathbf{q}_i, i = 0, \ldots, p-1$, the weighted mean quaternion, $\bar{\mathbf{q}}$, can be computed as follows:

1. Choose any of the $\mathbf{q}_i$'s as the initial mean quaternion, $\bar{\mathbf{q}}$.
2. Calculate the attitude difference, $\mathbf{q}_{\phi,i} = \mathbf{q}_i \star \bar{\mathbf{q}}^{-1}$.
3. Convert $\mathbf{q}_{\phi,i}$ into the corresponding rotation vector, $\boldsymbol{\phi}_i$.
4. Calculate the weighted mean of the rotation vectors, $\bar{\boldsymbol{\phi}} = \sum_{i=0}^{p-1} w_i^{\mathrm{m}} \boldsymbol{\phi}_i$.
5. Convert $\bar{\boldsymbol{\phi}}$ into the corresponding quaternion, $\mathbf{q}_\phi$.
6. Update the mean quaternion, $\bar{\mathbf{q}} := \mathbf{q}_\phi \star \bar{\mathbf{q}}$.
7. Repeat Steps 2 to 6 until $\|\hat{\boldsymbol{\phi}}\|$ falls below a specified threshold.

Now computation of the predicted covariance will be considered. Let

$$\boldsymbol{\mathcal{X}}'_{i,k|k-1} = \begin{bmatrix} \hat{\mathbf{r}}^{\mathrm{n}}_{i,k|k-1} \\ \hat{\mathbf{v}}^{\mathrm{n}}_{i,k|k-1} \\ (\hat{\mathbf{q}}^{\mathrm{n}}_{\mathrm{b}})_{i,k|k-1} \\ (\hat{\mathbf{b}}_{\mathrm{g}})_{i,k|k-1} \\ (\hat{\mathbf{b}}_{\mathrm{a}})_{i,k|k-1} \\ (\hat{\mathbf{s}}_{\mathrm{g}})_{i,k|k-1} \\ (\hat{\mathbf{s}}_{\mathrm{a}})_{i,k|k-1} \\ (\hat{\boldsymbol{\gamma}}_{\mathrm{g}})_{i,k|k-1} \\ (\hat{\boldsymbol{\gamma}}_{\mathrm{a}})_{i,k|k-1} \end{bmatrix}, \quad \Delta\boldsymbol{\mathcal{X}}'_{i,k|k-1} = \begin{bmatrix} \hat{\mathbf{r}}^{\mathrm{n}}_{i,k|k-1} - \hat{\mathbf{r}}^{\mathrm{n}}_{k|k-1} \\ \hat{\mathbf{v}}^{\mathrm{n}}_{i,k|k-1} - \hat{\mathbf{v}}^{\mathrm{n}}_{k|k-1} \\ \hat{\boldsymbol{\phi}}_{i,k|k-1} \\ (\hat{\mathbf{b}}_{\mathrm{g}})_{i,k|k-1} - (\hat{\mathbf{b}}_{\mathrm{g}})_{k|k-1} \\ (\hat{\mathbf{b}}_{\mathrm{a}})_{i,k|k-1} - (\hat{\mathbf{b}}_{\mathrm{a}})_{k|k-1} \\ (\hat{\mathbf{s}}_{\mathrm{g}})_{i,k|k-1} - (\hat{\mathbf{s}}_{\mathrm{g}})_{k|k-1} \\ (\hat{\mathbf{s}}_{\mathrm{a}})_{i,k|k-1} - (\hat{\mathbf{s}}_{\mathrm{a}})_{k|k-1} \\ (\hat{\boldsymbol{\gamma}}_{\mathrm{g}})_{i,k|k-1} - (\hat{\boldsymbol{\gamma}}_{\mathrm{g}})_{k|k-1} \\ (\hat{\boldsymbol{\gamma}}_{\mathrm{a}})_{i,k|k-1} - (\hat{\boldsymbol{\gamma}}_{\mathrm{a}})_{k|k-1} \end{bmatrix}, \tag{4.69}$$

where $\hat{\boldsymbol{\phi}}_{i,k|k-1}$ is the rotation vector corresponding to the quaternion $(\hat{\mathbf{q}}^{\mathrm{n}}_{\mathrm{b}})_{i,k|k-1} \star (\hat{\mathbf{q}}^{\mathrm{n}}_{\mathrm{b}})^{-1}_{k|k-1}$. Then, Eq. (4.48c) also has to be modified as follows:

$$\mathbf{P}_{k|k-1} \;=\; \sum_{i=0}^{p-1} w^{\mathrm{c}}_i \left[\Delta\boldsymbol{\mathcal{X}}'_{i,k|k-1}\right] \left[\Delta\boldsymbol{\mathcal{X}}'_{i,k|k-1}\right]^T. \tag{4.70}$$

Once the predicted state and covariance are computed, the scaled sigma points need to be regenerated. First, the square-root matrix $\mathbf{S}^{\mathrm{a}}_{k|k-1}$ can be obtained such that $\mathbf{S}^{\mathrm{a}}_{k|k-1}(\mathbf{S}^{\mathrm{a}}_{k|k-1})^T = \mathbf{p}^{\mathrm{a}}_{k|k-1}$, where

$$\mathbf{p}^{\mathrm{a}}_{k|k-1} = \begin{bmatrix} \mathbf{P}_{k|k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix}. \tag{4.71}$$

Then,

$$\begin{bmatrix} \Delta\mathbf{r}^{\mathrm{n}}_{i,k} \\ \Delta\mathbf{v}^{\mathrm{n}}_{i,k} \\ \boldsymbol{\phi}_{i,k} \\ \Delta(\mathbf{b}_{\mathrm{g}})_{i,k} \\ \Delta(\mathbf{b}_{\mathrm{a}})_{i,k} \\ \Delta(\mathbf{s}_{\mathrm{g}})_{i,k} \\ \Delta(\mathbf{s}_{\mathrm{a}})_{i,k} \\ \Delta(\boldsymbol{\gamma}_{\mathrm{g}})_{i,k} \\ \Delta(\boldsymbol{\gamma}_{\mathrm{a}})_{i,k} \\ \mathbf{w}_{i,k} \end{bmatrix} = \alpha\mathbf{S}^{\mathrm{a}}_{k|k-1}\boldsymbol{\mathcal{X}}^{\mathrm{a}}_{\mathrm{u},i} \text{ and } \boldsymbol{\mathcal{X}}^{\mathrm{a}\prime}_{i,k|k-1} = \begin{bmatrix} \hat{\mathbf{r}}^{\mathrm{n}}_{k|k-1} + \Delta\mathbf{r}^{\mathrm{n}}_{i,k} \\ \hat{\mathbf{v}}^{\mathrm{n}}_{k|k-1} + \Delta\mathbf{v}^{\mathrm{n}}_{i,k} \\ (\mathbf{q}_{\phi})_{i,k} \star (\hat{\mathbf{q}}^{\mathrm{n}}_{\mathrm{b}})_{k|k-1} \\ (\hat{\mathbf{b}}_{\mathrm{g}})_{k|k-1} + \Delta(\mathbf{b}_{\mathrm{g}})_{i,k} \\ (\hat{\mathbf{b}}_{\mathrm{a}})_{k|k-1} + \Delta(\mathbf{b}_{\mathrm{a}})_{i,k} \\ (\hat{\mathbf{s}}_{\mathrm{g}})_{k|k-1} + \Delta(\mathbf{s}_{\mathrm{g}})_{i,k} \\ (\hat{\mathbf{s}}_{\mathrm{a}})_{k|k-1} + \Delta(\mathbf{s}_{\mathrm{a}})_{i,k} \\ (\hat{\boldsymbol{\gamma}}_{\mathrm{g}})_{k|k-1} + \Delta(\boldsymbol{\gamma}_{\mathrm{g}})_{i,k} \\ (\hat{\boldsymbol{\gamma}}_{\mathrm{a}})_{k|k-1} + \Delta(\boldsymbol{\gamma}_{\mathrm{a}})_{i,k} \\ \mathbf{w}_{i,k} \end{bmatrix}, \tag{4.72}$$

where

$$(\mathbf{q}_{\phi})_{i,k} = \begin{bmatrix} \cos(0.5\|\boldsymbol{\phi}_{i,k}\|) \\ \dfrac{\sin(0.5\|\boldsymbol{\phi}_{i,k}\|)}{0.5\|\boldsymbol{\phi}_{i,k}\|} 0.5\boldsymbol{\phi}_{i,k} \end{bmatrix}.$$

Measurement Update

During the update stage, the sigma points are transformed through the measurement model:

$$\boldsymbol{\mathcal{Z}}_{i,k|k-1} = \mathbf{h}\left[\boldsymbol{\mathcal{X}}'_{i,k|k-1}\right]. \tag{4.73}$$

The predicted measurement vector is computed as follows:

$$\hat{\mathbf{z}}_{k|k-1} = \sum_{i=0}^{p-1} w^{\mathrm{m}}_{i} \boldsymbol{\mathcal{Z}}_{i,k|k-1}. \tag{4.74}$$

Then, the covariance matrix for the innovation sequence can be computed as follows:

$$\begin{aligned} \mathbf{P}_{\nu\nu,k} &= \sum_{i=0}^{p-1} w_i^{\mathrm{c}} \left[\Delta\boldsymbol{\mathcal{Z}}_{i,k|k-1}\right] \left[\Delta\boldsymbol{\mathcal{Z}}_{i,k|k-1}\right]^T + \mathbf{R}_k, \qquad (4.75) \\ \Delta\boldsymbol{\mathcal{Z}}_{i,k|k-1} &= \boldsymbol{\mathcal{Z}}_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1}, \qquad (4.76) \end{aligned}$$

where $\mathbf{R}_k$ is the measurement noise covariance matrix. The covariance matrix between the states and measurements is computed as follows:

$$\mathbf{P}_{\mathrm{xz},k} = \sum_{i=0}^{p-1} w_i^{\mathrm{c}} \left[\Delta\boldsymbol{\mathcal{X}}'_{i,k|k-1}\right] \left[\Delta\boldsymbol{\mathcal{Z}}_{i,k|k-1}\right]^T . \qquad (4.77)$$

Then, the Kalman gain matrix is computed as

$$\mathbf{K}_k = \mathbf{P}_{\mathrm{xz},k}\mathbf{P}^{-1}_{\nu\nu,k}, \qquad (4.78)$$

and the states can be updated by

$$\begin{bmatrix} \delta\hat{\mathbf{r}}^{\mathrm{n}}_k \\ \delta\hat{\mathbf{v}}^{\mathrm{n}}_k \\ \hat{\boldsymbol{\phi}}_k \\ \delta(\hat{\mathbf{b}}_{\mathrm{g}})_k \\ \delta(\hat{\mathbf{b}}_{\mathrm{a}})_k \\ \delta(\hat{\mathbf{s}}_{\mathrm{g}})_k \\ \delta(\hat{\mathbf{s}}_{\mathrm{a}})_k \\ (\delta\hat{\boldsymbol{\gamma}}_{\mathrm{g}})_k \\ (\delta\hat{\boldsymbol{\gamma}}_{\mathrm{a}})_k \end{bmatrix} = \mathbf{K}_k \left(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}\right) \text{ and } \hat{\mathbf{x}}_{k|k} = \begin{bmatrix} \hat{\mathbf{r}}^{\mathrm{n}}_{k|k-1} + \delta\hat{\mathbf{r}}^{\mathrm{n}}_k \\ \hat{\mathbf{v}}^{\mathrm{n}}_{k|k-1} + \delta\hat{\mathbf{v}}^{\mathrm{n}}_k \\ (\hat{\mathbf{q}}_{\phi})_k \star \hat{\mathbf{q}}^{\mathrm{n}}_{\mathrm{b},k|k-1} \\ (\hat{\mathbf{b}}_{\mathrm{g}})_{k|k-1} + \delta(\hat{\mathbf{b}}_{\mathrm{g}})_k \\ (\hat{\mathbf{b}}_{\mathrm{a}})_{k|k-1} + \delta(\hat{\mathbf{b}}_{\mathrm{a}})_k \\ (\hat{\mathbf{s}}_{\mathrm{g}})_{k|k-1} + \delta(\hat{\mathbf{s}}_{\mathrm{g}})_k \\ (\hat{\mathbf{s}}_{\mathrm{a}})_{k|k-1} + \delta(\hat{\mathbf{s}}_{\mathrm{a}})_k \\ (\hat{\boldsymbol{\gamma}}_{\gamma})_{k|k-1} + (\delta\hat{\boldsymbol{\gamma}}_{\mathrm{g}})_k \\ (\hat{\boldsymbol{\gamma}}_{\gamma})_{k|k-1} + (\delta\hat{\boldsymbol{\gamma}}_{\mathrm{a}})_k \end{bmatrix}, \qquad (4.79)$$

where $\mathbf{z}_k$ is the measurement vector and

$$(\hat{\mathbf{q}}_\phi)_k = \begin{bmatrix} \cos(0.5\|\hat{\boldsymbol{\phi}}_k\|) \\ \dfrac{\sin(0.5\|\hat{\boldsymbol{\phi}}_k\|)}{0.5\|\hat{\boldsymbol{\phi}}_k\|} 0.5\hat{\boldsymbol{\phi}}_k \end{bmatrix}. \tag{4.80}$$

The covariance update is the same as Eqs. (4.49g):

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{P}_{\nu\nu,k} \mathbf{K}_k^T. \tag{4.81}$$

The scaled sigma points need to be regenerated after updating the states and covariance. The square-root matrix $\mathbf{S}^{\mathrm{a}}_{k|k}$ can be obtained such that $\mathbf{S}^{\mathrm{a}}_{k|k}(\mathbf{S}^{\mathrm{a}}_{k|k})^T = \mathbf{p}^{\mathrm{a}}_{k|k}$, where

$$\mathbf{p}^{\mathrm{a}}_{k|k} = \begin{bmatrix} \mathbf{P}_{k|k} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix}. \tag{4.82}$$

Then,

$$\begin{bmatrix} \Delta\mathbf{r}^{\mathrm{n}}_{i,k} \\ \Delta\mathbf{v}^{\mathrm{n}}_{i,k} \\ \boldsymbol{\phi}_{i,k} \\ \Delta(\mathbf{b}_{\mathrm{g}})_{i,k} \\ \Delta(\mathbf{b}_{\mathrm{a}})_{i,k} \\ \Delta(\mathbf{s}_{\mathrm{g}})_{i,k} \\ \Delta(\mathbf{s}_{\mathrm{a}})_{i,k} \\ \Delta(\boldsymbol{\gamma}_{\mathrm{g}})_{i,k} \\ \Delta(\boldsymbol{\gamma}_{\mathrm{a}})_{i,k} \\ \mathbf{w}_{i,k} \end{bmatrix} = \alpha \mathbf{S}^{\mathrm{a}}_{k|k} \boldsymbol{\mathcal{X}}^{\mathrm{a}}_{\mathrm{u},i} \text{ and } \boldsymbol{\mathcal{X}}^{\mathrm{a}\prime}_{i,k|k-1} = \begin{bmatrix} \hat{\mathbf{r}}^{\mathrm{n}}_{k|k} + \Delta\mathbf{r}^{\mathrm{n}}_{i,k} \\ \hat{\mathbf{v}}^{\mathrm{n}}_{k|k} + \Delta\mathbf{v}^{\mathrm{n}}_{i,k} \\ (\mathbf{q}_\phi)_{i,k} \star (\hat{\mathbf{q}}^{\mathrm{n}}_{\mathrm{b}})_{k|k} \\ (\hat{\mathbf{b}}_{\mathrm{g}})_{k|k} + \Delta(\mathbf{b}_{\mathrm{g}})_{i,k} \\ (\hat{\mathbf{b}}_{\mathrm{a}})_{k|k} + \Delta(\mathbf{b}_{\mathrm{a}})_{i,k} \\ (\hat{\mathbf{s}}_{\mathrm{g}})_{k|k} + \Delta(\mathbf{s}_{\mathrm{g}})_{i,k} \\ (\hat{\mathbf{s}}_{\mathrm{a}})_{k|k} + \Delta(\mathbf{s}_{\mathrm{a}})_{i,k} \\ (\hat{\boldsymbol{\gamma}}_{\mathrm{g}})_{k|k} + \Delta(\boldsymbol{\gamma}_{\mathrm{g}})_{i,k} \\ (\hat{\boldsymbol{\gamma}}_{\mathrm{a}})_{k|k} + \Delta(\boldsymbol{\gamma}_{\mathrm{a}})_{i,k} \\ \mathbf{w}_{i,k} \end{bmatrix}, \tag{4.83}$$

where

$$(\mathbf{q}_\phi)_{i,k} = \begin{bmatrix} \cos(0.5\|\boldsymbol{\phi}_{i,k}\|) \\ \dfrac{\sin(0.5\|\boldsymbol{\phi}_{i,k}\|)}{0.5\|\boldsymbol{\phi}_{i,k}\|} 0.5\boldsymbol{\phi}_{i,k} \end{bmatrix}.$$

4.3.3 GPS Position Measurement

As the position state is expressed in terms of the geodetic latitude and longitude, a special treatment is required to avoid composing an ill-conditioned covariance matrix during position updates. Let us define a vector measurement function as follows:

$$\mathbf{h}[\mathbf{x}] = \mathrm{diag}(\begin{bmatrix} 1 & 1 & -1 \end{bmatrix}) \left[\mathbf{C}_{\mathrm{e}}^{\mathrm{n}} \left(\mathbf{r}_{\mathrm{GPS}}^{\mathrm{e}} - \mathbf{r}_{\mathrm{INS}}^{\mathrm{e}}\right) - \mathbf{C}_{\mathrm{b}}^{\mathrm{n}} \boldsymbol{\ell}_{\mathrm{GPS}}^{\mathrm{b}}\right], \tag{4.84}$$

where $\mathbf{r}_{\mathrm{GPS}}^{\mathrm{e}}$ is the measured GPS position in the e-frame; $\mathbf{r}_{\mathrm{INS}}^{\mathrm{e}}$ is the INS position in the e-frame converted from the position state using Eq. (2.20) and $\boldsymbol{\ell}_{\mathrm{GPS}}^{\mathrm{b}}$ is the lever arm of the GPS antenna in the b-frame. Let $\{\boldsymbol{\mathcal{Y}}_{i,k|k-1}\}$ be the transformed sigma points through Eq. (4.84) at time t_k. Then, the innovation sequence can be obtained by directly averaging the transformed sigma points:

$$\boldsymbol{\nu}_k = \mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1} = \sum_{i=0}^{p} w_i^{\mathrm{m}} \boldsymbol{\mathcal{Y}}_{i,k|k-1}. \tag{4.85}$$

The difference between the transformed sigma points and the predicted measurement can be computed as follows:

$$\Delta\boldsymbol{\mathcal{Z}}_{i,k|k-1} = \boldsymbol{\nu}_k - \boldsymbol{\mathcal{Y}}_{i,k|k-1}. \tag{4.86}$$

4.4 The Unscented Kalman Smoother

In this section, an unscented Kalman smoother will be developed using the forward and backward filtering approach. A backward system process model will be developed in Section 4.4.1 for the backward UKF. Implementation of the backward UKF is discussed in Section 4.4.2. Combination of the forward and backward solution is done inside the backward UKF.

4.4.1 Backward System Process Model

The backward system process model is required for the backward prediction, where backward sensor error prediction models, sensor error compensation and the backward INS mechanization are to be applied successively.

Sensor Output

Angle and velocity increments are given as follows, respectively:

$$\begin{aligned}\Delta\tilde{\boldsymbol{\theta}}_{k-1} &= \int_{t_{k-2}}^{t_{k-1}} \tilde{\boldsymbol{\omega}}_{\mathrm{ib}}^{\mathrm{b}} dt, &(4.87\mathrm{a})\\ \Delta\tilde{\mathbf{v}}_{\mathbf{f},k-1}^{\mathrm{b}} &= \int_{t_{k-2}}^{t_{k-1}} \tilde{\mathbf{f}}^{\mathrm{b}} dt, &(4.87\mathrm{b})\end{aligned}$$

where $\tilde{\cdot}$ denotes quantities corrupted by sensor errors.

Backward Sensor Error Prediction

Backward sensor error prediction models can be described as follows:

$$\begin{aligned}
\mathbf{b}_{\mathrm{g},k-1} &= \mathrm{diag}(\mathbf{d}_{\mathrm{gb}}^{\mathrm{B}})\mathbf{b}_{\mathrm{g},k} + \mathbf{w}_{\mathrm{gb},k-1}, && (4.88\mathrm{a})\\
\mathbf{b}_{\mathrm{a},k-1} &= \mathrm{diag}(\mathbf{d}_{\mathrm{ab}}^{\mathrm{B}})\mathbf{b}_{\mathrm{a},k} + \mathbf{w}_{\mathrm{ab},k-1}, && (4.88\mathrm{b})\\
\mathbf{s}_{\mathrm{g},k-1} &= \mathrm{diag}(\mathbf{d}_{\mathrm{gs}}^{\mathrm{B}})\mathbf{s}_{\mathrm{g},k} + \mathbf{w}_{\mathrm{gs},k-1}, && (4.88\mathrm{c})\\
\mathbf{s}_{\mathrm{a},k-1} &= \mathrm{diag}(\mathbf{d}_{\mathrm{as}}^{\mathrm{B}})\mathbf{s}_{\mathrm{a},k} + \mathbf{w}_{\mathrm{as},k-1}, && (4.88\mathrm{d})\\
\boldsymbol{\gamma}_{\mathrm{g},k-1} &= \mathrm{diag}(\mathbf{d}_{\mathrm{g}\gamma}^{\mathrm{B}})\boldsymbol{\gamma}_{\mathrm{g},k} + \mathbf{w}_{\mathrm{g}\gamma,k-1}, && (4.88\mathrm{e})\\
\boldsymbol{\gamma}_{\mathrm{a},k-1} &= \mathrm{diag}(\mathbf{d}_{\mathrm{a}\gamma}^{\mathrm{B}})\boldsymbol{\gamma}_{\mathrm{a},k} + \mathbf{w}_{\mathrm{a}\gamma,k-1}, && (4.88\mathrm{f})
\end{aligned}$$

where $\mathbf{d}_{\mathrm{gb}}^{\mathrm{B}}$, $\mathbf{d}_{\mathrm{ab}}^{\mathrm{B}}$, $\mathbf{d}_{\mathrm{gs}}^{\mathrm{B}}$, $\mathbf{d}_{\mathrm{as}}^{\mathrm{B}}$, $\mathbf{d}_{\mathrm{g}\gamma}^{\mathrm{B}}$, $\mathbf{d}_{\mathrm{a}\gamma}^{\mathrm{B}}$, $\mathbf{w}_{\mathrm{gb},k}$, $\mathbf{w}_{\mathrm{ab},k-1}$, $\mathbf{w}_{\mathrm{gs},k-1}$, $\mathbf{w}_{\mathrm{as},k-1}$, $\mathbf{w}_{\mathrm{g}\gamma,k-1}$ and $\mathbf{w}_{\mathrm{a}\gamma,k-1}$ are the parameters of the discrete-time backward sensor error prediction models, which describe the random constant, random walk, and Gauss-Markov process.

Error Compensation

The sensor error compensation is the the same as in Eqs. (4.57a) and (4.57b):

$$\begin{aligned}
\Delta\boldsymbol{\theta}_{k-1} &\approx \left[\mathbf{I} - \Xi(\mathbf{s}_{\mathrm{g}}^{0}, \boldsymbol{\gamma}_{\mathrm{g}}^{0}) - \Xi(\mathbf{s}_{\mathrm{g},k-1}, \boldsymbol{\gamma}_{\mathrm{g},k-1})\right]\\
&\qquad \left[\Delta\tilde{\boldsymbol{\theta}}_{k-1} - (\mathbf{b}_{\mathrm{g}}^{0} + \mathbf{b}_{\mathrm{g},k-1})\Delta t_{k-1} - \mathbf{w}_{\phi,k-1}\right], && (4.89\mathrm{a})
\end{aligned}$$

$$\begin{aligned}
\Delta\mathbf{v}_{\mathrm{f},k-1}^{\mathrm{b}} &\approx \left[\mathbf{I} - \Xi(\mathbf{s}_{\mathrm{a}}^{0}, \boldsymbol{\gamma}_{\mathrm{a}}^{0}) - \Xi(\mathbf{s}_{\mathrm{a},k-1}, \boldsymbol{\gamma}_{\mathrm{a},k-1})\right]\\
&\qquad \left[\Delta\tilde{\mathbf{v}}_{\mathrm{f},k-1}^{\mathrm{b}} - (\mathbf{b}_{\mathrm{a}}^{0} + \mathbf{b}_{\mathrm{a},k-1})\Delta t_{k-1} - \mathbf{w}_{\mathrm{v},k-1}\right], && (4.89\mathrm{b})
\end{aligned}$$

where $\Delta t_{k-1} = t_{k-1} - t_{k-2}$. $\Delta\boldsymbol{\theta}_k$ and $\Delta\mathbf{v}^{\mathrm{b}}_{\mathbf{f},k}$ are computed and stored in the previous cycle.

The rest of the backward system process model will be the same as the backward INS mechanization discussed in Eqs. (2.65a) to (2.66e), which will be repeated here briefly for completeness.

Backward Attitude Update

First, the rotation vectors corresponding to the b-frame and n-frame attitude change are computed, respectively:

$$\boldsymbol{\phi}_k \approx \Delta\boldsymbol{\theta}_k + \frac{1}{12}\Delta\boldsymbol{\theta}_{k-1} \times \Delta\boldsymbol{\theta}_k \tag{4.90a}$$

$$\boldsymbol{\zeta}_k = [\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{ie}} + \boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{en}}]_{k-1/2}\,\Delta t_k, \tag{4.90b}$$

where $\Delta t_k = t_k - t_{k-1}$ and the subscript '$k - 1/2$' denote quantities at the midway. Then, the attitude quaternion at time t_{k-1} can be obtained as follows:

$$\mathbf{q}^{\mathrm{n}(k)}_{\mathrm{b}(k-1)} = \mathbf{q}^{\mathrm{n}(k)}_{\mathrm{b}(k)} \star \mathbf{q}^{\mathrm{b}(k)}_{\mathrm{b}(k-1)}, \tag{4.90c}$$

$$\mathbf{q}^{\mathrm{n}(k-1)}_{\mathrm{b}(k-1)} = \mathbf{q}^{\mathrm{n}(k-1)}_{\mathrm{n}(k)} \star \mathbf{q}^{\mathrm{n}(k)}_{\mathrm{b}(k-1)}, \tag{4.90d}$$

where

$$\mathbf{q}^{\mathrm{b}(k)}_{\mathrm{b}(k-1)} = \begin{bmatrix} \cos\|0.5\boldsymbol{\phi}_k\| \\ -\dfrac{\sin\|0.5\boldsymbol{\phi}_k\|}{\|0.5\boldsymbol{\phi}_k\|}0.5\boldsymbol{\phi}_k \end{bmatrix} \text{ and } \mathbf{q}^{\mathrm{n}(k-1)}_{\mathrm{n}(k)} = \begin{bmatrix} \cos\|0.5\boldsymbol{\zeta}_k\| \\ \dfrac{\sin\|0.5\boldsymbol{\zeta}_k\|}{\|0.5\boldsymbol{\zeta}_k\|}0.5\boldsymbol{\zeta}_k \end{bmatrix}.$$

Backward Velocity Update

The backward update of the velocity can be written as

$$\mathbf{v}^{\mathrm{n}}_{k-1} = \mathbf{v}^{\mathrm{n}}_{k} - \Delta\mathbf{v}^{\mathrm{n}}_{\mathbf{f},k} - \Delta\mathbf{v}^{n}_{\mathrm{g/cor},k}. \tag{4.91a}$$

The velocity increment due to the specific force can be computed as follows:

$$\Delta\mathbf{v}^{\mathrm{n}}_{\mathbf{f},k} = \left[\mathbf{I} - (0.5\boldsymbol{\zeta}_k\times)\right]\mathbf{C}^{\mathrm{n}(k-1)}_{\mathrm{b}(k-1)}\Delta\mathbf{v}^{\mathrm{b}(k-1)}_{\mathbf{f},k}, \tag{4.91b}$$

$$\begin{aligned}\Delta\mathbf{v}^{\mathrm{b}(k-1)}_{\mathbf{f},k} \approx{}& \Delta\mathbf{v}^{\mathrm{b}}_{\mathbf{f},k} + \frac{1}{2}\Delta\boldsymbol{\theta}_k \times \Delta\mathbf{v}^{\mathrm{b}}_{\mathbf{f},k} \\ &+\frac{1}{12}\left(\Delta\boldsymbol{\theta}_{k-1} \times \Delta\mathbf{v}^{\mathrm{b}}_{\mathbf{f},k} + \Delta\mathbf{v}^{\mathrm{b}}_{\mathbf{f},k-1} \times \Delta\boldsymbol{\theta}_k\right),\end{aligned} \tag{4.91c}$$

where the DCM $\mathbf{C}^{\mathrm{n}(k-1)}_{\mathrm{b}(k-1)}$ is obtained from the corresponding quaternion $\mathbf{q}^{\mathrm{n}(k-1)}_{\mathrm{b}(k-1)}$. The gravity and Coriolis correction term can be computed as follows:

$$\Delta\mathbf{v}^{\mathrm{n}}_{\mathrm{g/cor},k} = \left[\mathbf{g}^{\mathrm{n}} - (2\boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{ie}} + \boldsymbol{\omega}^{\mathrm{n}}_{\mathrm{en}}) \times \mathbf{v}^{\mathrm{n}}\right]_{k-1/2}\Delta t_k. \tag{4.91d}$$

Backward Position Update

The quaternion $\mathbf{q}^{\mathrm{e}(k)}_{\mathrm{n}(k)}$ is computed from the geodetic coordinates (φ_k and λ_k). Then, the backward position updates can be written as follows:

$$\mathbf{q}^{\mathrm{e}(k)}_{\mathrm{n}(k-1)} = \mathbf{q}^{\mathrm{e}(k)}_{\mathrm{n}(k)} \star \mathbf{q}^{\mathrm{n}(k)}_{\mathrm{n}(k-1)}, \tag{4.92a}$$

$$\mathbf{q}^{\mathrm{e}(k-1)}_{\mathrm{n}(k-1)} = \mathbf{q}^{\mathrm{e}(k-1)}_{\mathrm{e}(k)} \star \mathbf{q}^{\mathrm{e}(k)}_{\mathrm{n}(k-1)}, \tag{4.92b}$$

$$h_{k-1} = h_k + v_{\mathrm{D},k-1/2}\Delta t_k, \tag{4.92c}$$

where

$$\mathbf{q}_{\mathrm{n}(k-1)}^{\mathrm{n}(k)} = \begin{bmatrix} \cos\|0.5\boldsymbol{\zeta}_k\| \\ -\dfrac{\sin\|0.5\boldsymbol{\zeta}_k\|}{\|0.5\boldsymbol{\zeta}_k\|}0.5\boldsymbol{\zeta}_k \end{bmatrix} \text{ and } \mathbf{q}_{\mathrm{e}(k)}^{\mathrm{e}(k-1)} = \begin{bmatrix} \cos\|0.5\boldsymbol{\xi}_k\| \\ \dfrac{\sin\|0.5\boldsymbol{\xi}_k\|}{\|0.5\boldsymbol{\xi}_k\|}0.5\boldsymbol{\xi}_k \end{bmatrix}.$$

Finally, φ_{k-1} and λ_{k-1} can be obtained from the quaternion $\mathbf{q}_{\mathrm{n}(k-1)}^{\mathrm{e}(k-1)}$.

The overall backward system process model will be referred to as

$$\mathbf{x}_{k-1} = \mathbf{m}_{\mathrm{b}}\left[t_{k-1}, \mathbf{x}_k^{\mathrm{a}}, \tilde{\mathbf{f}}^{\mathrm{b}}(t_{k-1}), \tilde{\boldsymbol{\omega}}_{\mathrm{ib}}^{\mathrm{b}}(t_{k-1})\right]. \tag{4.93}$$

4.4.2 Backward UKF and Smoothing for Aided INS

Like the forward UKF, the backward UKF also consists of initialization, backward prediction and measurement update stages. Hence, there has to be an initialization period at the end of the dataset. The backward estimates will be denoted as

$$\hat{\mathbf{x}}_{\mathrm{B},k|j} = E\left[\mathbf{x}_k|\mathbf{z}_N, \mathbf{z}_{N-1}, \cdots, \mathbf{z}_j\right], \tag{4.94}$$

where N is the total number of measurements.

Backward Initialization

The backward filter can be initialized as follows:

$$\hat{\mathbf{x}}_{\mathrm{B},N|N}^{\mathrm{a}} = \begin{bmatrix} \hat{\mathbf{x}}_{\mathrm{B},N|N} \\ \mathbf{0} \end{bmatrix} \text{ and } \mathbf{P}_{\mathrm{B},N|N}^{\mathrm{a}} = \begin{bmatrix} \mathbf{P}_{\mathrm{B},N|N} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{N-1} \end{bmatrix}, \tag{4.95}$$

where $\mathbf{P}_{\mathrm{B},N|N} = E\left[\delta\mathbf{x}_{\mathrm{B},N|N}\delta\mathbf{x}_{\mathrm{B},N|N}^T\right]$ and $\mathbf{Q}_N = E\left[\mathbf{w}_{N-1}\mathbf{w}_{N-1}^T\right]$. $\delta\mathbf{x}$ is defined in Eq. (4.62). Algorithm 1 needs to be applied to generate sigma points for zero mean and unit covariance: $\{\boldsymbol{\mathcal{X}}^{\mathrm{a}}_{\mathrm{u},i}\}$. Then, the weights for the mean, w^{m}_i, and covariance, w^{c}_i, are computed using Eqs. (4.8c) and (4.8d), respectively. The square-root matrix $\mathbf{S}^{\mathrm{a}}_{\mathrm{B},N|N}$ is computed such that $\mathbf{S}^{\mathrm{a}}_{\mathrm{B},N|N}\left(\mathbf{S}^{\mathrm{a}}_{\mathrm{B},N|N}\right)^T = \mathbf{P}^{\mathrm{a}}_{\mathrm{B},N|N}$. Then, the scaled spherical simplex sigma points can be generated as follows:

$$\begin{bmatrix} \Delta\mathbf{r}^{\mathrm{n}}_{i,N} \\ \Delta\mathbf{v}^{\mathrm{n}}_{i,N} \\ \boldsymbol{\phi}_{i,N} \\ \Delta\left(\mathbf{b}_{\mathrm{g}}\right)_{i,N} \\ \Delta\left(\mathbf{b}_{\mathrm{a}}\right)_{i,N} \\ \Delta\left(\mathbf{s}_{\mathrm{g}}\right)_{i,N} \\ \Delta\left(\mathbf{s}_{\mathrm{a}}\right)_{i,N} \\ \Delta\left(\boldsymbol{\gamma}_{\mathrm{g}}\right)_{i,N} \\ \Delta\left(\boldsymbol{\gamma}_{\mathrm{a}}\right)_{i,N} \\ \mathbf{w}_{i,N-1} \end{bmatrix} = \alpha\mathbf{S}^{\mathrm{a}}_{\mathrm{B},N|N}\boldsymbol{\mathcal{X}}^{\mathrm{a}}_{\mathrm{u},i} \text{ and } \boldsymbol{\mathcal{X}}^{\mathrm{a}\prime}_{i,\mathrm{B},N|N} = \begin{bmatrix} \hat{\mathbf{r}}^{\mathrm{n}}_{\mathrm{B},N|N} + \Delta\mathbf{r}^{\mathrm{n}}_{i,N} \\ \hat{\mathbf{v}}^{\mathrm{n}}_{\mathrm{B},N|N} + \Delta\mathbf{v}^{\mathrm{n}}_{i,N} \\ \left(\mathbf{q}_{\phi}\right)_{i,N} \star (\hat{\mathbf{q}}^{\mathrm{n}}_{\mathrm{b}})_{\mathrm{B},N|N} \\ \left(\mathbf{b}_{\mathrm{g}}\right)_{\mathrm{B},N|N} + \Delta\left(\mathbf{b}_{\mathrm{g}}\right)_{i,N} \\ \left(\mathbf{b}_{\mathrm{a}}\right)_{\mathrm{B},N|N} + \Delta\left(\mathbf{b}_{\mathrm{a}}\right)_{i,N} \\ \left(\mathbf{s}_{\mathrm{g}}\right)_{\mathrm{B},N|N} + \Delta\left(\mathbf{s}_{\mathrm{g}}\right)_{i,N} \\ \left(\mathbf{s}_{\mathrm{a}}\right)_{\mathrm{B},N|N} + \Delta\left(\mathbf{s}_{\mathrm{a}}\right)_{i,N} \\ \left(\boldsymbol{\gamma}_{\mathrm{g}}\right)_{\mathrm{B},N|N} + \Delta\left(\boldsymbol{\gamma}_{\mathrm{g}}\right)_{i,N} \\ \left(\boldsymbol{\gamma}_{\mathrm{g}}\right)_{\mathrm{B},N|N} + \Delta\left(\boldsymbol{\gamma}_{\mathrm{a}}\right)_{i,N} \\ \mathbf{w}_{i,N-1} \end{bmatrix}, \quad (4.96)$$

where

$$\mathbf{w}_{i,k} = \begin{bmatrix} (\mathbf{w}^T_{\mathrm{v}})_{i,k} & (\mathbf{w}^T_{\phi})_{i,k} & (\mathbf{w}^T_{\mathrm{gb}})_{i,k} & (\mathbf{w}^T_{\mathrm{ab}})_{i,k} & (\mathbf{w}^T_{\mathrm{gs}})_{i,k} & (\mathbf{w}^T_{\mathrm{as}})_{i,k} & (\mathbf{w}^T_{\mathrm{g}\gamma})_{i,k} & (\mathbf{w}^T_{\mathrm{a}\gamma})_{i,k} \end{bmatrix}^T$$

and

$$\left(\mathbf{q}_{\phi}\right)_{i,N} = \begin{bmatrix} \cos(0.5\|\boldsymbol{\phi}_{i,N}\|) \\ \dfrac{\sin(0.5\|\boldsymbol{\phi}_{i,N}\|)}{0.5\|\boldsymbol{\phi}_{i,N}\|}0.5\boldsymbol{\phi}_{i,N} \end{bmatrix}.$$

Backward Prediction

In the backward prediction, each of the sigma points goes through the backward system process:

$$\boldsymbol{\mathcal{X}}'_{i,\mathrm{B},k|k+1} = \mathbf{m}_\mathrm{b}\left[t_k, \boldsymbol{\mathcal{X}}^{\mathrm{a}\prime}_{i,\mathrm{B},k+1|k+1}, \mathbf{f}^\mathrm{b}(t_k), \boldsymbol{\omega}^\mathrm{b}_\mathrm{ib}(t_k)\right]. \tag{4.97}$$

The predicted states $\hat{\mathbf{x}}_{\mathrm{B},k|k+1}$ and the covariance $\mathbf{P}_{\mathrm{B},k|k+1}$ can be constructed with the same procedures as in the forward UKF. The predicted position can be computed in the e-frame as follows:

$$\hat{\mathbf{r}}^\mathrm{e}_{\mathrm{B},k|k+1} = \sum_{i=0}^{p-1} w_i^\mathrm{m}\ \hat{\mathbf{r}}^\mathrm{e}_{i,\mathrm{B},k|k+1}, \tag{4.98}$$

where $\hat{\mathbf{r}}^\mathrm{e}_{i,\mathrm{B},k|k+1}$ is the position vector in the e-frame computed from the geodetic coordinates in the ith sigma point: $\hat{\mathbf{r}}^\mathrm{n}_{i,\mathrm{B},k|k+1}$. Then, the algorithms discussed in Section 2.2.2 can be used to obtain the geodetic coordinates of the predicted position: $\hat{\mathbf{r}}^\mathrm{n}_{i,\mathrm{B},k|k+1}$. Algorithm 1 is used to obtain the predicted attitude quaternion $(\hat{\mathbf{q}}^\mathrm{n}_\mathrm{b})_{\mathrm{B},k|k+1}$. For the other states,

$$\begin{bmatrix} \hat{\mathbf{v}}^\mathrm{n}_{\mathrm{B},k|k+1} \\ (\hat{\mathbf{b}}_\mathrm{g})_{\mathrm{B},k|k+1} \\ (\hat{\mathbf{b}}_\mathrm{a})_{\mathrm{B},k|k+1} \\ (\hat{\mathbf{s}}_\mathrm{g})_{\mathrm{B},k|k+1} \\ (\hat{\mathbf{s}}_\mathrm{a})_{\mathrm{B},k|k+1} \\ (\hat{\boldsymbol{\gamma}}_\mathrm{g})_{\mathrm{B},k|k+1} \\ (\hat{\boldsymbol{\gamma}}_\mathrm{a})_{\mathrm{B},k|k+1} \end{bmatrix} = \sum_{i=0}^{p-1} w_i^\mathrm{m} \begin{bmatrix} \hat{\mathbf{v}}^\mathrm{n}_{i,\mathrm{B},k|k+1} \\ (\hat{\mathbf{b}}_\mathrm{g})_{i,\mathrm{B},k|k+1} \\ (\hat{\mathbf{b}}_\mathrm{a})_{i,\mathrm{B},k|k+1} \\ (\hat{\mathbf{s}}_\mathrm{g})_{i,\mathrm{B},k|k+1} \\ (\hat{\mathbf{s}}_\mathrm{a})_{i,\mathrm{B},k|k+1} \\ (\hat{\boldsymbol{\gamma}}_\mathrm{g})_{i,\mathrm{B},k|k+1} \\ (\hat{\boldsymbol{\gamma}}_\mathrm{a})_{i,\mathrm{B},k|k+1} \end{bmatrix}. \tag{4.99}$$

Thus, the backward predicted state vector, $\hat{\mathbf{x}}_{\mathrm{B},k|k+1}$, is obtained. To obtain the predicted covariance, let us first compute

$$\Delta \boldsymbol{\mathcal{X}}'_{i,\mathrm{B},k|k+1} = \begin{bmatrix} (\hat{\mathbf{r}}_i^{\mathrm{n}})_{\mathrm{B},k|k+1} - \hat{\mathbf{r}}^{\mathrm{n}}_{\mathrm{B},k|k+1} \\ (\hat{\mathbf{v}}_i^{\mathrm{n}})_{\mathrm{B},k|k+1} - \hat{\mathbf{v}}^{\mathrm{n}}_{\mathrm{B},k|k+1} \\ \hat{\boldsymbol{\phi}}_{i,\mathrm{B},k|k+1} \\ (\hat{\mathbf{b}}_{\mathrm{g}})_{i,\mathrm{B},k|k+1} - (\hat{\mathbf{b}}_{\mathrm{g}})_{\mathrm{B},k|k+1} \\ (\hat{\mathbf{b}}_{\mathrm{a}})_{i,\mathrm{B},k|k+1} - (\hat{\mathbf{b}}_{\mathrm{a}})_{\mathrm{B},k|k+1} \\ (\hat{\mathbf{s}}_{\mathrm{g}})_{i,\mathrm{B},k|k+1} - (\hat{\mathbf{s}}_{\mathrm{g}})_{\mathrm{B},k|k+1} \\ (\hat{\mathbf{s}}_{\mathrm{a}})_{i,\mathrm{B},k|k+1} - (\hat{\mathbf{s}}_{\mathrm{a}})_{\mathrm{B},k|k+1} \\ (\hat{\boldsymbol{\gamma}}_{\mathrm{g}})_{i,\mathrm{B},k|k+1} - (\hat{\boldsymbol{\gamma}}_{\mathrm{g}})_{\mathrm{B},k|k+1} \\ (\hat{\boldsymbol{\gamma}}_{\mathrm{a}})_{i,\mathrm{B},k|k+1} - (\hat{\boldsymbol{\gamma}}_{\mathrm{a}})_{\mathrm{B},k|k+1} \end{bmatrix}, \tag{4.100}$$

where $\hat{\boldsymbol{\phi}}_{i,\mathrm{B},k|k+1}$ is the rotation vector corresponding to the quaternion $(\hat{\mathbf{q}}^{\mathrm{n}}_{\mathrm{b}})_{i,\mathrm{B},k|k+1} \star (\hat{\mathbf{q}}^{\mathrm{n}}_{\mathrm{b}})^{-1}_{\mathrm{B},k|k+1}$. The predicted covariance can be computed as follows:

$$\mathbf{P}_{\mathrm{B},k|k+1} = \sum_{i=0}^{p-1} w_i^{\mathrm{c}} \left[\Delta \boldsymbol{\mathcal{X}}'_{i,\mathrm{B},k|k+1}\right] \left[\Delta \boldsymbol{\mathcal{X}}'_{i,\mathrm{B},k|k+1}\right]^T. \tag{4.101}$$

The sigma points need to be regenerated after obtaining predicted states and covariance. The square-root matrix $\mathbf{S}^{\mathrm{a}}_{\mathrm{B},k|k+1}$ is computed such that $\mathbf{S}^{\mathrm{a}}_{\mathrm{B},k|k+1} \left(\mathbf{S}^{\mathrm{a}}_{\mathrm{B},k|k+1}\right)^T = \mathbf{P}^{\mathrm{a}}_{\mathrm{B},k|k+1}$, where

$$\mathbf{P}^{\mathrm{a}}_{\mathrm{B},k|k+1} = \begin{bmatrix} \mathbf{P}_{\mathrm{B},k|k+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{k-1} \end{bmatrix}. \tag{4.102}$$

Then, the scaled spherical simplex sigma points can be generated as follows:

$$\begin{bmatrix} \Delta\mathbf{r}^{\mathrm{n}}_{i,k} \\ \Delta\mathbf{v}^{\mathrm{n}}_{i,k} \\ \boldsymbol{\phi}_{i,k} \\ \Delta(\mathbf{b}_{\mathrm{g}})_{i,k} \\ \Delta(\mathbf{b}_{\mathrm{a}})_{i,k} \\ \Delta(\mathbf{s}_{\mathrm{g}})_{i,k} \\ \Delta(\mathbf{s}_{\mathrm{a}})_{i,k} \\ \Delta(\boldsymbol{\gamma}_{\mathrm{g}})_{i,k} \\ \Delta(\boldsymbol{\gamma}_{\mathrm{a}})_{i,k} \\ \mathbf{w}_{i,k-1} \end{bmatrix} = \alpha\mathbf{S}^{\mathrm{a}}_{\mathrm{B},k|k+1}\boldsymbol{\mathcal{X}}^{\mathrm{a}}_{\mathrm{u},i} \text{ and } \boldsymbol{\mathcal{X}}^{\mathrm{a}\prime}_{i,\mathrm{B},k|k+1} = \begin{bmatrix} \hat{\mathbf{r}}^{\mathrm{n}}_{\mathrm{B},k|k+1} + \Delta\mathbf{r}^{\mathrm{n}}_{i,k} \\ \hat{\mathbf{v}}^{\mathrm{n}}_{\mathrm{B},k|k+1} + \Delta\mathbf{v}^{\mathrm{n}}_{i,k} \\ (\mathbf{q}_{\phi})_{i,k} \star (\hat{\mathbf{q}}^{\mathrm{n}}_{\mathrm{b}})_{\mathrm{B},k|k+1} \\ (\hat{\mathbf{b}}_{\mathrm{g}})_{\mathrm{B},k|k+1} + \Delta(\mathbf{b}_{\mathrm{g}})_{i,k} \\ (\hat{\mathbf{b}}_{\mathrm{a}})_{\mathrm{B},k|k+1} + \Delta(\mathbf{b}_{\mathrm{a}})_{i,k} \\ (\hat{\mathbf{s}}_{\mathrm{g}})_{\mathrm{B},k|k+1} + \Delta(\mathbf{s}_{\mathrm{g}})_{i,k} \\ (\hat{\mathbf{s}}_{\mathrm{a}})_{\mathrm{B},k|k+1} + \Delta(\mathbf{s}_{\mathrm{a}})_{i,k} \\ (\hat{\boldsymbol{\gamma}}_{\mathrm{g}})_{\mathrm{B},k|k+1} + \Delta(\boldsymbol{\gamma}_{\mathrm{g}})_{i,k} \\ (\hat{\boldsymbol{\gamma}}_{\mathrm{a}})_{\mathrm{B},k|k+1} + \Delta(\boldsymbol{\gamma}_{\mathrm{a}})_{i,k} \\ \mathbf{w}_{i,k-1} \end{bmatrix}, \quad (4.103)$$

where

$$(\mathbf{q}_{\phi})_{i,k} = \begin{bmatrix} \cos(0.5\|\boldsymbol{\phi}_{i,k}\|) \\ \dfrac{\sin(0.5\|\boldsymbol{\phi}_{i,k}\|)}{0.5\|\boldsymbol{\phi}_{i,k}\|}0.5\boldsymbol{\phi}_{i,k} \end{bmatrix}.$$

Computation of Smoothed Solution

Given the forward solutions ($\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$) and the backward solutions ($\hat{\mathbf{x}}_{\mathrm{B},k|k+1}$ and $\mathbf{P}_{\mathrm{B},k|k+1}$), the smoothed solutions can be constructed as follows. First, we compute the smoothed covariance:

$$\mathbf{P}_{k|N} = \left(\mathbf{P}^{-1}_{k|k} + \mathbf{P}^{-1}_{\mathrm{B},k|k+1}\right)^{-1}. \quad (4.104)$$

Then, the difference between the two estimates is computed:

$$\Delta \mathbf{x}_k = \begin{bmatrix} \Delta \mathbf{r}^{\mathrm{n}}_k \\ \Delta \mathbf{v}^{\mathrm{n}}_k \\ \boldsymbol{\phi}_k \\ \Delta \mathbf{b}_{\mathrm{g},k} \\ \Delta \mathbf{b}_{\mathrm{a},k} \\ \Delta \mathbf{s}_{\mathrm{g},k} \\ \Delta \mathbf{s}_{\mathrm{a},k} \\ \Delta \boldsymbol{\gamma}_{\mathrm{g},k} \\ \Delta \boldsymbol{\gamma}_{\mathrm{a},k} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{r}}^{\mathrm{n}}_{\mathrm{B},k|k+1} - \hat{\mathbf{r}}^{\mathrm{n}}_{k|k} \\ \hat{\mathbf{v}}^{\mathrm{n}}_{\mathrm{B},k|k+1} - \hat{\mathbf{v}}^{\mathrm{n}}_{k|k} \\ \boldsymbol{\phi}_k \\ (\hat{\mathbf{b}}_{\mathrm{g}})_{\mathrm{B},k|k+1} - (\hat{\mathbf{b}}_{\mathrm{g}})_{k|k} \\ (\hat{\mathbf{b}}_{\mathrm{a}})_{\mathrm{B},k|k+1} - (\hat{\mathbf{b}}_{\mathrm{a}})_{k|k} \\ (\hat{\mathbf{s}}_{\mathrm{g}})_{\mathrm{B},k|k+1} - (\hat{\mathbf{s}}_{\mathrm{g}})_{k|k} \\ (\hat{\mathbf{s}}_{\mathrm{a}})_{\mathrm{B},k|k+1} - (\hat{\mathbf{s}}_{\mathrm{a}})_{k|k} \\ (\hat{\boldsymbol{\gamma}}_{\mathrm{g}})_{\mathrm{B},k|k+1} - (\hat{\boldsymbol{\gamma}}_{\mathrm{g}})_{k|k} \\ (\hat{\boldsymbol{\gamma}}_{\mathrm{a}})_{\mathrm{B},k|k+1} - (\hat{\boldsymbol{\gamma}}_{\mathrm{a}})_{k|k} \end{bmatrix}, \tag{4.105}$$

where $\boldsymbol{\phi}_k$ is the rotation vector corresponding to the quaternion $(\hat{\mathbf{q}}^{\mathrm{n}}_{\mathrm{b}})_{\mathrm{B},k|k+1} \star (\hat{\mathbf{q}}^{\mathrm{n}}_{\mathrm{b}})^{-1}_{k|k}$. While the updated solution was used from the forward solution, the predicted solution is used from the backward solution to maintain the independence between the two solutions in the combination as shown in Figure 4.3. The white and red circles represent the backward predicted and updated solutions, respectively.

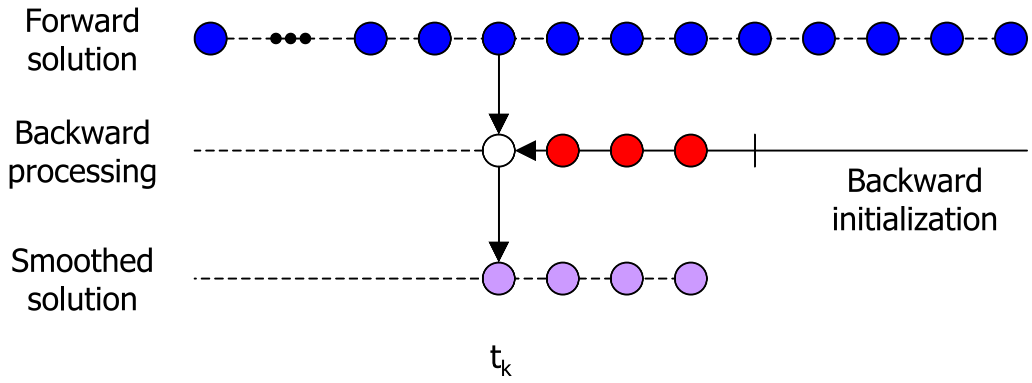

Figure 4.3: Combination of the forward and backward solutions

Finally, the smoothed estimates can be computed as follows:

$$\begin{bmatrix} \Delta\hat{\mathbf{r}}_k^{\mathrm{n}} \\ \Delta\hat{\mathbf{v}}_k^{\mathrm{n}} \\ \hat{\boldsymbol{\phi}}_k \\ \Delta\hat{\mathbf{b}}_{\mathrm{g},k} \\ \Delta\hat{\mathbf{b}}_{\mathrm{a},k} \\ \Delta\hat{\mathbf{s}}_{\mathrm{g},k} \\ \Delta\hat{\mathbf{s}}_{\mathrm{a},k} \\ \Delta\hat{\boldsymbol{\gamma}}_{\mathrm{g},k} \\ \Delta\hat{\boldsymbol{\gamma}}_{\mathrm{a},k} \end{bmatrix} = \mathbf{P}_{k|N}\mathbf{P}_{\mathrm{B},k|k+1}^{-1}\Delta\mathbf{x}_k \text{ and } \hat{\mathbf{x}}_{k|N} = \begin{bmatrix} \hat{\mathbf{r}}_{k|k}^{\mathrm{n}} + \Delta\hat{\mathbf{r}}_k^{\mathrm{n}} \\ \hat{\mathbf{v}}_{k|k}^{\mathrm{n}} + \Delta\hat{\mathbf{v}}_k^{\mathrm{n}} \\ (\hat{\mathbf{q}}_\phi)_k \star (\hat{\mathbf{q}}_{\mathrm{b}}^{\mathrm{n}})_{k|k} \\ \left(\hat{\mathbf{b}}_{\mathrm{g}}\right)_{k|k} + \Delta\hat{\mathbf{b}}_{\mathrm{g},k} \\ \left(\hat{\mathbf{b}}_{\mathrm{a}}\right)_{k|k} + \Delta\hat{\mathbf{b}}_{\mathrm{a},k} \\ (\hat{\mathbf{s}}_{\mathrm{g}})_{k|k} + \Delta\hat{\mathbf{s}}_{\mathrm{g},k} \\ (\hat{\mathbf{s}}_{\mathrm{a}})_{k|k} + \Delta\hat{\mathbf{s}}_{\mathrm{a},k} \\ \left(\hat{\boldsymbol{\gamma}}_{\mathrm{g}}\right)_{k|k} + \Delta\hat{\boldsymbol{\gamma}}_{\mathrm{g},k} \\ (\hat{\boldsymbol{\gamma}}_{\mathrm{a}})_{k|k} + \Delta\hat{\boldsymbol{\gamma}}_{\mathrm{a},k} \end{bmatrix}, \tag{4.106}$$

where

$$(\hat{\mathbf{q}}_\phi)_k = \begin{bmatrix} \cos\|0.5\hat{\boldsymbol{\phi}}_k\| \\ \dfrac{\sin\|0.5\hat{\boldsymbol{\phi}}_k\|}{\|0.5\hat{\boldsymbol{\phi}}_k\|}0.5\hat{\boldsymbol{\phi}}_k \end{bmatrix}. \tag{4.107}$$

Measurement Update

The measurement update procedures of the forward UKF can also be shared. The sigma points are transformed through the measurement model:

$$\boldsymbol{\mathcal{Z}}_{i,\mathrm{B},k|k+1} = \mathbf{h}\left[\boldsymbol{\mathcal{X}}'_{i,\mathrm{B},k|k+1}\right]. \tag{4.108}$$

The predicted measurement vector is computed as follows:

$$\hat{\mathbf{z}}_{\mathrm{B},k|k+1} = \sum_{i=0}^{p-1} w_i^{\mathrm{m}} \boldsymbol{\mathcal{Z}}_{i,\mathrm{B},k|k+1}. \tag{4.109}$$

The covariance matrix for the innovations sequence can be computed as follows:

$$\begin{aligned} \mathbf{P}_{\nu\nu,\mathrm{B},k} &= \sum_{i=0}^{p-1} w_i^{\mathrm{c}} \left[\Delta\boldsymbol{\mathcal{Z}}_{i,\mathrm{B},k|k+1}\right]\left[\Delta\boldsymbol{\mathcal{Z}}_{i,\mathrm{B},k|k+1}\right]^T + \mathbf{R}_k, && (4.110) \\ \Delta\boldsymbol{\mathcal{Z}}_{i,\mathrm{B},k|k+1} &= \boldsymbol{\mathcal{Z}}_{i,\mathrm{B},k|k+1} - \hat{\mathbf{z}}_{\mathrm{B},k|k+1}, && (4.111) \end{aligned}$$

where $\mathbf{R}_k$ is the measurement noise covariance matrix. The covariance matrix between the states and measurements is computed as follows:

$$\mathbf{P}_{\mathrm{xz},\mathrm{B},k} = \sum_{i=0}^{p-1} w_i^{\mathrm{c}} \left[\Delta\boldsymbol{\mathcal{X}}'_{i,\mathrm{B},k|k+1}\right]\left[\Delta\boldsymbol{\mathcal{Z}}_{i,\mathrm{B},k|k+1}\right]^T. \tag{4.112}$$

Then, the Kalman gain matrix is computed as follows:

$$\mathbf{K}_k = \mathbf{P}_{\mathrm{xz},\mathrm{B},k}\mathbf{P}_{\nu\nu,\mathrm{B},k}^{-1}. \tag{4.113}$$

The states can be updated as follows:

$$\begin{bmatrix} \delta\hat{\mathbf{r}}^{\mathrm{n}}_{\mathrm{B},k} \\ \delta\hat{\mathbf{v}}^{\mathrm{n}}_{\mathrm{B},k} \\ \hat{\boldsymbol{\phi}}_{\mathrm{B},k} \\ \delta(\hat{\mathbf{b}}_{\mathrm{g}})_{\mathrm{B},k} \\ \delta(\hat{\mathbf{b}}_{\mathrm{a}})_{\mathrm{B},k} \\ \delta(\hat{\mathbf{s}}_{\mathrm{g}})_{\mathrm{B},k} \\ \delta(\hat{\mathbf{s}}_{\mathrm{a}})_{\mathrm{B},k} \\ \delta(\hat{\boldsymbol{\gamma}}_{\mathrm{g}})_{\mathrm{B},k} \\ \delta(\hat{\boldsymbol{\gamma}}_{\mathrm{a}})_{\mathrm{B},k} \end{bmatrix} = \mathbf{K}_k \left(\mathbf{z}_k - \hat{\mathbf{z}}_{\mathrm{B},k|k+1}\right) \text{ and } \hat{\mathbf{x}}_{\mathrm{B},k|k} = \begin{bmatrix} \hat{\mathbf{r}}^{\mathrm{n}}_{\mathrm{B},k|k+1} + \delta\hat{\mathbf{r}}^{\mathrm{n}}_{\mathrm{B},k} \\ \hat{\mathbf{v}}^{\mathrm{n}}_{\mathrm{B},k|k+1} + \delta\hat{\mathbf{v}}^{\mathrm{n}}_{\mathrm{B},k} \\ (\hat{\mathbf{q}}_{\phi})_{\mathrm{B},k} \star (\hat{\mathbf{q}}^{\mathrm{n}}_{\mathrm{b}})_{\mathrm{B},k|k+1} \\ (\hat{\mathbf{b}}_{\mathrm{g}})_{\mathrm{B},k|k+1} + \delta(\hat{\mathbf{b}}_{\mathrm{g}})_{\mathrm{B},k} \\ (\hat{\mathbf{b}}_{\mathrm{a}})_{\mathrm{B},k|k+1} + \delta(\hat{\mathbf{b}}_{\mathrm{a}})_{\mathrm{B},k} \\ (\hat{\mathbf{s}}_{\mathrm{g}})_{\mathrm{B},k|k+1} + \delta(\hat{\mathbf{s}}_{\mathrm{g}})_{\mathrm{B},k} \\ (\hat{\mathbf{s}}_{\mathrm{a}})_{\mathrm{B},k|k+1} + \delta(\hat{\mathbf{s}}_{\mathrm{a}})_{\mathrm{B},k} \\ (\hat{\boldsymbol{\gamma}}_{\gamma})_{\mathrm{B},k|k+1} + \delta(\hat{\boldsymbol{\gamma}}_{\mathrm{g}})_{\mathrm{B},k} \\ (\hat{\boldsymbol{\gamma}}_{\gamma})_{\mathrm{B},k|k+1} + \delta(\hat{\boldsymbol{\gamma}}_{\mathrm{a}})_{\mathrm{B},k} \end{bmatrix}, \tag{4.114}$$

where $\mathbf{z}_k$ is the measurement vector and

$$(\hat{\mathbf{q}}_{\phi})_{\mathrm{B},k} = \begin{bmatrix} \cos(0.5\|\hat{\boldsymbol{\phi}}_{\mathrm{B},k}\|) \\ \dfrac{\sin(0.5\|\hat{\boldsymbol{\phi}}_{\mathrm{B},k}\|)}{0.5\|\hat{\boldsymbol{\phi}}_{\mathrm{B},k}\|} 0.5\hat{\boldsymbol{\phi}}_k \end{bmatrix}. \tag{4.115}$$

The covariance update can be updated as

$$\mathbf{P}_{\mathrm{B},k|k} = \mathbf{P}_{\mathrm{B},k|k+1} - \mathbf{K}_k \mathbf{P}_{\nu\nu,\mathrm{B},k} \mathbf{K}_k^T. \tag{4.116}$$

The scaled sigma points need to be regenerated after updating the states and covariance. The square-root matrix $\mathbf{S}^{\mathrm{a}}_{\mathrm{B},k|k}$ can be obtained such that $\mathbf{S}^{\mathrm{a}}_{\mathrm{B},k|k}(\mathbf{S}^{\mathrm{a}}_{\mathrm{B},k|k})^T = \mathbf{p}^{\mathrm{a}}_{\mathrm{B},k|k}$, where

$$\mathbf{p}^{\mathrm{a}}_{\mathrm{B},k|k} = \begin{bmatrix} \mathbf{P}_{\mathrm{B},k|k} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{k-1} \end{bmatrix}. \tag{4.117}$$

Then,

$$\begin{bmatrix} \Delta\mathbf{r}^{\mathrm{n}}_{i,k} \\ \Delta\mathbf{v}^{\mathrm{n}}_{i,k} \\ \boldsymbol{\phi}_{i,k} \\ \Delta(\mathbf{b}_{\mathrm{g}})_{i,k} \\ \Delta(\mathbf{b}_{\mathrm{a}})_{i,k} \\ \Delta(\mathbf{s}_{\mathrm{g}})_{i,k} \\ \Delta(\mathbf{s}_{\mathrm{a}})_{i,k} \\ \Delta(\boldsymbol{\gamma}_{\mathrm{g}})_{i,k} \\ \Delta(\boldsymbol{\gamma}_{\mathrm{a}})_{i,k} \\ \mathbf{w}_{i,k-1} \end{bmatrix} = \alpha\mathbf{S}^{\mathrm{a}}_{\mathrm{B},k|k}\boldsymbol{\mathcal{X}}^{\mathrm{a}}_{\mathrm{u},i} \text{ and } \boldsymbol{\mathcal{X}}^{\mathrm{a}\prime}_{i,\mathrm{B},k|k} = \begin{bmatrix} \hat{\mathbf{r}}^{\mathrm{n}}_{\mathrm{B},k|k} + \Delta\mathbf{r}^{\mathrm{n}}_{i,k} \\ \hat{\mathbf{v}}^{\mathrm{n}}_{\mathrm{B},k|k} + \Delta\mathbf{v}^{\mathrm{n}}_{i,k} \\ (\mathbf{q}_{\phi})_{i,k} \star (\hat{\mathbf{q}}^{\mathrm{n}}_{\mathrm{b}})_{\mathrm{B},k|k} \\ (\hat{\mathbf{b}}_{\mathrm{g}})_{\mathrm{B},k|k} + \Delta(\mathbf{b}_{\mathrm{g}})_{i,k} \\ (\hat{\mathbf{b}}_{\mathrm{a}})_{\mathrm{B},k|k} + \Delta(\mathbf{b}_{\mathrm{a}})_{i,k} \\ (\hat{\mathbf{s}}_{\mathrm{g}})_{\mathrm{B},k|k} + \Delta(\mathbf{s}_{\mathrm{g}})_{i,k} \\ (\hat{\mathbf{s}}_{\mathrm{a}})_{\mathrm{B},k|k} + \Delta(\mathbf{s}_{\mathrm{a}})_{i,k} \\ (\hat{\boldsymbol{\gamma}}_{\mathrm{g}})_{\mathrm{B},k|k} + \Delta(\boldsymbol{\gamma}_{\mathrm{g}})_{i,k} \\ (\hat{\boldsymbol{\gamma}}_{\mathrm{a}})_{\mathrm{B},k|k} + \Delta(\boldsymbol{\gamma}_{\mathrm{a}})_{i,k} \\ \mathbf{w}_{i,k-1} \end{bmatrix}, \tag{4.118}$$

where

$$(\mathbf{q}_{\phi})_{i,k} = \begin{bmatrix} \cos(0.5\|\boldsymbol{\phi}_{i,k}\|) \\ \dfrac{\sin(0.5\|\boldsymbol{\phi}_{i,k}\|)}{0.5\|\boldsymbol{\phi}_{i,k}\|} 0.5\boldsymbol{\phi}_{i,k} \end{bmatrix}.$$

Chapter 5

Tests and Results

In this chapter, the performance of the EKF/RTS approach and the UKF/UKS approach will be compared using datasets collected from MEMS IMUs mounted in land vehicles. The DGPS position solutions from Waypoint Consulting, Inc.'s GrafNav$^{\mathrm{TM}}$ software are used as the measurement updates for the EKF and the UKF in all datasets.

5.1 The First Dataset

A prototype MEMS IMU from Inertial Science, Inc. was used in the first test. Table B.1 in Appendix B lists some of the specifications for the 'Rev. C' model of the IMU. As the attitude difference between the MEMS IMU and the reference IMU is very small, this dataset will be used mainly to test the alignment performance of the EKF and the UKF.

The reference trajectory used in this test was provided by Applanix Corp. This reference, the smoothed best estimates (SBET), was generated by processing the data from a tactical-grade IMU (Northrop Grumman LN-200) using Applanix's POSPac^{TM} software. The accuracy of the reference attitude is known to be 0.02 ° for roll and pitch and 0.025 ° for heading when there are no GPS outages.

The vehicle used in the test was a van driven with low dynamics along the trajectory shown in Figure 5.1. Figure 5.2 shows the accuracy of the DGPS solution. For most of the time, horizontal and vertical position accuracies of the DGPS solution were about 2–3 cm and 4–9 cm, respectively.

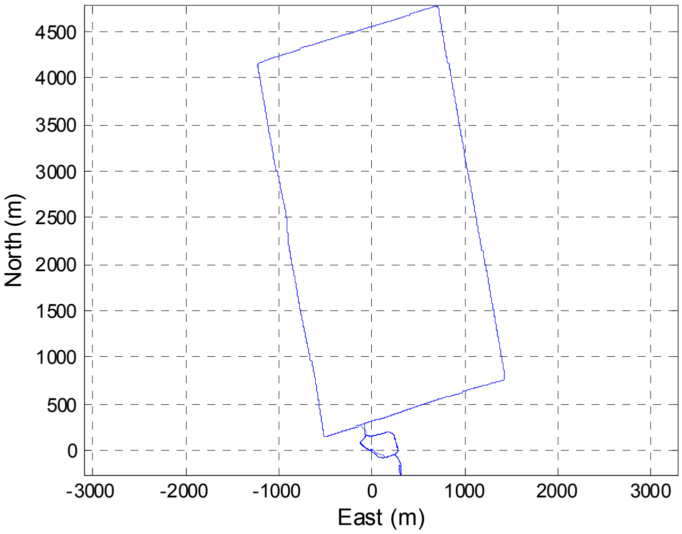

Figure 5.1: Trajectory of the first dataset

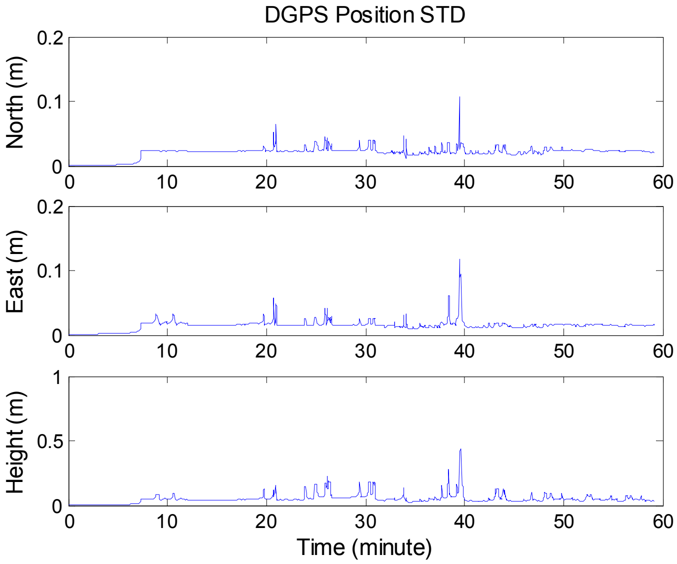

Figure 5.2: DGPS position accuracy of the first dataset

The majority of the tests will be done using this dataset. Tuning of the EKF and the UKF will be investigated in Section 5.1.1. In-motion alignment performance of the two filters is compared in Section 5.1.2. The effect of ZIHR measurements is tested in Section 5.1.3. In Section 5.1.4, behaviour of a low-cost INS during GPS outages will be investigated. Finally, the effect of smoothing is demonstrated in Section 5.1.5.

5.1.1 Filter Tuning

Both filters have 21 states including biases and scale factors. The EKF uses the small $\boldsymbol{\psi}$-angle error model described in Section 3.1.2. Table 5.1 lists the parameters used in the test. For land vehicle applications, using low-cost gyroscopes, stationary outputs of gyroscopes can be considered as the initial biases. This is because the Earth's rotation rate (≈15 °/h) is negligibly small against the large biases of the gyroscopes. Hence, although the actual gyro biases are about 200–1000 deg/h, the large biases can be estimated (for instance, by averaging) while the vehicle is stationary and can be treated as known errors. Therefore, the initial uncertainties of the gyro biases can be reduced from the start and $\sigma = 100\,{}^{\mathrm{deg}}\!/\mathrm{h}$ is used in the Gauss-Markov model.

Table 5.1: Tuning parameters for the prototype ISIS IMU

VRW	$0.6\,{}^{\mathrm{m/s}}\!/\sqrt{\mathrm{h}}$
ARW	$3.5\,{}^{\mathrm{deg}}\!/\sqrt{\mathrm{h}}$
Gyro bias	Gauss-Markov with $\sigma = 100\,{}^{\mathrm{deg}}\!/\mathrm{h}$, $T = 1\,\mathrm{hour}$
Accel. bias	Gauss-Markov with $\sigma = 0.1\,{}^{\mathrm{m}}\!/s^2$, $T = 1\,\mathrm{hour}$
Gyro scale factor	Gauss-Markov with $\sigma = 1000\,\mathrm{PPM}$, $T = 4\,\mathrm{hour}$
Accel. scale factor	Gauss-Markov with $\sigma = 1000\,\mathrm{PPM}$, $T = 4\,\mathrm{hour}$

Table 5.2 lists the percentages of the errors included in the 1σ, 2σ and 3σ envelopes, for which 68%, 95% and 99% correspond in Gaussian distribution, respectively. A "conservative" Kalman filter has larger σ-bounds than needed, and hence more than 68% of errors are included inside the σ-bound. Therefore, it can be said that the parameters in Table 5.1 resulted in a consevative Kalman filter except for the heading.

Table 5.2: Error envelope of the EKF

		1σ (%)	2σ (%)	3σ (%)
Position	North	73.3	95.8	99.5
	East	73.2	96.8	98.9
	Height	92.6	97.6	99.2
Velocity	North	90.0	99.0	99.8
	East	88.3	98.9	99.9
	Down	93.8	98.8	99.5
Attitude	Roll	80.0	99.2	100.0
	Pitch	86.3	99.2	99.6
	Heading	63.5	93.9	99.2

The tuning parameters in Table 5.1 have been applied to the UKF and additional parameters specific to the UKF are listed in Table 5.3. The resulting error envelope of the UKF, listed in Table 5.4, is similar to that of the EKF.

Table 5.3: Parameters of the UKF

Weight of 0th point	$w_0 = 0.5$
Scaling parameters	$\alpha = 0.05$, $\beta = 2$

Table 5.4: Error envelope of the UKF

		1σ (%)	2σ (%)	3σ (%)
Position	North	73.9	95.9	99.5
	East	73.2	96.8	98.9
	Height	92.7	97.6	99.2
Velocity	North	90.1	99.0	99.8
	East	88.2	98.8	99.9
	Down	94.0 98.8	99.5	
Attitude	Roll	80.7	99.2	100.0
	Pitch	87.4	99.2	99.6
	Heading	64.1	94.3	99.2

5.1.2 In-Motion Alignment

As the INS cannot have a static period upon initialization, the large gyro biases cannot be estimated as was done in Section 5.1.1. Therefore, a Gauss-Markov model with σ=1000 $^{\mathrm{deg}}/_{\mathrm{h}}$ will be used in this section to model the gyro biases. Other parameters will be the same as those listed in Table 5.1.

To initialize the INS during motion, the position, velocity, attitude, and their uncertainties have to be obtained from the DGPS solution. If the position is initialized directly with that of the DGPS solution, then the remaining uncertainty will be the lever-arm. The velocities can be derived from the position increment as follows:

$$v_{\mathrm{N}} = \Delta\varphi_m(R_{\mathrm{M}} + h_{m-1/2})/\Delta t_m, \tag{5.1a}$$

$$v_{\mathrm{E}} = \Delta\lambda_m(R_{\mathrm{N}} + h_{m-1/2})\cos\varphi_{m-1/2}/\Delta t_m, \tag{5.1b}$$

$$v_{\mathrm{D}} = -\Delta h_m/\Delta t_m, \tag{5.1c}$$

where the subscript m denotes the GPS measurement epochs, $\Delta\varphi_m = \varphi_m - \varphi_{m-1}$, $\Delta\lambda_m = \lambda_m - \lambda_{m-1}$, $\Delta h_m = h_m - h_{m-1}$, and $\Delta t_m = t_m - t_{m-1}$. As shown in Figure 5.3, the velocity errors of Eqs. (5.1a)–(5.1c) are, in most cases, within 1.5 $^{\mathrm{m}}/_{\mathrm{s}}$ and, in extreme cases, over 2 $^{\mathrm{m}}/_{\mathrm{s}}$ which also includes the uncompensated lever-arm effects.

Tilt sensors can be used to initialize the roll and pitch. For land vehicles on a level surface, however, the roll can be initialized with zero. The pitch and heading can be initialized using the DGPS position-derived velocity with Eqs. (2.73a) and (2.73b), respectively. Figures 5.4 and 5.5 show that the initial pitch and heading

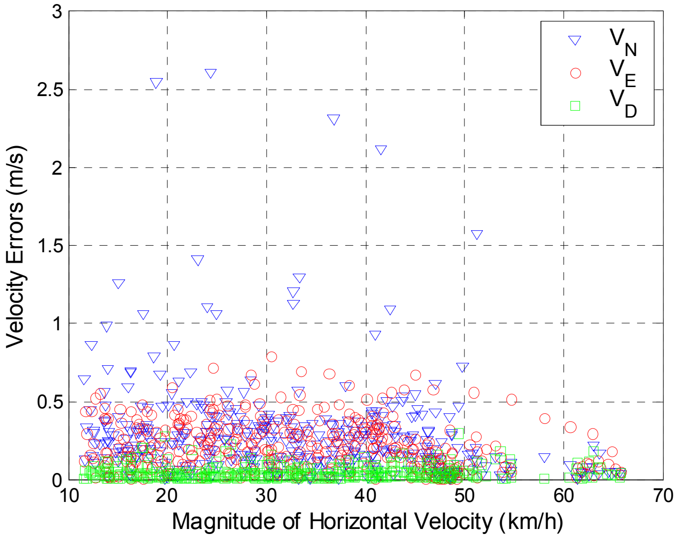

Figure 5.3: DGPS position-derived velocity errors

errors computed from the DGPS position-derived velocity can be up to 3° and 6°, respectively, when speeds are between 10 and 55 km/h. For speeds higher than this, the pitch and heading errors are smaller than 1°. Therefore, small error assumption holds for land vehicles in general.

In airborne or shipborne applications, the velocity may not be parallel to the longitudinal axis of the vehicle, in which case the initial pitch and heading computed from Eqs. (2.73a) and (2.73b) can have large errors. Initial roll errors may also be large if the vehicle is in high-dynamic motion. Therefore, a test was conducted by intentionally adding large initial attitude errors (40°) for roll, pitch and heading. The speed at the start of the in-motion alignment was about 11.5 km/h and the results are shown in Figure 5.6. In the UKF, the roll and pitch converged within 10 s and the heading within 50 s. The EKF, however, required a much longer transient period of

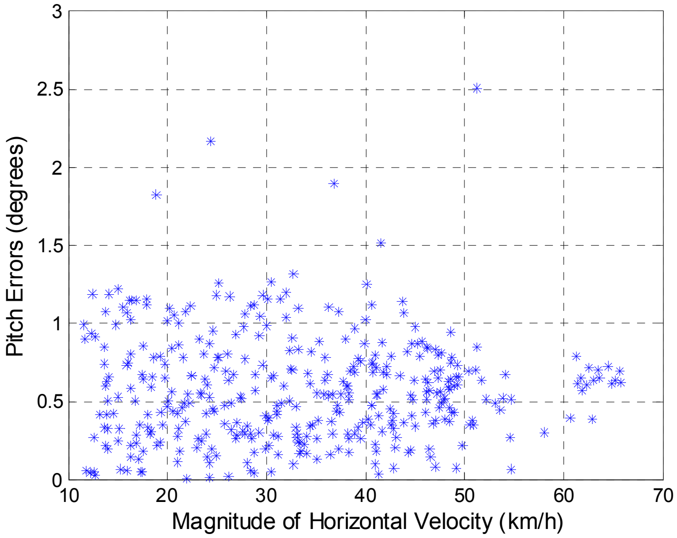

Figure 5.4: DGPS position-derived pitch errors

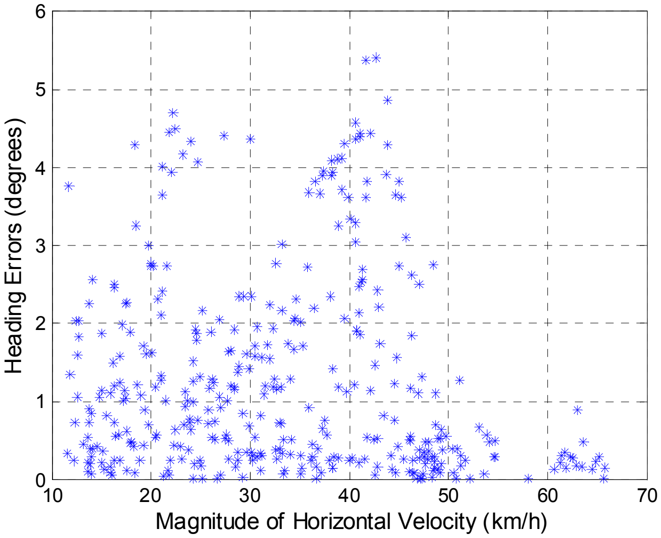

Figure 5.5: DGPS position-derived heading errors

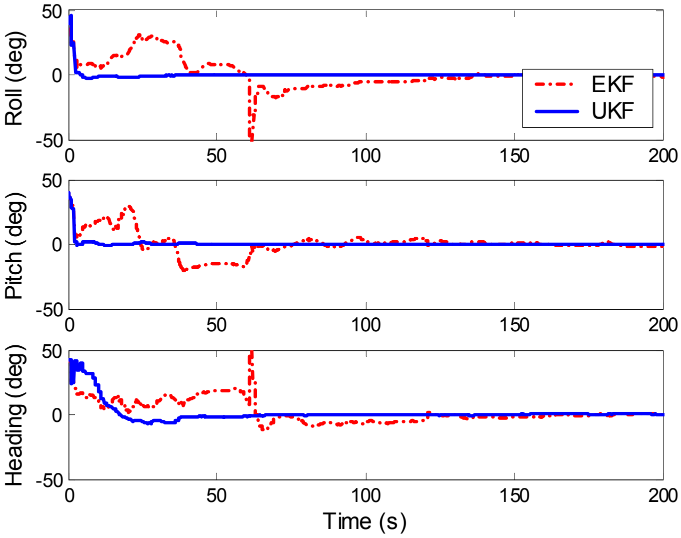

Figure 5.6: Attitude errors during in-motion alignment with 40° initial attitude errors about 200 seconds.

If the vehicle is not accelerating, then the roll and pitch can be determined by leveling and only the heading may have large uncertainty. Hence, the two algorithms have been tested with a 60° initial heading error and the results are shown in Figure 5.7. Again, the UKF showed faster convergence than the EKF. Therefore, the UKF performs better than the EKF when large attitude uncertainties exist and this is the key difference between the two filters.

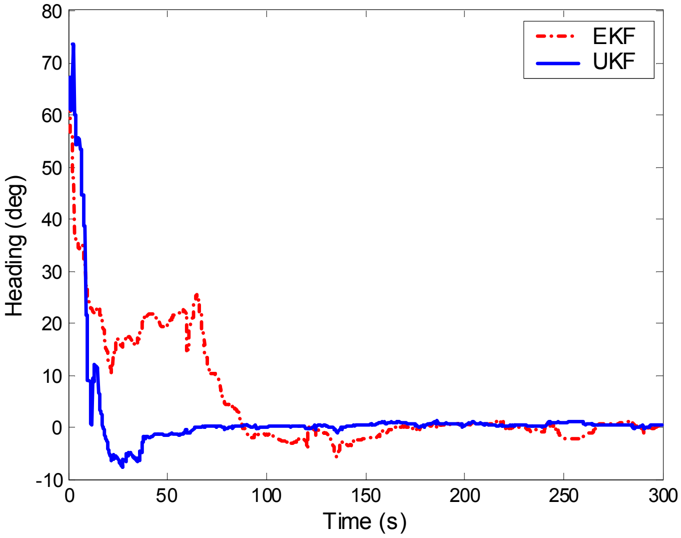

Figure 5.7: Heading errors during in-motion alignment with a 60° initial heading error

5.1.3 Zero Integrated Heading Rate Measurement

A 2 minute ZUPT is applied after 365 seconds from the in-motion alignment using the EKF with small initial attitude errors. As shown in Figure 5.8, the heading drifts over 3 degrees during the ZUPT period. Therefore, ZUPTs are not enough for low-cost INSs and the ZIHR measurements, discussed in Section 3.3.2, can be applied together to fix the heading during ZUPTs. The benefit of ZIHR measurements is clearly visible in Figure 5.8.

To apply ZIHR measurements, heading outputs from the INS at two different time epochs are required. Assume that the heading at time t_{k-1} is stored and a ZIHR measurement is scheduled to occur at time t_k. The attitude feedback can happen at time t_m: $t_{k-1} < t_m < t_k$. In this case, the heading at time t_m can be

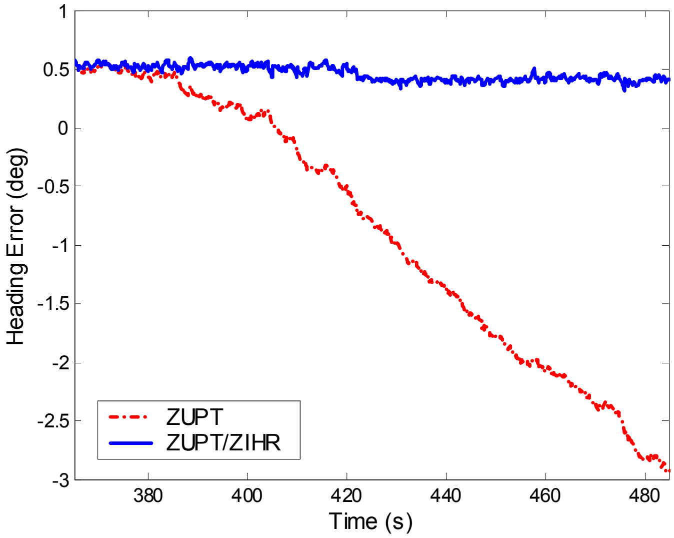

Figure 5.8: Heading fix during a ZUPT

stored after the feedback and the ZIHR measurement must be rescheduled. If the integration time interval is fixed, then the next ZIHR measurement should occur at time $t_{m+1} = t_m + t_k - t_{k-1}$.

5.1.4 GPS Outage and Attitude Error

Figures 5.9 and 5.10 show the position and attitude errors, respectively, when an artificial 60 second GPS outage is introduced in the EKF. The position error reached about 120 m at the end of the GPS outage. Also, tilt errors reached about 1 degrees, which causes the projection of gravity onto horizontal channels. Therefore, tilt errors are dominating error sources during GPS outages in a low-cost INS.

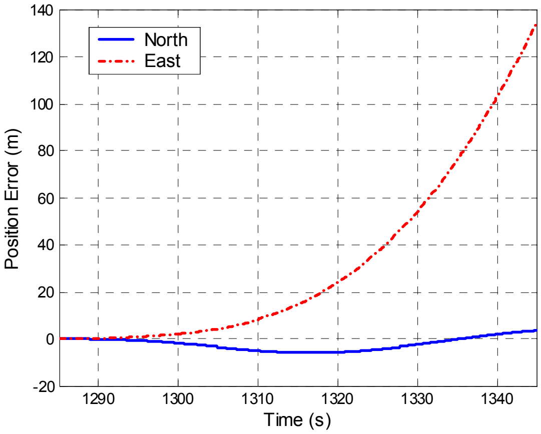

Figure 5.9: Position errors during a 60s GPS outage

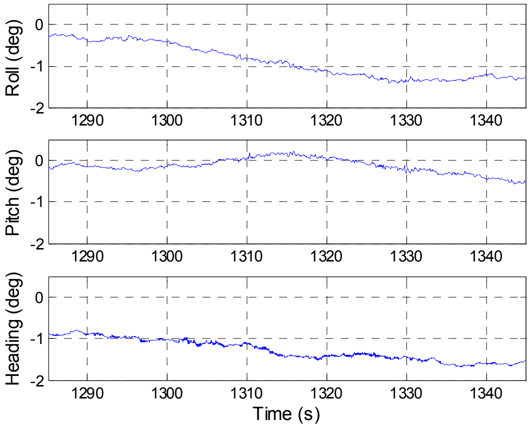

Figure 5.10: Attitude errors during a 60s GPS outage

5.1.5 GPS Outages and Smoothing

To examine the performance of the filters and smoothers, 30 second GPS outages were intentionally introduced and the position errors of the algorithms are listed in Table 5.5. Although the position errors of both filters reached over 50 m, those of the smoothers remained at the 1 m level. The EKF/RTS and the UKF/UKS approaches yielded similar performance. Therefore, both approaches can be considered to be the same if the error dynamics are linear.

Table 5.5: Position errors during 30s GPS outages (ISI IMU)

Outage	Direction		EKF	UKF	RTS	UKS
1	North (m)	Max.	16.922	24.240	0.794	0.759
		RMS	8.143	10.933	0.473	0.454
	East (m)	Max.	1.079	3.382	0.156	0.156
		RMS	0.613	2.162	0.070	0.070
	Height (m)	Max.	1.235	2.328	0.066	0.044
		RMS	0.610	1.083	0.032	0.024
2	North (m)	Max.	18.108	1.030	1.422	1.257
		RMS	7.336	0.475	0.774	0.670
	East (m)	Max.	84.760	72.983	0.297	0.297
		RMS	34.963	30.228	0.186	0.186
	Height (m)	Max.	3.756	3.672	0.229	0.169
		RMS	1.648	1.583	0.124	0.088
3	North (m)	Max.	33.509	4.372	0.911	1.442
		RMS	13.361	2.548	0.444	0.660
	East (m)	Max.	69.142	45.326	1.168	1.168
		RMS	28.232	17.436	0.632	0.632
	Height (m)	Max.	1.045	0.244	0.231	0.220
		RMS	0.572	0.103	0.113	0.103

5.2 The Second Dataset

The second dataset used in testing was collected using a prototype MEMS IMU developed by the mobile multi-sensor research group at the University of Calgary. This device is shown in Figure 5.11. Since sensors from Analog Devices, Inc. were used, this IMU will be referred to as the ADI sensor triads (El-Sheimy and Niu, 2004). The specifications of this IMU are listed in Table B.2 of Appendix B.

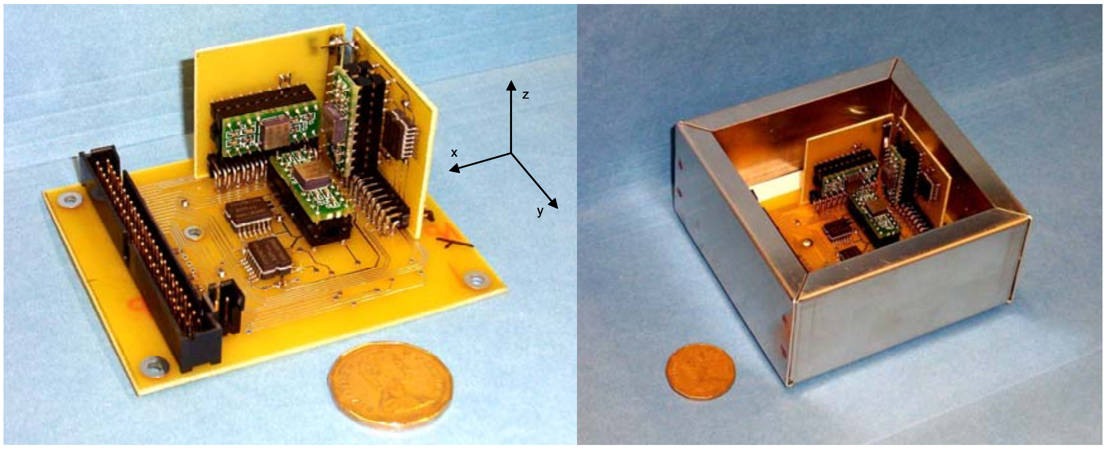

Figure 5.11: ADI sensor triads developed by the mobile multi-sensor research group, the University of Calgary (El-Sheimy and Niu, 2004)

The purpose of testing this dataset is to investigate the effect of odometer aiding during GPS outages. Figures 5.12 and 5.13 show the test trajectory and the position accuracy of the DGPS solution, respectively. Smoothed solutions from a navigation-grade IMU (Honeywell C-IMU) will be used as the reference in this dataset. Again, the POSPac$^{\text{TM}}$ software was used to generate the reference solution.

An odometer was also developed and installed together with the ADI sensor triads

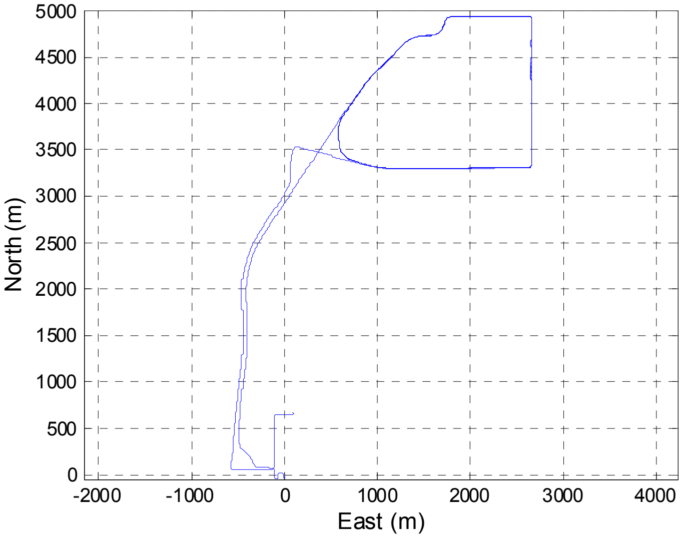

Figure 5.12: Trajectory of the second data set

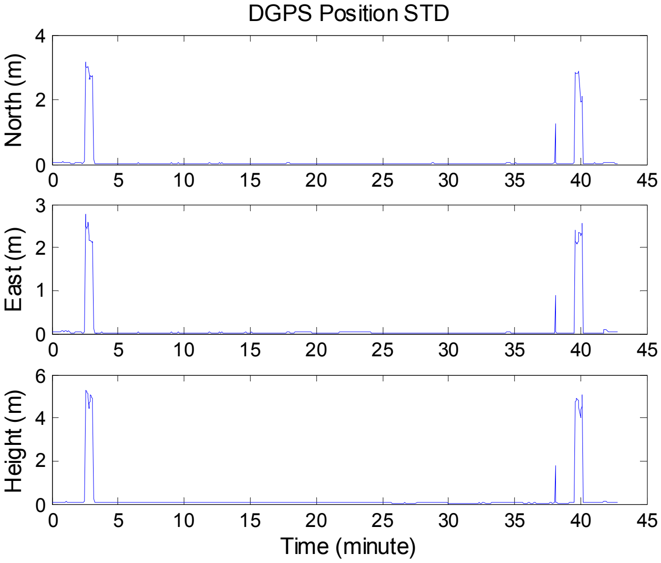

Figure 5.13: DGPS accuracy of the second dataset

on the vehicle (Niu and El-Sheimy, 2004). Figure 5.14 shows the configuration of the odometer. A magnet was attached on the rim of the left-rear wheel. A Hall sensor was used to generate pulses whenever the magnet passed it.

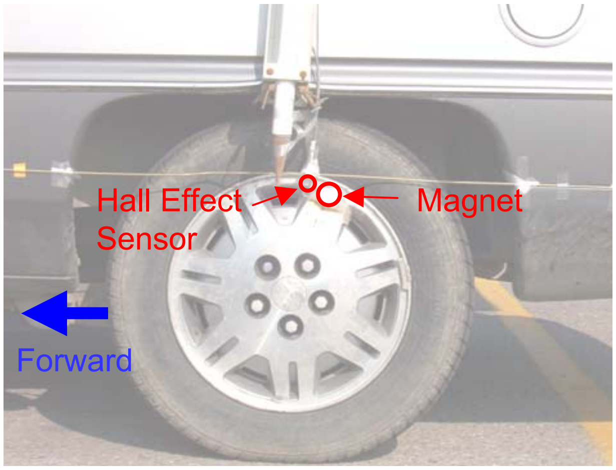

Figure 5.14: Odometer (Courtesy of Dr. Xiaoji Niu)

Each revolution of the wheel is equivalent to about 2 m. The along-track speed can be obtained by dividing the travel distance by the time interval between adjacent pulses. Unfortunately, since only one magnet was attached, the resolution of the odometer was very low; therefore, it had large quantization noise. Furthermore, pulses were frequently lost and the sensor was not able to sense the direction of movement. In spite of these limitations, the benefit of odometer aiding in the EKF during artificial GPS outages can be significant. As shown in Tables 5.6 and 5.7, the odometer aiding decreased the position error during GPS outages (about 4 metres in maximum) to a level smaller than 1 meter.

Table 5.6: Position errors of the EKF during 10 s GPS outages (ADI sensor triads/Odometer)

		INS (m)			INS/Odometer (m)		
Outage		North	East	Height	North	East	Height
1	Max.	1.463	0.651	0.928	0.132	0.539	0.335
	RMS	0.688	0.439	0.578	0.080	0.215	0.219
2	Max.	0.810	0.863	2.033	0.763	0.615	0.166
	RMS	0.438	0.462	1.116	0.324	0.260	0.078
3	Max.	4.123	0.410	0.391	0.068	0.094	0.218
	RMS	1.887	0.158	0.295	0.046	0.056	0.188
4	Max.	0.282	0.500	0.362	0.125	0.153	0.671
	RMS	0.200	0.207	0.283	0.078	0.125	0.377

Table 5.7: Position errors of the EKF during 30 s GPS outages (ADI sensor triads/Odometer)

		INS (m)			INS/Odometer (m)		
Outage		North	East	Height	North	East	Height
1	Max.	1.491	0.660	0.925	0.132	0.542	0.341
	RMS	0.699	0.444	0.576	0.080	0.216	0.222
2	Max.	0.789	0.872	2.008	0.761	0.614	0.166
	RMS	0.434	0.461	1.106	0.324	0.260	0.078
3	Max.	4.045	0.406	0.392	0.067	0.095	0.216
	RMS	1.853	0.156	0.297	0.044	0.057	0.185
4	Max.	0.288	0.509	0.368	0.127	0.154	0.668
	RMS	0.206	0.211	0.289	0.081	0.126	0.375

Chapter 6

Conclusions and Recommendations

6.1 Summary

The main objective of this dissertation is to investigate unscented estimation techniques for aided low-cost inertial navigation. Various types of aiding models are investigated in Section 2.5 to overcome the limitation in the accuracy of a low-cost IMU. Since choosing an appropriate INS error model is an essential part in an LKF/EKF design, INS error models are discussed in Section 3.1. Choosing an appropriate estimation method is also important. Hence, the EKF and the RTS smoother, state-of-the-art estimation techniques, are developed in Chapter 3. A measurement model for the ZIHR is developed in Section 3.3.2 to fix heading drift of a low-cost INS during ZUPTs, which is useful for wheeled vehicles. In Section 5.1.3, the ZIHR measurement model is tested successfully and it is discussed that measurement rescheduling is required if the attitude feedback occurs in between the integration time interval. In Section 5.2, it is shown that, even with a primitive design, odometer measurement updates can improve the position accuracy significantly during GPS outages.

The research work contributed the following new developments to the field of inertial navigation:

1. In Section 4.3, a general system process model is developed for a quaternion-based UKF for an aided INS. An efficient model to apply GPS position measurement updates is developed in Section 4.3.3. It is demonstrated in Section 5.1.2 that the UKF converges faster than the EKF when the initial attitude uncertainties are large. Hence, the UKF can recover faster than the EKF from large attitude errors due to GPS outages.

2. A quaternion-based backward UKF is also developed and the method to combine forward and backward solutions is discussed to yield unscented smoothed solutions. It is demonstrated in Section 5.1.5 that the performance of the UKS is as good as that of the RTS smoother.

The following sections contain conclusions drawn from the testing results and recommendations for further research.

6.2 Conclusions

The ultimate difference between the UKF and the EKF is in the way they deal with nonlinearities. Nonlinear error dynamics typically happen in a navigation system when attitude errors are large. The UKF, however, cannot deal with complete uncertainties in attitude because it works based upon averaging sigma points; for instance, the average of three directions -120°, 0°, and +120° cannot be defined.

For land vehicles, roll and pitch errors are usually small but heading errors can grow quickly if large uncertainties exist in gyro outputs. Thus, the UKF is preferable if the heading error can be large in an operational scenario. However, the benefit of the second-order accuracy of the UKF becomes negligibly small when the small attitude error assumption is valid. Therefore, the EKF/RTS approach should still be chosen if the attitude errors are guaranteed to be small during the operation.

In the EKF, it is designer's responsibility to choose or develop an appropriate INS error model. This is not a trivial task. As discussed in Section 3.1, many INS error models currently exist and virtually infinite number of error models can be created. There can be specific difficulties in the implementation for each INS error model and the implementation is also dependent on the aiding sensors. However, in the UKF, only the general system process and measurement models need to be defined. Hence, the UKF is capable of unifying INS error models in a broad range of attitude uncertainties. Therefore, the system development stage can be simplified by using the UKF.

6.3 Recommendations for Further Research

1. Investigation of the change of the PDF during a navigation mission is warranted. Julier and LaViola (2003) argued that the PDF was not Gaussian but rather it is a von Mises distribution if the normalization constraint of quaternions was enforced. Although particle filters may not be practical in navigation due to the heavy computational load, they can be used in investigating the PDF. The optimal values of the parameters α and β in the UKF, which control the effect of higher-order moments, can also be determined through investigation of the PDF.

2. Although some of the noise statistics can be obtained through the Allan variance analysis, it only provides statistics at sensor level and not at the overall system level. Furthermore, the Allan variance analysis does not always work well for low-cost inertial sensors (Scherzinger, 2004). The system noise statistics can also vary according to the temperature change. Thus, the system noise statistics in this dissertation are chosen empirically. Although some studies have been conducted on the adaptation of system noise intensity, development of efficient adaptation algorithms is still an open research area. For a recent publication on this subject, refer to Powell (2002).

3. If the UKF can handle complete uncertainties in attitude, then it can entirely unify all the INS error models developed so far. Therefore, more research is required to extend the UKF to handle complete attitude uncertainties: especially, uncertainties in the heading.

4. The forward and backward filtering approach has been developed in this dissertation for unscented smoothing. For an efficient computation, however, an RTS-style implementation of unscented smoother needs to be investigated.

5. Although Reif et al. (1999) showed that the estimation error of the EKF was bounded when the initial estimation error and disturbing noise were small enough, stability of nonlinear estimators are largely unknown. Therefore, the stability of the UKF requires investigation.

6. Covariance formulation of the UKF discussed in Chapter 4 requires a Cholesky factorization whenever a transformation occurs, which increases the computational load significantly. Hence, in actual hardware implementation, square-root formulation will be preferred, which stores and propagates square-root of the covariance matrix. To implement a square-root UKF, a Cholesky update algorithm (rank 1 update to Cholesky factorization) is required. Wan and van der Merwe (2001) provides a generic structure of the square-root UKF. Cholesky update algorithms can be found in Gill et al. (1974).

Appendix A

Matrix Representation of the Psi-Angle Error Model

The system dynamics for the psi-angle error model can be written in matrix form as

$$\delta\dot{\mathbf{x}} = \mathbf{F}\delta\mathbf{x} + \mathbf{G}\mathbf{w}. \tag{A.1}$$

For instance, the error state vector can be defined as

$$\delta\mathbf{x} = \begin{bmatrix} (\delta\mathbf{r}^{\mathrm{c}})^T & (\delta\mathbf{v}^{\mathrm{c}})^T & \boldsymbol{\psi}^T & \mathbf{b}_{\mathrm{g}}^T & \mathbf{b}_{\mathrm{a}}^T & \mathbf{s}_{\mathrm{g}}^T & \mathbf{s}_{\mathrm{a}}^T & \boldsymbol{\gamma}_{\mathrm{g}}^T & \boldsymbol{\gamma}_{\mathrm{a}}^T \end{bmatrix}^T, \tag{A.2}$$

where $\delta\mathbf{r}^{\mathrm{c}}$ and $\delta\mathbf{v}^{\mathrm{c}}$ can be written as follows:

$$\delta\mathbf{r}^{\mathrm{c}} = \begin{bmatrix} \delta r_{\mathrm{N}} & \delta r_{\mathrm{E}} & \delta r_{\mathrm{D}} \end{bmatrix}^T, \tag{A.3}$$

$$\delta\mathbf{v}^{\mathrm{c}} = \begin{bmatrix} \delta v_{\mathrm{N}}^{\mathrm{c}} & \delta v_{\mathrm{E}}^{\mathrm{c}} & \delta v_{\mathrm{D}}^{\mathrm{c}} \end{bmatrix}^T; \tag{A.4}$$

$\boldsymbol{\psi}$ denotes the attitude errors; $\mathbf{b}_{\mathrm{g}}$, the gyro biases; $\mathbf{b}_{\mathrm{a}}$, the accelerometer biases; $\mathbf{s}_{\mathrm{g}}$ represents the gyro scale factors; $\mathbf{s}_{\mathrm{a}}$ denotes the accelerometer scale factors; and $\boldsymbol{\gamma}_{\mathrm{g}}$ and $\boldsymbol{\gamma}_{\mathrm{a}}$ are nonorthogonalities of the gyro and accelerometer triad, respectively. Then, the system dynamics matrix can be written as

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{11} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{21} & \mathbf{F}_{22} & (\mathbf{f}^{\mathrm{c}}\times) & \mathbf{0} & \mathbf{C}_{\mathrm{b}}^{\mathrm{p}} & \mathbf{0} & \mathbf{F}_{27} & \mathbf{0} & \mathbf{C}_{\mathrm{b}}^{\mathrm{p}}\boldsymbol{\Gamma}_{\mathrm{a}} \\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{33} & -\mathbf{C}_{\mathrm{b}}^{\mathrm{p}} & \mathbf{0} & \mathbf{F}_{36} & \mathbf{0} & -\mathbf{C}_{\mathrm{b}}^{\mathrm{p}}\boldsymbol{\Gamma}_{\mathrm{g}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{44} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{55} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{66} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{77} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{88} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{99} \end{bmatrix}, \tag{A.5}$$

where $\mathbf{F}_{11} = -(\boldsymbol{\omega}_{\mathrm{ec}}^{\mathrm{c}}\times)$, $\mathbf{F}_{21} = \mathrm{diag}(-\omega_{\mathrm{s}}^2, -\omega_{\mathrm{s}}^2, 2\omega_{\mathrm{s}}^2)$, $\mathbf{F}_{22} = -(2\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{c}} \times + \boldsymbol{\omega}_{\mathrm{ec}}^{\mathrm{c}}\times)$, $\mathbf{F}_{27} = \mathbf{C}_{\mathrm{b}}^{\mathrm{p}}\mathrm{diag}(\mathbf{f}^{\mathrm{b}})$, $\mathbf{F}_{33} = -(\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{c}} \times + \boldsymbol{\omega}_{\mathrm{ec}}^{\mathrm{c}}\times)$, $\mathbf{F}_{36} = -\mathbf{C}_{\mathrm{b}}^{\mathrm{p}}\mathrm{diag}(\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}})$, $\mathbf{F}_{44} = \mathrm{diag}(\mathbf{c}_{\mathrm{gb}})$, $\mathbf{F}_{55} = \mathrm{diag}(\mathbf{c}_{\mathrm{ab}})$,$\mathbf{F}_{66} = \mathrm{diag}(\mathbf{c}_{\mathrm{gs}})$, $\mathbf{F}_{77} = \mathrm{diag}(\mathbf{c}_{\mathrm{as}})$, $\mathbf{F}_{88} = \mathrm{diag}(\mathbf{c}_{\mathrm{g}\gamma})$, $\mathbf{F}_{99} = \mathrm{diag}(\mathbf{c}_{\mathrm{a}\gamma})$,

$$\boldsymbol{\Gamma}_{\mathrm{g}} = \begin{bmatrix} \omega_{\mathrm{y}} & \omega_{\mathrm{z}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega_{\mathrm{x}} & \omega_{\mathrm{z}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \omega_{\mathrm{x}} & \omega_{\mathrm{y}} \end{bmatrix}, \quad \boldsymbol{\Gamma}_{\mathrm{a}} = \begin{bmatrix} f_{\mathrm{y}} & f_{\mathrm{z}} & 0 & 0 & 0 & 0 \\ 0 & 0 & f_{\mathrm{x}} & f_{\mathrm{z}} & 0 & 0 \\ 0 & 0 & 0 & 0 & f_{\mathrm{x}} & f_{\mathrm{y}} \end{bmatrix};$$

$\mathbf{c}_{\mathrm{gb}}$, $\mathbf{c}_{\mathrm{ab}}$, $\mathbf{c}_{\mathrm{gs}}$, $\mathbf{c}_{\mathrm{as}}$, $\mathbf{c}_{\mathrm{g}\gamma}$, and $\mathbf{c}_{\mathrm{a}\gamma}$ are the coefficients describing the error models for the gyro biases, accelerometer biases, gyro scale factors, accelerometer scale factors, gyro triad non-orthogonalities and accelerometer triad non-orthogonalities, respectively. The system noise vector is defined as

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_{\mathbf{v}}^{T} & \mathbf{w}_{\boldsymbol{\psi}}^{T} & \mathbf{w}_{\mathrm{gb}}^{T} & \mathbf{w}_{\mathrm{ab}}^{T} & \mathbf{w}_{\mathrm{gs}}^{T} & \mathbf{w}_{\mathrm{as}}^{T} & \mathbf{w}_{\mathrm{g}\gamma}^{T} & \mathbf{w}_{\mathrm{a}\gamma}^{T} \end{bmatrix}^{T}, \tag{A.6}$$

where $\mathbf{w}_{\mathbf{v}}$ and $\mathbf{w}_{\boldsymbol{\psi}}$ are the velocity and the attitude noise, respectively; $\mathbf{w}_{\mathrm{gb}}$ and $\mathbf{w}_{\mathrm{ab}}$ represent the bias noise of the gyros and accelerometers, respectively; $\mathbf{w}_{\mathrm{gs}}$ and $\mathbf{w}_{\mathrm{as}}$ are noise of the gyro and accelerometer scale factor, respectively; $\mathbf{w}_{\mathrm{g}\gamma}$ and $\mathbf{w}_{\mathrm{a}\gamma}$ are noise of the gyro-triad and accelerometer-triad non-orthogonalities . The noise input mapping matrix can, then, be written as

$$\mathbf{G} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_{\mathrm{b}}^{\mathrm{p}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{\mathrm{b}}^{\mathrm{p}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}. \tag{A.7}$$

Appendix B

Specifications of the IMUs

The specifications of IMUs used in this dissertation are introduced in this section. Table B.1 lists the specifications of a MEMS IMU made by the Inertial Science, Inc. (http://www.inertialscience.com).

Table B.2 describes the specifications of the sensor triads built by the mobile multi-sensor research group at the University of Calgary, that integrates three ADXL105 accelerometers and three ADXRS150 gyroscopes made by Analog Devices, Inc. The listed biases are the residual values after laboratory calibrations while the other specifications are obtained from the specification sheets of the sensors provided by Analog Devices, Inc. (http://www.analog.com).

Table B.1: Specifications of the Rev. C model of the ISIS-IMU

	Accelerometers	Gyroscopes
Random walk	$< 0.1\,{}^{\mathrm{m/s}}/_{\sqrt{\mathrm{h}}}$	$< 0.5\,{}^{\mathrm{deg}}/_{\sqrt{\mathrm{h}}}$
Scale factor	< 2000 ppm	< 2000 ppm
Nonlinearity	< 5000 ppm	< 2000 ppm
Misalignment	< 2 mRad	–
Short term bias stability	$\pm$ 2 mg	$< 0.01\,{}^{\mathrm{deg}}/_{\mathrm{s}}$
Long term bias stability	–	$< 1\,{}^{\mathrm{deg}}/_{\mathrm{s}}$ (RMS)

Table B.2: Specifications of the ADI sensor triads

	Accelerometers	Gyroscopes
Range	± 5 g	$\pm 150\,{}^{\mathrm{deg}}/_{\mathrm{s}}$
Random walk	0.135–0.195 ${}^{\mathrm{m/s}}/_{\sqrt{\mathrm{h}}}$	$3\,{}^{\mathrm{deg}}/_{\sqrt{\mathrm{h}}}$
Nonlinearity	0.2%	0.1%
Bias	$0.05\,{}^{\mathrm{m}}/_{s^2}$	$0.5\,{}^{\mathrm{deg}}/_{\mathrm{s}}$

Appendix C

INS Error Analysis

If the vehicle is not accelerating, the east channel error equations can be simplified as follows:

$$\delta \dot{v}_{\mathrm{E}} = g\phi_{\mathrm{N}} + \delta f_{\mathrm{E}}, \tag{C.1a}$$

$$\dot{\phi}_{\mathrm{N}} = -\frac{1}{R}\delta v_{\mathrm{E}} + \delta\omega_{\mathrm{N}}, \tag{C.1b}$$

where δf_{E} and $\delta\omega_{\mathrm{N}}$ are the east channel accelerometer bias and the north channel gyro bias, respectively. Here, the biases will be considered as constants. Differentiating Eq. (C.1a) with respect to time yields

$$\delta \ddot{v}_{\mathrm{E}} = g\dot{\phi}_{\mathrm{N}} = -\frac{g}{R}\delta v_{\mathrm{E}} + g\delta\omega_{\mathrm{N}}, \tag{C.2}$$

where the gravity is also treated as a constant for simplicity. A Laplace transformation of the above equation gives

$$s^2\delta v_{\mathrm{E}}(s) - s\delta v_{\mathrm{E}}(0) - \delta\dot{v}_{\mathrm{E}}(0) = -\frac{g}{R}\delta v_{\mathrm{E}}(s) + g\delta\omega_{\mathrm{N}}\frac{1}{s}. \tag{C.3}$$

From Eq. (C.1a) we can write

$$\delta\dot{v}_{\mathrm{E}}(0) = g\phi_{\mathrm{N}}(0) + \delta f_{\mathrm{E}}. \tag{C.4}$$

Substituting the above into Eq. (C.3) and rearranging gives

$$\delta v_{\mathrm{E}}(s) = \frac{1}{s^2+\omega_{\mathrm{s}}^2}\left(\delta v_{\mathrm{E}}(0)s + g\phi_{\mathrm{N}}(0) + \delta f_{\mathrm{E}} + \frac{g\delta\omega_{\mathrm{N}}}{s}\right), \tag{C.5}$$

where $\omega_{\mathrm{s}} = \sqrt{g/R}$ is the Schuler frequency with a period of about 84.4 minutes. Using the following Laplace transform pairs

$$\begin{aligned}
\delta v_{\mathrm{E}}(0)\frac{s}{s^2+\omega_{\mathrm{s}}^2} &\iff \delta v_{\mathrm{E}}(0)\cos\omega_{\mathrm{s}}t, && \text{(C.6a)}\\
\frac{g\phi_{\mathrm{N}}(0)+\delta f_{\mathrm{E}}}{s^2+\omega_{\mathrm{s}}^2} &\iff (g\phi_{\mathrm{N}}(0)+\delta f_{\mathrm{E}})\frac{\sin\omega_{\mathrm{s}}t}{\omega_{\mathrm{s}}}, && \text{(C.6b)}\\
g\delta\omega_{\mathrm{N}}\frac{1}{s^2+\omega_{\mathrm{s}}^2}\frac{1}{s} &\iff g\delta\omega_{\mathrm{N}}\int_0^\infty \frac{\sin\omega_{\mathrm{s}}\tau}{\omega_{\mathrm{s}}}u(t-\tau)d\tau \\
&= g\delta\omega_{\mathrm{N}}\int_0^t \frac{\sin\omega_{\mathrm{s}}(t-\tau)}{\omega_{\mathrm{s}}}d\tau = g\delta\omega_{\mathrm{N}}\frac{1-\cos\omega_{\mathrm{s}}t}{\omega_{\mathrm{s}}^2}, && \text{(C.6c)}
\end{aligned}$$

the inverse Laplace transformation of Eq. (C.5) can be written as

$$\delta v_{\mathrm{E}}(t) = \delta v_{\mathrm{E}}(0)\cos\omega_{\mathrm{s}}t + (g\phi_{\mathrm{N}}(0)+\delta f_{\mathrm{E}})\frac{\sin\omega_{\mathrm{s}}t}{\omega_{\mathrm{s}}} + g\delta\omega_{\mathrm{N}}\frac{1-\cos\omega_{\mathrm{s}}t}{\omega_{\mathrm{s}}^2}. \tag{C.7}$$

In a non-accelerating case, the north channel errors can be written as

$$\begin{aligned}
\delta\dot{v}_{\mathrm{N}} &= -g\phi_{\mathrm{E}} + \delta f_{\mathrm{N}}, && \text{(C.8a)}\\
\dot{\phi}_{\mathrm{E}} &= -\frac{1}{R}\delta v_{\mathrm{N}} + \delta\omega_{\mathrm{E}}. && \text{(C.8b)}
\end{aligned}$$

Following similar procedures as applied for the east channel, the north channel error can be obtained as follows:

$$\delta v_{\mathrm{N}}(t) = \delta v_{\mathrm{N}}(0)\cos\omega_{\mathrm{s}}t + (-g\phi_{\mathrm{N}}(0)+\delta f_{\mathrm{N}})\frac{\sin\omega_{\mathrm{s}}t}{\omega_{\mathrm{s}}} - g\delta\omega_{\mathrm{E}}\frac{1-\cos\omega_{\mathrm{s}}t}{\omega_{\mathrm{s}}^2}. \tag{C.9}$$

References

Altmann, S. L. (1986). *Rotations, Quaternions, and Double Groups.* Oxford University Press, Oxford, New York.

Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. I. (2003). An Introduction to MCMC for Machine Learning. *Machine Learning*, 50:5–43.

Benson Jr., D. O. (1975). A Comparison of Two Approaches to Pure-Inertial and Doppler-Inertial Error Analysis. *IEEE Transactions on Aerospace and Electronic Systems*, AES-11(4):447–455.

Bergman, N. (1999). *Recursive Bayesian Estimation: Navigation and Tracking Applications.* PhD thesis, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden.

Bortz, J. E. (1971). A New Mathematical Formulation for Strapdown Inertial Navigation. *IEEE Transactions on Aerospace and Electronic Systems*, AES-7(1):61–66.

Britting, K. R. (1971). *Inertial Navigation Systems Analysis.* John Wiley & Sons, Inc.

Brown, R. G. and Hwang, P. Y. C. (1992). *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley & Sons, Inc., second edition.

Brown, R. G. and Hwang, P. Y. C. (1997). *Introduction to Random Signals and Applied Kalman Filtering: with MatLab® Exercises and Solutions.* John Wiley & Sons, Inc., third edition.

Chiang, K.-W. and El-Sheimy, N. (2002). INS/GPS Integration Using Neural Networks for Land Vehicle Navigation Applications. In *Proceedings of the ION GPS*, pages 535–544, Portland, Oregon.

Crassidis, J. L. and Markley, F. L. (2003). Unscented Filtering for Spacecraft Attitude Estimation. *Journal of Guidance, Control, and Dynamics*, 26(4):536–542.

Doucet, A. (1998). On sequential simulaion-based methods for Bayesian filtering. Technical Report CUED/F-INFENG/TR 310, Cambridge University Engineering Department.

El-Sheimy, N., Abdel-Hamid, W., and Lachapelle, G. (2004). An Adaptive Neuro-Fuzzy Model for Bridging GPS Outages in MEMS-IMU/GPS Land Vehicle Navigation. In *Proceedings of the ION GNSS*, pages 1088–1095, Long Beach, California.

El-Sheimy, N. and Niu, X. (2004). The Development of Low-Cost MEMS-Based IMU for Land Vehicle Navigation Applications. Presentation in GEOIDE Annual Meeting, Gatineau, Québec, Canada (CD).

Farrell, J. A. and Barth, M. (1998). *The Global Positioning System & Inertial Navigation*. McGraw–Hill.

Gelb, A., editor (1974). *Applied Optimal Estimation*. The M. I. T. Press.

Gill, P. E., Golub, G. H., Murray, W., and Saunders, M. A. (1974). Methods for Modifying Matrix Factorizations. *Mathematics of Computation*, 28(126):505–535.

Godsill, S. J., Doucet, A., and West, M. (2000). Monte Carlo Smoothing for Non-Linear Time Series. In *Symposium on Frontiers of Time Series Modelling*, Tokyo, Japan. Institute of Statistical Mathematics.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. *IEE Proceedings Part F: Communications, Radar, and Signal Processing*, 140(2):107–113.

Grewal, M. S. and Andrews, A. P. (1993). *Kalman Filtering: Theory and Practice*. Prentice-Hall, Inc.

Grewal, M. S. and Andrews, A. P. (2001). *Kalman Filtering: Theory and Practice Using MATLAB®*. John Wiley & Sons, Inc.

Haykin, S. (1996). *Adaptive Filter Theory*. Prentice Hall, Upper Saddle River, New Jersey, Third edition.

Hirokawa, R., Nakakuki, K., Sato, K., and Ishihara, R. (2004). Autonomous Vehicle Navigation with Carrier Phase DGPS and Laser-Scanner Augmentation. In *Proceedings of ION GNSS*, pages 1115–1123, Long Beach, California.

Hou, H. and El-Sheimy, N. (2003). Inertial Sensors Errors Modeling Using Allan Variance. In *Proceedings of ION GPS/GNSS*, pages 2860–2867, Portland, Oregon.

Julier, S. J. (2003). The Spherical Simplex Unscented Transformation. In *Proceedings of the IEEE American Control Conference*, pages 2430–2434, Denver, Colorado.

Julier, S. J. and Durrant-Whyte, H. F. (2003). On The Role of Process Models in Autonomous Land Vehicle Navigation Systems. *IEEE Transactions on Robotics and Automation*, 19(1):1–14.

Julier, S. J. and LaViola, J. J. (2003). Kalman Filtering Under Nonlinear Constraints. Unpublished.

Julier, S. J. and Uhlmann, J. K. (1996). A General Method for Approximating Nonlinear Transformations of Probability Distributions. Technical report, Department of Engineering Science, University of Oxford, Oxford, OX1 3PJ UK.

Julier, S. J. and Uhlmann, J. K. (2002a). Reduced Sigma Point Filters for the Propagation of Means and Covariances through Nonlinear Transformations. In *Proceedings of the IEEE American Control Conference*, pages 887–892, Anchorage AK, USA.

Julier, S. J. and Uhlmann, J. K. (2002b). The Scaled Unscented Transformation. In

Proceedings of the IEEE American Control Conference, pages 4555–4559, Anchorage AK, USA.

Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F. (1995). A New Approach for Filtering Nonlinear Systems. In *Proceedings of the American Control Conference*, pages 1628–1632.

Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F. (2000). A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482.

Kitagawa, G. (1996). Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, 5(1):1–25.

Kitagawa, G. and Higuchi, T. (2001). Preface: Special Issue on Nonlinear Non-Gaussian Models and Related Filtering Methods. *Annals of the Institute of Statistical Mathematics*, 53(1):iii.

Kraft, E. (2003). A Quaternion-Based Unscented Kalman Filter for Orientation Tracking. In *Proceedings of Fusion*, Cairns, Australia.

Kuipers, J. B. (1999). *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality.* Princeton University Press, Princeton, New Jersey.

Lefebvre, T., Bruyninckx, H., and de Schutter, J. (2002). Comment on “A New

Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators". *IEEE Transactions on Automatic Control*, 47(8):1406–1408.

Lerro, D. and Bar-Shalom, Y. (1993). Tracking with Debiased Consistent Converted Measurements Versus EKF. *IEEE Transactions on Aerospace and Electronic Systems*, 29(3):1015–1022.

Liu, J. S. and Chen, R. (1995). Blind Deconvolution via Sequential Imputations. *Journal of the American Statistical Association*, 90(403):567–576.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo Methods for Dynamic Systems. *Journal of the American Statistical Association*, 93(443):1022–1031.

Maybeck, P. S. (1994a). *Stochastic Models, Estimation, and Control: Volume 1*. Navtech Book & Software Store.

Maybeck, P. S. (1994b). *Stochastic Models, Estimation, and Control: Volume 2*. Navtech Book & Software Store.

McGreevy, J. (1986). *Fundamentals of Strapdown Inertial Navigation*. Litton Guidance & Control Systems. Document No. 406372, Revision A.

Meditch, J. S. (1969). *Stochastic Optimal Linear Estimation and Control*. McGraw-Hill.

Miller, R. B. (1983). A New Strapdown Attitude Algorithm. *Journal of Guidance, Control, and Dynamics*, 6(4):287–291.

Niu, X. and El-Sheimy, N. (2004). Testing an Integrated GPS, MEMS-IMU, and Odometer System for Land Vehicle Navigation. Internal Report, Mobile Multi-Sensor Research Group, The University of Calgary.

Nørgaard, M., Poulsen, N. K., and Ravn, O. (2000). New Developments in State Estimation of Nonlinear Systems. *Automatica*, 36:1672–1638.

Pennec, X. (1998). Computing the Mean of Geometric Features: Application to the Mean Rotation. Technical Report Rapport de recherche n° 3371, Institut National de Recherche en Informatique et en Automatique, Sophia Antipolis Cedex, France.

Powell, T. D. (2002). Automated Tuning of an Extended Kalman Filter Using the Downhill Simplex Algorithm. *Journal of Guidance, Control, and Dynamics*, 25(5):901–908.

Reif, K., Günther, S., Yaz, E., and Unbehauen, R. (1999). Stochastic Stability of the Discrete-Time Extended Kalman Filter. *IEEE Transactions on Automatic Control*, 44(4):714–728.

Rogers, R. M. (1997). IMU In-Motion Alignment Without Benefit of Attitude Initialization. *Navigation: Journal of The Institute of Navigation*, 44(3):301–311.

Rogers, R. M. (2000). *Applied Mathematics in Integrated Navigation Systems.* American Institute of Aeronautics and Astonautics, Inc.

Savage, P. G. (2000). *Strapdown Analytics: Part 1.* Strapdown Associates, Inc., Maple Plain, Minnesota.

Scherzinger, B. M. (1996). Inertial Navigator Error Models for Large Heading Uncertainty. In *IEEE Position Location and Navigation Symposium*, pages 477–484.

Scherzinger, B. M. (2002). INS Error Models (Unpublished). Applanix Corporation.

Scherzinger, B. M. (2004). *Estimation with Application to Navigation: Lecture Notes ENGO 699.11.* Dept. of Geomatics Eng., The University of Calgary, Calgary, Canada.

Scherzinger, B. M. and Reid, D. B. (1994). Modified Strapdown Inertial Navigator Error Models. In *IEEE Position Location and Navigation Symposium*, pages 426–430, Las Vegas, Nevada.

Schwarz, K.-P. (1999). *Fundamentals of Geodesy: Lecture Notes ENGO 421.* Dept. of Geomatics Eng., The University of Calgary, Calgary, Canada.

Schwarz, K.-P. and Wei, M. (2000). *INS/GPS Integration for Geodetic Applications: Lecture Notes ENGO 623.* Dept. of Geomatics Eng., The University of Calgary, Calgary, Canada.

Shin, E.-H. (2001). Accuracy Improvement of Low Cost INS/GPS for Land Applications. UCGE Reports Number 20156, The University of Calgary, Calgary, Alberta, Canada.

Shin, E.-H. (2004). A Quaternion-Based Unscented Kalman Filter for the Integration of GPS and MEMS INS. In *Proceedings of ION GNSS*, pages 1060–1068, Long Beach, CA.

Shin, E.-H. and El-Sheimy, N. (2004). An Unscented Kalman Filter for In-Motion Alignment of Low-Cost IMUs. In *Proceedings of IEEE Position, Location, and Navigation Symposium*, pages 273–279, Monterey, CA.

Shuster, M. D. (1993). A Survey of Attitude Representations. *The Journal of the Astronautical Sciences*, 41(4):439–517.

Stewart, R. W. and Chapman, R. (1990). Fast Stable Kalman Filter Algorithms Utilising the Square Root. In *Proceedings of the IEEE International Conference on Acustics, Speech, and Signal Processing*, pages 1815–1818, Albuquerque, New Mexico.

Sukkarieh, S. (2000). *Low Cost, High Integrity, Aided Inertial Navigation Systems for Autonomous Land Vehicles*. Ph.D. Thesis, Australian Centre for Field Robotics, Dept. of Mechanical and Mechatronic Engineering, The University of Sydney, Sydney, Australia.

Tehrani, M. M. (1983). Ring Laser Gyro Data Analysis with Cluster Sampling Technique. In *Proceedings of SPIE*, volume 412, pages 207–220.

Titterton, D. H. and Weston, J. L. (1997). *Strapdown Inertial Navigation Technology*. Peter Peregrinus Ltd.

Van der Merwe, R., Doucet, A., de Freitas, N., and Wan, E. (2000). The Unscented Particle Filter. Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department.

Wan, E. A. and van der Merwe, R. (2001). *Kalman Filtering and Neural Networks, Haykin, S. (Ed.)*, chapter 7. John Wiley & Sons, New York.