Kenneth Gade

Inertial Navigation — Theory and Applications

Thesis for the degree of Doctor Philosophiae

Trondheim, January 2018

Norwegian University of Science and Technology
Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Doctor Philosophiae

Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

ISBN 978-82-326-2870-4 (printed version)
ISBN 978-82-326-2871-1 (electronic version)
ISSN 1503-8181

ITK-report: 2018-1-W

Doctoral theses at NTNU, 2018:38

Printed by Skipnes Kommunikasjon as

To Helene, Alida and Linnea

Summary

Navigation is defined as the process of estimating the six degrees of freedom. We have seen an increased demand for navigation the last decade, and important reasons for the growth are the increased availability of low cost inertial measurement units (IMUs) and global navigation satellite system (GNSS) receivers, and the increased use of autonomous vehicles.

When working with navigation in general, and when designing and implementing navigation systems in particular, a precise notation system is of utmost importance. Kinematical quantities such as velocity, acceleration, orientation, and angular velocity must be unambiguously specified both in documentation and program code. Five properties of a good notation system are identified, and a notation system fulfilling these properties is presented. The notation system includes a usage of sub- /superscripts that follow simple rules when the equations are correct, and hence the system contributes strongly to correct deductions and implementations. The sub- /superscripts and unambiguousness also lead to better understanding of quantities such as linear velocity, and misunderstandings/errors during exchange of code and/or equations are greatly reduced.

Position calculations are a central part of any navigation system, and common concerns are imprecise calculations (e.g. when using an ellipsoidal Earth model or when using map projections), complex implementations, and singularities. In addition, separating the horizontal and vertical position is often desirable. By representing horizontal position with the normal vector to the Earth ellipsoid (called *n*-vector) this separation is kept, while avoiding common problems with other such representations, e.g. the singularities and discontinuity of latitude/longitude and the distortion of map projections. Further, since the *n*-vector is a 3D vector, the powerful vector algebra can be used to solve many calculations intuitively and with few code lines. A code library solving many of the most common position calculations using *n*-vector has been made available for download (for several programming languages).

Estimating heading with sufficient accuracy is often the most challenging part when designing a low cost navigation system, and the necessary theory to support this task has not been available, making it even more challenging. A study of the theory behind heading estimation has thus been made, and based on this theory, different methods to find heading have been categorized by means of consistent mathematical principles. Using this categorization system, we have identified seven different methods to find heading for practical navigation systems. The methods are magnetic and gyrocompass, two methods based on observations, multi-antenna GNSS, and two methods based on vehicle motion. With the aid of this theory and list of methods, designing navigation systems where heading is a challenge can now be done with full understanding and insight into the task. The possible ways to find heading for a given system are immediately identified, and no method is overlooked.

During navigation research and development, the support of appropriate software is vital. The aim to design one common software solution for a range of different navigation tasks was the motivation behind the development of a tool called NavLab. Important areas of usage include research and development, simulation studies, post-processing of logged sensor data, sensor evaluation, and decision basis for sensor purchase and mission planning. It has turned out that a generic design and implementation is feasible, and NavLab has been used to navigate a variety of different maritime, land and air vehicles. Users include research groups, commercial companies, military users and universities.

For underwater navigation, and in particular for autonomous underwater vehicles (AUVs), several different techniques have been used in NavLab to reduce the horizontal position estimation uncertainty. When feasible, the underwater vehicle can go to the surface for a GNSS fix, or be followed by a surface vehicle that combines GNSS with acoustic positioning and transmits the result. However, in practice an AUV must often handle long periods without position aid, and thus the drift of the core navigation system is of great importance. This core system often consists of an IMU and a Doppler velocity log (DVL), where the DVL is usually the most important sensor to limit the drift. In cases of DVL dropouts, the use of a vehicle model in the estimator significantly reduces the position drift, compared to a system in free inertial drift. This is the case even with high-end IMUs. For low-cost systems without a DVL, a vehicle model is vital, and it can also be used together with a DVL to improve the navigation system integrity.

Position drift can be avoided altogether by deploying one or more underwater transponders that provide range measurements to the underwater vehicle. We have developed a method where accurate position is estimated by means of only one single transponder. The method is implemented in NavLab, and it has demonstrated a position accuracy which is close to the performance achieved when the AUV is followed by a surface ship with acoustic positioning.

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) in partial fulfilment of the requirements for the degree of *Doctor Philosophiae*. The thesis contains eight research papers, which will be referenced as **Paper I** through **Paper VIII** (listed in Section 1.2).

The writing of the papers and this thesis has been done during my employment at the Norwegian Defence Research Establishment (FFI). My master thesis (about inertial navigation) was written for FFI during the autumn of 1996, and I started working at FFI directly after that. My main task at FFI has been the development of aided inertial navigation systems, first for the HUGIN AUVs, and later for a range of different applications at sea, on land and in air. We have also worked closely with the industry, and our navigation technology is being used in several commercial products.

At FFI, our highest priority is to develop and implement high performance navigation systems, while the writing of publications is prioritized lower. In addition, much of the developed technology and results cannot be published since it is either “business confidential” or military classified. Still, there has been time for some publications, and collecting several of them for a Dr.Philos. (Doctor Philosophiae) thesis seemed in my case to be the best way to obtain a doctoral degree while working at FFI. The main difference from a Ph.D. is that a Dr.Philos. is without supervision and outside an organized Ph.D.-program.

Acknowledgements

First, I would like to thank Bjørn Jalving, who employed me at FFI to develop the HUGIN navigation system, and this was the start of today’s navigation group at FFI. Bjørn was the ideal leader, always very inspiring, encouraging and interested in my work, and he supported me 100% when I wanted to develop a more general navigation system rather than a system dedicated to AUVs only. We also shared the same goal of building a larger navigation group, which could solve a range of different tasks within navigation, for many different applications. Bjørn was the leader of the navigation group until he started working for Kongsberg Maritime in 2006.

A few years after I was employed at FFI, the group started growing, and soon I had several colleagues who also became deeply immersed in the exciting topic of navigation. I would like to thank you all (in alphabetical order); Einar Berglund, Ove Kent Hagen, Magne Mandt, Kristian Svartveit and Kjetil Bergh Ånonsen, for all your valuable contributions into the group. It is amazing to look back at what we have accomplished together as a team. Not only are you very skilled professionals within navigation, you are also great colleagues and an important reason for me looking forward to go to work every day. I am also very grateful to the other good colleagues at FFI, and the HUGIN team in particular, for making FFI a great place to work.

Outside FFI, Kongsberg Maritime has been our most important partner, having made several commercial products from our navigation technology and contributing to the navigation development as well. I am grateful to our colleagues at Kongsberg Maritime for our excellent collaboration, and it is clearly inspiring to see that you bring our technology out to customers worldwide. In addition, I would like to express my gratitude to Professor Oddvar Hallingstad at the University of Oslo — I have really appreciated our long and interesting discussions about mathematics and notation.

Finally, I would thank the persons who are the most important to me; my closest family. I am immensely grateful for all your support through all these years.

Contents

List of Figures

List of Tables

List of abbreviations

Abbreviation	**Explanation**
AR	Augmented reality
AUV	Autonomous underwater vehicle
DPCA	Displaced phase-center antenna
DVL	Doppler velocity log
ECEF	Earth-centered-earth-fixed
FFI	Norwegian Defence Research Establishment (in Norwegian: Forsvarets forskningsinstitutt)
FOG	Fiber optic gyro
GNSS	Global navigation satellite system
LBL	Long baseline
MEMS	Microelectromechanical systems
MRU	Motion reference unit
IMU	Inertial measurement unit
ROV	Remotely operated vehicle
RLG	Ring laser gyro
SAS	Synthetic aperture sonar
USBL	Ultra-short baseline
UTM	Universal Transverse Mercator
UAV	Unmanned aerial vehicle
UGV	Unmanned ground vehicle
USV	Unmanned surface vehicle
WGS-72	World Geodetic System 1972
WGS-84	World Geodetic System 1984

Chapter 1

Introduction

Several definitions of the term *navigation* exist, but here we will define navigation as the process of estimating the six degrees of freedom[1] (including their derivatives) of a rigid body (i.e. any vehicle or device). The uncertainties of the estimates are often also part of the output from the navigation, and the navigation can be performed in real time, or in post processing.

1.1 Increased demand for navigation

The need for navigation in a wide range of applications is well known. However, it is interesting to observe that we experience an *increase* in the demand for navigation. We have seen an increased demand for navigation systems over the last decade, both in the civilian industry and in the military, and there are at least four reasons for this.

- **The availability of key navigation sensors has increased:** The development of microelectromechanical systems (MEMS) inertial measurement units (IMUs) has led to the availability of navigation systems that are inexpensive, small, with low weight and low power consumption. Global navigation satellite system (GNSS) receivers have also become lighter, smaller and cheaper, and it is now feasible to make navigation systems for many more applications than before, e.g. for cameras, small low-cost vehicles or personnel.
- **Increased use of autonomous vehicles:** The increased use of unmanned and autonomous vehicles gives increased demand for navigation systems for two reasons. Firstly, with a human (pilot, driver etc.) on board, several types of vehicles did not

[1] I.e. position and orientation in three-dimensional space (three degrees of freedom each).

need a navigation system, but when replacing the human with an automated system, a navigation system is usually required. Secondly, the removal of the humans often means that the number and variety of vehicles can be increased, with an increased demand for navigation systems as a result.

- **Imaging sensors get higher resolution:** The development within cameras and (synthetic aperture) sonars and radars has given significantly better resolution of their images. When georeferencing the images from these sensors, the required accuracy of the georeferencing is typically given by the resolution, resulting in an increased demand for high accuracy navigation of the sensor platform.
- **More processing power available:** The fourth reason we have identified that is leading to an increased demand for navigation development, is the rapid growth of computer processing power. With more processing power available, complex and computer intensive navigation algorithms are becoming feasible. One example is the use of one or more cameras attached to the navigating vehicle, imaging Earth-fixed features. With enough processing power, the movement of the features in successive images can be observed and/or features can be recognized, giving valuable input to the navigation system. Also for other sensors, such as IMUs and Doppler velocity logs, advanced navigation algorithms with multiple states and complex error models can be implemented, giving higher navigation accuracy at the cost of computing power. In general, we have seen an increased number of requests to design navigation systems where low hardware cost is a high priority, and the required navigation accuracy is achieved by developing complex and computer intensive navigation algorithms.

1.2 List of publications

The following eight research papers, denoted **Paper I** through **Paper VIII**, are included in this thesis:

Paper I Gade, K. (2010). A Nonsingular Horizontal Position Representation. *The Journal of Navigation*, vol. 63, no. 3, pp. 395-417. doi:10.1017/ S0373463309990415

Paper II Gade, K. (2016). The Seven Ways to Find Heading. *The Journal of Navigation*, vol. 69, no. 5, pp. 955-970. doi:10.1017/S0373463316000096

Paper III Gade, K. and Jalving, B. (1999). An Aided Navigation Post Processing Filter for Detailed Seabed Mapping. *Modeling, Identification and Control*, vol. 20, no. 3, pp. 165-176. doi: 10.4173/mic.1999.3.2. First published in Proceedings of AUV '98, Cambridge, MA, USA, Aug. 20-21, 1998

Paper IV Gade, K. (2005). NavLab, a Generic Simulation and Post-processing Tool for Navigation. *Modeling, Identification and Control*, vol. 26, no. 3, pp. 135-150. doi: 10.4173/mic.2005.3.2. First published in European Journal of Navigation, vol. 2, no. 4, pp. 51-59, 2004

Paper V Jalving, B., Gade, K., Hagen, O. K. and Vestgård, K. (2004). A Toolbox of Aiding Techniques for the HUGIN AUV Integrated Inertial Navigation System. *Modeling, Identification and Control*, vol. 25, no. 3, pp. 173-190. doi: 10.4173/mic.2004.3.3. First published in Proceedings from MTS/IEEE Oceans 2003, San Diego, CA, USA

Paper VI Jalving, B., Gade, K., Svartveit, K., Willumsen, A. and Sørhagen, R. (2004). DVL Velocity Aiding in the HUGIN 1000 Integrated Inertial Navigation System. *Modeling, Identification and Control*, vol. 25, no. 4, pp. 223-236. doi: 10.4173/mic.2004.4.2. First published in proceedings from ADCPs in Action 2004, Nice, France

Paper VII Hegrenæs, Ø., Hallingstad, O. and Gade, K. (2007). Towards Model-Aided Navigation of Underwater Vehicles. *Modeling, Identification and Control*, vol. 28, no. 4, pp. 113-123. doi:10.4173/mic.2007.4.3. First published in proceedings of UUST '07, Durham, USA, 2007

Paper VIII Hegrenæs, Ø., Gade, K., Hagen, O. K. and Hagen, P. E. (2009). Underwater Transponder Positioning and Navigation of Autonomous Underwater Vehicles. *Proceedings of the IEEE Oceans Conference*, Biloxi, MS, USA

The papers are included at the end of this thesis (from page 55).

In addition to the papers included, the author has co-authored the following publications on the topic of inertial navigation:

- Kjørsvik, N., Gjevestad, J. G., Brøste, E., Gade, K. and Hagen, O.K. (2010). Tightly Coupled Precise Point Positioning and Inertial Navigation Systems. *Proceedings from EuroCOW 2010*, European Spatial Data Research, February 2010, Castelldefels, Spain.
- Hansen, R. E., Sæbø, T. O., Gade, K. and Chapman, S. (2003). Signal Processing for AUV Based Interferometric Synthetic Aperture Sonar. *Proceedings of the MTS/IEEE Oceans 2003*, vol. 5, pp. 2438-2444. San Diego, CA, USA
- Jalving, B., Bovio, E. and Gade, K. (2003). Integrated inertial navigation systems for AUVs for REA applications. *Proceedings from MREP 2003*, NATO SACLANT Undersea Research Centre, May 2003, Italy.
- Nypan, T., Gade, K. and Hallingstad, O. (2002). Vehicle positioning by database comparison using the Box-Cox metric and Kalman filtering. *Proceedings of the VTC Spring*, pp. 1650-1654, AL, USA
- Wang, L., Bellettini, A., Hollett, R., Tesei, A., Pinto, M., Chapman, S. and Gade, K. (2001). InSAS'00: Interferometric SAS and INS aided SAS imaging. *Proceedings of the MTS/IEEE Oceans 2001*, pp. 179-187, Honolulu, HI, USA
- Nypan, T., Gade, K. and Maseng, T. (2001). Location using Estimated Impulse Responses in a Mobile Communication System. *Proceedings of the NORSIG,* Trondheim, Norway
- Hagen, P. E., Hansen, R. E., Gade, K. and Hammerstad, E. (2001). Interferometric Synthetic Aperture Sonar for AUV Based Mine Hunting: The SENSOTEK project. *Proceedings from Unmanned Systems 2001*, Baltimore, MD, USA
- Mandt, M., Gade, K. and Jalving, B. (2001). Integrating DGPS-USBL position measurements with inertial navigation in the HUGIN 3000 AUV. *Proceedings of the 8th Saint Petersburg International Conference on Integrated Navigation Systems*, pp. 63-74. Saint Petersburg, Russia
- Hafskjold, B., Jalving, B., Hagen, P. E. and Gade, K. (2000). Integrated Camera-Based Navigation. *The Journal of Navigation*, vol. 53, no. 2, pp. 237-245
- Jalving, B. and Gade, K. (1998). Positioning Accuracy for the HUGIN Detailed Seabed Mapping UUV. *Proceedings of the IEEE Oceans '98*, pp. 108-112. Nice, France

1.3 Thesis structure

A unified notation system that is used throughout this thesis and in all the included publications is presented in Chapter 2. In Chapter 3, two fundamental topics within navigation are discussed. First, position calculations and an alternative representation for horizontal position are presented. The second topic is heading estimation, where seven different methods to find heading are defined. Chapter 4 introduces a general navigation software tool called NavLab. Finally, underwater navigation is the topic of Chapter 5, where different ways to limit the positional drift is the main focus.

The topics of the eight included papers (listed in Section 1.2) are covered from Chapter 3 to Chapter 5, and the list below shows the main connection between each of the papers and the chapters/sections.

1. Introduction
2. A Unified Notation for Kinematics
3. Fundamental Topics within Navigation
    - 3.1. Position calculations **(Paper I)**
    - 3.2. Heading estimation **(Paper II)**
        - 3.2.1. Example **(Paper III)**
        - 3.2.2. Usage of the list of methods
4. General Navigation Software **(Paper IV)**
5. Underwater Navigation **(Paper V)**
    - 5.1. Core underwater navigation system **(Paper VI)**
        - 5.1.1. Aiding with a vehicle model **(Paper VII)**
        - 5.1.2. Velocity measurements from a sonar array
    - 5.2. Acoustic positioning from a surface ship
    - 5.3. Range from underwater transponders **(Paper VIII)**
    - 5.4. Terrain referenced navigation

1.4 Contributions

The main contributions of this thesis are the following:

Chapter 2
(Notation system)

Developed a unified, stringent and unambiguous notation system

The importance of the notation system used when working with navigation is often underestimated. Hence, five properties of a good notation system are identified, and a notation system fulfilling the five properties is presented. The system is unambiguous, and it includes mechanisms to ensure correct deductions and correct implementations in program code. It also improves the understanding and greatly reduces the chance for errors when exchanging code and/or equations. The notation system is an important foundation for the remainder of the thesis; more details are given in Chapter 2.

Paper I
(*n*-vector)

Introduced a non-singular position representation that simplifies many of the common position calculations

Common concerns for position calculations have been imprecise calculations (e.g. when using an ellipsoidal Earth model or when using map projections), complex implementations, and singularities. In addition, separating the horizontal and vertical position is often desired. By representing horizontal position with *n*-vector, this separation is kept, while avoiding common problems with other such representations, e.g. the singularities and discontinuity of latitude/longitude and the distortion of map projections. Further, since the *n*-vector is a 3D vector, the powerful vector algebra can be used to solve many calculations intuitively and with few code lines (i.e. solutions to common position calculations, that are exact, simple to implement and valid for all Earth positions, are found). For more details, see **Paper I**. A web-page with a simplified presentation and a downloadable code library is also available, as described in Section 3.1.

Paper II
(Heading estimation)

Introduced new fundamental theory for heading estimation, defining the possible ways to find heading

In low cost navigation systems, the greatest challenge is often the heading accuracy, since magnetic compasses typically are too inaccurate for the purpose. A general theory of heading estimation is presented, and based on consistent mathematical principles, seven different methods to find heading are defined. The theory and list of methods has turned out to be a game changer when it comes to the design of navigation systems where heading is a challenge. For a given system, the possible ways to find heading are now immediately identified, and we can confidently determine which sensors to add and what maneuvers are required to fulfill the heading requirement. For more details, see **Paper II** and Section 3.2.

Paper III
(Dedicated navigation system)

Designed and implemented a dedicated navigation system

An autonomous underwater vehicle (AUV) underwater navigation system without access to raw inertial data was designed. Only a low cost IMU was available, and heading was found by utilizing the velocity vector (which corresponds to Method 6 to find heading when using the list of methods in **Paper II**). The performance of the navigation system was verified using recorded data, as described in **Paper III**.

Paper IV
(NavLab)

Designed and implemented NavLab (general navigation software)

In **Paper IV** it is shown how one generic and flexible tool can be designed to solve a variety of different navigation tasks. The advantages achieved by the use of smoothing are discussed and demonstrated, and different ways to verify estimator performance are presented. Following the suggested design, a general navigation simulation and post-processing tool, called NavLab, is developed. NavLab is used for a range of different purposes, by international industry, military, research groups and academia. For more details, see Chapter 4.

Paper V
(Underwater navigation techniques)

Developed and implemented several underwater navigation techniques

Several different techniques for aiding inertial underwater navigation systems are developed, and **Paper V** gives an overview of the strengths and weaknesses of these techniques. The paper also describes how to combine the techniques in various typical AUV-scenarios, and their performances are demonstrated in HUGIN AUV missions. Chapter 5 contains more details on this topic.

Paper VI (Doppler velocity log)

Analyzed Doppler velocity log error contributions in theory and by using recorded data

The Doppler velocity log (DVL) is often the most important sensor for limiting the drift in an underwater navigation system. In **Paper VI** the DVL error sources, and how they contribute to the total error, are studied, both in theory and by the use of recorded data.

Paper VII (Vehicle model)

Aided the underwater navigation with a vehicle model

Including a vehicle model improves the robustness, integrity and in some cases the accuracy of an underwater navigation system. **Paper VII** presents this aiding technique and it includes AUV-results showing the navigation performance for cases of DVL-dropouts or low DVL-rate.

Paper VIII (Range measurements)

Developed a method that achieves accurate position by using range measurements from a single transponder

In classical long baseline (LBL) systems, several transponders within range are needed to calculate the vehicle position. A method is developed that can estimate accurate position by means of one transponder only (several transponders can also be used, which improves the accuracy further). The accuracy is achieved by integrating the range measurements tightly with the core navigation system, and utilizing the vehicle movement. High accuracy (and robustness) has been demonstrated repeatedly, see **Paper VIII**.

Chapter 2

A Unified Notation for Kinematics

In a practical navigation system, there are usually multiple available sensors, with different positions and orientations, measuring different quantities. Based on this input, the calculated navigation output is often needed for high accuracy applications, such as georeferencing recorded data (e.g. images from camera, sonar or radar). To fulfil the high standards for accuracy, it is of utmost importance to first be able to precisely describe the input measurements, and then continue to use precise descriptions throughout the estimation process. Finally, the output, i.e. the estimates from the navigation, must also be precisely described and well defined to be used correctly. To obtain these precise descriptions, an unambiguous and consistent notation for kinematics is needed.

Section 2.1 will present some important properties for a good notation system, and then a notation system fulfilling the requirements is presented, by first introducing some basic concepts in Section 2.2. The basic concepts form the theoretical foundation for the notation system, and Section 2.3 presents the suggested notation system, while notation rules are given in Section 2.5.

2.1 Properties of a good notation system

After more than twenty years of navigation system development, our experience is that it is difficult to overstate the importance of the notation system. We have identified five properties that a good notation system should have:

1. Any quantity/equation should be unambiguous on its own, i.e. it should be possible to understand precisely what it expresses without having to read additional text. This property is very important both for written publications and computer programs, since ambiguities typically lead to errors in equations and implementations. When errors are

made, the writer and/or the readers normally do not fully understand the *precise* meaning of a quantity.

2. The notation must clearly indicate all coordinate frames that are involved in a particular quantity (e.g. for an angular velocity it must be clear which frame is rotating relative to which reference).
3. The notation should have an inherent “mechanism” to avoid errors in equations. Usually this is achieved by means of sub- /superscripts that follow simple rules when the quantities are used correctly.
4. The notation should be able to specify if it is the *position* or *orientation* (or both) of a coordinate frame that matters. In most cases either the position or the orientation is significant, but in some cases both are significant, which e.g. is the case for one of the coordinate frames involved in a standard (linear) velocity. This is the reason why linear velocity is often not fully understood, and errors often are made. A notation that is able to distinguish between the three variants *position only*, *orientation only*, and *position and orientation* of a coordinate frame, makes it possible to improve the understanding of the quantities and to describe the kinematical relations very precisely.
5. The notation should include coordinate-free (also called component-free or geometrical) vectors. Since most relations do not depend on the coordinate frame in which the vectors are decomposed/resolved, such information is redundant and obscures the relevant relation.

A notation system that fulfils these five properties has been developed over the years, by considering the efficiency and precision both in theoretical works and in practical implementations. The system and its basic concepts are presented in the following.

2.2 Basic concepts

We define a *particle* to be a physical object whose size can be neglected, and thus a *given particle* uniquely defines a *position* in the three dimensional space. When establishing a mathematical model of our world, the particle will be represented by a *point*, denoted $\underset{\cdot}{X}$ (the reason for using a capital letter for a point will be clear in Section 2.2.1). The point is an element of an affine space, denoted $\mathbb{A}$, i.e. $\underset{\cdot}{X} \in \mathbb{A}$. Any affine space is associated with a *vector space* (Crampin and Pirani, 1986), denoted $\mathbb{V}$. Vectors are denoted $\vec{x}$, where $\vec{x} \in \mathbb{V}$. A vector defines *direction* and *magnitude* in the mathematical model. Note that the vectors are coordinate-free (also called geometrical), i.e. they define direction/magnitude in the mathematical model with no reference to other quantities (decomposed/resolved vectors will be discussed in Section 2.2.2.). Coordinate-free vectors are frequently used in the literature, see e.g. Britting (1971) and McGill and King (1995).

The basic operations defined for an affine space and the associated vector space are

- difference between two points, giving a vector in the associated vector space, e.g. $\underset{\cdot}{X} - \underset{\cdot}{Y} = \vec{z}$.
- addition of a point and a vector, giving a new point, e.g. $\underset{\cdot}{Y} + \vec{z} = \underset{\cdot}{X}$.

Note that a point represents position without any reference, and can thus be said to represent *absolute position*. This is in the same manner as a given particle (or a specified position at a given physical object), uniquely defines a position in the physical world. Similarly, a coordinate-free vector defines direction and magnitude without any reference.

2.2.1 Coordinate frame

We define a *rigid body* as a collection of particles whose distances relative to each other are constant (according to the needed accuracy of the model in use). This collection of particles defines position and orientation (with six degrees of freedom).

A representation of a rigid body that, in this setting, is more convenient than a collection of points, is a *coordinate frame*. A coordinate frame is defined as a combination of the following:

- A point, defining the position of the coordinate frame, also called the *origin* of the coordinate frame.
- 3 linearly independent vectors, defining the orientation of the coordinate frame. The vectors have fixed lengths, fixed relative directions, a defined order, and are called the basis vectors of the coordinate frame.

We see that the coordinate frame has six degrees of freedom as desired. A capital underlined letter, e.g. $\underline{B}$, is used to represent a coordinate frame. Even though a coordinate frame can represent a physical rigid body, it is not restricted to this use, and it is often convenient to introduce several coordinate frames in the mathematical model in addition to those corresponding to rigid bodies (e.g. a North-East-Down coordinate frame).

There will be cases where it is useful to treat and denote the position and orientation of a coordinate frame $\underline{B}$ separately. The *position* of $\underline{B}$, i.e. its origin, is denoted $\underset{\cdot}{B}$ (the bar is replaced with a dot). $\underset{\cdot}{B}$ is simply a point, i.e. an element of an affine space, $\underset{\cdot}{B} \in \mathbb{A}$. $\underline{B}$'s *orientation* is represented by letting an arrow replace the bar, i.e. $\underset{\rightarrow}{B}$, and hence this symbol represents the basis vectors. Assuming the basis vectors are given by the tuple $\left(\vec{b}_{\underset{\rightarrow}{B},1}, \vec{b}_{\underset{\rightarrow}{B},2}, \vec{b}_{\underset{\rightarrow}{B},3}\right)$, we have

$$\underset{\rightarrow}{B} = \left(\vec{b}_{\underset{\rightarrow}{B},1}, \vec{b}_{\underset{\rightarrow}{B},2}, \vec{b}_{\underset{\rightarrow}{B},3} \right) \in \mathbb{V}^3, \tag{2.1}$$

where $\mathbb{V}^3 = \mathbb{V} \times \mathbb{V} \times \mathbb{V}$, and $\times$ indicates the Cartesian product of sets (Munkres, 2000). Since a coordinate frame $\underline{B}$ consists of both a point and basis vectors, we have

$$\underline{B} = \left(\underset{\cdot}{B}, \underset{\rightarrow}{B} \right) \in \mathbb{A} \times \mathbb{V}^3. \tag{2.2}$$

The possibility to specify the position and orientation of a coordinate frame independently gives a compact notation to specify relations between two coordinate frames. E.g. if two coordinate frames $\underline{A}$ and $\underline{B}$ have different origins, this is expressed by

$$\underset{\cdot}{A} \neq \underset{\cdot}{B} \tag{2.3}$$

(while (2.3) says nothing about their relative orientation). The relation between two coordinate frames will often change as a function of time, and hence the frame relations will typically include time specifications. E.g. if coordinate frames $\underline{A}$ and $\underline{B}$ have the same orientation at time t_1 and the same position (origin) at time t_2, this can be expressed as

$$\underset{\rightarrow}{A}(t_1) = \underset{\rightarrow}{B}(t_1)\,,\ \underset{\cdot}{A}(t_2) = \underset{\cdot}{B}(t_2)\,. \tag{2.4}$$

Another example is when two coordinate frames always have the same orientation, e.g. if a platform ($\underline{B}$) is aligned and stabilized relative to a reference ($\underline{A}$) (while their *translational* relation (their relative position, velocity and acceleration) is unspecified). In that case we have

$$\underset{\rightarrow}{B}(t) = \underset{\rightarrow}{A}(t)\ \forall t \in \mathbb{R}\,. \tag{2.5}$$

2.2.2 Decomposed vectors

If the basis vectors of $\underline{A}$ are given by $\underset{\rightarrow}{A} = \left(\vec{b}_{\underset{\rightarrow}{A},1}, \vec{b}_{\underset{\rightarrow}{A},2}, \vec{b}_{\underset{\rightarrow}{A},3} \right)$, we have that the general vector $\vec{x}$ can be expressed as a linear combination of the basis vectors

$$\vec{x} = x_1 \vec{b}_{\underset{\rightarrow}{A},1} + x_2 \vec{b}_{\underset{\rightarrow}{A},2} + x_3 \vec{b}_{\underset{\rightarrow}{A},3}\,, \tag{2.6}$$

where $x_i, i \in \{1,2,3\}$, are three scalars. The vector $\vec{x}$ *decomposed* (or resolved/represented) in $\underset{\rightarrow}{A}$ can now be expressed as

$$\boldsymbol{x}^{\underset{\rightarrow}{A}} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{2.7}$$

$\boldsymbol{x}^{\underline{A}}$ is called a *decomposed vector* (also sometimes called a coordinate vector or an algebraic vector). In contrast to coordinate-free vectors, decomposed vectors are well suited for computer implementation.

Coordinate-free vectors will be preferred in all expressions and relations, since the frame of decomposition normally does not affect the general expression and thus is redundant information. In a deduction for example, coordinate-free vectors will be used, and the final answer will be decomposed in a selected coordinate frame only if the equation shall be implemented in a computer program.

2.3 Position, orientation and their derivatives

When working with position and orientation (and their derivatives) in practice, *relative* quantities are normally used; i.e. we are expressing a position or orientation of one coordinate frame relative to another (where one can be thought of as a reference). Thus the position, velocity, orientation etc. defined below are all relative quantities, and the right subscript will always specify the two coordinate frames involved. For instance a general relation x, depending on the relative position and orientation between $\underline{A}$ and $\underline{B}$ will be denoted $x_{\underline{A}\underline{B}}$.

2.3.1 Position

Absolute position is represented by a point, while relative position is defined by a *point difference*. The position of coordinate frame $\underline{B}$ relative to $\underline{A}$ is defined by the vector created by subtracting the point $\underset{\cdot}{A}$ from $\underset{\cdot}{B}$ in the affine space,

$$\boxed{\vec{p}_{\underset{\cdot}{A}\underset{\cdot}{B}} \triangleq \underset{\cdot}{B} - \underset{\cdot}{A}}. \tag{2.8}$$

The length and direction of $\vec{p}_{\underset{\cdot}{A}\underset{\cdot}{B}}$ is such that it goes from $\underset{\cdot}{A}$ to $\underset{\cdot}{B}$. Note that the subscript indicates that only the *positions* of $\underline{A}$ and $\underline{B}$ are included, i.e. (2.8) is not affected by the orientation of $\underline{A}$ or $\underline{B}$.

When decomposing (2.8) in $\underset{\rightarrow}{A}$, $\boldsymbol{p}^{\underline{A}}_{\underset{\cdot}{A}\underset{\cdot}{B}}$ will simply express the coordinates of the point $\underset{\cdot}{B}$ relative to frame $\underline{A}$. From the notation $\boldsymbol{p}^{\underline{A}}_{\underset{\cdot}{A}\underset{\cdot}{B}}$ we see that only the position of $\underline{B}$ matters, while both the position and orientation of $\underline{A}$ matter.

2.3.1.1 Simplified notation

An effective notation should not include more symbols than strictly needed to make it unambiguous. When coordinate frames are used as sub- or superscripts in the notation presented here, their position and/or the quantity they describe will usually specify whether it is the position or orientation of the coordinate frame that matters. For instance, the right superscript always indicates where the vector is decomposed, thus this superscript will always contain the orientation. Similarly, for the subscript of a position vector, it is always the positions of the coordinate frames that matter. In these cases it is sufficient to write the coordinate frame letter (without any arrow or dot below it) in sub- and superscripts. Thus the position vector defined in (2.8) can be written as $\vec{p}_{AB}$, and decomposed in $\underline{C}$ this vector can be simply written as $\boldsymbol{p}_{AB}^{C}$ (instead of $\boldsymbol{p}_{\underset{\cdot}{A}\underset{\cdot}{B}}^{\underset{\rightarrow}{C}}$). Also when referring to a given coordinate frame in text, the underline (or arrow/dot) can normally be omitted, unless for cases where emphasizing either the position or orientation properties (or both) of the coordinate frame is needed.

The simplification improves the readability, without introducing any ambiguities. We will explicitly state the position/orientation (by using the arrow or dot) in sub- and superscripts primarily when it is needed to emphasize which of the two is relevant, or when extra precision is needed. All definitions will have full precision notation.

2.3.2 Velocity

If we observe the change of the vector $\vec{p}_{AB}$ from an (arbitrary) coordinate frame C, we can express its time derivative as

$$\boxed{{}^{\underset{\rightarrow}{C}}\vec{v}_{\underset{\cdot}{A}\underset{\cdot}{B}} \triangleq \frac{{}^{\underset{\rightarrow}{C}}d}{dt}\left(\vec{p}_{\underset{\cdot}{A}\underset{\cdot}{B}}\right)}. \tag{2.9}$$

Note that only a change in a *vector* is observed, and since a vector does not have a position, the position of C does not matter. This is indicated by using $\underset{\rightarrow}{C}$ as leading superscript, and ${}^{\underset{\rightarrow}{C}}\vec{v}_{\underset{\cdot}{A}\underset{\cdot}{B}}$ describes how the vector $\vec{p}_{\underset{\cdot}{A}\underset{\cdot}{B}}$ changes *observed from* coordinate frame C. Thus this is a more general quantity than the standard understanding of the term velocity, and hence we call ${}^{\underset{\rightarrow}{C}}\vec{v}_{\underset{\cdot}{A}\underset{\cdot}{B}}$ *generalized velocity*. Note that as any other coordinate-free vector, this vector can also be decomposed in an arbitrary coordinate frame, and hence we can construct a velocity vector that depends on four different coordinate frames; ${}^{\underset{\rightarrow}{C}}\boldsymbol{v}_{\underset{\cdot}{A}\underset{\cdot}{B}}^{\underset{\rightarrow}{D}}$ (while for the most common velocities, a maximum of three different frames are involved, see Table 2.5).

In the standard understanding of velocity, the position vector originates from the same frame as we observe its change, i.e. we often have ${}^{\underset{\rightarrow}{A}}\vec{v}_{\underset{\cdot}{A}\underset{\cdot}{B}}$. The *standard velocity* expresses how the point $\underset{\cdot}{B}$ (the orientation of B is not relevant) moves observed from $\underline{A}$ (both the orientation and position of A are relevant). The standard velocity has a simpler (yet unambiguous) notation,

$$\boxed{\vec{v}_{\underline{A}\underset{\cdot}{B}} \triangleq {}^{\underset{\rightarrow}{A}}\vec{v}_{\underset{\cdot}{A}\underset{\cdot}{B}}}. \tag{2.10}$$

As we can see, the first letter in the subscript of $\vec{v}_{\underline{A}\underset{\cdot}{B}}$ includes both the position and orientation of A. The difference between $\underline{A}$ and $\underset{\cdot}{B}$ for standard velocity is often not fully understood, and this is a common source of error. Thus the underline must be kept also in the simplified notation to emphasize this difference, i.e. we use $\vec{v}_{\underline{A}B}$ for standard velocity in the simplified notation.

2.3.3 Acceleration

Observing the change of vector $\vec{p}_{AB}$ from coordinate frame C as in (2.9), but now differentiating twice gives

$$\boxed{{}^{\underset{\rightarrow}{C}}\vec{a}_{\underset{\cdot}{A}\underset{\cdot}{B}} \triangleq {}^{\underset{\rightarrow}{C}}\frac{d^2}{dt^2}\left(\vec{p}_{\underset{\cdot}{A}\underset{\cdot}{B}}\right)}, \tag{2.11}$$

which we call *generalized acceleration*. As with velocity, acceleration is usually observed from the same frame as the differentiated position vector originates. Hence we also define a more compact symbol for the *standard acceleration*,

$$\boxed{\vec{a}_{\underline{A}\underset{\cdot}{B}} \triangleq {}^{\underset{\rightarrow}{A}}\vec{a}_{\underset{\cdot}{A}\underset{\cdot}{B}}}. \tag{2.12}$$

2.3.4 Orientation

Absolute orientation can be represented by a tuple of basis vectors, e.g. $\underset{\rightarrow}{A}$, while in practice, the *relative orientation* between two coordinate frames is often needed. The orientation of an arbitrary coordinate frame B relative to A can, according to Euler's theorem, always be described as one (simple) rotation[1] of an angle, β_{AB}, about a fixed axis, $\vec{k}_{AB}$. The sign of β_{AB} is found from the right hand rule. Thus, the orientation of B relative to A can be described by

$$\left(\vec{k}_{AB}, \beta_{AB}\right), \quad \left|\vec{k}_{AB}\right| = 1, \ \beta_{AB} \in [0, \pi]. \tag{2.13}$$

[1] Rotation of a temporary frame *T* that initially has the same orientation as *A* and ends up having the same orientation as *B*.

(2.13) is called the *axis-angle representation*.

The *product* of the axis and angle is often of interest, giving a vector called the *axis-angle product*,

$$\boxed{\vec{\theta}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}} \triangleq \vec{k}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}} \cdot \beta_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}}}. \tag{2.14}$$

2.3.4.1 Alternative orientation representation: Rotation matrix

Many alternative parameterizations exist for representing orientation (see for instance Craig (1989) or Kane et al. (1983)). The most important representation in this context is the *rotation matrix*, which is thus included here.

Assume two arbitrary coordinate frames A and B. An arbitrary (nonzero) vector $\vec{x}_1$ is rotated an angle β_{AB} about an axis $\vec{k}_{AB}$ getting a new vector $\vec{x}_2$ (where $\left(\vec{k}_{AB}, \beta_{AB}\right)$ is the axis-angle representation of the rotation). Thus $\vec{x}_1$ will relate to A as $\vec{x}_2$ relates to B, i.e. in decomposed form we have

$$\boldsymbol{x}_1^A = \boldsymbol{x}_2^B . \tag{2.15}$$

We seek an entity to multiply with $\vec{x}_1$ to get $\vec{x}_2$, i.e. we seek a *dyadic*. A dyadic consists of sums of pairs of coordinate-free vectors such that scalar pre- or post-multiplication with a coordinate-free vector gives a new (coordinate-free) vector (se e.g. Kane et al. (1983) or Egeland and Gravdahl (2002) for more about dyadics). To find the dyadic, we will first find the relation between $\vec{x}_1$ and $\vec{x}_2$ expressed by means of $\vec{k}_{AB}$ and β_{AB} (from (2.13)). This relation can be found by simple vector algebra/geometrical inspections (see e.g. Goldstein (1980)),

$$\vec{x}_2 = \cos\beta_{AB}\left(\vec{x}_1\right) + \sin\beta_{AB}\left(\vec{k}_{AB} \times \vec{x}_1\right) + \left(1 - \cos\beta_{AB}\right)\vec{k}_{AB}\left(\vec{k}_{AB} \cdot \vec{x}_1\right), \tag{2.16}$$

where × denotes the cross product and · denotes the dot product. (2.16) can be rewritten as

$$\vec{x}_2 = \vec{R}_{AB} \cdot \vec{x}_1 \tag{2.17}$$

where $\vec{R}_{AB}$ is called a *rotation dyadic*. In agreement with (2.16) and (2.17) we define $\vec{R}_{AB}$ by

$$\boxed{\vec{R}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}} \triangleq \cos\beta_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}}\vec{I} + \sin\beta_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}}\vec{S}\left(\vec{k}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}}\right) + \left(1 - \cos\beta_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}}\right)\vec{k}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}}\vec{k}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}}}, \tag{2.18}$$

where $\vec{I}$ is the identity dyadic and $\vec{S}\left(\vec{k}_{AB}\right)$ denotes the skew symmetric dyadic form of $\vec{k}_{AB}$. We now have a dyadic $\vec{R}_{AB}$ that rotates an arbitrary coordinate-free vector $\vec{x}$ from A to B such that (2.15) is fulfilled. To get a rotation matrix, the dyadic (2.18) is decomposed in the arbitrary frame C, obtaining what we can call a *generalized rotation matrix*,

$$\boxed{\boldsymbol{R}_{AB}^{C} \triangleq \cos\beta_{AB}\boldsymbol{I} + \sin\beta_{AB}\boldsymbol{S}\left(\boldsymbol{k}_{AB}^{C}\right) + \left(1-\cos\beta_{AB}\right)\boldsymbol{k}_{AB}^{C}\left(\boldsymbol{k}_{AB}^{C}\right)^{T}}. \tag{2.19}$$

A generalized rotation matrix is rotating vectors decomposed in (the arbitrary) frame C, from A to B. Hence, the rotation (2.17) decomposed in C, is

$$\boldsymbol{x}_2^C = \boldsymbol{R}_{AB}^C \boldsymbol{x}_1^C. \tag{2.20}$$

In practice, vectors multiplied by $\boldsymbol{R}_{AB}^C$ will usually be decomposed in either A or B, and hence we define the (standard) *rotation matrix* as

$$\boxed{\boldsymbol{R}_{AB} \triangleq \boldsymbol{R}_{AB}^{A} = \boldsymbol{R}_{AB}^{B}}. \tag{2.21}$$

The two latter are equal since the axis of rotation is fixed in both frames, i.e. $\boldsymbol{k}_{AB}^A = \boldsymbol{k}_{AB}^B$.

Note that in the deduction we have used $\vec{R}_{AB}$ as an *active* rotation to rotate $\vec{x}_1$ to a *new* vector $\vec{x}_2$, such that (2.15) is fulfilled. Active rotations of $\vec{x}_1$ and $\vec{x}_2$ decomposed in A and B respectively, are given by

$$\begin{aligned} \boldsymbol{x}_2^A &= \boldsymbol{R}_{AB}\boldsymbol{x}_1^A \\ \boldsymbol{x}_2^B &= \boldsymbol{R}_{AB}\boldsymbol{x}_1^B \end{aligned}. \tag{2.22}$$

If we substitute using (2.15), we get the *passive* use of $\boldsymbol{R}_{AB}$, e.g. decomposing a vector in a desired system (which is the most common usage in navigation),

$$\begin{aligned} \boldsymbol{x}_2^A &= \boldsymbol{R}_{AB}\boldsymbol{x}_2^B \\ \boldsymbol{x}_1^A &= \boldsymbol{R}_{AB}\boldsymbol{x}_1^B \end{aligned}. \tag{2.23}$$

Note that many authors place the A as superscript in $\boldsymbol{R}_{AB}$ (since the rotation matrix $\boldsymbol{R}_{AB}$ can be constructed from the three basis vectors of B decomposed in A). However, over the years we have chosen to place both A and B in the subscript due to the following reasons:

- When deducing and defining the rotation matrix this notation is most natural, and for the generalized rotation matrix the superscript has a different meaning (see (2.19) and (2.20)).
- The subscript usage where the two letters of the subscript show the two frames involved, follows the general notation system introduced in the start of Section 2.3, and is the same as used in all other quantities, such as position and (angular) velocity.
- In Section 2.5 notation rules are summarized, and with both frames as subscripts, the rules for cancelling intermediate frames and negating a variable are very similar for rotation matrices, position, angular velocity etc. (see Sections 2.5.1 and 2.5.2).
- When implementing $\boldsymbol{R}_{AB}$ as a variable in a computer program (i.e. with plain text only), there is no doubt about the order of the frames (e.g. R_AB is used). With $\boldsymbol{R}_B^A$ it turns out that some programmers will follow the order which is most common for vectors (subscript(s) first, then superscript), while others find the top-down order most natural. This has led to uncertainty when implementing code and misinterpretation when reading code.
- In Section 2.5.3 we get a simple rule of closest frames, which is also very useful in computer implementations (one simple rule specifies the order of the subscripts for the various equations (2.40) to (2.43)).

2.3.5 Angular velocity

The angular velocity of B relative to A is defined by

$$\boxed{\begin{aligned}\vec{\omega}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}} \triangleq {} & \vec{b}_{\underset{\rightarrow}{B},1}\left({}^{\underset{\rightarrow}{A}}\frac{d}{dt}\left(\vec{b}_{\underset{\rightarrow}{B},2}\right)\cdot\vec{b}_{\underset{\rightarrow}{B},3}\right)+\vec{b}_{\underset{\rightarrow}{B},2}\left({}^{\underset{\rightarrow}{A}}\frac{d}{dt}\left(\vec{b}_{\underset{\rightarrow}{B},3}\right)\cdot\vec{b}_{\underset{\rightarrow}{B},1}\right)\\ & +\vec{b}_{\underset{\rightarrow}{B},3}\left({}^{\underset{\rightarrow}{A}}\frac{d}{dt}\left(\vec{b}_{\underset{\rightarrow}{B},1}\right)\cdot\vec{b}_{\underset{\rightarrow}{B},2}\right)\end{aligned}}, \tag{2.24}$$

where $\vec{b}_{\underset{\rightarrow}{B},i},\ i\in\{1,2,3\}$ are the basis vectors of B.

From (2.24), the relation between the angular velocity and the derivative of an arbitrary vector $\vec{x}$ is found to be

$${}^{A}\frac{d}{dt}\left(\vec{x}\right)={}^{B}\frac{d}{dt}\left(\vec{x}\right)+\vec{\omega}_{AB}\times\vec{x}\,, \tag{2.25}$$

a relation that is sometimes called the Coriolis equation (Kelly, 2013). In fact the definition, (2.24) is constructed to give (2.25), and this definition is used e.g. by Kane and Levinson (1985).

When the angular velocity is decomposed in A or B, it has a simple relation to the derivative of the rotation matrix,

$$\dot{\boldsymbol{R}}_{AB} = \boldsymbol{R}_{AB}\boldsymbol{S}(\boldsymbol{\omega}_{AB}^{B}) = \boldsymbol{S}(\boldsymbol{\omega}_{AB}^{A})\boldsymbol{R}_{AB}\,, \tag{2.26}$$

where $\boldsymbol{S}(\;)$ is the skew-symmetric form of the input vector. (2.26) can be proven in several ways (Groves, 2013; Egeland and Gravdahl, 2002) and some authors (e.g. Spong and Vidyasagar, 1989 or Egeland and Gravdahl, 2002) uses (2.26) to define the angular velocity.

2.3.6 Angular acceleration

Angular acceleration is defined by

$$\boxed{\vec{\alpha}_{\underset{\to}{A}\underset{\to}{B}} \triangleq {}^{\underset{\to}{A}}\frac{d}{dt}\left(\vec{\omega}_{\underset{\to}{A}\underset{\to}{B}}\right) = {}^{\underset{\to}{B}}\frac{d}{dt}\left(\vec{\omega}_{\underset{\to}{A}\underset{\to}{B}}\right)}. \tag{2.27}$$

The fact that the derivative of $\vec{\omega}_{AB}$ is the same in both A and B can be seen from (2.25).

2.4 Summary of the notation system

This section summarizes the notation system introduced above, and it also includes examples of coordinate frames and quantities that are commonly used in navigation.

When specification of only the position or orientation (or both) of a coordinate frame is needed, the symbols in Table 2.1 are used.

Quantity	Description
$\underline{A}$	Coordinate frame $\underline{A}$, with *six degrees of freedom*. $\underline{A}$ can represent a rigid body, and consists of a point and the basis vectors; $\underline{A} = \left(\underset{\cdot}{A}, \underset{\to}{A}\right)$.
$\underset{\cdot}{A}$	The *position* (origin) of coordinate frame $\underline{A}$, i.e. $\underset{\cdot}{A}$ is a point (member of an affine space), and has *three degrees of freedom*.
$\underset{\to}{A}$	The *orientation* of coordinate frame $\underline{A}$, i.e. $\underset{\to}{A}$ has *three degrees of freedom* and consists of the basis vectors; $\underset{\to}{A} = \left(\vec{b}_{\underset{\to}{A},1}, \vec{b}_{\underset{\to}{A},2}, \vec{b}_{\underset{\to}{A},3}\right)$.

Table 2.1. A coordinate frame, with its position and orientation.

A summary of the notation for the most central quantities for translational movement is given in Table 2.2, while the rotational quantities are summarized in Table 2.3.

Simplified notation	Full precision notation	Definition	Description
$\vec{p}_{AB}$	$\vec{p}_{\underset{\cdot}{A}\underset{\cdot}{B}}$	$\vec{p}_{\underset{\cdot}{A}\underset{\cdot}{B}} \triangleq \underset{\cdot}{B} - \underset{\cdot}{A}$	**Position vector**. The vector whose length and direction is such that it goes from the origin of A to the origin of B.
${}^{C}\vec{v}_{AB}$	${}^{\underset{\rightarrow}{C}}\vec{v}_{\underset{\cdot}{A}\underset{\cdot}{B}}$	${}^{\underset{\rightarrow}{C}}\vec{v}_{\underset{\cdot}{A}\underset{\cdot}{B}} \triangleq {}^{\underset{\rightarrow}{C}}\frac{d}{dt}\left(\vec{p}_{\underset{\cdot}{A}\underset{\cdot}{B}}\right)$	**Generalized velocity**. The derivative of $\vec{p}_{AB}$, relative to coordinate frame C.
$\vec{v}_{\underline{A}B}$	$\vec{v}_{\underline{A}\underset{\cdot}{B}}$	$\vec{v}_{\underline{A}\underset{\cdot}{B}} \triangleq {}^{\underset{\rightarrow}{A}}\vec{v}_{\underset{\cdot}{A}\underset{\cdot}{B}}$	**Standard velocity**. The velocity of the origin of coordinate frame B relative to coordinate frame A. The underline is kept also in the simplified notation to emphasize the asymmetry between $\underline{A}$ and $\underset{\cdot}{B}$, which is important to keep in mind when using the notation rules presented in Section 2.5.
${}^{C}\vec{a}_{AB}$	${}^{\underset{\rightarrow}{C}}\vec{a}_{\underset{\cdot}{A}\underset{\cdot}{B}}$	${}^{\underset{\rightarrow}{C}}\vec{a}_{\underset{\cdot}{A}\underset{\cdot}{B}} \triangleq {}^{\underset{\rightarrow}{C}}\frac{d^2}{dt^2}\left(\vec{p}_{\underset{\cdot}{A}\underset{\cdot}{B}}\right)$	**Generalized acceleration**. The double derivative of $\vec{p}_{AB}$, relative to coordinate frame C.
$\vec{a}_{\underline{A}B}$	$\vec{a}_{\underline{A}\underset{\cdot}{B}}$	$\vec{a}_{\underline{A}\underset{\cdot}{B}} \triangleq {}^{\underset{\rightarrow}{A}}\vec{a}_{\underset{\cdot}{A}\underset{\cdot}{B}}$	**Standard acceleration**. The acceleration of the origin of coordinate frame B relative to coordinate frame A.

Table 2.2. Kinematical quantities for translational movement, for the general coordinate frames A, B, and C.

Simplified notation	**Full precision notation**	**Definition**	**Description**
$\vec{\theta}_{AB}$	$\vec{\theta}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}}$	$\vec{\theta}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}} \triangleq \vec{k}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}} \cdot \beta_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}}$	**Axis-angle product**. $\vec{k}_{AB}$ is the axis of rotation and β_{AB} is the angle rotated.
$\boldsymbol{R}_{AB}^{C}$	$\boldsymbol{R}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}}^{\underset{\rightarrow}{C}}$	Equation (2.19)	**Generalized rotation matrix**. Rotates a vector decomposed in C from frame A to frame B.
$\boldsymbol{R}_{AB}$	$\boldsymbol{R}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}}$	$\boldsymbol{R}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}} \triangleq \boldsymbol{R}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}}^{\underset{\rightarrow}{A}} = \boldsymbol{R}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}}^{\underset{\rightarrow}{B}}$	**Standard rotation matrix**. Used mostly to represent orientation and decompose vectors in different frames.
$\vec{\omega}_{AB}$	$\vec{\omega}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}}$	Equation (2.24)	**Angular velocity**. The angular velocity of coordinate frame B, relative to coordinate frame A.
$\vec{\alpha}_{AB}$	$\vec{\alpha}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}}$	$\vec{\alpha}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}} \triangleq {}^{\underset{\rightarrow}{A}}\frac{d}{dt}\left(\vec{\omega}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}}\right) = {}^{\underset{\rightarrow}{B}}\frac{d}{dt}\left(\vec{\omega}_{\underset{\rightarrow}{A}\underset{\rightarrow}{B}}\right)$	**Angular acceleration.** The angular acceleration of coordinate frame B, relative to coordinate frame A.

Table 2.3. Rotational kinematical quantities, for the general coordinate frames A*,* B*,* and C*.*

All the vectors in Table 2.2 and Table 2.3 are *coordinate-free*, indicated by an arrow above the letter. Any coordinate-free vector can be *decomposed* in any coordinate frame. When decomposed in a coordinate frame (getting a column vector with three scalars), the vector is written in bold, without arrow, and the frame of decomposition is indicated with the right superscript. For example, $\vec{p}_{AB}$ decomposed in C is written $\boldsymbol{p}_{AB}^{C}$.

The A, B, and C-frames used above are arbitrary coordinate frames, while Table 2.4 lists specific coordinate frames used throughout this thesis.

Symbol	Name	Description
I	Inertial	The coordinate frame is an inertial frame of reference.
E	Earth	The coordinate frame is Earth-fixed, with origin in the geometrical center of the reference ellipsoid used (often called earth-centered-earth-fixed, ECEF).
B	Body	The coordinate frame is fixed to the vehicle/device to be navigated.
N	North-East-Down	A local level coordinate frame with the origin directly beneath or above the vehicle (B), at Earth's surface (surface of ellipsoid model). The x-axis points towards north, the y-axis points towards east (both are horizontal), and the z-axis is pointing down. Note: When moving relative to the Earth, the frame rotates about its z-axis to allow the x-axis to always point towards north. When getting close to a pole this rotation rate will increase, being infinite at the poles. The poles are thus singularities and the direction of the x- and y-axes is undefined there.
L	Local level, Wander azimuth	A local level coordinate frame with the origin directly beneath or above the vehicle (B), at Earth's surface (surface of ellipsoid model). The z-axis is pointing down and hence L is equal to N except for the rotation about the z-axis. The rotation rate about the z-axis is defined to be zero (i.e. $\boldsymbol{\omega}_{EL,\mathrm{z}}^{L} = 0$), and thus L is non-singular. L is often chosen to be equal to N initially (if outside the poles), and as the vehicle moves there will in general be a non-zero angle between the x-axis of L and the north direction; this angle is called the wander azimuth angle.

Table 2.4. Common coordinate frames used in this thesis. They are all orthonormal and right handed.

With the coordinate frames in Table 2.4 introduced, examples of quantities that are very common within navigation are listed in Table 2.5.

Simplified notation	Full precision notation	Description
$\vec{p}_{EB}$	$\vec{p}_{\underset{\cdot}{E}\underset{\cdot}{B}}$	The position of (the origin of) B relative to (the origin of) E, coordinate-free.
$\boldsymbol{p}_{EB}^{E}$	$\boldsymbol{p}_{\underset{\cdot}{E}\underset{\cdot}{B}}^{\underset{\rightarrow}{E}}$	The position of (the origin of) B relative to (the origin of) E, decomposed in E. This vector is often called the “ECEF-vector” and the three elements are called the “ECEF coordinates”.
$\boldsymbol{p}_{BS}^{B}$	$\boldsymbol{p}_{\underset{\cdot}{B}\underset{\cdot}{S}}^{\underset{\rightarrow}{B}}$	The position of a sensor, S, mounted on the vehicle, given relative to the vehicle reference frame B. This vector is often called the lever arm of the sensor, and when assuming a rigid body, the vector is modelled as fixed.
$\vec{v}_{\underline{E}B}$	$\vec{v}_{\underline{E}\underset{\cdot}{B}}$	The velocity of (the origin of) B relative to $\underline{E}$, coordinate-free. When someone uses the (ambiguous) term “the velocity of object B”, they often mean this vector.
$\boldsymbol{v}_{\underline{E}B}^{E}$	$\boldsymbol{v}_{\underline{E}\underset{\cdot}{B}}^{\underset{\rightarrow}{E}}$	The velocity of (the origin of) B relative to $\underline{E}$, decomposed in E. In practice, this quantity can be obtained from GNSS, e.g. by utilizing the Doppler shift.
$\boldsymbol{v}_{\underline{E}B}^{B}$	$\boldsymbol{v}_{\underline{E}\underset{\cdot}{B}}^{\underset{\rightarrow}{B}}$	The velocity of (the origin of) B relative to $\underline{E}$, decomposed in B. This vector is typically measured by a body fixed sensor observing Earth-fixed objects (or the ground). Examples of sensors giving $\boldsymbol{v}_{\underline{E}B}^{B}$ are cameras, acoustic Doppler velocity logs or Doppler radars.
$\boldsymbol{v}_{\underline{E}B}^{N}$	$\boldsymbol{v}_{\underline{E}\underset{\cdot}{B}}^{\underset{\rightarrow}{N}}$	The velocity of (the origin of) B relative to $\underline{E}$, decomposed in N (i.e. the north, east and down components of the velocity vector).
$\vec{\omega}_{IB}$	$\vec{\omega}_{\underset{\rightarrow}{I}\underset{\rightarrow}{B}}$	The angular velocity of B relative to I, coordinate-free. This is the vector that is measurable by gyros.
$\boldsymbol{\omega}_{IB}^{B}$	$\boldsymbol{\omega}_{\underset{\rightarrow}{I}\underset{\rightarrow}{B}}^{\underset{\rightarrow}{B}}$	The angular velocity of B relative to I, decomposed in B. This is the measurement[1] from (strapdown) gyros.
$\boldsymbol{R}_{NB}$	$\boldsymbol{R}_{\underset{\rightarrow}{N}\underset{\rightarrow}{B}}$	The orientation of B relative to N, represented as a rotation matrix. This rotation matrix contains the same information as the vehicle’s roll, pitch and yaw angles.

Table 2.5. Examples of quantities common in navigation.

2.5 Notation rules

The quantities defined have properties that give simple rules for their usage when following the notation system introduced above.

[1] In practice, a set of gyros often return an incremental rotation (called “delta theta”), but in principle it is the shown angular velocity that is measured.

2.5.1 Negating a quantity (switching the order of the subscripts)

Switching the order of the subscripts gives the opposite position vector,

$$\vec{p}_{AB} = -\vec{p}_{BA}\,. \tag{2.28}$$

This is also the case for generalized velocity,

$$^{C}\vec{v}_{AB} = -\,^{C}\vec{v}_{BA}\,. \tag{2.29}$$

For standard velocity, $\vec{v}_{\underline{A}B}$, switching the order of the subscripts *does not* give the negative vector, which is indicated by the underline (see also the comment in Section 2.3.2).

For acceleration, we have similar relations, i.e. switching of subscripts negates the generalized acceleration,

$$^{C}\vec{a}_{AB} = -\,^{C}\vec{a}_{BA}\,, \tag{2.30}$$

while this is not the case for the standard acceleration, $\vec{a}_{\underline{A}B}$.

Switching the subscripts of the axis-angle product gives the negative vector,

$$\vec{\theta}_{AB} = -\vec{\theta}_{BA}\,. \tag{2.31}$$

For a rotation matrix we have that

$$\boldsymbol{R}_{AB} = \left(\boldsymbol{R}_{BA}\right)^{T} \tag{2.32}$$

where the T indicates matrix transpose. It should be noted that for rotation matrices the transpose is the inverse, i.e. $\boldsymbol{R}_{AB}\boldsymbol{R}_{BA} = \boldsymbol{I}$, and hence we again get that quantities with opposite order of subscripts cancel each other (the vectors of equations (2.28) to (2.31) cancel each other when summed).

For angular velocity, we have that

$$\vec{\omega}_{AB} = -\vec{\omega}_{BA}\,. \tag{2.33}$$

And finally, a similar relation is also true for angular acceleration,

$$\vec{\alpha}_{AB} = -\vec{\alpha}_{BA}\,. \tag{2.34}$$

2.5.2 Cancelling an intermediate coordinate frame

With three (or more) coordinate frames involved, the cancelling of an intermediate coordinate frame is a very useful rule.

For position, we have that

$$\vec{p}_{AC} = \vec{p}_{AB} + \vec{p}_{BC}\,, \tag{2.35}$$

where the two B's in the subscripts that are closest to each other are cancelled.

For velocity, a similar relation is valid for generalized velocity, i.e.

$${}^{C}\vec{v}_{AD} = {}^{C}\vec{v}_{AB} + {}^{C}\vec{v}_{BD}\,. \tag{2.36}$$

And again, the underline of the standard velocity indicates that such a relation is not true for $\vec{v}_{\underline{A}B}$.

Acceleration has similar properties, where

$${}^{C}\vec{a}_{AD} = {}^{C}\vec{a}_{AB} + {}^{C}\vec{a}_{BD} \tag{2.37}$$

holds for generalized acceleration, whereas for standard acceleration, $\vec{a}_{\underline{A}B}$, no such relation is valid.

Adding axis-angle product vectors does not cancel intermediate frames. For $\vec{v}_{\underline{A}B}$ and $\vec{a}_{\underline{A}B}$, the asymmetry in the subscript coordinate frames (indicated by the underline) was the reason why intermediate frames did not cancel, while for $\vec{\theta}_{AB}$ this is not the case (as no such asymmetry is present). Instead the reason is simply the complexity of rotations in three dimensional Euclidian space (3D rotations do not commute; see e.g. Mirman (1995)).

For the rotation matrix however, we can cancel intermediate coordinate frames with matrix multiplication,

$$\boldsymbol{R}_{AC} = \boldsymbol{R}_{AB}\boldsymbol{R}_{BC}\,. \tag{2.38}$$

Also for angular velocity, we have such a relation,

$$\vec{\omega}_{AC} = \vec{\omega}_{AB} + \vec{\omega}_{BC}\,. \tag{2.39}$$

Finally, adding angular acceleration vectors does not cancel the intermediate frames, and this can be shown by using (2.25) (and this equation can also be used to show (2.39)).

2.5.3 The rule of closest frames for rotation matrices

The two previous sections gave rules for the subscript usage when negating quantities or when cancelling intermediate coordinate frames. For the rotation matrix however, there are many other usages, not covered by (2.32) and (2.38). The passive use of the rotation matrix, presented in (2.23), is the most common in navigation (and for this reason the rotation matrix is sometimes just called the *coordinate transformation matrix* (Groves, 2013)).

A rule to decide the order of the subscripts when decomposing a vector is needed. To find this rule we can look at the equation that relates a general vector $\vec{x}$ decomposed in A or B. From (2.23) we have

$$\boldsymbol{x}^{A} = \boldsymbol{R}_{AB}\boldsymbol{x}^{B}\,. \tag{2.40}$$

From this equation we see that the B in the subscript of $\boldsymbol{R}_{AB}$ is closest to the B in which the vector is decomposed. We can call this “the rule of closest frames” for rotation matrixes, which for this case says that the frame closest to the vector for post multiplication is always the same as the frame where the vector is decomposed.

The rule of closest frames is also valid for other common relations involving rotation matrices. The first example to include is its relation with the angular velocity, i.e.

$$\dot{\boldsymbol{R}}_{AB} = \boldsymbol{R}_{AB}\boldsymbol{S}(\boldsymbol{\omega}_{AB}^{B}) = \boldsymbol{S}(\boldsymbol{\omega}_{AB}^{A})\boldsymbol{R}_{AB}\,. \tag{2.41}$$

We see that when the skew symmetric matrix contains the vector decomposed in B, a rotation matrix pre-multiplied must have its subscript B closest to the vector. In the variant where the vector is decomposed in A, the post-multiplied rotation matrix has its subscript A closest to the vector.

The next example is the similarity transform of a skew symmetric form, i.e. we have

$$\boldsymbol{S}(\boldsymbol{x}^{A}) = \boldsymbol{R}_{AB}\boldsymbol{S}(\boldsymbol{x}^{B})\boldsymbol{R}_{BA}\,. \tag{2.42}$$

Again, we see that for both the rotation matrices, their order of subscripts is such that coordinate frames B are always closest to the vector decomposed in B.

The final example included for the closest frames rule is for a 3x3 covariance matrix representing an uncertainty ellipsoid (confidence ellipsoid) in B, i.e. we would write it $\boldsymbol{W}^{B}$. If we want to transform this matrix to A, we would use

$$\boldsymbol{W}^{A} = \boldsymbol{R}_{AB}\boldsymbol{W}^{B}\boldsymbol{R}_{BA}\,, \tag{2.43}$$

where the subscripts of the rotation matrices follow the closest frames rule since the matrix is represented in B. Transformations like (2.43) are common in navigation (and estimation in general) since the covariance matrix is diagonal when the axes of representation are aligned with the semi-principal axes of the ellipsoid (i.e. parallel with the eigenvectors).

2.6 Conclusion

When developing navigation systems on a daily basis in a team, the importance of a good notation system becomes particularly clear, and the advantages can be summarized as follows:

- **Ensuring correct deductions:** The notation rules give an effective mechanism to avoid
  - errors when setting up equations.
  - errors in deductions.
  - wrong usage of measurements.
- **Ensuring correct implementation:** The equations often end up in program code, and it is important to have strict rules for how to port the notation to plain text variables (rules that maintain the precision and unambiguousness). When implementing an estimator e.g., a precise notation is critical to maintain optimality and stability throughout the code. With the simple notation rules, a quick look at the implemented code is sufficient to reveal any errors.
- **Improving the understanding:** Due to properties 2 and 4 of Section 2.1, and the introduction of generalized quantities, the notation system improves the users' understanding of the described quantities.
- **Avoid errors when exchanging code and/or equations:** With an ambiguous notation system, errors are common when exchanging code and/or equations, and even when revisiting one's own work done a few years ago, misunderstandings may arise.
- **No need to invent new symbols for new quantities:** Both when deducing equations and when programming, new quantities are constructed based on existing quantities. The new quantities need a symbol/variable name, and when the notation system is extensive, the name of the new quantity is already given from this system. Hence, the development gets more effective since no time or attention is needed to invent new symbols.

Chapter 3

Fundamental Topics within Navigation

With the notation system defined, it is now possible to present more navigation specific topics, and in this chapter, two topics that are both fundamental within navigation will be discussed. The first is *position calculations*, and the full description of this theory is given in **Paper I**. The second topic, *heading estimation*, is thoroughly described in **Paper II**.

3.1 Position calculations

The ability to *calculate accurate geographic positions* is critical within navigation, as well as within other fields such as geodesy. However, from many years of experience, we have seen that when performing global position calculations, one or more of the following concerns are often involved:

1) Approximations, e.g.
  a) distortion in map projections
  b) assuming spherical Earth when an ellipsoidal model should be used
  Errors often increase with increasing distances
2) Complex implementations (many, and often complex, lines of program code needed)
3) Equations and/or code not valid/accurate for all Earth positions, e.g.
  a) Latitude/longitude:
    i) Singularities at Poles
    ii) Discontinuity at the ±180° meridian
  b) Map projections: usually valid for a limited area, e.g. the Universal Transverse Mercator (UTM, Snyder, 1987)
4) Iterations (iterations are required to achieve the needed accuracy)

To overcome these difficulties, we will start by looking at how position is represented. Two of the most common representations of global position are latitude/longitude (and height) and the position vector decomposed in the Earth-centered-earth-fixed (ECEF) coordinate frame, $\boldsymbol{p}_{EB}^{E}$. This vector is now called the "ECEF-vector" (see also Table 2.5).

A major difference between these two representations is that the latitude/longitude representation is separating the vertical and horizontal positions, which is not the case for the ECEF-vector. This separation is both intuitive and has several practical advantages. Three examples where separation is clearly useful are:

- In navigation systems, where horizontal and vertical position are often measured by different sensors at different points in time
- In a vehicle autopilot, where horizontal and vertical position are often controlled independently
- For ships and several land vehicles, where many calculations only consider the horizontal position

In these examples (and in many other cases) we need a quantity for representing horizontal position independently of the vertical height/depth (e.g. when comparing two horizontal positions). Thus, it should be possible to represent horizontal position without considering the vertical position, and vice versa. If the ECEF-vector is used, the horizontal and vertical positions are not separated as desired.

Due to the above reasons, position representations that separate horizontal and vertical directions are used extensively in a wide range of applications. In addition to latitude/longitude, other common representations with this property are the UTM (and other map projections) and a local vector relative to a local Cartesian "flat Earth" coordinate frame (e.g. North-East-Down).

However, all these representations (which separate the vertical and horizontal directions) have significant disadvantages when performing many position calculations (as discussed in **Paper I**). Hence, we seek a representation that separates the vertical and horizontal directions, but that also has good mathematical properties for position calculations. In **Paper I** the outward pointing normal vector to the Earth reference ellipsoid is introduced as a horizontal position representation, and it is called *n*-vector. Figure 3.1 shows that *n*-vector corresponds to standard (geodetic) latitude.

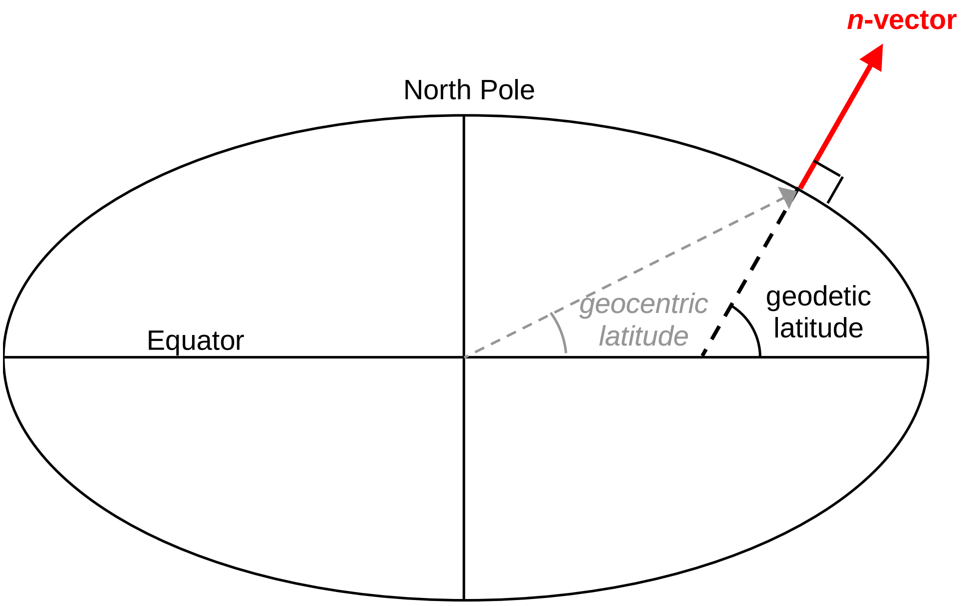


Figure 3.1. Earth reference ellipsoid with n-vector, standard (geodetic) latitude and geocentric latitude.

The *n*-vector representation is non-singular for all Earth positions, and it has no discontinuities. Its mathematical properties make many position calculations quite simple, and one example is the fact that the *n*-vector is a 3D vector. This means that the powerful vector algebra can be used to solve many position calculations intuitively and with few code lines.

In Table 3.1, six important properties of a position representation are summarized for latitude/longitude, for the *n*-vector and for the ECEF-vector.

Property	**Latitude/longitude**	***n*-vector**	**ECEF-vector**
Horizontal position can be expressed independently of height/depth	Yes	Yes	No
Non-singular	No	Yes	Yes
No discontinuities	No	Yes	Yes
General position calculations are often simple	No	Yes	Yes
Geocentric	*No*	*No*	*Yes*
Geodetic	*Yes*	*Yes*	*No*

Table 3.1. A simplified summary of six important properties for latitude/longitude, n-vector and the ECEF-vector. The colors used are: Green (Yes): Normally an advantage. Red (No): Normally a disadvantage. Black (italic): Advantage/disadvantage is depending on application.

It should be noted that since the ECEF-vector is geocentric, its relation to standard (geodetic) latitude is complex. On the other hand, this relation is very simple for *n*-vector which is also geodetic (normal to the ellipsoid surface). Thus, calculations that are based on latitude and longitude are usually very simple to replace with *n*-vector calculations. The same is not the case for the ECEF-vector.

Using *n*-vector, the vertical direction vector (true up/down direction) is readily available, as opposed to the ECEF-vector, where this direction is complex to calculate. The use of the vertical direction vector makes several calculations very easy; e.g. finding a point that is *x* meters above/below another position, finding horizontal vectors (such as the north and east vectors), and finding vertical components of vectors (see equations (7) to (10) in **Paper I**).

3.1.1 Practical usage

When solving position calculations in practice, computer programs are normally used, and hence it is very useful to have a program library available. We have written a web page (Gade, 2017), that provides examples and a downloadable *n*-vector library. Ten examples of common position calculations are included, and they are shown in Table 3.2 and Table 3.3.

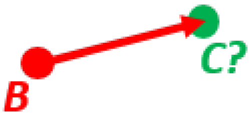
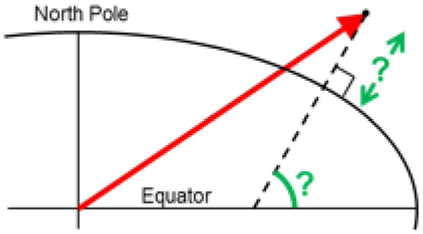
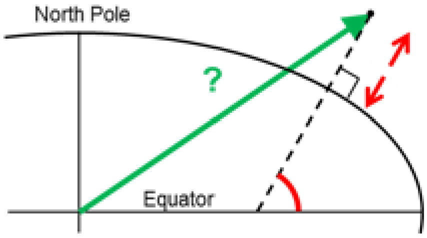
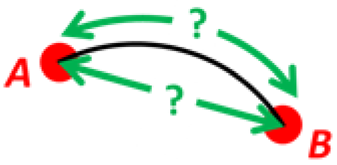
#	Simple description	Simple figure
1.	**A and B to delta** Given two positions A and B. Find the exact vector from A to B in meters north, east and down, and find the direction (azimuth/bearing) to B, relative to north. Use WGS-84 ellipsoid.	
2.	**B and delta to C** Given the position of vehicle B and a bearing and distance to an object C. Find the exact position of C. Use WGS-72 ellipsoid.	
3.	**ECEF-vector to geodetic latitude** Given an ECEF-vector of a position. Find the geodetic latitude, longitude and height.	
4.	**Geodetic latitude to ECEF-vector** Given geodetic latitude, longitude and height. Find the ECEF-vector.	
5.	**Surface distance (great circle distance)** Given position A and B. Find the surface distance (i.e. great circle distance) and the Euclidean distance between A and B.	

Table 3.2. Examples 1-5 of position calculations provided on Gade (2017). Red color indicates the information that is given, while green is what to find.

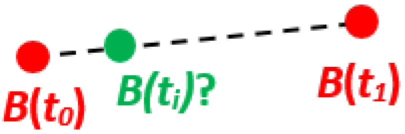
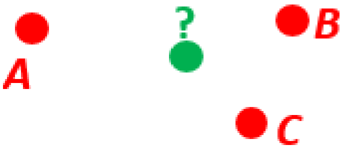
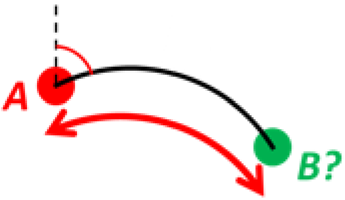
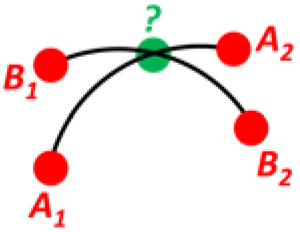
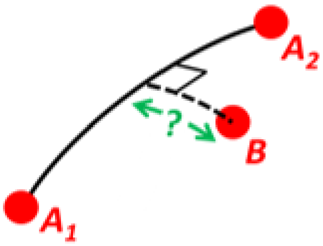
#	Simple description	Simple figure
6.	**Interpolated position** Given the position of B at time t_0 and t_1. Find an interpolated position at time t_i.	
7.	**Mean position (center/midpoint)** Given three positions A, B, and C. Find the mean position (center/midpoint).	
8.	***A* and azimuth/distance to *B*** Given position A and an azimuth/bearing and a (great circle) distance. Find the destination point B.	
9.	**Intersection of two paths** Given path A going through A_1 and A_2, and path B going through B_1 and B_2. Find the intersection of the two paths.	
10.	**Cross-track distance (cross-track error)** Given path A going through A_1 and A_2, and a point B. Find the cross-track distance/cross-track error between B and the path.	

Table 3.3. Examples 6-10 of position calculations provided on Gade (2017). Red color indicates the information that is given, while green is what to find.

On Gade (2017) it is shown (with equations and pseudocode) how the ten examples are solved using *n*-vector, and functions from the downloadable library are used when necessary.

The original program library was written in MATLAB (The MathWorks, 2017), mainly by the author. This library has been extensively used for many years[1] in many different

[1] The first *n*-vector files in the library are from 1999, and in 2004 important functionality was added.

applications, by different groups (e.g. research groups, academia, military, and industry). Due to the widespread usage, the library has been translated by other authors to other programming languages as well (e.g. C#, C++, Python and JavaScript), and these libraries are also available for download.

With **Paper I** and the web page with downloadable code, we consider the concerns listed in the start of Section 3.1 as addressed, both in theory and when performing practical position calculations.

3.2 Heading estimation

Having found solutions for position calculations, the next fundamental topic to discuss is heading estimation. Before looking at heading in particular, we will make some general considerations about the estimation of the six degrees of freedom.

Of the six degrees of freedom, not all are equally difficult to estimate in navigation near Earth. Due to the presence of the gravity vector, position is often separated into horizontal and vertical position, and for orientation we similarly have that estimating heading is typically different from estimating roll and pitch.

Roll and pitch are often estimated with sufficient accuracy (when the specific force measured by the accelerometers is dominated by the gravity vector), while for many applications a magnetic compass is too inaccurate and unreliable to find heading. For position, there are many applications where the horizontal position is clearly more challenging to estimate than the vertical position, since the latter often can be found from pressure sensors or radar/laser altimeters.

Hence, the three most challenging degrees of freedom are often the heading and the horizontal position. However, the actual challenge of estimating these is clearly very dependent on the availability of GNSS and high accuracy gyros (sufficiently accurate for gyrocompassing). Consequently, we can divide navigation systems into four categories, based on the availability of GNSS and accurate gyros, see Table 3.4 (which is from **Paper II**).

Green/italic = Often satisfactory Red/underlined = Challenging		GNSS (or similar) available	
		Yes	**No**
Gyros with sufficient accuracy for gyro-compassing	**Yes**	**Category A1:** ***Heading*** ***Horizontal position*** Typical cases: Large/expensive vehicles (not submerged), e.g. airplanes, ships, helicopters	**Category A2:** ***Heading*** **Horizontal position** Typical cases: Underwater navigation of large/expensive vehicles, e.g. submarines and AUVs
	No	**Category B1:** **Heading** ***Horizontal position*** Typical cases: Light/small/cheap applications in air, land or at sea, e.g. unmanned aerial vehicles (UAVs), boats, robots, cameras, personnel	**Category B2:** **Heading** **Horizontal position** Typical cases: GNSS denied light/small/cheap applications, e.g. indoor navigation, applications under GNSS jamming, low-cost underwater navigation

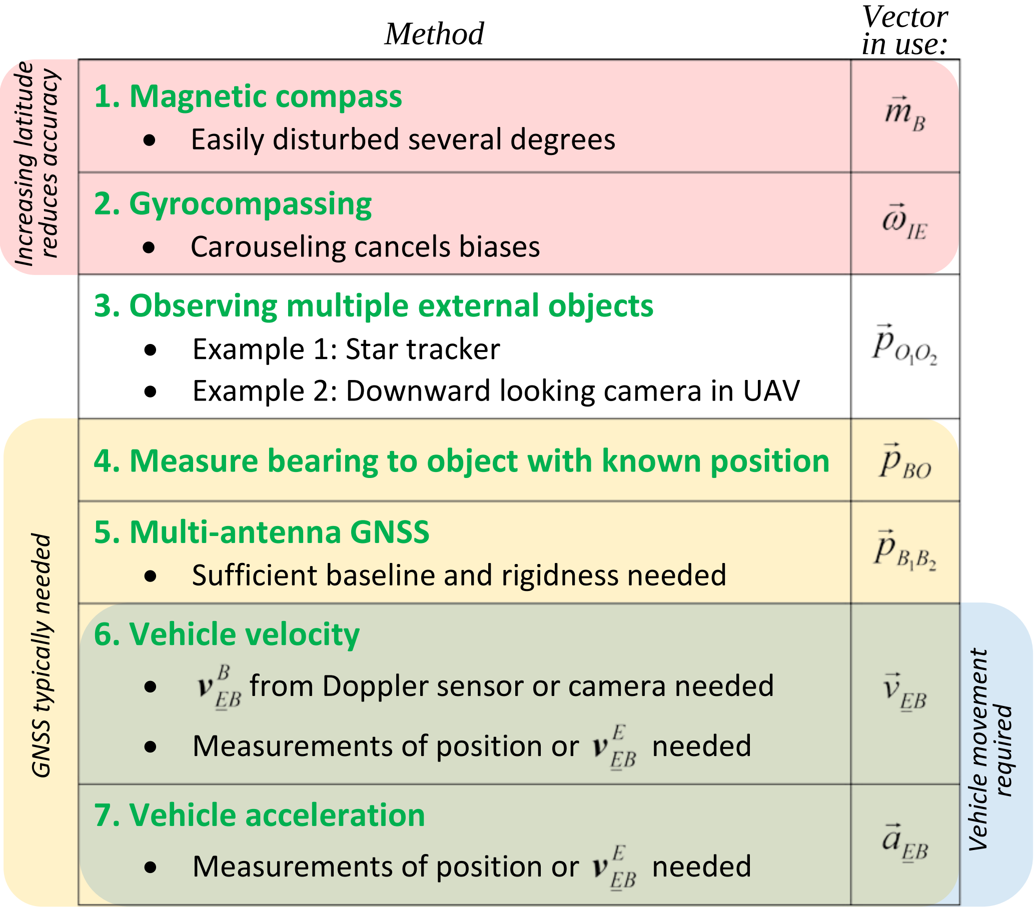
Table 3.4. The four categories (A1, A2, B1, and B2) of inertial navigation systems, broken down by the availability of GNSS and the accuracy of gyros (the table is from ***Paper II****).*

As mentioned in Section 1.1 there has been a rapid growth of applications using MEMS IMUs, and thus we have seen a significant increase in the number of navigation systems belonging to the B-categories (B1 and B2). A common characteristic of these navigation systems is that heading estimation is often a great challenge, and it is not clear how to achieve sufficient heading accuracy.

Despite this increased demand for methods to find heading, it has been difficult to find a list of possible methods in the literature. Thus, we have studied the topic in detail ourselves, and it turned out that it is indeed possible to develop a general theory for heading estimation, and to establish a corresponding list. The different methods have been categorized by means of consistent mathematical principles, but the list is also intuitive, which makes it useful in practice.

As described in **Paper II**, in order to estimate heading, a vector that is known both relative to the vehicle and relative to the Earth (i.e. decomposed in B and E) is needed. If this vector has a horizontal component, heading can be estimated. Thus, when trying to establish a system to categorize the different possible methods for heading estimation, we found it most intuitive to define one method for each type of vector.

Paper II has identified seven different vectors in use in practical navigation systems, and thus seven corresponding methods of heading estimation are defined. A simplified summary of the seven methods is given in Figure 3.2. In addition to the symbols defined in Chapter 2, the figure uses coordinate frames B_1 and B_2 for positions fixed to the vehicle (B). Coordinate frames O, O_1, and O_2 are external objects, and $\vec{m}_B$ is the magnetic field vector at position B. More details are available in **Paper II**.



Method	Vector in use:
1. Magnetic compass • Easily disturbed several degrees	$\vec{m}_B$
2. Gyrocompassing • Carouseling cancels biases	$\vec{\omega}_{IE}$
3. Observing multiple external objects • Example 1: Star tracker • Example 2: Downward looking camera in UAV	$\vec{p}_{O_1O_2}$
4. Measure bearing to object with known position	$\vec{p}_{BO}$
5. Multi-antenna GNSS • Sufficient baseline and rigidness needed	$\vec{p}_{B_1B_2}$
6. Vehicle velocity • $\boldsymbol{v}_{\underline{E}B}^{B}$ from Doppler sensor or camera needed • Measurements of position or $\boldsymbol{v}_{\underline{E}B}^{E}$ needed	$\vec{v}_{\underline{E}B}$
7. Vehicle acceleration • Measurements of position or $\boldsymbol{v}_{\underline{E}B}^{E}$ needed	$\vec{a}_{\underline{E}B}$

Increasing latitude reduces accuracy
GNSS typically needed
Vehicle movement required

Figure 3.2. A simplified summary of the seven methods of heading estimation, and some key features/examples of each method (figure from ***Paper II****).*

3.2.1 Example: Finding heading for a navigation system of Category B2

An AUV without sufficient gyro accuracy for gyrocompassing has a navigation system of Category B2 (of Table 3.4). An example of such a vehicle was the HUGIN I AUV (Størkersen et al., 1998; Kristensen and Vestgård, 1998). The vehicle was fitted with a version of the Seatex Motion Reference Unit (MRU, Kongsberg Seatex, 2017) where the raw gyro and accelerometer measurements were not available. Without the raw IMU measurements, it was not feasible to design a full general inertial navigation system (as described in Chapter 4), and a dedicated navigation system was designed instead, as described in **Paper III**.

The vehicle had a magnetic compass, and thus Method 1 was available to find heading. However, this did not give the required heading accuracy, and another method was needed.

Methods 2 to 5 were not feasible, and Method 7 did not give sufficient accuracy due to low acceleration and position measurements with low accuracy and rate.

However, the vehicle was fitted with a DVL, and hence an accurate measurement of $\boldsymbol{v}_{\underline{E}B}^{B}$ was available. The vehicle also kept a forward velocity (with a horizontal component) at all times, and even with the relatively inaccurate position measurements, Method 6 gave a heading accuracy of about 0.5° in post processing, see **Paper III** for details.

3.2.2 Usage of the list of methods

The theory and list of methods from **Paper II** have been used by FFI the last couple of years, and below are six examples of applications where we have used the list to find heading when designing their navigation system:

- Low cost augmented reality system for military vehicles
- Camera with navigation unit
- Navigation system for an unmanned ground vehicle (UGV)
- Lightweight target localization system (with laser rangefinder). (The documentation of this application is neither classified nor confidential; Hovde (2017).)
- Low cost navigation system for a remotely operated vehicle (ROV)
- Augmented reality for soldier helmets

For each of these six examples, we used the list in the same manner as described in Appendix A of **Paper II**.

The use of the list and theory has turned out to be a game changer when it comes to the design of low cost navigation systems (and also for high end systems where heading is a challenge, e.g. systems requiring rapid initial alignment at sea). The reason is that the insight and understanding of how to estimate heading and the knowledge of the possibilities available have increased drastically. For a given system, the possible ways to find heading are now immediately identified, and we can confidently determine which sensors to add and what maneuvers are required to fulfill the heading requirement.

Chapter 4

General Navigation Software

Having covered the necessary fundamental theory, the next topic is navigation software, which is important since the right choice of software is essential during navigation system development. Topics covered include how to design generic navigation software, and in which manners the software can support a range of different tasks within navigation.

4.1 NavLab

For many applications real-time navigation software is required, but for these applications an offline-tool is also often very useful. Important usages of an offline-tool include system design, test, tuning and verification of performance; see **Paper IV** and Section 4.2 for more details.

There are also many cases where the post processed navigation is of interest itself, e.g. when a vehicle has observed objects or made maps (using camera, radar or sonar), and the objects/maps need to be georeferenced. Due to the possibility of performing smoothing (Gelb, 1974; Minkler and Minkler, 1993), a better estimate is available post mission than in real time.

Based on the above, the need for a navigation post processing software tool is clear, and the goal is to design a tool that can cover all the above mentioned needs (and several others). **Paper IV** presents a generic tool called NavLab (programmed in MATLAB), that covers these needs. NavLab consists of a Simulator part and an Estimator part, and its main structure is shown in Figure 4.1. With the Simulator, any vehicle trajectory can be simulated, and corresponding sensor measurements are simulated (by using models of the sensor errors). The simulated sensor measurements have the same format as measurements logged from a real vehicle, and thus the Estimator can be run with either simulated or real measurements.

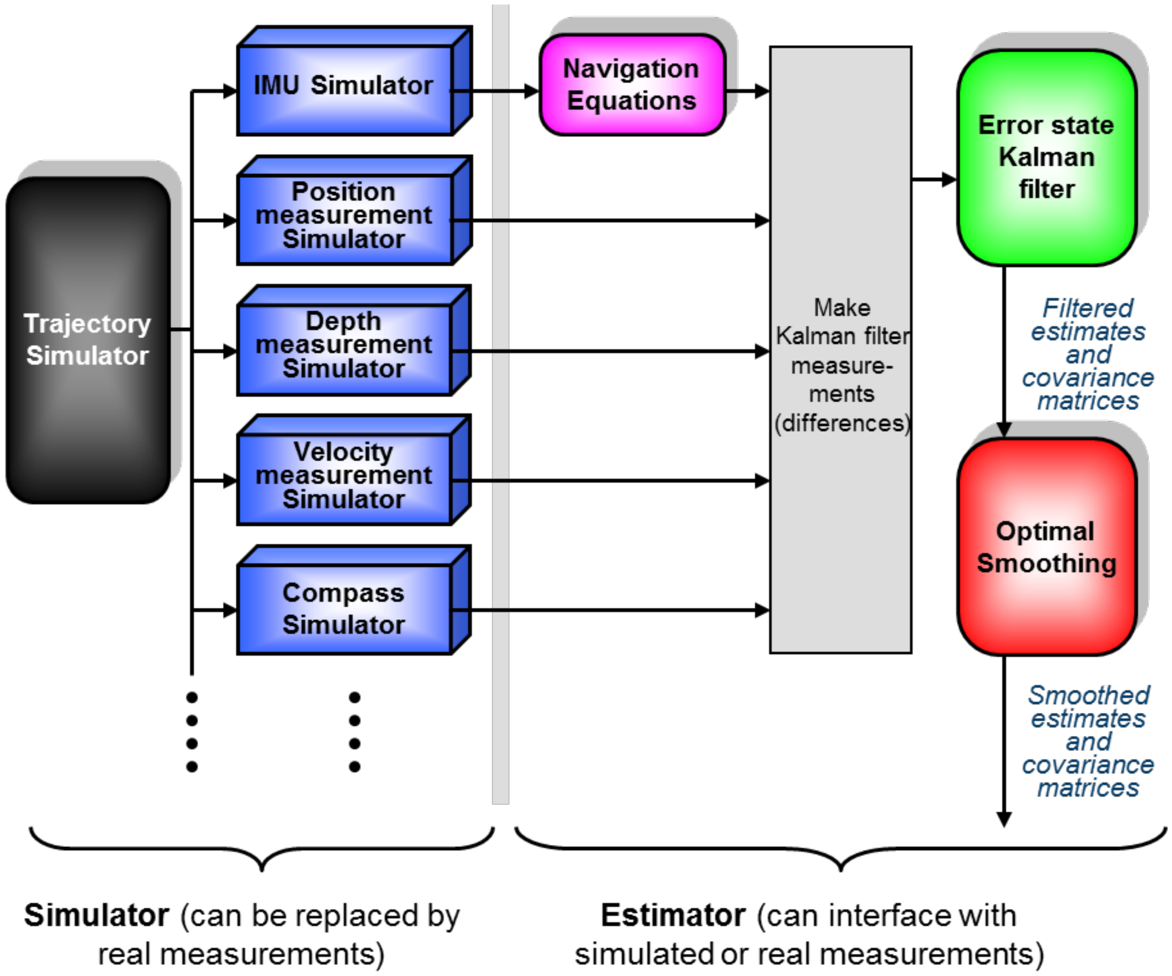

Trajectory Simulator
IMU Simulator
Position measurement Simulator
Depth measurement Simulator
Velocity measurement Simulator
Compass Simulator
Navigation Equations
Make Kalman filter measure-ments (differences)
Error state Kalman filter
Filtered estimates and covariance matrices
Optimal Smoothing
Smoothed estimates and covariance matrices
Simulator (can be replaced by real measurements)
Estimator (can interface with simulated or real measurements)

Figure 4.1. The NavLab main structure (figure from ***Paper IV****)*

Note that the colors used in Figure 4.1 are used for all NavLab plots, and are also consistent through the publications included in this thesis. The following colors are used:

- **Black:** True values (true position, velocity, orientation etc.)
- **Blue:** Measured values
- **Magenta:** Values calculated by navigation equations (equations integrating the IMU measurements to velocity, position and orientation)
- **Green:** Kalman filtered estimates
- **Red:** Smoothed estimates

4.2 Possible NavLab usage

The flexibility of NavLab has made it useful for a wide range of different areas, as discussed in Section 4 of **Paper IV**. A short summary of the different usages is presented here:

- **Navigation system research and development**
    - New aiding techniques and algorithms are implemented and tested.
- **Analysis of a given navigation system**
    - Behavior under different maneuvers/trajectories is analyzed.
    - Robustness against the use of wrong models is studied.
- **Teaching navigation theory**
    - Everything from basic principles to complex mechanisms of an aided inertial navigation system can be demonstrated and visualized.
- **Decision basis**
    - **Sensor purchase:** By entering parameters found in the specifications of the relevant sensors into NavLab, one can simulate the navigation performance, in order to decide which sensors that should be purchased to achieve the required accuracy for the given scenarios.
    - **Mission planning:** For a given vehicle, different mission alternatives can be simulated in advance, to ensure sufficient navigation accuracy. Examples of questions that can be answered are: How often are GNSS-fixes needed? Can a sensor be used with low rate or turned off for a period to save power? Which observability maneuvers are needed?
- **Post-processed navigation from logged sensor data**
    - NavLab has been extensively used for post-processing of logged sensor data, e.g. by survey companies producing underwater maps.
    - The use of post processing means that faulty data sets (e.g. caused by a sensor partially failing) often can be recovered.
- **Sensor evaluation**
    - The performance of each sensor is evaluated in realistic scenarios[1] (far better evaluation is achieved post-mission than in real-time due to the accuracy and robustness of the smoothing).
- **Improving real-time navigation**
    - A post-processing tool is useful also when only real-time navigation is needed, mainly due to the improved accuracy and robustness of the smoothing. Examples of usage include:
        - Sensor calibration (e.g. estimating sensor misalignment)
        - Finding the best Kalman filter tuning (from empirical data)
        - Evaluating the performance of the real-time estimator (no extra sensors needed, all state estimates are evaluated for the entire mission)

NavLab users include research groups, commercial companies, military users and universities.

[1] This is in contrast to sensor evaluations done in a laboratory, where the conditions are often less realistic.

4.3 Real time navigation

For real-time navigation, the algorithms in NavLab have been ported from MATLAB to C++ (not by the author), and put in a real-time framework. Real-time specific algorithms, such as the handling of delayed measurements (Mandt, Gade and Jalving, 2001) are also included. In this manner, the results achieved in real-time are very close to the estimates from the Kalman filter in NavLab (without smoothing). The real-time navigation software is called NavP, and it is described e.g. in **Paper V** and in Hagen, Ånonsen and Mandt (2010).

4.4 Applications

NavLab and NavP were both designed as general navigation systems, being able to navigate any vehicle or device with an IMU. Some examples of vehicles that have been navigated with NavLab and/or NavP are:

Marine applications:

- Autonomous underwater vehicles (AUVs)
- Remotely operated vehicles (ROVs)
- Ships
- Drilling rigs
- Unmanned surface vehicle (USV)

Land applications:

- Unmanned ground vehicle (UGV)
- Personnel/soldiers
- Augmented reality (AR) for military vehicles
- Portable ground-penetrating radar
- Cell phones
- Cars

Air applications:

- Airplanes
- Helicopters
- Missiles
- F-16 (fighter aircraft). An attached pod was navigated.
- Unmanned aerial vehicles (UAVs)

A range of different IMUs have been used for the various applications, from low cost MEMS IMUs, to high-end IMUs with fiber optic gyros (FOGs) or ring laser gyros (RLGs).

With almost 20 years of NavLab usage, it is clear that it is indeed possible to design one general navigation tool for a wide range of usages, and that such a tool is vital in research and development of navigation systems.

Chapter 5

Underwater Navigation

After presenting NavLab, it is now time to study more practical navigation applications. This chapter will focus on underwater navigation, which is the main topic of **Papers V** to **VIII**, and in all of these papers NavLab is used as the main tool.

The lack of GNSS under water means that the navigation systems belong to the right column of Table 3.4, and the main challenges are the accuracy of the horizontal position and possibly the heading accuracy. This chapter will discuss various possibilities to improve these accuracies for different underwater applications.

Underwater navigation systems usually consist of an IMU and a pressure sensor; in addition many underwater vehicles have a DVL. The combination of an IMU, a pressure sensor and a DVL is here called the core navigation system, and it will have unlimited positional drift, see Section 5.1 for more details.

To reduce/avoid the drift, different aiding techniques can be applied, and their feasibility will depend on the given scenario. A navigation system handling different scenarios should thus be flexible, and able to utilize a variety of aiding techniques. **Paper V** describes the flexibility of the navigation system developed for the HUGIN AUVs and gives an overview of the pros and cons of the different techniques, and how to combine them in various common AUV-scenarios.

5.1 Core underwater navigation system

The core underwater navigation system typically consists of an IMU, a DVL, a pressure sensor, and for navigation systems of Category B2 (of Table 3.4), a magnetic compass may be of relevance. The accuracy of the core navigation system is important for the overall navigation accuracy, especially for applications that experience long periods without any external (horizontal) position input.

The core navigation system can provide an estimate of $\boldsymbol{v}_{\underline{E}B}^{E}$ with limited uncertainty, and the error of this quantity will determine the (horizontal) positional drift. $\boldsymbol{v}_{\underline{E}B}^{E}$ is made from two components, $\boldsymbol{v}_{\underline{E}B}^{B}$ and $\boldsymbol{R}_{EB}$, where $\boldsymbol{v}_{\underline{E}B}^{B}$ is measured by the DVL and the most significant error of $\boldsymbol{R}_{EB}$ is the heading error. The accuracy of the DVL is clearly of great importance and a thorough discussion of the error sources of the DVL and the performance of a core navigation system in different scenarios is given in **Paper VI**. For along-track positional drift, the DVL is the main error source, while the cross-track drift caused by the DVL may be larger or smaller than the drift from the heading error, depending on the heading accuracy of the given application. The accuracy of the DVL output is depending greatly on whether it has bottom track or not.

For cases where bottom track is not achieved, most DVLs will provide velocity relative to the surrounding water, and then the error in the sea current estimate is normally the main error source of the core navigation system. A good estimate of the sea current can be achieved in a period with bottom track, by letting the DVL alternate between measuring velocity relative to the bottom and relative to the water. When a good estimate of the sea current is available, a period without bottom track will give far less drift than in a case where the sea current is unknown, assuming that the sea current is relatively constant during the period. Estimates of the sea current can also be obtained in periods without bottom track, if some position measurements are available. Hence, even position measurements of very low frequency may be of great importance for a core navigation system running a water-referenced DVL, as long as the sea current does not change much between the measurements.

5.1.1 Aiding with a vehicle model

The DVL is often critical for the navigation accuracy, and in cases of DVL failure the core navigation system will experience free inertial drift in horizontal velocity and position. However, it is possible to reduce this drift by means of a hydrodynamic vehicle model, and even with a relatively high-end IMU, the free inertial velocity error will quickly become larger than the accuracy we can obtain from such a model. Thus, using a hydrodynamic vehicle model can be crucial in cases of DVL failure or dropouts. Such a model can also be used to improve the robustness and integrity, for example by letting the estimated velocity of the navigation system be continuously monitored by comparing it to the velocity calculated from the vehicle model. Finally, a vehicle model can be required for low cost vehicles where a DVL is too expensive and/or too large.

Paper VII presents a vehicle model, and shows how an underwater navigation system can be aided with such a model. With an error-state structure of the Kalman filter, the vehicle model can run in parallel with the estimator, and its output can be modelled as a velocity measurement. In several cases, a mere addition of software with a vehicle model (no additional instrumentation needed) can significantly reduce the navigation uncertainty.

5.1.2 Velocity measurements from a sonar array

A DVL is not the only sensor that can provide high accuracy measurements of velocity relative to the seabed. AUVs used for high accuracy seabed mapping are often equipped with sonar arrays intended for synthetic aperture sonar (SAS, Hayes and Gough, 2009; Hansen, 2011). These arrays can also be used to calculate displacement of the AUV by correlating the response of successive pings (Bellettini and Pinto, 2002), a technique called displaced phase-center antenna (DPCA). Correlation between elements (in space) gives a surge displacement, while correlation in time of overlapping elements (or more precisely overlapping phase centers) gives a sway displacement. These DPCA displacements can be used to aid the inertial navigation, as described in Hagen et al. (2001). The first reported results of aiding inertial navigation with such sonar displacements were given in Wang et al. (2001). In Hansen et al. (2003) different strategies of combining DPCA and inertial navigation are compared by evaluating the contrast of the resulting SAS images.

5.2 Acoustic positioning from a surface ship

To restrain the unlimited drift of the core navigation system, (horizontal) position measurements are ideal. If the underwater vehicle can go to surface, a GNSS fix is obviously a simple method. When submerged (where GNSS is not directly available), a common solution is to let the underwater vehicle be followed by a surface platform with GNSS and acoustic positioning. A typical implementation of this is a surface ship measuring the relative position of the underwater vehicle using ultra-short baseline (USBL) acoustic positioning. The USBL position measurements will have decreasing accuracy with increasing water depths, and the magnitude of the different error contributions are discussed in Jalving and Gade (1998). The global position of the underwater vehicle is calculated on board the ship, and for the HUGIN vehicles, a subset of these calculated measurements are transmitted to the AUV. These position measurements will be significantly delayed, and this must be handled by the AUV real-time navigation system (NavP), see Mandt, Gade and Jalving (2001) for more details.

Acoustic positioning from a surface ship has significant limitations for real time navigation, but for post processed navigation, the measurements can be better utilized. Post mission, the delay is no issue. In addition, the position measurements stored on the ship can be used, and these are typically of much higher rate than the subset that was transmitted to the underwater vehicle in real time. Examples of the performance achieved post mission using acoustic positioning from a surface ship are available in **Papers III**, **IV** and **V**.

5.3 Range from underwater transponders

Following an AUV with a USBL-equipped surface ship is expensive and there are several scenarios where the use of underwater transponders for positioning, as illustrated in Figure 5.1, is a far better alternative. Examples are pipeline inspection and other areas where repeated dives are needed. Also in areas with heavy surface traffic, avoiding the surface is clearly beneficial.

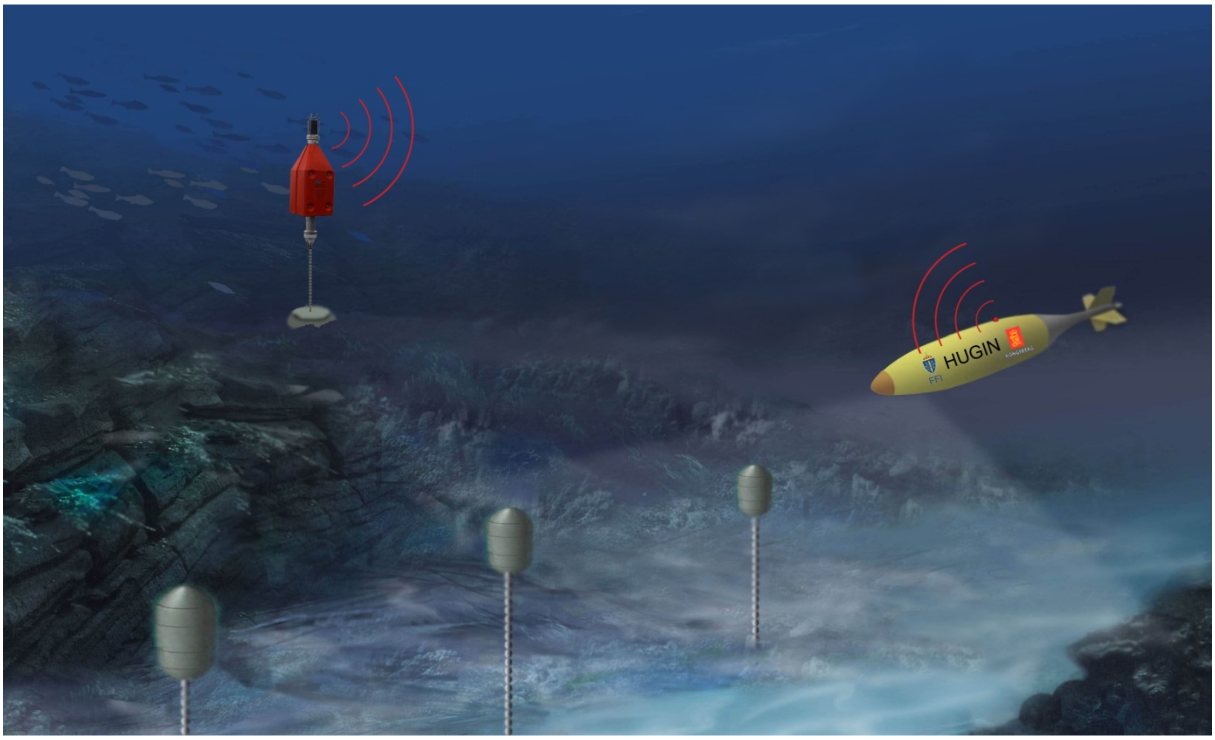


Figure 5.1. AUV measuring range to an underwater transponder.

Underwater transponders can be deployed and boxed-in (positioned) with a USBL-equipped ship, and they typically have a battery life of several years. When battery life is soon ending, or if the transponder is no longer needed in the current position, an acoustic command can instruct the release of a disposable weight, and the transponder floats to the surface for reuse.

When an underwater vehicle interrogates the transponder, the range from the transponder is found from two-way travel time (or one-way travel time, in cases with synchronized clocks (Eustice et al., 2007)). In classical long baseline (LBL) systems, three transponders within range are needed to calculate the vehicle position (assuming the depth of the vehicle is known). **Paper VIII** presents a solution where accurate position is achieved with the use of only one single transponder within range (several transponders can also be used, improving the accuracy further). The accuracy is achieved by integrating the range measurements tightly with the core navigation system (in NavLab), and utilizing the vehicle movement. The performance of the range aiding can be verified by using a surface ship with USBL as reference, and in **Paper VIII** an accuracy close to the USBL accuracy was demonstrated. A similar performance has been achieved in several other trials, and one of these trials is described in Section 4.3 in **Paper V**.

5.4 Terrain referenced navigation

A chapter about underwater navigation is not complete without mentioning terrain referenced navigation. In general, if a vehicle is moving through a varying Earth-fixed field, and has a sensor whose output is a function of these variations, the sensor can be used for position estimation. If a database/map of the field exists, the vehicle position can be found by correlation. Examples of fields that can be utilized are the magnetic field (Goldenberg, 2006; Storms, Shockley, and Raquet, 2010) and the varying channel impulse response of cell phones in urban areas (Nypan, Gade, and Maseng, 2001; Nypan, Gade, and Hallingstad, 2002). More common techniques are based on observation of Earth-fixed features, and cameras, lasers or radars are often used for this purpose above water. Under water, acoustic waves are usually preferred, and a common method is to compare measurements from single- or multibeam echosounders with an existing bathymetric map (Nygren and Jansson, 2004; Ånonsen, 2010; Di Massa, 1997). For cases where no map exists, mapping the bathymetry (and/or intensity of reflected signals) can still be useful to limit the positional drift, if the same area is visited more than once (Williams, Dissanayke, and Durrant-Whyte, 2001; Newman, Leonard, and Rikoski, 2005).

Bibliography

Britting, K. R. (1971). *Inertial Navigation Systems Analysis*. Wiley Interscience

Bellettini, A. and Pinto, M. A. (2002). Theoretical accuracy of synthetic aperture sonar micronavigation using a displaced phase-center antenna, *IEEE J. Oceanic Eng.* vol. 27, no. 4, pp. 780–789

Craig, J. J. (1989). *Introduction to Robotics: Mechanics and Control*. 2nd edition. Addison-Wesley Publishing Company, Inc.

Crampin, M. and Pirani, F. A. E. (1986). *Applicable Differential Geometry*. Cambridge University Press

Di Massa, D. E. (1997). *Terrain-Relative Navigation for Autonomous Underwater Vehicles*. Ph.D. thesis, Massachusetts Institute of Technology

Egeland, O. and Gravdahl, J. T. (2002). *Modeling and simulation for automatic control*. Marine Cybernetics AS

Eustice, R. M., Whitcomb, L. L., Singh, H. and Grund, M. (2007). Experimental results in synchronous-clock one-way-travel-time acoustic navigation for autonomous underwater vehicles. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4257–4264, Rome, Italy

Gade, K. and Jalving, B. (1999). An Aided Navigation Post Processing Filter for Detailed Seabed Mapping. *Modeling, Identification and Control*, vol. 20, no. 3, pp. 165-176

Gade, K. (2005). NavLab, a Generic Simulation and Post-processing Tool for Navigation. *Modeling, Identification and Control*, vol. 26, no. 3, pp. 135-150

Gade, K. (2010). A Nonsingular Horizontal Position Representation. *The Journal of Navigation*, vol. 63, no. 3, pp. 395-417

Gade, K. (2016). The Seven Ways to Find Heading. *The Journal of Navigation*, vol. 69, no. 5, pp. 955-970

Gade, K. (2017). The *n*-vector page, http://www.navlab.net/nvector Accessed 7 June 2017.

Gelb, A. (1974). *Applied optimal estimation*. MIT press

Goldenberg, F. (2006). Geomagnetic Navigation beyond the Magnetic Compass. In *Position, Location, and Navigation Symposium*, 2006 IEEE/ION, pp. 684-694

Goldstein, H. (1980). *Classical Mechanics*. 2nd edition. Addison-Wesley

Groves, P. D. (2013). *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. 2nd edition. Artech house

Hagen, O. K., Ånonsen, K. B. and Mandt, M. (2010). The HUGIN Real-Time Terrain Navigation System, *Proceedings of the MTS/IEEE Oceans 2010 Conference*. Seattle, WA, USA

Hagen, P. E., Hansen, R. E., Gade, K. and Hammerstad, E. (2001). Interferometric Synthetic Aperture Sonar for AUV Based Mine Hunting: The SENSOTEK project. *Proceedings from Unmanned Systems 2001*, Baltimore, MD, USA

Hansen, R. E., Sæbø, T. O., Gade, K. and Chapman, S. (2003). Signal Processing for AUV Based Interferometric Synthetic Aperture Sonar. *Proceedings of the MTS/IEEE Oceans 2003*, vol. 5, pp. 2438-2444. San Diego, CA, USA

Hansen, R. E. (2011). Introduction to Synthetic Aperture Sonar. *Sonar Systems*, ed. by N. Z. Kolev, pp. 3-28. Intech, http://www.intechopen.com/books/sonar-systems Accessed 7 June 2017.

Hayes, M. P. and Gough, P. T. (2009). Synthetic aperture sonar: A review of current status, *IEEE J. Oceanic Eng.* vol. 34, no. 3, pp. 207–224

Hegrenæs, Ø., Hallingstad, O. and Gade, K. (2007). Towards Model-Aided Navigation of Underwater Vehicles. *Modeling, Identification and Control*, vol. 28, no. 4, pp. 113-123

Hegrenæs, Ø., Gade, K., Hagen, O. K. and Hagen, P. E. (2009). Underwater Transponder Positioning and Navigation of Autonomous Underwater Vehicles. *Proceedings of the IEEE Oceans 2009*, Biloxi, MS, USA

Hovde, S. T. (2017). Compact Sensor System for Target Localization. *Master thesis*. Faculty of mathematics and natural sciences. University of Oslo.

Jalving, B. and Gade, K. (1998). Positioning Accuracy for the HUGIN Detailed Seabed Mapping UUV. *Proceedings of the IEEE Oceans '98*, pp. 108-112. Nice, France

Jalving, B., Gade, K., Hagen, O. K. and Vestgård, K. (2004). A Toolbox of Aiding Techniques for the HUGIN AUV Integrated Inertial Navigation System. *Modeling, Identification and Control*, vol. 25, no. 3, pp. 173-190

Jalving, B., Gade, K., Svartveit, K., Willumsen, A. and Sørhagen, R. (2004). DVL Velocity Aiding in the HUGIN 1000 Integrated Inertial Navigation System. *Modeling, Identification and Control*, vol. 25, no. 4, pp. 223-236

Kane, T. R., Likins, P. W. and Levinson, D. A. (1983). *Spacecraft dynamics*. McGraw-Hill

Kane, T. R. and Levinson, D. A. (1985). *Dynamics: Theory and Applications*. McGraw-Hill

Kelly, A. (2013). *Mobile Robotics – Mathematics, Models, and Methods*. Cambridge University Press

Kongsberg Seatex (2017). Motion Reference Unit (MRU) product line, www.km.kongsberg.com/mru Accessed 7 June 2017.

Kristensen, J. and Vestgård, K. (1998). Hugin - an untethered underwater vehicle for seabed surveying. *Proceedings of the IEEE OCEANS '98*, pp. 118-123. Nice, France

Mandt, M., Gade, K. and Jalving, B. (2001). Integrating DGPS-USBL position measurements with inertial navigation in the HUGIN 3000 AUV. *Proceedings of the 8th Saint Petersburg International Conference on Integrated Navigation Systems*, pp. 63-74. Saint Petersburg, Russia

MathWorks (2017). MATLAB, www.mathworks.com Accessed 7 June 2017.

McGill, D. J. and King, W. W. (1995). *Engineering Mechanics*. 3rd edition, PWS-KENT, Boston

Minkler, G. and Minkler, J. (1993). *Theory and applications of Kalman filtering*. Magellan Book Company

Mirman, R. (1995). *Group theory: An intuitive approach*. World Scientific Publishing Co Inc.

Munkres, J. R. (2000). *Topology* (2nd edition). Prentice Hall.

Newman, P. M., Leonard, J. J. and Rikoski, R. J. (2005). Towards constant-time SLAM on an autonomous underwater vehicle using synthetic aperture sonar. In: Dario P., Chatila R. (eds) *Robotics Research*. *The Eleventh International Symposium*. pp. 409-420. Springer Tracts in Advanced Robotics. Springer, Berlin, Heidelberg.

Nygren, I. and Jansson, M. (2004). Terrain navigation for underwater vehicles using the correlator method. *IEEE Journal of Oceanic Engineering*, vol. 29, no. 3, pp.906-915

Nypan, T., Gade, K. and Maseng, T. (2001). Location using Estimated Impulse Responses in a Mobile Communication System. *Proceedings of the NORSIG*, Trondheim, Norway

Nypan, T., Gade, K. and Hallingstad, O. (2002). Vehicle positioning by database comparison using the Box-Cox metric and Kalman filtering. *Proceedings of the VTC Spring*, pp. 1650-1654, AL, USA

Snyder, J. P. (1987). *Map projections--A working manual* (U.S. Geological survey professional paper 1395). United States Government Printing Office

Spong, M. W. and Vidyasagar, M. (1989). *Robot dynamics and control*. John Wiley & sons

Storms, W., Shockley, J. and Raquet, J. (2010). Magnetic field navigation in an indoor environment. *Proceedings of the Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*, IEEE, Kirkkonummi, Finland

Størkersen, N., Kristensen, J., Indreeide, A., Seim, J. and Glancy, T. (1998). Hugin - UUV for seabed surveying, *Sea Technology*, vol. 39, no. 2, pp.99-104

Wang, L., Bellettini, A., Hollett, R., Tesei, A., Pinto, M., Chapman, S. and Gade, K. (2001). InSAS'00: Interferometric SAS and INS aided SAS imaging. *Proceedings of the MTS/IEEE Oceans 2001*, pp. 179-187, Honolulu, HI, USA

Williams, S., Dissanayake, G. and Durrant-Whyte, H. (2001). Towards terrain-aided navigation for underwater robotics. *Advanced Robotics*, vol. 15, no. 5, pp.533-549

Ånonsen, K. B. (2010). *Advances in terrain aided navigation for underwater vehicles*. Ph.D. thesis, Norwegian University of Science and Technology, Trondheim, Norway

Papers

The eight papers, listed in Section 1.2, are included in the following pages.

Paper I

Gade, K. (2010). A Nonsingular Horizontal Position Representation. *The Journal of Navigation*, vol. 63, no. 3, pp. 395-417

THE JOURNAL OF NAVIGATION (2010), **63**, 395–417. © The Royal Institute of Navigation
doi:10.1017/S0373463309990415

A Non-singular Horizontal Position Representation

Kenneth Gade

(*Norwegian Defence Research Establishment* (*FFI*))
(Email: Kenneth.Gade@ffi.no)

Position calculations, e.g. adding, subtracting, interpolating, and averaging positions, depend on the representation used, both with respect to simplicity of the written code and accuracy of the result. The latitude/longitude representation is widely used, but near the pole singularities, this representation has several complex properties, such as error in latitude leading to error in longitude. Longitude also has a discontinuity at ±180°. These properties may lead to large errors in many standard algorithms. Using an ellipsoidal Earth model also makes latitude/longitude calculations complex or approximate. Other common representations of horizontal position include UTM and local Cartesian 'flat Earth' approximations, but these usually only give approximate answers, and are complex to use over larger distances. The normal vector to the Earth ellipsoid (called *n*-vector) is a non-singular position representation that turns out to be very convenient for practical position calculations. This paper presents this representation, and compares it with other alternatives, showing that *n*-vector is simpler to use and gives exact answers for all global positions, and all distances, for both ellipsoidal and spherical Earth models. In addition, two functions based on *n*-vector are presented, that further simplify most practical position calculations, while ensuring full accuracy.

KEY WORDS

1. Position representations. 2. Position calculations. 3. Implementation simplicity. 4. Non-singular representation.

1. INTRODUCTION. Calculations involving global position, i.e. position relative to the Earth, are central in many fields, such as navigation, radar/sonar calculations, geodesy, and vehicle guidance and control. In these calculations, the position can be represented by different mathematical quantities, each with its own properties. There are several well-known representations for global position, such as latitude/longitude, UTM (Universal Transverse Mercator (Snyder, 1987)), and Cartesian 3D vector (Earth-Centred-Earth Fixed). These position representations will be discussed in Section 3, focusing on their limitations, and how their properties may induce significant errors in common calculations. Many of the problems and limitations of these alternatives are avoided if using the normal vector to the Earth ellipsoid (called *n*-vector) to represent the position. Although this representation has been briefly mentioned in some texts (e.g. Aeronautical Systems Div Wright-Patterson AFB OH, 1986) a thorough presentation of this alternative,

including comparisons with the more well known representations is not found in the literature. This paper will present the *n*-vector alternative by first discussing the geometrical properties of *n*-vector in Section 4. Section 5 presents various *n*-vector calculations, illustrating the fact that calculations involving *n*-vector are in general remarkably simple. To simplify implementation further, two functions are presented, which turn out to cover a majority of practical position calculations. In Section 6, several examples comparing the use of latitude/longitude with *n*-vector for specific calculations are studied. The practical usefulness of *n*-vector in real-life applications is the topic of Section 7, where the experience is that research groups prefer using *n*-vector in many of their position calculations after becoming familiar with it.

2. NOTATION.

A unified and stringent notation is of utmost importance when describing the kinematics of multiple rotating systems, and a full notation system with definitions of the central quantities has been developed by the author (to be published). A short, simplified extract from this system, with only the symbols relevant in this paper, is given below.

2.1. *Coordinate frame.* A coordinate frame is defined as a combination of a point (origin), representing position, and a set of basis vectors, representing orientation. Thus, a coordinate frame has 6 degrees of freedom and can be used to represent the position and orientation of a rigid body. A listing of the specific coordinate frames relevant in this paper is found in Appendix A.

2.2. *General notation.* A general vector can be represented in two different ways (McGill and King, 1995), (Britting, 1971):

- $\vec{x}$ (Lower case letter with arrow): Coordinate free/geometrical vector (not decomposed in any coordinate frame)
- $\boldsymbol{x}^A$ (Bold lower case letter with right superscript): Vector decomposed/represented in a specific frame (column matrix with three scalars)

The physical world to be described by the kinematics is modelled in terms of coordinate frames. Hence, quantities such as position, angular velocity, etc. relate one coordinate frame to another. To make a quantity unique, the two frames in question are given as right subscript, as shown in Table 1. Note that in most examples in Table 1, only the position or the orientation of the frame is relevant, and the context should make it clear which of the two properties is relevant (for instance, only the *orientation* of a frame written as right superscript is relevant, since it denotes the frame of decomposition, where only the direction of the basis vectors matters). If both the position and the orientation of the frame are relevant, the frame is underlined to emphasize that fact. For generality, the vectors in Table 1 are written in coordinate free form, but before implementation in a computer, they must be decomposed in a selected frame (e.g. $\boldsymbol{\omega}_{AB}^{C}$ is the angular velocity $\vec{\omega}_{AB}$ decomposed in frame *C*).

3. STANDARD POSITION REPRESENTATIONS.

Before introducing *n*-vector, the standard position representations are discussed as a background for comparison.

Table 1. Symbols used to describe basic relations between two coordinate frames.

Quantity	Symbol	Description
Position vector	$\vec{p}_{AB}$	A vector whose length and direction is such that it goes from the origin of frame A to the origin of frame B, i.e. the position of B relative to A.
Velocity vector	$\vec{v}_{\underline{A}B}$	The velocity of the origin of frame B, relative to frame A. The underline indicates that both the position and orientation of A is relevant (whereas only the position of B matters).
Rotation matrix	$\boldsymbol{R}_{AB}$	A 3x3 direction cosine matrix (DCM) describing the orientation of frame B relative to frame A.
Angular velocity	$\vec{\omega}_{AB}$	The angular velocity of frame B relative to frame A.

3.1. *Cartesian 3D vector*. When representing the position of a general coordinate frame B relative to a reference coordinate frame A, the most intuitive quantity to use is the position vector from A to B, decomposed in A, $\boldsymbol{p}_{AB}^{A}$. This paper focuses on global positioning, and using the frames defined in Appendix A, we can represent the position of a body frame (B) relative to the Earth (E), by using $\boldsymbol{p}_{EB}^{E}$. This (Cartesian) vector is often referred to as Earth Centred Earth Fixed (ECEF) vector. While this representation is non-singular and intuitive, there are many situations where other representations are more practical when positioning an object relative to the Earth reference ellipsoid.

3.2. *Separating horizontal and vertical components*. For many position calculations, it is desirable and most intuitive to treat *horizontal* and *vertical* positions independently. This is for instance useful in a navigation system, where horizontal and vertical position are usually measured by different sensors at different points in time, or in a vehicle autopilot, where horizontal and vertical position are often controlled independently. In such applications, we usually compare two horizontal positions, and thus we need a quantity for representing horizontal position independently of the vertical height/depth. It should thus be possible to represent horizontal position without considering the vertical position, and vice versa. If the vector $\boldsymbol{p}_{EB}^{E}$ is used, the horizontal and vertical positions are not separated as desired.

3.2.1. *Latitude and longitude*. A common solution for obtaining separate horizontal and vertical positions is the use of latitude, longitude and height/depth (related to a *reference ellipsoid*, discussed in Section 4.1). However, this representation has a severe limitation; the two singularities at latitudes $\pm 90^\circ$, where longitude is undefined. In addition, when getting close to the singularities, the representation exhibits considerable non-linearities and extreme latitude dependency, leading to reduced accuracy in many algorithms, as exemplified in Section 6. Thus, these coordinates are not suitable for algorithms that should be able to calculate positions far north or far south. In addition, calculations near $\pm 180^\circ$ longitude become complicated due to the discontinuity.

3.2.2. *Local Cartesian coordinate frame* (*flat Earth assumption*). Another common solution for separating the horizontal and vertical components is to introduce a local Earth-fixed Cartesian coordinate frame, with two axes forming a horizontal tangent plane to the reference ellipsoid at a specified tangent point. Assuming several calculations are needed in a limited area, position calculations can be performed relative to this system to get approximate horizontal and vertical components. This coordinate frame is not used as a global position representation (since the local origin (tangent point) must still be represented relative to the Earth), but is rather a way to get horizontal and vertical directions in the local position calculations.

However, the local Cartesian representation corresponds to a local flat Earth assumption and does not give exact horizontal and vertical directions for positions that are not directly above or below the tangent point. The further away from the tangent point the calculations are done, the greater the error in the horizontal and vertical directions. In an application with e.g. moving vehicles, the system typically has to be repositioned regularly in order to minimize these errors.

Finally, most local Cartesian frames are aligned with the north/east directions at the tangent point, and are often treated as a linearization of the meridians and parallels. However, when getting close to the poles the linearization is sufficiently accurate only for a very small area. Here, an error in the assumed position can give errors in the assumed north/east directions as well. At the pole points the north and east directions are undefined. Thus, calculations operating with a north/east aligned coordinate frame can generate significant errors in the polar regions.

3.2.3. *UTM and UPS.* Horizontal position can also be represented by defining an Earth fixed coordinate system based on a map projection (i.e., a mapping of points on a curved surface to a plane) valid in a limited geographical area. One such system is Universal Transverse Mercator (UTM), specifying 60 longitude zones, covering the globe except for the polar regions (Snyder, 1987). For the polar regions, a similar system, Universal Polar Stereographic (UPS), defines horizontal positions (Hager et al., 1989). While these systems are well-defined and the coordinate values approximately correspond to metres, they have an inherent distortion due to the projection and thus a corresponding error in many calculations (e.g. a difference vector between two UTM coordinates will give a length (in metres) and direction (relative to north) that both have errors compared to the true values). In addition, general calculations get very complex when crossing zones (Hager et al., 1989).

3.2.4. *Rotation matrix.* In a set of navigation equations, integrating measurements from an inertial measurement unit, horizontal position is often stored together with an azimuth angle in a rotation matrix (Savage, 2000). Although it has nice properties with respect to the pole singularities (similar to *n*-vector), this matrix representation is not suited for pure horizontal position representation. More about this alternative is found in Section 5.5.

4. *n*-VECTOR.

We will seek an alternative for representing horizontal position. The vertical position representation (height/depth from the reference ellipsoid) is very convenient and will still be used. It should be noted that the terms *horizontal* and *vertical* directions implicitly introduce a reference surface, and the terms are valid for a given point at the surface. The horizontal direction is given by the surface tangent plane (2D) and the vertical direction is the normal to the surface (1D). Thus, the task of finding a non-singular representation of horizontal position can in general be viewed as finding a suitable representation of 2D position on a surface.

We define a surface as *associated* with a coordinate frame, if the surface is fixed relative to the coordinate frame. We also define a surface as *strictly convex* if it is or can be extended to a closed surface whose enclosed volume is strictly convex. Realising that the 2D position on a strictly convex surface can be uniquely

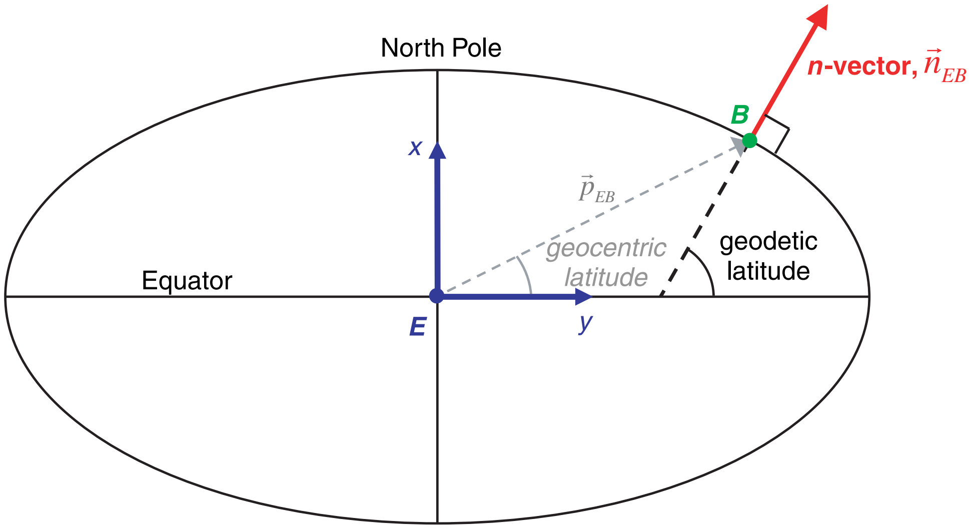


Figure 1. Earth reference ellipsoid with n-vector, geodetic and geocentric latitude. Two axes and the origin of the E-frame (blue) and the origin of the B-frame (green) are also shown.

represented by the normal vector to the surface, leads to the idea of using this vector as a position representation.

- Definition of n-vector:

A strictly convex and differentiable surface is associated with coordinate frame A. A coordinate frame B is located at the surface. The n-vector representation of the position of B relative to A is defined as the outward pointing normal vector of the surface at the B-position, with unit length. The n-vector is denoted as $\vec{n}_{AB}$.

The surface might consist of several patches/pieces, as long as they can be extended to a closed surface with a strictly convex interior. Since the surface is differentiable, the boundary of each piece is not part of the surface (the edges are open). In the definition above, B is at the surface since n-vector is a representation of horizontal position only. How to use the n-vector for 3D positions is discussed in Section 4.3.

4.1. *Reference ellipsoid.* Note that the alternatives for global positioning discussed in Section 3 are defined relative to a *reference ellipsoid*, e.g. WGS-84 (National Imagery and Mapping Agency, 2000). When using n-vector to represent global position (i.e. position relative to the Earth frame, E), it must also relate to a reference ellipsoid. The reference ellipsoid is a surface that is both strictly convex and differentiable, and for n-vector this ellipsoid is the surface associated with E. The vehicle/object to be positioned is typically denoted B (Body), and thus the n-vector for global positioning is denoted $\vec{n}_{EB}$. Hence, $\vec{n}_{EB}$ conveniently represents horizontal position at the Earth surface without singularities.

Figure 1 shows n-vector as the normal to the reference ellipsoid surface. Note also, as shown, that n-vector corresponds to *geodetic* latitude (Snyder, 1987). Geodetic latitude is the latitude most commonly used, and when using the term *latitude* in the rest of this document, this always means geodetic latitude.

4.2. *One-to-one property.* n-vector is a one-to-one representation, i.e. any normal vector corresponds to one unique surface position, and any surface position

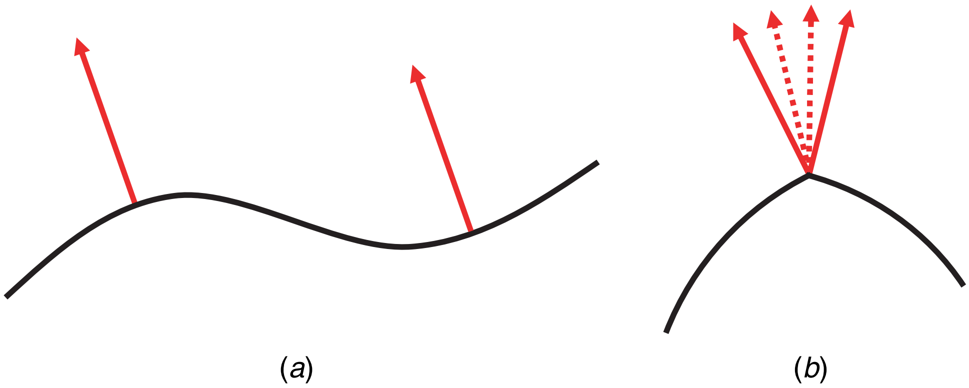


Figure 2. *a.* The surface is differentiable, but not strictly convex: One *n*-vector corresponds to many positions. *b.* The surface is strictly convex, but not differentiable: Many *n*-vectors correspond to one position.

corresponds to one unique *n*-vector. This one-to-one property is not held by various other representations, such as latitude/longitude (many-to-one), roll/pitch/yaw (many-to-one) or quaternions (Zipfel, 2000) (two-to-one).

- *Note 1*. If the surface is closed, any unit vector will be a valid *n*-vector, and thus it will correspond to one unique position.
- *Note 2*. The one-to-one property of *n*-vector is due to the strict convexness and differentiability of the associated surface:

  a. If the surface had not been strictly convex, *n*-vector would be one-to-many. (In addition, the lack of strict convexness means that the term 'outward pointing' is not defined all over the surface, and in general *n*-vector cannot be uniquely defined for such surfaces.)
  b. If the surface had not been differentiable, *n*-vector would be many-to-one.

The two cases are illustrated in Figure 2.

4.3. *3D positions*. So far, *n*-vector has been used to represent a position on the reference surface, but in the same manner as for latitude/longitude, a 3D position can be represented by adding a height/depth parameter (above/below the nearest part of the reference ellipsoid surface). Mathematically, this is a very convenient combination since the *n*-vector defines the exact direction in space where the height/depth is valid (e.g., if we need the vector from the ellipsoid surface to a given 3D position, it is simply found as the product of *n*-vector and the height). When B is not at the surface, the B in $\vec{n}_{EB}$ represents the horizontal position of B. As mentioned in Section 3.2, it is usually most convenient with separate horizontal and vertical position representations. However, this is not the case if working with positions near[1] the Earth's centre, since the horizontal and vertical directions are not defined there.

5. *n*-VECTOR CALCULATIONS.

We have seen that from a geometrical point of view $\vec{n}_{EB}$ works very well as a representation of horizontal position. The

[1] Positions where the depth is equal to or greater than the ellipsoid radius of curvature.

overall usefulness also depends on how easy it is to work with *n*-vector in practical calculations. In most practical calculations, *n*-vector is decomposed in the *E*-frame, i.e. $\boldsymbol{n}_{EB}^{E}$. If, for example, we want to represent the South Pole, we see from the definition of the *E*-frame in Appendix A that $\mathbf{n}_{EB}^{E} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$. It should also be noted that *n*-vector has approximately the same direction as the position vector $\boldsymbol{p}_{EB}^{E}$ (see Figure 1 where the ellipticity is exaggerated). If assuming spherical Earth, the directions are equal, i.e.

$$\mathbf{n}_{EB}^{E} = \frac{\mathbf{p}_{EB}^{E}}{\left|\mathbf{p}_{EB}^{E}\right|} \tag{1}$$

where | | denotes the vector length (vector 2-norm).

Note that for all equations in this paper that are valid only for spherical Earth, this is stated directly before the equation. All other equations in this paper are valid for both ellipsoidal and spherical Earth.

5.1. *Simplified notation.* For the general quantities in Table 1 there are many possible frames (such as the position of a sensor relative to a vehicle), but for global position, the frames are often *E* and *B*. If all positioning is relative to Earth (*E*) and only one object is positioned, the subscript is redundant and can be omitted, so we can simply use $\boldsymbol{n}^{E}$. This is similar to situations where it is sufficient to use only the variables *latitude* and *longitude* (without further specification).

5.2. *Converting to or from latitude and longitude.* The relations between latitude/longitude and *n*-vector are found in this section. Note that the relations are valid for any reference ellipsoid or sphere.

Mathematically, the longitude and latitude are the first two angles of a *x-y-z* Euler angle representation of the orientation of a local level frame (such as *N* or *L*, see Appendix A) relative to *E*. The latitude (λ) and longitude (μ) have the following dynamical intervals:

$$\begin{aligned} &\lambda \in [-\pi/2, \pi/2] \\ &\mu \in (-\pi, \pi] \end{aligned} \tag{2}$$

5.2.1. *From latitude and longitude to n-vector.* By observing simple geometry, we get the following relation:

$$\boldsymbol{n}^{E} = \begin{bmatrix} \sin(\lambda) \\ \sin(\mu)\cdot\cos(\lambda) \\ -\cos(\mu)\cdot\cos(\lambda) \end{bmatrix} \tag{3}$$

This equation has no singularities, i.e., one can always calculate a unique *n*-vector from a set of latitude and longitude. At the poles ($\lambda = \pm\pi/2$), a zero factor, cos(λ), in both the *y* and *z* components makes the actual value of the (undefined) longitude irrelevant. It is also clear from (3) how the discontinuity of longitude at $\pm\pi$ is eliminated when using *n*-vector, since both cos( ) and sin( ) are continuous for an angle going through this value. Note that the order and signs of the vector elements in (3) obviously depend on the choice of *E*-frame axes, see Appendix A.

5.2.2. *From n-vector to latitude and longitude.* From the geometry, we get the following relations:

$$\lambda = \arcsin\left(n_x^E\right) \tag{4}$$

$$\mu = \arctan\left(n_y^E, -n_z^E\right) \tag{5}$$

where arctan(b,a) is the four quadrant version of arctan(b/a). The x, y and z subscripts indicate the three components of *n*-vector. The longitude singularity at the poles is apparent in the arctan( ) expression, which is undefined for the input (0,0) (however, in practical programming languages, a default output is usually returned also in this case, making it possible to convert back to *n*-vector with (3) also at the poles).

- Implementation considerations. Equation (4) is not recommend for implementation, since the arcsin( ) function is numerically inaccurate near the poles[2] and also will return imaginary results if the input should be outside ±1 due to numerical inaccuracy. An equivalent alternative that is robust against numerical errors is given in (6).

$$\lambda = \arctan\left(n_x^E, \sqrt{\left(n_y^E\right)^2 + \left(n_z^E\right)^2}\right) \tag{6}$$

Note that the four quadrant arctan(,) is used even when we know that latitude is limited to two of the quadrants (see (2)) to avoid division by zero at the poles. Because the second parameter is non-negative, this function will always return answers in the correct quadrants.

5.2.3. *Quaternion comparison.* Orientation (three degrees of freedom) is often represented by three Euler angles or three other parameters (Craig, 1989). However, all three-parameter orientation representations have singularities (Stuelplnagel, 1964), and by using 4-parameter quaternions (Zipfel, 2000), the singularities are avoided. The quaternion has a restriction of unit length to ensure it only has three degrees of freedom.

Similarly, a surface position has two degrees of freedom, and can be represented by two parameters with singularities, such as latitude and longitude. *n*-vector adds a third parameter to avoid the singularities, and also has a restriction of unit length. Converting between Euler angles and quaternions yields equations similar to (3), (4), and (5), i.e. the quaternion is found by products of sin( ) and cos( ), and the reverse equations consist of arctan( ) and arcsin( ) functions.

5.3. *n-vector relations.* This section includes several useful calculations involving *n*-vector, illustrating its properties.

5.3.1. *Horizontal and vertical parts of an arbitrary vector.* Using *n*-vector, it is easy to find the vertical and horizontal parts of an arbitrary vector $\boldsymbol{k}^E$,

$$\boldsymbol{k}_{vertical}^E = \left(\boldsymbol{n}^E \cdot \boldsymbol{k}^E\right)\boldsymbol{n}^E \tag{7}$$

[2] The arcsin( ) is in general numerically inaccurate for calculating angles close to $\pm\pi/2$, just as arccos( ) is inaccurate close to zero and π. Arctan( ) is accurate for all angles.

The horizontal part is found by subtracting the vertical part,

$$\boldsymbol{k}^{E}_{horizontal} = \boldsymbol{k}^{E} - \left(\boldsymbol{n}^{E} \cdot \boldsymbol{k}^{E}\right)\boldsymbol{n}^{E} \tag{8}$$

5.3.2. *North and east directions.* Outside the poles, the (horizontal) directions of north and east are often of interest. The east direction (normal to the meridian plane) is simply given by

$$\boldsymbol{k}^{E}_{east} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \times \boldsymbol{n}^{E} \tag{9}$$

Similarly, the north direction (normal to the transverse plane[3]) is given by

$$\boldsymbol{k}^{E}_{north} = \boldsymbol{n}^{E} \times \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \times \boldsymbol{n}^{E} \tag{10}$$

Since the triple cross product is associative when the first and last vectors are equal, no parentheses are needed to specify the order of operation.

The rotation matrix relating the *N* and *E* frames is useful in many calculations, and is found from (9) and (10),

$$\boldsymbol{R}_{EN} = \begin{bmatrix} \dfrac{\boldsymbol{k}^{E}_{north}}{\left|\boldsymbol{k}^{E}_{north}\right|} & \dfrac{\boldsymbol{k}^{E}_{east}}{\left|\boldsymbol{k}^{E}_{east}\right|} & -\boldsymbol{n}^{E} \end{bmatrix} \tag{11}$$

5.3.3. *Angular velocity of local frame.* The angular velocity of a local frame *L* (see Appendix A) relative to *E*, $\boldsymbol{\omega}^{E}_{EL}$, is a useful quantity, for instance in a navigation system. From the linear velocity of *B* relative to *E*, $\boldsymbol{v}^{E}_{\underline{E}B}$, and the meridian and transverse radius of curvature at the current height ($r_{roc,meridian}$, $r_{roc,transverse}$), this angular velocity is given by

$$\boldsymbol{\omega}^{E}_{EL} = \boldsymbol{n}^{E} \times \left(\frac{\boldsymbol{v}^{E}_{\underline{EB},\,north}}{r_{roc,\,meridian}} + \frac{\boldsymbol{v}^{E}_{\underline{EB},\,east}}{r_{roc,\,transverse}} \right) \tag{12}$$

Equation (12) is valid for an ellipsoidal Earth model, while for a spherical model, the relation is simply

$$\boldsymbol{\omega}^{E}_{EL} = \boldsymbol{n}^{E} \times \left(\frac{\boldsymbol{v}^{E}_{\underline{EB}}}{r_{roc}} \right) \tag{13}$$

where r_{roc} is the radius of curvature, i.e. Earth radius + height.

5.3.4. *Derivative of n-vector and height/depth.* The angular velocity found in (12) or (13) can be used directly to describe the derivative (with respect to time) of *n*-vector (also for elliptical Earth),

$$\dot{\boldsymbol{n}}^{E} = \boldsymbol{\omega}^{E}_{EL} \times \boldsymbol{n}^{E} \tag{14}$$

The vertical part of the angular velocity vector does not affect the update of *n*-vector, and hence the angular velocity of any local level coordinate frame could be used

[3] The transverse plane is normal to the meridian plane and contains *n*-vector.

in (14), for instance of N (i.e. $\boldsymbol{\omega}_{EN}^{E}$). The derivative of n-vector is useful for instance when integrating velocity to get position as n-vector (as will be done in Section 6.5).

If integrating to get 3D position, an update of the height/depth would also be needed. Updating height (h) is simple when knowing n-vector,

$$\dot{h} = \boldsymbol{n}^{E} \cdot \boldsymbol{v}_{\underline{E}B}^{E} \tag{15}$$

5.3.5. *Surface distance.* If assuming spherical Earth, the surface distance (length of geodesic) between two positions (given by $\boldsymbol{n}_{EA}^{E}$ and $\boldsymbol{n}_{EB}^{E}$), is easy to find by utilizing the properties of the dot and cross products,

$$\begin{aligned} s_{AB} &= \arccos\left(\boldsymbol{n}_{EA}^{E} \cdot \boldsymbol{n}_{EB}^{E}\right) \cdot r_{roc} \\ &= \arcsin\left(\left|\boldsymbol{n}_{EA}^{E} \times \boldsymbol{n}_{EB}^{E}\right|\right) \cdot r_{roc} \end{aligned} \tag{16}$$

where s_{AB} is the surface distance between A and B. For implementation, note that the arccos( ) expression is ill-conditioned for small angles, and the arcsin( ) expression is ill-conditioned for angles near $\pi/2$ (and not valid above $\pi/2$). Full numerical accuracy for all angles is achieved by combining the two expressions into an arctan(,) expression as before (see Section 5.2.2.).

5.3.6. *Horizontal geographical mean.* if m horizontal positions are given as n-vectors, the geographical mean, B_{GM}, is simply given by (assuming spherical Earth)

$$\boldsymbol{n}_{EB_{GM}}^{E} = \text{unit}\left(\sum_{i=1}^{m} \boldsymbol{n}_{EB_i}^{E}\right) \tag{17}$$

where $\boldsymbol{n}_{EB_i}^{E}$ is the i'th position, and 'unit( )' makes the input vector's length/magnitude equal to one. Equation (17) gives the exact answer for any set of global positions (except in the case where the horizontal mean is undefined, i.e. for a set of antipodal positions, cancelling each other).

5.4. *n-vector and Cartesian position vectors.* In many practical calculations, there is a need to combine global position with position differences given as Cartesian vectors (typically relative positions within a limited area). If assuming spherical Earth, the relations are simple when using n-vector. For elliptical Earth, exact calculations have almost the same complexity as (geodetic) latitude and longitude, since n-vector is also a geodetic quantity. However, such calculations can be easily handled in practice by using two general functions:

1. 'A and $B \Rightarrow$ delta position': Two global positions A and B are given as n-vectors ($\boldsymbol{n}_{EA}^{E}$ and $\boldsymbol{n}_{EB}^{E}$) with heights. The function calculates the position vector from A to B, $\boldsymbol{p}_{AB}^{E}$.
2. 'A and delta position $\Rightarrow B$': One global position is given as $\boldsymbol{n}_{EA}^{E}$ with height, and a position vector to the point B is given ($\boldsymbol{p}_{AB}^{E}$). The function calculates $\boldsymbol{n}_{EB}^{E}$ with height.

The implementation of the functions is described in Appendix B. It turns out that most calculations involving global position and local position vectors can easily be solved using these two functions (e.g. calculations involving bearing/elevation/range from a sonar/radar or when an estimated error or a lever-arm should be subtracted from a global position).

5.5. *Using the orientation of a local coordinate frame to represent position.* In (11) we saw that $\boldsymbol{R}_{EN}$ has minus *n*-vector as the last column, and thus this matrix contains horizontal position information. If replacing the singular *N*-frame with a non-singular *L*-frame (given in Appendix A), $\boldsymbol{R}_{EL}$ will be a non-singular position representation (also with minus *n*-vector as last column). $\boldsymbol{R}_{EL}$ is the matrix mentioned in Section 3.2.4., and since this matrix is often of interest in an inertial navigation system, it is also used for horizontal position representation (Savage, 2000). It has the same qualities as *n*-vector with respect to the pole singularities, but as a rotation matrix, it has six extra elements with one extra degree of freedom (the wander azimuth angle described in Appendix A) that for most position calculations are not relevant.

6. COMPARING LATITUDE/LONGITUDE AND *n*-VECTOR.

The latitude and longitude angles are Euler angles (see Section 5.2.) and they have singularities just like any set of three Euler angles representing orientation. For orientation, if calculations near or at the singular points might be relevant, the Euler angles are usually replaced with a quaternion or a rotation matrix, see for example (Fortescue et al., 2003), (Levine, 2000), (Phillips, 2004), or (Obaidat and Papadimitriou, 2003). The replacement is in these references motivated by the *common knowledge* that singularities give problems, and they do not study what would actually happen if trying to perform calculations using a singular representation near or at a singular point. In the following, we will demonstrate some of these problems by looking at some examples where we use latitude and longitude near or at a pole. Other important reasons for using *n*-vector, which are also important far away from the poles, are the ease of use, and the exact results obtained. Some examples will focus on these properties as well.

6.1. *Example 1, relative position.* A common calculation is to convert positions given by latitude/longitude (located within a limited area), to relative positions in a local metric grid, often with north and east axes (e.g. when an estimated position is compared with a measured position). In practice, such calculations often involve the equations

$$\begin{aligned} \Delta_{north} &= (\lambda_B - \lambda_A) r_{roc} \\ \Delta_{east} &= (\mu_B - \mu_A) r_{roc} \cos(\lambda_C) \end{aligned} \tag{18}$$

for two positions A and B, where λ_C typically is one of the involved latitudes, or an average.

We immediately see that the discontinuity of longitude at ±180° can lead to large errors in (18) (but this problem can be handled by adding specific code). Problems that are more serious would appear if trying to use (18) near one of the poles. If the two latitudes were at opposite sides of a pole, the north distance would be wrong. Near a pole, the east distance in (18) will be along a clearly curved line and it will also depend heavily on the latitude used. If one of the positions is at the pole, the longitude is undefined and calculating delta east is problematic. In fact, when near a pole, the north and east directions will vary considerably even within a limited area, and at the polar point, the directions are undefined.

With *n*-vector, the vector difference is decomposed in E, with no problems for any positions. It is found by a simple vector difference multiplied with r_{roc} for spherical Earth, or by the function in Section 5.4 for ellipsoidal Earth. If the delta north and east components are desired (away from the poles), the difference vector is simply multiplied with (11).

6.2. *Example 2, surface distance.* A typical calculation for many applications is to find the surface distance (length of geodesic) between two horizontal positions. Even if assuming spherical Earth, calculating the great circle distance between two positions requires several steps to find exactly from latitudes and longitudes (an approximation is often found by square summing the deltas in (18)). The resulting expression found in many books, such as (Longley et al., 2005), (Weisstein, 2003) and (Hofmann-Wellenhof et al., 2003) gives the result as an arccos expression, see first part of (19). However, as discussed in Section 5.2.2 an implementation finding an angle from arccos will give numerical problems for small angles. In (Sinnott, 1984) an arcsin expression accurate for small angles is found (assuming spherical Earth). The two expressions for surface distance, s_{AB}, are

$$\begin{aligned} s_{AB} &= \arccos(\sin(\lambda_A)\sin(\lambda_B) + \cos(\lambda_A)\cos(\lambda_B)\cos(\mu_A - \mu_B)) \cdot r_{roc} \\ &= 2 \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\lambda_B - \lambda_A}{2}\right) + \cos(\lambda_A)\cos(\lambda_B)\sin^2\left(\frac{\mu_B - \mu_A}{2}\right)}\right) \cdot r_{roc} \end{aligned} \tag{19}$$

If using *n*-vector rather than latitude/longitude, we saw from (16) that it is easy to find the (non-singular) *n*-vector versions of both the arccos and arcsin expressions in (19).

6.3. *Example 3, horizontal geographical mean.* In several applications, it is interesting to find the geographical mean of multiple horizontal positions. If the positions are given as latitudes/longitudes, even when assuming spherical Earth, taking the arithmetical mean will give an answer that is only approximately correct for a small area, away from the poles and the $\pm 180^\circ$-line. Finding the exact answer is complicated when using latitudes and longitudes. On the other hand, if the positions are given as *n*-vectors, the geographical mean is simply given by (17).

6.4. *Example 4, interpolated position.* A variant of the above example is the calculation of an interpolated position. Using the standard formula for linear interpolation on the latitude/longitude coordinates will not give positions that are at the shortest path (geodesic) between the two original positions. Errors will increase near the poles and at larger distances. In addition, positions at each side of $\mu = \pm 180^\circ$ will give wrong answers. With *n*-vector, the standard formula for interpolation gives the correct result for all positions.

6.5. *Example 5, integrating velocity.* In several applications, such as dead-reckoning systems and simulators, the velocity of a vehicle/object is typically integrated to give global position. We shall investigate the error build-up in the integration process when using latitude/longitude or *n*-vector as the position representation.

The velocity vector to be integrated is $\vec{v}_{\underline{E}B}$, and when the vehicle position is represented by latitude/longitude, it is updated using north and east velocity. These are achieved by decomposing the velocity vector in the N frame, i.e. $\boldsymbol{v}^N_{\underline{E}B}$, and the derivatives of latitude and longitude can be found by (assuming spherical Earth)

$$\begin{aligned} \dot{\mu} &= \frac{v^N_{\underline{E}B,y}}{\cos(\lambda) \cdot r_{roc}} \\ \dot{\lambda} &= \frac{v^N_{\underline{E}B,x}}{r_{roc}} \end{aligned} \tag{20}$$

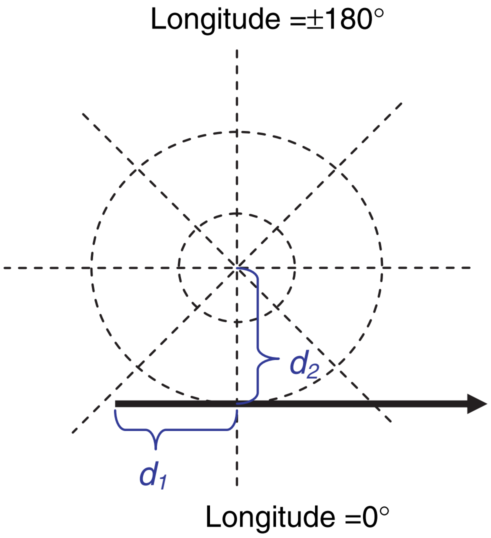


Figure 3. View from directly above the North Pole. The trajectory (passing the pole at d_2 metres distance) is shown in solid (with an arrowhead showing the direction). Dotted straight lines: constant longitudes. Dotted circles: constant latitudes.

We assume correct initial position and velocity input, and thus we only study the errors arising from the integration process itself.

6.5.1. *Part 1 – Demonstrating the error effects.* To visualise the error effects appearing when integrating latitude/longitude close to a pole, we will use an example where much error arises during few time steps. A ship such as an LCC (large crude carrier) has low dynamics, and we assume that 1 Hz forward or backward Euler integration is sufficiently accurate to integrate its position. We use spherical Earth model in this example and first assume that the ship follows a great circle with a speed of 7·5 m/s and zero height. It passes the North Pole with a minimum distance of 10 metres, as illustrated with d_2 in Figure 3, and we will look at a 50 seconds interval, where there are 20 seconds (corresponding to d_1) before passing at the closest distance.

To investigate the drift in an algorithm, a true trajectory is needed as a reference. With spherical Earth and no vehicle turning, it is easy to calculate the true trajectory analytically i.e. to directly calculate an exact true position and velocity for any point in time (we use the fact that seen from the *E* frame, the trajectory follows a great circle (geodesic) that is tilted relative to the meridians). The true trajectory is shown together with the results from the Euler algorithms in Figure 4.

Looking at the upper part of Figure 4, we see that the Euler algorithms follow the true latitude relatively well until the pole passing. Due to the high curvature in latitude during the pole passing, the algorithms get an error of about ±7·5 metres as shown in Figure 5 (which shows the errors converted to metres).

For longitude (Figure 4, lower part), the Euler algorithms also get problems when the graph starts curving, but far more serious is the error from the latitude dependency, see (20). The too large value in the forward Euler latitude in the last half of the pole passing causes the longitude to be increased far too much based on the velocity in (20). The opposite is true for the backward Euler method. Another weakness of the latitude/longitude representation is the high rate of change of longitude when close to the pole. A longitudinal error of only 0·01 metres at the closest point in the example

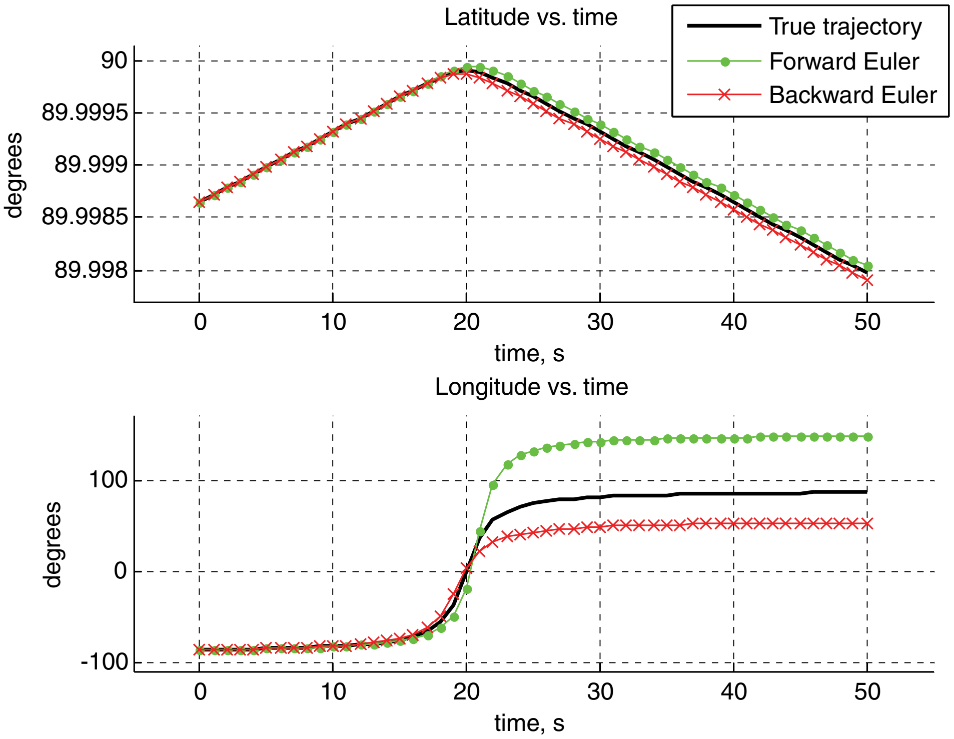


Figure 4. Latitude and longitude versus time. The true longitude and latitude are calculated directly from an exact function.

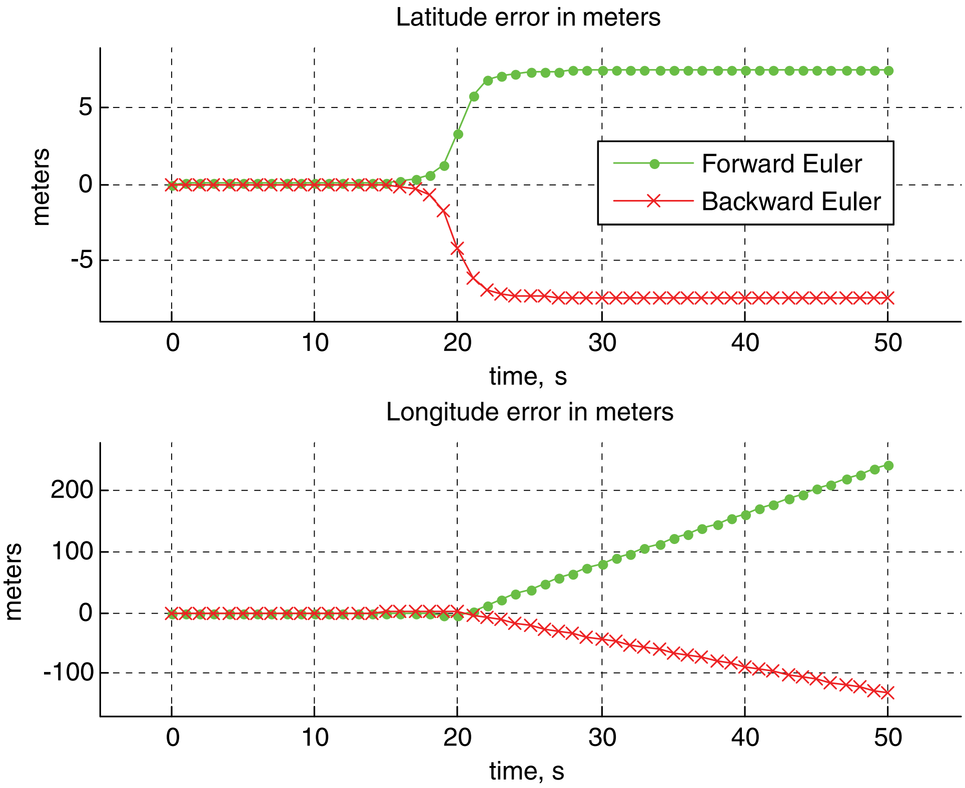


Figure 5. Error from the forward and backward Euler methods when using the latitude and longitude representation (calculated trajectory minus true trajectory).

corresponds to an error of about 6400 metres at the equator. Thus, small insignificant position errors generated close to the poles are magnified by many orders of magnitude when moving away from the pole. In the above example, the errors that arose during the pole-passing are magnified as shown in part 2 of Figure 5. Even without these effects, small errors in the initial position, in the velocity or in the timing (such errors are not included in the example), would be scaled to significant levels when increasing the distance from the pole.

6.5.1.1. *Higher order integration method.* Looking at the errors from forward or backward Euler, it is tempting to try a second order integration method like the trapezoid method. As expected, this reduces the error significantly (down to 56 metres/14° in longitude), but the error is probably still too large for most applications. Using a higher rate than 1 Hz also improves the result (as studied in Section 6.5.2) but for any rate, we could pass the pole at a shorter distance, and the error would again be unacceptable. This illustrates that the fundamental problem is the singularity of the latitude/longitude representation, and as we will see in the next section, replacing latitude/longitude with the non-singular *n*-vector is a far better solution than using a more complex integration method or a higher rate.

6.5.1.2. *Calculating position using n-vector.* Updating *n*-vector is done using the velocity decomposed in the *E*-frame, $\boldsymbol{v}^E_{\underline{E}B}$. Note that this is a more realistic and better suited velocity input than $\boldsymbol{v}^N_{\underline{E}B}$. $\boldsymbol{v}^E_{\underline{E}B}$ can be obtained by measuring Doppler shift from GPS or from underwater transponders with known position. While $\boldsymbol{v}^E_{\underline{E}B}$ has no error due to own position error, $\boldsymbol{v}^N_{\underline{E}B}$ is decomposed in the north and east directions, and in the polar regions these directions themselves will have errors given directly by the error in our assumed position. Combining (13) and (14) we find that the derivative of *n*-vector is calculated by (assuming spherical Earth)

$$\dot{\boldsymbol{n}}^E = \boldsymbol{n}^E \times \left(\frac{\boldsymbol{v}^E_{\underline{E}B}}{r_{roc}} \right) \times \boldsymbol{n}^E \tag{21}$$

Note that even if the full 3D vector $\boldsymbol{v}^E_{\underline{E}B}$ is used, only the horizontal component will contribute due to the cross product with $\boldsymbol{n}^E$.

Using the derivative as input, updating *n*-vector with the forward and backward Euler methods gives the result shown in Figure 6. The difference from the true trajectory is too small to be visible in this figure, but as we did for latitude and longitude, we can calculate the error in metres, shown in Figure 7. This is done using the calculated and true *n*-vector in (16) or by multiplying the difference vector with r_{roc}, which gives the same result for small angles. By comparing with Figure 5, we see that replacing latitude and longitude with *n*-vector has reduced the accumulated error from ca. 228 metres (great circle error, found by using (3) and (16)) to only $2{\cdot}1 \times 10^{-9}$ metres (both numbers are the largest error from either the forward or backward Euler method). The latter is at the level of errors from the computer's numerical precision used in the test, i.e. IEEE 754 double precision, which near the surface of the Earth gives a precision of $r_{roc}/2^{52} \approx 1{\cdot}4 \times 10^{-9}$ metres.

6.5.2. *Part 2 – Sensitivity analysis.* The trajectory used in Part 1 was practically straight (only curving due to the Earth's curvature), and thus errors arising in the Euler methods when the vehicle is turning were not included. The LCC in the example typically makes a heading change at 0·3°/s, and a 60° turn (simplified to follow a circle

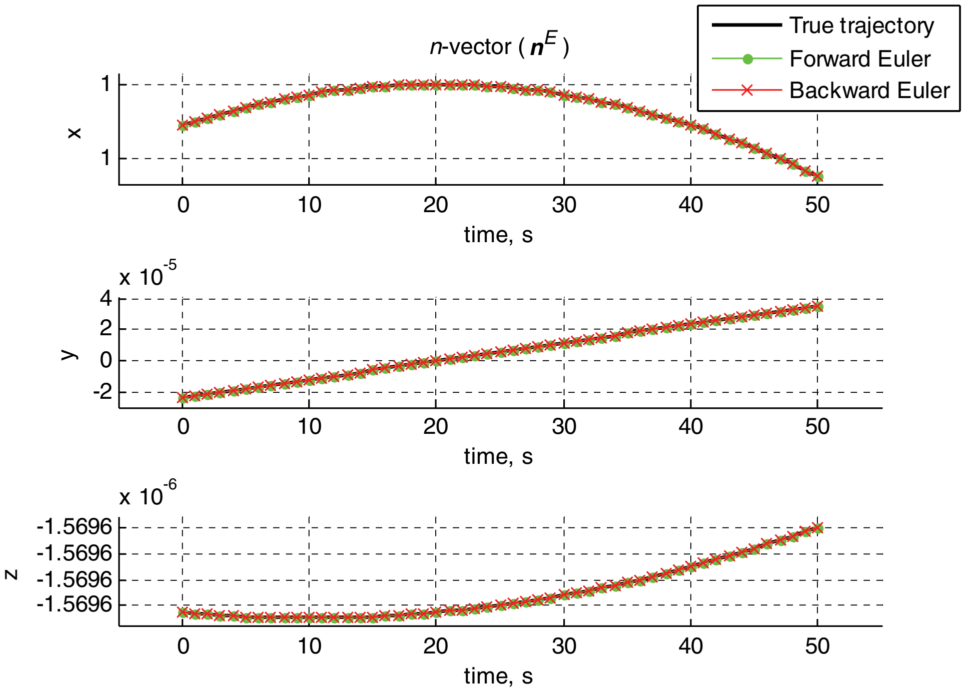


Figure 6. The components of *n*-vector versus time. The true *n*-vector is calculated directly from an exact function. (The errors of the Euler methods are too small to be visible in this plot.)

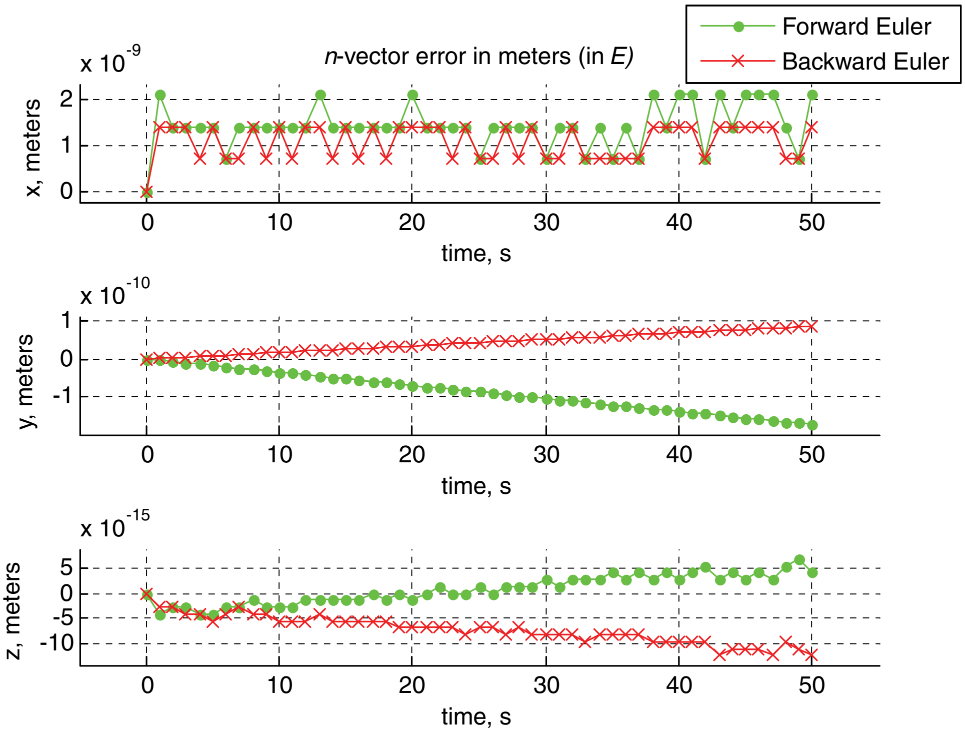


Figure 7. Error from the forward and backward Euler methods when using the *n*-vector representation.

segment) will generate an error of ±3·75 metres for 2D Cartesian coordinates with the 1 Hz Euler methods. The error is not dependent on the turning rate, but on the speed and the net number of degrees turned.

We will now expand the scenario to start 10 minutes before the Pole passing (corresponding to d_1 in Figure 3), include two 30° starboard turns and last for 1 hour, where the turns start at 15 and 30 minutes. The distance to the Pole (d_2 in Figure 3) will be varied, and we will look at the final error as a function of this distance. The results from similar straight trajectories (i.e. no vehicle turning) are also included for comparison. For the straight trajectories, the true trajectory is found analytically as in Part 1, while for the curved trajectories the truth is found using NavLab (Gade, 2004) running at 100 Hz (NavLab uses the trapezoid method and has no pole singularities).

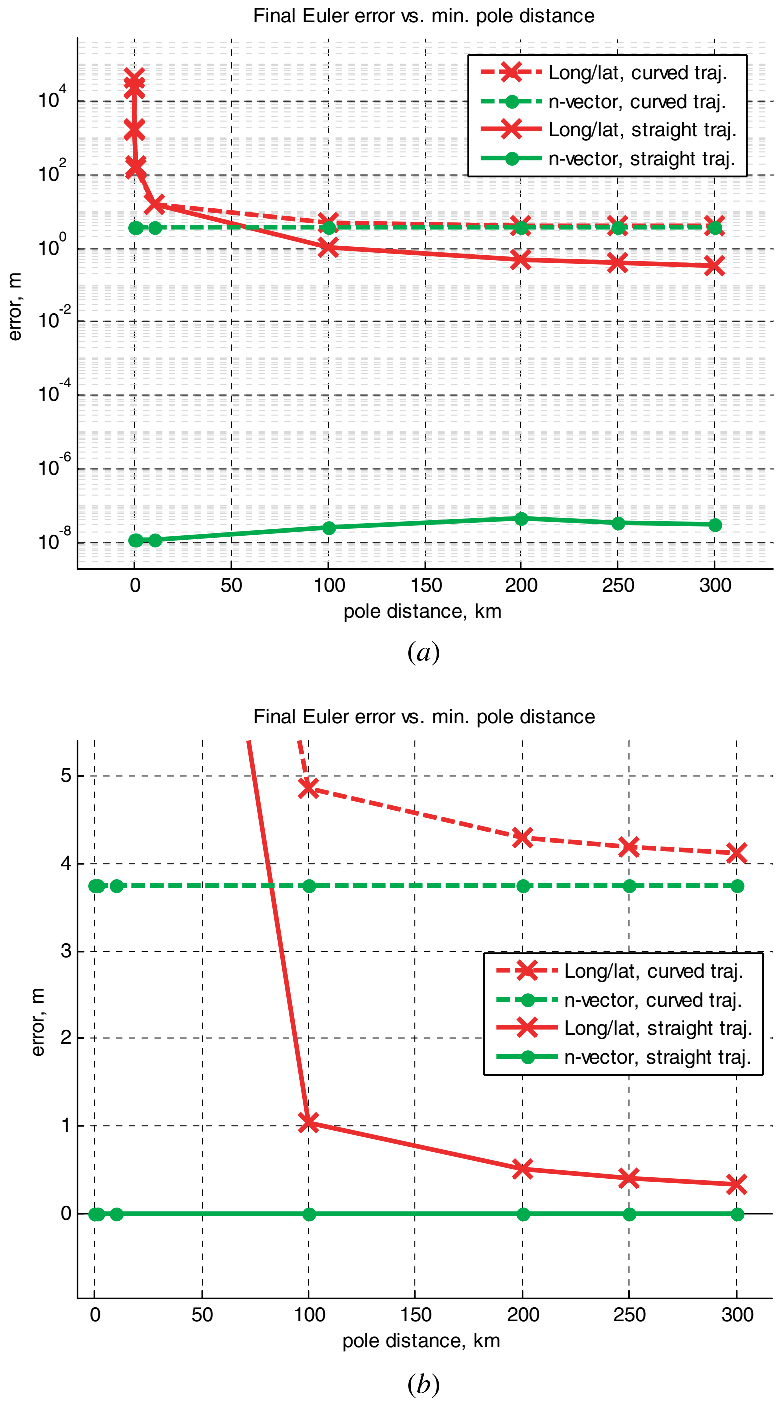
Figure 8 shows the result, and for the curved trajectory, the error in latitude/longitude is 4·1 metres when passing the Pole at 300 km distance, while passing at 5 metres distance gives an error of about 42 km. For *n*-vector, the only visible error is the expected 3·75 metres arising from the turning, independent of the distance to the Pole. For the straight trajectory, we get similar results, but the error of 3·75 metres from the turning is removed (and the small changes in *n*-vector error visible in Figure 8a is because the computer's numerical precision gives different error accumulation at different locations).

A common rule of thumb when considering error sources is to ignore an error source if it is below 10% of a known error. To reduce the extra error generated from the use of the latitude/longitude representation to this level, compared to the error from the turning, requires a distance of about 250 km from the pole in this example. Note that in a practical application, there will be other error sources, other integration periods and other dynamics, and thus the distance where the extra error from the latitude/longitude representation could be neglected will vary from application to application.

6.5.2.1. *Sensitivity of integration period.* We could also do a sensitivity analysis by reducing the integration period to see if this would give acceptable performance for the 10 metres Pole distance scenario. To be able to use the analytical true trajectory, the straight trajectory from above is used. The result is shown in Figure 9 where increasing the rate to 100 Hz reduces the latitude/longitude error to 170 metres. For *n*-vector, the higher rate first reduces the error in a similar manner, but for very high rates, the total number of iterations is considerably increased, and thus the accumulation of round-off errors increases the total error.

6.5.3. *Conclusion.* Integrating position using latitude and longitude can give unacceptably large errors when close to a Pole, particularly due to the coupling of error from latitude to longitude. In the given example, a distance in the order of 100 km from the Pole was needed to be able to safely neglect this additional error source. By using *n*-vector instead, no additional error is introduced (the error here was determined by the actual curvature of the trajectory or by computer precision for straight trajectory).

6.6. *Example 6. Change reference of a position vector.* In many applications, an object's (3D) position given relative to one frame needs to be expressed relative to another frame. For example, this calculation is needed if a vehicle position measured by one radar should be expressed relative to a second radar. The global positions of both radars are given and a classical approach to this problem is to assume that each radar has an associated *N*-frame. The vector to the vehicle is given relative to and decomposed in one *N*-frame, and should be found relative to and decomposed in the second *N*-frame. Using the elliptical Earth model, a classical approach is based on



(*a*)

(*b*)

Figure 8. Final error in the Euler methods (the method with the largest error is plotted) at different distances to the pole (d_2). In part *a*, the magnitude of the spike is compressed by using a logarithmic scale at the *y*-axis. Part *b* is zoomed in without including the large spike and has a linear *y*-axis.

latitude and longitude (Moore and Blair, 2000) and gives the answer using 16 lines of code. Solving the same problem using *n*-vector and the functions in Section 5.4 is intuitive, and only 4 lines of code need be written. Counting the code lines inside the functions used, and multiplying for repeating use of functions gives a total of 10 code lines.

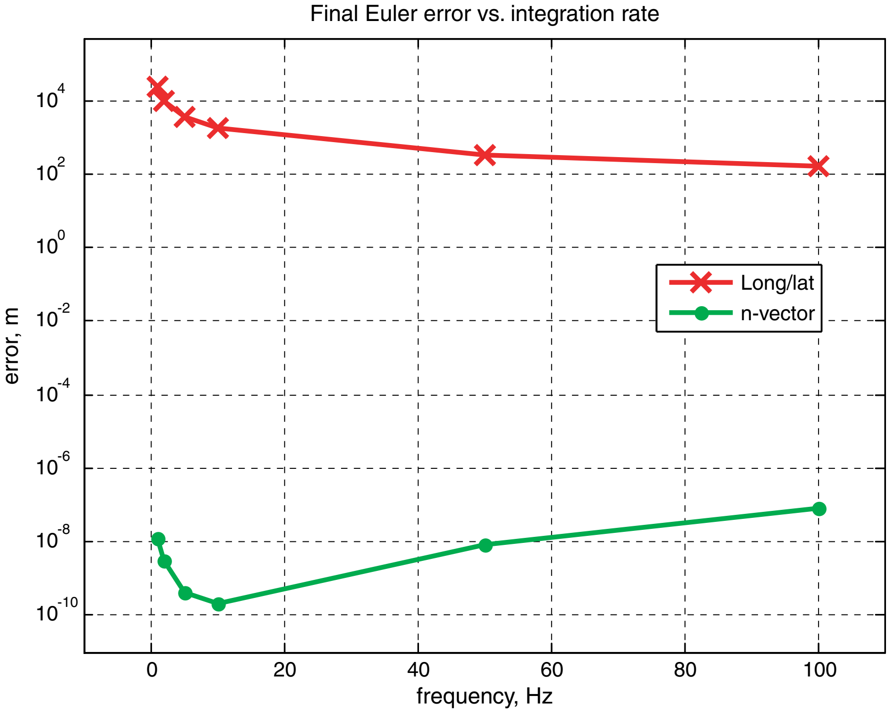


Figure 9. Final error in forward or backward Euler (the one with largest error is plotted) at different integration rates for straight trajectory, with a minimum pole distance (d_2) of 10 metres. The y-axis is logarithmic.

The above problem used N-frames, and is thus inherently singular at any pole. Another variant of the problem would be if the original and final vectors were given in the radar frames (where they are measured), and the orientation was given relative to E (which is the natural measurement from a multi-antenna GPS), rather than N. This variant is also discussed in (Moore and Blair, 2000) and in the classical solution new lines of code are needed for this variant (since the solution still goes via the N-frames). For n-vector, the total code now consists of only two lines, and in addition, the solution is completely non-singular.

If local level frames are desired also in the polar regions, non-singular code is easily achieved with the n-vector approach by replacing N with a non-singular frame. For the classical approach, this is not possible, since it is based on latitude and longitude.

7. *n*-VECTOR USAGE. In practice, it turns out that n-vector can be used for most position calculations where global position is involved. After presenting this alternative for various research groups at FFI and collaborating universities and research institutes, n-vector has replaced other alternatives in numerous applications. Examples of military usage include position calculations for radars, passive submarine sonars and active ship sonars, where the position calculations include target tracking. For navigation applications, n-vector is central for the position calculations in NavLab, HAIN (Marthiniussen et al. 2004) and the HUGIN real-time navigation system (Hagen et al., 2003) and (Jalving et al., 2004). Applications include real-time and post-processing implementations in Matlab, C++ and C#, where thousands of hours of sensor data have been processed using n-vector, since 1999.

7.1. *Using n-vector for attitude representation.*

n-vector is usually decomposed in *E* for horizontal position representation. However, if *n*-vector is decomposed in *B*, it will serve as a convenient and non-singular representation of roll and pitch. This is useful in several situations, since roll and pitch are often treated together, separately from yaw (heading). Actually $\boldsymbol{n}^B$ relates to roll and pitch, in the same way as $\boldsymbol{n}^E$ relates to latitude and longitude. Since the focus of this paper is on position representation, the treatment of $\boldsymbol{n}^B$ as orientation representation will not be covered here.

8. CONCLUSIONS.

Calculations involving global position often include the use of latitude/longitude, local north/east grids or map projections. The latter two alternatives involve limitations and approximations, and we have seen that the latitude/longitude representation has several complex properties, such as error in latitude leading to error in longitude, a discontinuity at longitude $= \pm 180^\circ$, and longitude rate going towards infinity at the poles. Numerous examples have shown that the use of *n*-vector to represent horizontal position gives one or more of the following advantages compared to the traditional approaches (for elliptical or spherical Earth model):

- There are no singularities at the poles or problems at longitude $= \pm 180^\circ$ (the code works equally well for all global positions).
- An exact answer is returned (no approximations are made leading to increasing errors with increasing distances).
- Fewer lines of code are needed to solve typical position calculations.
- Implementation of the code is intuitive (no need to look up a specific procedure).
- No if-statements or iterations are needed in the code.

At FFI and collaborating universities and research institutes, *n*-vector has successfully been replacing other alternatives in numerous military and civilian applications and commercial products since 1999.

ACKNOWLEDGEMENTS

The author would like to thank everyone that has suggested topics and improvements of this paper, in particular scientists and engineers at the Norwegian Defence Research Establishment (FFI), the University Graduate Centre and Kongsberg Maritime.

REFERENCES

Aeronautical Systems Div Wright-Patterson AFB OH (1986). *Specification for USAF Standard Form, Fit and Function Medium Accuracy Inertial Navigation Unit* (SNU-84.1).

Britting, K.R. (1971). *Inertial Navigation Systems Analysis*. Wiley Interscience.

Craig, J.J. (1989). *Introduction to Robotics*. Addison-Wesley Publishing Company, Boston, 2nd edn.

Fortescue, P.W., Stark, J. and Swinerd, G. (2003). *Spacecraft Systems Engineering*. John Wiley and Sons, 3rd edn.

Gade, B.H., and Gade, K. (2007). *n-vector – formulas with derivations*. FFI/RAPPORT 2007/00633, Norwegian Defence Research Establishment (FFI).

Gade, K. (2004). NavLab, a Generic Simulation and Post-processing Tool for Navigation. *European Journal of Navigation*, **2**, 51–59.

Hagen, P.E., Storkersen, N., and Vestgard, K. (2003). The HUGIN AUVs – multi-role capability for challenging underwater survey operations. *EEZ International*.

Hager, J.W., Behensky, J.F., and Drew, B.W. (1989). *The Universal Grids: Universal Transverse Mercator (UTM) and Universal Polar Stereographic (UPS)*. DMA Technical Manual 8358.2, Defence Mapping Agency.

Hofmann-Wellenhof, B., Wieser, M., and Lega, K. (2003). *Navigation: Principles of Positioning and Guidance*. Springer.

Jalving, B., Gade, K., Hagen, O.K., and Vestgard, K. (2004). A Toolbox of Aiding Techniques for the HUGIN AUV Integrated Inertial Navigation System. *Modeling, Identification and Control*, **25**, 173–190.

Levine, W.S. (2000). *Control System Applications*. CRC Press.

Longley, P.A., Goodchild, M.F., Maguire, D.J. and Rhind, D.W. (2005). *Geographic Information Systems and Science*. John Wiley and Sons, 2nd edn.

Marthiniussen, R, Faugstadmo, J. E. and Jakobsen, H. P. (2004). HAIN an integrated acoustic positioning and inertial navigation system. *Proceedings from MTS/IEEE Oceans 2004*, Kobe, Japan.

McGill, D.J., and King, W.W. (1995). *Engineering Mechanics*. PWS-KENT, Boston, 3rd edn.

Moore, J.R. and Blair. W.D. (2000). Practical Aspects of Multisensor Tracking, in *Multitarget-Multisensor Tracking: Applications and Advances*, Volume III, Eds: Bar-Shalom, Y. and Blair, W.D., Artech House.

National Imagery and Mapping Agency (2000). *Department of Defense World Geodetic System 1984: Its Definition and Relationships With Local Geodetic Systems*. NIMA Technical Report TR8350.2, 3rd edn.

Obaidat, M.S. and Papadimitriou, G.I. (2003). *Applied System Simulation: Methodologies and Applications*. Springer.

Phillips, W.F. (2004). *Mechanics of Flight*. John Wiley and Sons.

Savage, P.G. (2000). *Strapdown Analytics*. Strapdown Associates, Inc., Maple Plain.

Sinnott, R.W. (1984). Virtues of the Haversine. *Sky and Telescope*, **68**, 159.

Snyder, J.P. (1987). *Map Projections – A Working Manual*. U. S. Geological Survey Professional Paper 1395. U. S. Government Printing Office.

Strang, G., and Borre, K. (1997). *Linear Algebra, Geodesy, and GPS*. Wellesley-Cambridge Press, Wellesley.

Stuelpnagel, J. (1964). On the Parametrization of the Three-Dimensional Rotation Group, SIAM Review, **6**, 422–430.

Vermeille, H. (2004). Computing geodetic coordinates from geocentric coordinates. *Journal of Geodesy*, **78**, 94–95.

Weisstein, E.W. (2003). *CRC Concise Encyclopedia of Mathematics*. CRC Press.

Zipfel, P.H. (2000). *Modeling and Simulation of Aerospace Vehicle Dynamics*. AIAA Education Series, Reston.

APPENDIX A: RELEVANT COORDINATE FRAMES.

The coordinate frames relevant for this paper are defined in Table 2, and illustrated in Figure 10 (all right handed and orthonormal).

APPENDIX B: KERNEL FUNCTIONS.

The functions described in Section 5.4 are very easy to implement if using two *kernel functions*. The kernel functions are the back and forth conversions between the two non-singular alternatives for global position given in this paper: *n*-vector (with height) and the position vector ($\boldsymbol{p}_{EB}^{E}$).

B.1. *From n-vector to position vector.* Going from *n*-vector (and height) to $\boldsymbol{p}_{EB}^{E}$ is done with a single equation that can be found from the geometry (Gade and Gade, 2007),

$$\boldsymbol{p}_{EB}^{E} = \frac{b}{\sqrt{(n_x^E)^2 + \frac{a^2}{b^2}(n_y^E)^2 + \frac{a^2}{b^2}(n_z^E)^2}} \begin{bmatrix} n_x^E \\ \frac{a^2}{b^2} n_y^E \\ \frac{a^2}{b^2} n_z^E \end{bmatrix} + h \cdot \boldsymbol{n}^E \tag{22}$$

where *a* and *b* are the semi-major and semi-minor axes of the ellipsoid model in use.

It should be noted that this task is almost the same as if we were going from latitude and longitude (and height/depth) to $\boldsymbol{p}_{EB}^{E}$ instead. The equation for the latter is given in textbooks such as (Strang and Borre, 1997). Thus, (22) can also be found by substituting *n*-vector components (using (3)) in the standard equation (the substitution is very simple, since the equation already contains only terms that are equal to the three components in (3)). Thus, replacing latitude/longitude with *n*-vector makes the standard equation shorter in addition to removing the singular quantities.

Table 2. Coordinate frame definitions.

Symbol	Description
E	***Name***: **E**arth ***Position***: The origin coincides with Earth's centre (geometrical centre of ellipsoid model). ***Orientation***: The *x*-axis is along the Earth's rotation axis, pointing north (the *yz*-plane coincides with the equatorial plane), the *y*-axis points towards longitude +90° (east). ***Comments***: The frame is Earth-fixed (rotates and moves with the Earth). The choice of axis directions ensures that at zero latitude and longitude, *N* (described below) has the same orientation as *E*. If roll/pitch/yaw are zero, also *B* (described below) has this orientation. Note that these properties are not valid for another common choice of the axis directions, denoted *e* (lower case), which has *z* pointing north and *x* pointing to latitude = longitude = 0.
B	***Name***: **B**ody (typically of a vehicle) ***Position***: The origin is in the vehicle's reference point. ***Orientation***: The *x*-axis points forward, the *y*-axis to the right (starboard) and the *z*-axis in the vehicle's down direction. ***Comments***: The frame is fixed to the vehicle.
N	***Name***: **N**orth-East-Down (local level) ***Position***: The origin is directly beneath or above the vehicle (*B*), at Earth's surface (surface of ellipsoid model). ***Orientation***: The *x*-axis points towards north, the *y*-axis points towards east (both are horizontal), and the *z*-axis is pointing down. ***Comments***: When moving relative to the Earth, the frame rotates about its *z*-axis to allow the *x*-axis to always point towards north. When getting close to the poles this rotation rate will increase, being infinite at the poles. The poles are thus singularities and the direction of the *x*- and *y*-axes are not defined here. Hence, this coordinate frame is not suitable for general calculations.
L	***Name***: **L**ocal level, Wander azimuth ***Position***: The origin is directly beneath or above the vehicle (*B*), at Earth's surface (surface of ellipsoid model). ***Orientation***: The *z*-axis is pointing down. Initially, the *x*-axis points towards north, and the *y*-axis points towards east, but as the vehicle moves they are not rotating about the *z*-axis (their angular velocity relative to the Earth has zero component along the *z*-axis). (Note: Any initial horizontal direction of the *x*- and *y*-axes is valid for *L*, but if the initial position is outside the poles, north and east are usually chosen for convenience.) ***Comments***: The *L*-frame is equal to the *N*-frame except for the rotation about the *z*-axis, which is always zero for this frame (relative to *E*). Hence, at a given time, the only difference between the frames is an angle between the *x*-axis of *L* and the north direction; this angle is called the *wander azimuth* angle. The *L*-frame is well suited for general calculations, as it is non-singular.

B.2. *From position vector to n-vector.* Going from $\boldsymbol{p}_{EB}^{E}$ to *n*-vector is again a similar problem as going from $\boldsymbol{p}_{EB}^{E}$ to latitude and longitude. The solution of the latter is believed by many to require iterations (see for example (Zipfel, 2000) and (Strang and Borre, 1997)), but direct and exact (closed-form) solutions are available (Vermeille, 2004). Also in this solution, replacing latitude/longitude with *n*-vector gives a shorter and non-singular equation (Gade and Gade, 2007),

$$\boldsymbol{n}^{E} = \frac{1}{\sqrt{d^{2} + {p_{EB,x}^{E}}^{2}}} \begin{bmatrix} p_{EB,x}^{E} \\ \dfrac{k}{(k+e^{2})} p_{EB,y}^{E} \\ \dfrac{k}{(k+e^{2})} p_{EB,z}^{E} \end{bmatrix} \qquad (23)$$

$$h = \frac{k+e^{2}-1}{k} \sqrt{d^{2} + {p_{EB,x}^{E}}^{2}}$$

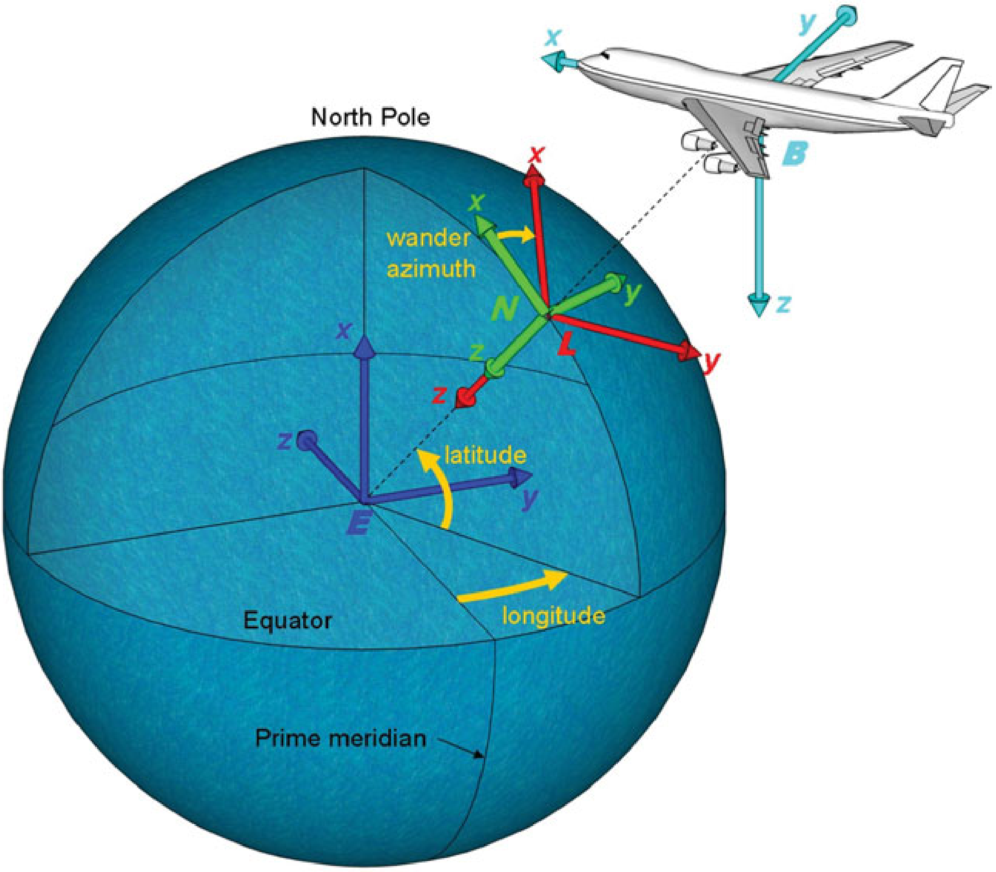


Figure 10. Coordinate frames *E*, *N*, *L* and *B* (figure uses spherical Earth).

where *e* is the eccentricity of the Earth ellipsoid, given by $e=\sqrt{1-\frac{b^2}{a^2}}$. Further

$$d=\frac{k\sqrt{{p_{EB,y}^{E}}^{2}+{p_{EB,z}^{E}}^{2}}}{k+e^2},\quad k=\sqrt{u+v+w^2}-w,\quad w=e^2\frac{u+v-q}{2v},$$

$$v=\sqrt{u^2+e^4q},\quad u=r\left(1+t+\frac{1}{t}\right),\quad t=\sqrt[3]{1+s+\sqrt{s(2+s)}},$$

$$s=\frac{\mathrm{e}^4pq}{4r^3},\quad r=\frac{p+q-e^4}{6},\quad p=\frac{{p_{EB,y}^{E}}^{2}+{p_{EB,z}^{E}}^{2}}{a^2}\quad \textit{and}\ q=\frac{1-e^2}{a^2}{p_{EB,x}^{E}}^{2}.$$

Note that by avoiding iterations, (23) runs faster than standard iterative solutions. Equation (23) is e.g. 2 to 3 times faster than the solution found in the current Matlab version ('ecef2geodetic.m' in the Mapping Toolbox), depending on the number of iterations needed (Gade and Gade, 2007).

Paper II

Gade, K. (2016). The Seven Ways to Find Heading. *The Journal of Navigation*, vol. 69, no. 5, pp. 955-970

THE JOURNAL OF NAVIGATION (2016), **69**, 955–970. © The Royal Institute of Navigation 2016
doi:10.1017/S0373463316000096

The Seven Ways to Find Heading

Kenneth Gade

(*Norwegian Defence Research Establishment* (*FFI*))
(E-mail: Kenneth.Gade@ffi.no)

A magnetic compass has too large a heading error for many applications, and it is often not obvious how to achieve an accurate heading, in particular for low-cost navigation systems. However, there are several different methods available for finding heading, and their feasibility depends on the given scenario. Some of the methods may seem very different, but they can all be related and categorised into a list by studying the vector that each method is using when achieving heading. A list of possible methods is very useful when ensuring that all relevant methods are being considered for a given application. For practical navigation, we have identified seven different vectors in use for heading estimation, and we define seven corresponding methods. The methods are magnetic and gyrocompass, two methods based on observations, multi-antenna Global Navigation Satellite Systems (GNSS), and two methods based on vehicle motion.

KEYWORDS

1. Heading estimation. 2. Categorisation of methods. 3. Vectors.

Submitted: 30 March 2015. Accepted: 6 February 2016. First published online: 4 April 2016.

1. INTRODUCTION.

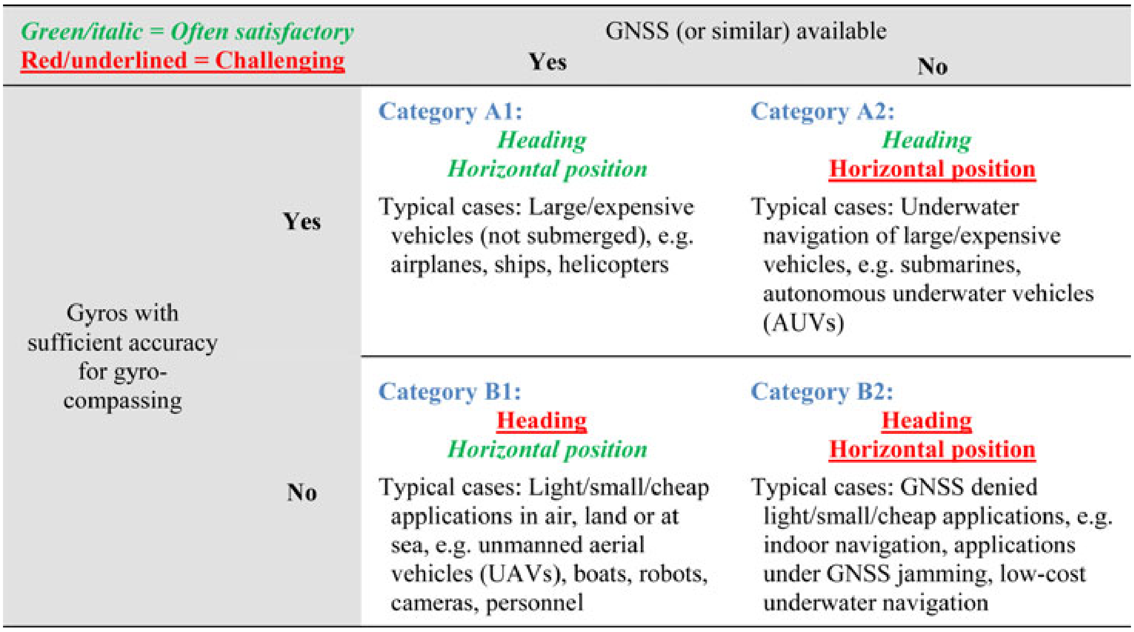
The goal of most navigation systems is to estimate the six degrees of freedom to a required accuracy. The challenge of estimating each of these six will depend on the given scenario, but some common cases can be described.

For inertial navigation systems that are not in free fall, the gravity vector typically dominates the specific force measurement of the accelerometers, and thus roll and pitch are often estimated to a sufficient accuracy. Similarly, the vertical position is commonly obtained from a pressure sensor or a radar/laser altimeter (and for surface-bound vehicles such as ships and cars, it may not be needed).

The challenge of estimating the remaining three degrees of freedom, heading and horizontal position, depends greatly on two factors. For heading, the gyro accuracy determines if heading can be found with sufficient accuracy from gyrocompassing or not (given that a magnetic compass often lacks both the accuracy and reliability to fulfil the heading requirement). For horizontal position, the availability of a Global Navigation Satellite System (GNSS) will clearly be of great importance. By combining these two factors, inertial navigation systems can roughly be divided into four categories, as shown in Table 1.

The increasing availability of Microelectromechanical Systems (MEMS) Inertial Measurement Units (IMUs) has led to a significant growth in light, low-cost navigation systems. However, in most cases, today's MEMS-gyros lack the accuracy

Table 1. The four categories (A1, A2, B1, and B2) of inertial navigation systems, broken down by the availability of GNSS and accuracy of gyros.



Green/italic = Often satisfactory Red/underlined = Challenging		GNSS (or similar) available	
		Yes	No
Gyros with sufficient accuracy for gyro-compassing	Yes	Category A1: *Heading* *Horizontal position* Typical cases: Large/expensive vehicles (not submerged), e.g. airplanes, ships, helicopters	Category A2: *Heading* Horizontal position Typical cases: Underwater navigation of large/expensive vehicles, e.g. submarines, autonomous underwater vehicles (AUVs)
	No	Category B1: Heading *Horizontal position* Typical cases: Light/small/cheap applications in air, land or at sea, e.g. unmanned aerial vehicles (UAVs), boats, robots, cameras, personnel	Category B2: Heading Horizontal position Typical cases: GNSS denied light/small/cheap applications, e.g. indoor navigation, applications under GNSS jamming, low-cost underwater navigation

needed for gyrocompassing, and thus MEMS systems belong to Category B1 or B2 of Table 1 where heading is typically a challenge. Another trend is that GNSS-receivers have become smaller and cheaper, at the same time as GNSS other than the Global Positioning System (GPS) are becoming available. Hence, many of today's low-cost navigation systems belong to Category B1, where five degrees of freedom are often found with sufficient accuracy, while heading is the main challenge.

GNSS can be utilised in several ways to find heading, and the increased GNSS availability makes such heading methods more relevant. Similarly, heading methods that utilise a camera are increasingly attractive for low-cost applications. This is due both to the falling price of cameras and to the increased availability of low cost image processing power.

There are several different and seemingly unrelated ways to find heading, and it is not obvious how to categorise them. Dedicated instruments giving heading/orientation such as magnetic compasses, gyro compasses, star trackers, and multi-antenna GNSS are available. Heading can also be determined by specific procedures or observations, and in addition, there are several scenarios where an integrated navigation system is able to estimate heading based on other measurements and/or specific manoeuvres. In the final case, several underlying methods may be used together, where one method may be dominating the heading estimation for one period, while a second method is the only one that provides heading during another period.

A categorisation of methods to find heading would clearly be useful for the general understanding of heading estimation, and specifically for understanding the underlying methods being available for an integrated navigation system. From a categorisation, we would also achieve a list of available methods, which would be of practical use for those involved with heading estimation. Since the methods are clearly feasible in different scenarios, it is important for a navigation system designer to consider all methods that are valuable in a given scenario, but without a list, it is difficult to guarantee this.

Table 2. Symbols used to describe basic relations between two coordinate frames.

Quantity	Symbol	Description
Position vector	$\vec{p}_{AB}$	A vector whose length and direction is such that it extends from the origin of frame A to the origin of frame B, i.e. the position of B relative to A
Velocity vector	$\vec{v}_{\underline{A}B}$	The velocity of the origin of frame B, relative to frame A. The underline indicates that both the position and orientation of A is relevant (whereas only the position of B matters). Thus, it emphasises that the order of the frames cannot be switched to get the negative vector (which they can for position and angular velocity).
Acceleration vector	$\vec{a}_{\underline{A}B}$	The acceleration of the origin of frame B, relative to frame A
Rotation matrix	$\boldsymbol{R}_{AB}$	A 3 × 3 Direction Cosine Matrix (DCM) describing the orientation of frame B relative to frame A
Angular velocity	$\vec{\omega}_{AB}$	The angular velocity of frame B relative to frame A

The aim of this paper is to suggest a system to categorise the different possible ways to find heading. After the notation is introduced in Section 2, the proposed system is presented in Section 3, together with general theory for heading estimation. Based on the categorisation system, we get a list of methods, which is presented in Section 4, before Section 5 concludes the paper.

Estimated heading may be useful for different units/devices, such as vehicles, instruments, cameras etc. For simplicity, we will use the term *vehicle* for the unit whose heading we want to find.

2. NOTATION.

The notation used is based on Gade (2010).

A general vector can be represented in two different ways (McGill and King, 1995 or Britting, 1971):

$\vec{x}$ (Lower case letter with arrow): Coordinate free/geometrical vector (not decomposed in any coordinate frame).

$\boldsymbol{x}^A$ (Bold lower case letter with right superscript): Vector decomposed/represented in a specific coordinate frame (column matrix with three scalars).

A coordinate frame is defined as a combination of a point (origin), representing position, and a set of basis vectors, representing orientation. Thus, a coordinate frame has six degrees of freedom and can be used to represent the position and orientation of a rigid body. Quantities such as position, angular velocity etc., relate one coordinate frame to another. To make a quantity unique, the two frames in question are given as right subscript, as shown in Table 2.

Note that in most examples in Table 2, only the position or the orientation of the frame is relevant, and the context should make it clear which of the two properties is relevant. For instance, only the orientation of a frame written as right superscript is relevant, since it denotes the frame of decomposition, where only the direction of the basis vectors matters. If both the position and the orientation of the frame are relevant, the frame is underlined to emphasise that fact.

For generality, the vectors in Table 2 are written in coordinate free form, but before implementation on a computer, they must be decomposed in a selected coordinate frame (e.g. the position vector $\vec{p}_{AB}$ decomposed in frame C is $\boldsymbol{p}_{AB}^{C}$).

Table 3. Coordinate frames in use.

Coordinate frame	Description
I	*Inertial* space
E	*Earth*-fixed coordinate frame (moves and rotates with the Earth)
B	*Body*-fixed coordinate frame, attached to the navigating vehicle. If the vehicle has a "forward direction", the x-axis is used for this direction.
O	An "external" *object*, i.e. not part of the navigating vehicle

The *A, B,* and *C*-frames used above were just three arbitrary coordinate frames, while the frames used in the remainder of the paper will have a meaning given by Table 3.

3. HEADING ESTIMATION.

This section presents theory that is common for the different methods of finding heading.

3.1. *Definition of heading.* Heading, sometimes called yaw or azimuth, is one of three rotational degrees of freedom, which is natural to define for land, sea, and air navigation, due to the direction of gravity. By heading, we mean the orientation about the vertical direction vector (where vertical is defined as the normal to the reference ellipsoid). The heading can be represented in several ways, e.g. as a scalar, such as in the Euler angles roll, pitch and yaw (which have singularities at certain positions/orientations[1]). Heading can also be represented by a rotation matrix or quaternion containing the full orientation. How the heading is represented is not important for this paper, since the discussion is valid for any representation.

3.2. *Categorising methods for heading estimation.* When investigating the various methods for heading estimation, it turns out that for each method, a vector is utilised to find the heading. It is also clear that different methods use different vectors, and based on this it seems sensible to define different categories of methods based on which vector is used.

Following this system, we find that there are seven different vectors in common use in practical navigation systems, and thus we define seven corresponding methods. The seven methods are presented in Section 4.

3.3. *Basic principles of heading calculation.* The vector used to find heading (or orientation in the general case) is denoted by $\vec{x}$. The vector must have a known (or measurable) direction relative to the Earth (E) and for now we assume that also the length is known, such that $\boldsymbol{x}^E$ is known. We also need to know the vector relative to the vehicle (B), i.e. $\boldsymbol{x}^B$. The relation between these vectors is

$$\boldsymbol{x}^E = \boldsymbol{R}_{EB}\boldsymbol{x}^B \tag{1}$$

where we want to calculate $\boldsymbol{R}_{EB}$, the vehicle orientation. Note that the length of $\vec{x}$ is not needed to find $\boldsymbol{R}_{EB}$, it is sufficient to know the direction of $\boldsymbol{x}^E$ and $\boldsymbol{x}^B$ (if the lengths are unknown, unit vectors can be used).

[1] The yaw angle (relative North) is singular at the Poles and for pitch = ±90°, while the basic challenge of finding orientation about the vertical direction vector is not affected by these singularities.

Of the three degrees of freedom in $\boldsymbol{R}_{EB}$, it is not possible to find the orientation about an axis parallel to $\vec{x}$ by using Equation (1), but the two remaining degrees of freedom are determined. For example, the gravity vector is known relative to E (when the position is approximately known) and for a stationary vehicle, it is measured relative to B by accelerometers. This gives roll and pitch, while heading (rotation about the gravity vector[2]) is not found. To find the heading we thus need a vector with a non-zero horizontal component, i.e.

$$\vec{x}_{horizontal} \neq \vec{0} \tag{2}$$

3.4. *Accuracy of the heading calculation.* In practice there will be errors in the knowledge about both $\boldsymbol{x}^E$ and $\boldsymbol{x}^B$, and we can write

$$\begin{aligned} \hat{\boldsymbol{x}}_E^E &= \boldsymbol{x}^E + \delta\boldsymbol{x}_E^E \\ \hat{\boldsymbol{x}}_B^B &= \boldsymbol{x}^B + \delta\boldsymbol{x}_B^B \end{aligned} \tag{3}$$

where the hat indicates a quantity with error. $\boldsymbol{x}^E$ and $\boldsymbol{x}^B$ are the true vectors and the delta-terms are the error vectors. The subscript E shows that the error originates from determining $\vec{x}$ in E, and similarly for subscript B. In coordinate free form Equation (3) can be written

$$\begin{aligned} \hat{\vec{x}}_E &= \vec{x} + \overrightarrow{\delta x_E} \\ \hat{\vec{x}}_B &= \vec{x} + \overrightarrow{\delta x_B} \end{aligned} \tag{4}$$

Note that the two error-vectors, $\overrightarrow{\delta x_E}$ and $\overrightarrow{\delta x_B}$ are generally not correlated, as they originate from different calculations/measurements. $\overrightarrow{\delta x_B}$ is often an error from an on-board (internal) sensor, while $\overrightarrow{\delta x_E}$ usually is given by error in external information. Both errors will contribute to the final heading error, but their relative importance will vary greatly between the seven methods, and this will be discussed for each method in Section 4.

Only the components of the error-vectors that are both horizontal and normal to $\vec{x}$ will contribute to the heading error (when assuming small angular errors). If we let the operator $()_{\mathrm{contr}}$ return the length of the contributing component, we can write the standard deviation of these contributing errors $\sigma\left((\overrightarrow{\delta x_E})_{contr}\right)$ and $\sigma\left((\overrightarrow{\delta x_B})_{contr}\right)$. The standard deviation of the resulting heading error ($\delta\psi$) is then given by (uncorrelated errors assumed, first order approximation)

$$\sigma(\delta\psi) \approx \frac{\sqrt{\sigma\left((\overrightarrow{\delta x_E})_{contr}\right)^2 + \sigma\left((\overrightarrow{\delta x_B})_{contr}\right)^2}}{|\vec{x}_{horizontal}|} \tag{5}$$

[2] The direction of the (plumb bob) gravity vector (i.e. the gravitation plus the centripetal acceleration due to Earth's rotation) is very close to the vertical direction (ellipsoid normal).

Table 4. Method 1 (magnetic compass), some key properties.

Advantages	Disadvantages
Low-cost, light and small sensor	Easily disturbed (i.e. low robustness and reliability) which can give unacceptable heading errors for many applications
Self-contained	Reduced accuracy at higher latitudes

4. THE SEVEN METHODS. The seven vectors and corresponding methods commonly used to find heading will be presented in the following subsections. An example of how the list of methods can be used to study a specific navigation system is found in Appendix A.

4.1. *Method 1: The magnetic vector field of the Earth.* The most basic method for finding heading is probably by means of a magnetic compass. For any position of B, the (total) magnetic vector field will give a vector $\vec{m}_B$ (this vector is determined by the position of B, in contrast to the kinematical vectors in Table 2 that are constructed from a relation between two coordinate frames). A magnetometer can measure the direction (relative to B) of the magnetic vector with high accuracy, thus $\boldsymbol{m}_B^B$ is accurately found (and the error contribution from $\overrightarrow{\delta x_B}$ is small). However, the magnetic vector at the position of the magnetometer will have contributions from other sources than the known (and relatively weak) magnetic field of the Earth, and in many cases there will be a large uncertainty in $\boldsymbol{m}_B^E$ (i.e. the contribution from $\overrightarrow{\delta x_E}$ is significant). In addition, $\vec{x}_{horizontal}$ will be short at high latitudes (far north or south).

While the predictable global declination can be compensated for quite easily (e.g. by the International Geomagnetic Reference Field model (IAGA, 2010)), local deviations often limit the practical accuracy of a magnetic compass. Naturally occurring magnetic material in the ground can give deviations of tens of degrees (Leaman, 1997; Langley, 2003), and similar magnitudes of error are also common in urban areas (Godha et al., 2005). Vehicles travelling at a significant distance from the ground, e.g. in air or deep water, are far less vulnerable to these deviations, but their compass may still be degraded by other effects, such as rapid changes in the solar wind. Geomagnetic storms give greatest distortion at higher latitudes, but can sometimes also give significant errors at medium latitudes, such as a 7° change during 20 hours in Scotland (Thomson et al., 2005).

Ferrous materials or electromagnetic interference from the vehicle itself may give significant errors due to the short distance to the compass. The static part of these may be corrected for by calibration procedures, but contributions from changes in the vehicle's magnetic signature (e.g. from changing electrical currents) must still be considered (Healey et al., 1998). Some key properties of Method 1 are summarised in Table 4.

4.2. *Method 2: The angular velocity of the Earth.* The second vector to consider is the angular velocity of the Earth relative to inertial space, $\vec{\omega}_{IE}$. In contrast to the previous method, the direction of this vector is very well known in E, and it defines the locations of the geographic North- and South Pole. Hence the error from $\overrightarrow{\delta x_E}$ is usually negligible.

One possible way to find $\boldsymbol{\omega}_{IE}^B$ is by means of a camera that is detecting change in the direction to celestial objects, e.g. the apparent movement of stars is mainly given by $\vec{\omega}_{IE}$ for an Earth-fixed camera. However, in practical navigation $\boldsymbol{\omega}_{IE}^B$ is usually measured by means of gyroscopes, and the method is thus called gyrocompassing. For the rest of

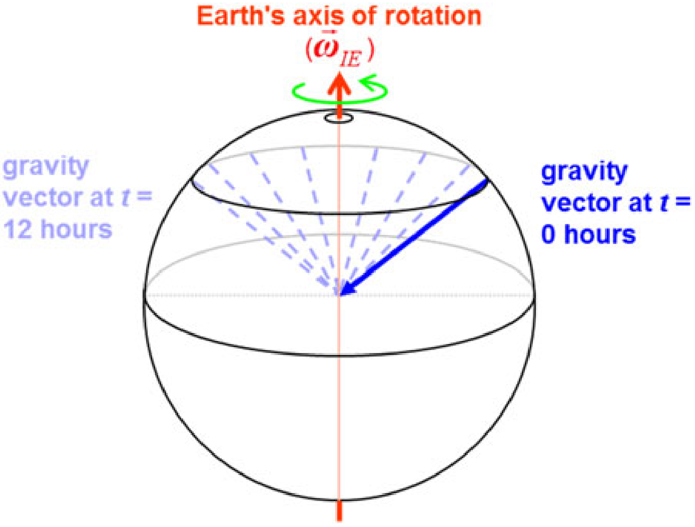


Figure 1. The gravity vector will rotate relative to the inertial space (figure assumes low/zero velocity relative to Earth).

this section we will assume that gyroscopes are used to find $\boldsymbol{\omega}_{IE}^{B}$ (usually ring laser, fibre optic or spinning mass gyros).

The uncertainty of this method is dominated by $\overrightarrow{\delta x_B}$, i.e. error in $\boldsymbol{\omega}_{IE}^{B}$. The vector $\boldsymbol{\omega}_{IE}^{B}$ is found from the relation

$$\boldsymbol{\omega}_{IE}^{B} = \boldsymbol{\omega}_{IB}^{B} - \boldsymbol{\omega}_{EB}^{B}. \tag{6}$$

The first term ($\boldsymbol{\omega}_{IB}^{B}$) is measured by the gyros, and for stationary scenarios ($\vec{\omega}_{EB}(t) = \vec{0}\ \forall\ t$) the gyro error is the main error source. In such scenarios, the gyro error can be averaged to improve the heading accuracy (the ideal averaging time is given by the bottom point of the Allan variance plot (Allan, 1966) of the gyro). The constant part of the gyro error can be cancelled if the gyros can be mechanically rotated, e.g. rotating 180° about the down-axis. This can be very useful e.g. for MEMS gyros, which typically have large constant biases, and thus rotation can make gyrocompassing possible (Renkoski, 2008; Iozan et al., 2010). How different carouseling schemes affect the heading accuracy is discussed in Renkoski (2008). Rotation to improve the heading accuracy can also be obtained by turning the vehicle, when feasible.

For a moving vehicle, gyrocompassing is more challenging since $\boldsymbol{\omega}_{EB}^{B}$ (the second term of Equation (6)) usually has a significant uncertainty. Using a strapdown IMU and knowledge of the vehicle's velocity ($\vec{v}_{EB}$), the heading can still be found by utilising the movement of the gravity vector relative to inertial space, as shown in Figure 1.

The length of $\vec{x}_{horizontal}$ will obviously decrease at higher latitudes, and from geometry and Equation (5) we see that the uncertainty of Method 2 will be proportional to 1/cos(*latitude*), and the accuracy of marine gyrocompasses is often in the order of 0·1°/cos(*latitude*). In Table 5 a short (simplified) summary of Method 2 is given.

4.3. *Method 3: Vector between external objects.* In the third method, heading is obtained by observing at least two objects (O_1 and O_2), that form a vector $\vec{p}_{O_1O_2}$. If the global positions of the two objects are known, the vector is known in E, $\boldsymbol{p}_{O_1O_2}^{E}$. Finding the direction of $\boldsymbol{p}_{O_1O_2}^{B}$ may be done with a camera (with known orientation

Table 5. Method 2 (gyrocompassing), some key properties.

Advantages	Disadvantages
Finds true north	High-accuracy gyros are needed (or a carouseling mechanism)
Self-contained	Reduced accuracy at high latitudes ($\propto 1/\cos(latitude)$)
Not affected by magnetic disturbances	Takes time to find the initial heading (from minutes to hours, depending on several factors, such as vehicle movement and gyro technology)
	Uncertainty of the vehicle velocity will reduce the heading accuracy (a 1 m/s error in north/south-velocity gives a heading error of 0·12° at the Equator).

Table 6. Method 3 (vector between external objects), some key properties.

Advantages	Disadvantages
May find heading very accurately	An imaging sensor, such as a camera or a sonar, must typically be available
	At least two objects (or a feature) must be observed and recognised
	A database of objects must be available

relative to the vehicle), e.g. a downward-looking camera in an Unmanned Aerial Vehicle (UAV) recognising O_1 and O_2. A feature with a known direction, such as a building or a road, may also be recognised and used, but here we simply consider it to consist of one or more vectors known in *E*. The sensor does not need to be a camera; the principle can also be used by other imaging sensors such as a sonar (Lucido et al., 1998).

To find heading, $\vec{p}_{O_1O_2}$ must obviously have a horizontal component, but the direction of observation does not need to be vertical. Consider the concept of leading lights (range lights) for ships, i.e. one low and one high lighthouse that are vertically aligned when the ship is positioned at the correct bearing. In this case $\vec{p}_{O_1O_2}$ points directly towards (or away from) the ship, and the direction of $\boldsymbol{p}^B_{O_1O_2}$ is easily found e.g. by means of an alidade or a camera.

This method can also be used with celestial objects, such as the stars, where the direction of $\boldsymbol{p}^E_{O_1O_2}$ will be known when time is known. Star trackers are commonly used to determine the orientation of spacecraft, but can also be used from the surface of Earth (Samaan et al., 2008).

For Method 3, a high heading accuracy can be obtained from one pair of objects, and with more than two objects, the accuracy will improve further. An example of star tracker accuracy is 0·02° (around boresight, Dzamba et al., 2014). The method is summarised in Table 6.

4.4. *Method 4: Vector from own vehicle to external object.* Methods 1 to 3 may work without knowing the vehicle's own position, but for the rest of the methods, knowledge about vehicle position (or change in position) is needed for the heading calculation. The first of these methods is related to Method 3, but with the knowledge of vehicle (*B*) position, only one external object *O* is needed, i.e. we use the vector $\vec{p}_{BO}$. When we also know the position of *O*, $\boldsymbol{p}^E_{BO}$ is found. The direction of the horizontal part of $\vec{p}_{BO}$ relative to the vehicle's orientation is often called the bearing of the object, and when the bearing is measured, we have the needed part of $\boldsymbol{p}^B_{BO}$.

To measure the bearing in practice, a camera may be used, recognising an object in the picture. There are also other possibilities, e.g. a radio transmitter with a known

Table 7. Method 4 (vector from vehicle to object), some key properties.

Advantages	Disadvantages
Possibility of long vector can give high accuracy	Vehicle position is needed Identifiable object with known position must be observed Sensor with ability to measure bearing to observed object is needed

position can be used as the object, if bearing can be measured by the receiving antenna. Underwater, hydrophones can measure the bearing to an acoustic transmitter or a river outlet with known position.

The object used does not need to be Earth-fixed, as long as the position of the object is known, such that we can determine the direction of $\boldsymbol{p}_{BO}^{E}$. A relevant example is when a second vehicle is travelling close enough to be observed, and the position of that vehicle can be received through a communication channel. A similar situation (that is most relevant on land) is when a GNSS receiver is placed in an observable position, a procedure that is used e.g. to determine the heading of instruments for land surveying (Leica Geosystems, 2008) and for aiming (Rockwell Collins, 2015).

Celestial objects usually have a known position relative to Earth when time is known, and are thus suitable for Method 4, e.g. the direction to the Sun or a satellite transmitting radio signals can be used to find the heading when own position is known (unless when directly above, where $\vec{x}_{horizontal} = \vec{0}$). An example where heading is estimated with an accuracy of about 1° by means of the Sun is found in Lalonde et al. (2010).

If three or more different objects (with different horizontal bearings) are available for observation, horizontal position of B can first be calculated (the "Snellius-Pothenot Surveying Problem" (Dorrie, 1965)), and subsequently any of the objects can be used to find the heading (using $\vec{p}_{BO}$). Some key properties of Method 4 are summarised in Table 7.

4.5. *Method 5: Body-fixed vector.* The two previous methods were using $\vec{p}_{O_1O_2}$ and $\vec{p}_{BO}$, and thus it is now natural to look at the vector $\vec{p}_{B_1B_2}$, i.e. using two separate positions B_1 and B_2 on the vehicle. If we can measure the position of both B_1 and B_2 e.g. from two GNSS receivers or other means, $\boldsymbol{p}_{B_1B_2}^{E}$ is found. $\boldsymbol{p}_{B_1B_2}^{B}$ is typically known from the mounting (and it is constant when assuming a rigid body).

In the example of two GNSS-receivers, heading can be found with two independent receivers if the baseline ($\vec{x}_{horizontal}$) is sufficiently large. However, $\boldsymbol{p}_{B_1B_2}^{E}$ can be found much more accurately (allowing a shorter baseline), by utilising the phase difference of the GNSS carrier signal at two (or more) antennae (resolving the integer ambiguities). An accuracy of about 0·3° can be achieved with a baseline of 0·5 m (Hemisphere GNSS, 2015).

The $\vec{p}_{B_1B_2}$-method can also be used without GNSS, consider e.g. an upward looking camera at ground level with known orientation, observing both B_1 and B_2 (e.g. two recognisable lights) at a vehicle flying above the camera. A similar example is given in Hauschild et al. (2012) where the heading of a satellite, which has two transmitting antennae with a baseline of 1·3 m, was estimated. Ground stations can measure the phase difference of the waves from the two antennae, and can thus estimate the direction of $\boldsymbol{p}_{B_1B_2}^{E}$.

Table 8. Method 5 (multi-antenna GNSS/body-fixed vector), some key properties.

Advantages	Disadvantages
Heading (and possibly roll/pitch) is found with good accuracy at all latitudes	Good GNSS coverage is (typically) required, and multipath may reduce the accuracy Sufficient space on the vehicle is needed Lack of vehicle rigidness reduces accuracy

For Method 5, $\overrightarrow{\delta x_B}$ is usually found accurately by measurements of the installed B_1 and B_2 at the vehicle, and if the vehicle is sufficiently rigid, this error will be small. $\overrightarrow{\delta x_E}$ will typically be larger, but a long baseline can make the resulting heading error very small, giving the method a high potential accuracy. A short summary of Method 5 is given in Table 8.

4.6. *Method 6: Vehicle velocity vector.* So far, the methods have been independent of own movement, but if the vehicle has a horizontal velocity component, the velocity vector $\vec{v}_{\underline{EB}}$ can also be used to find heading.

Finding $\boldsymbol{v}^B_{\underline{EB}}$ can be done in several ways, where one option is using a Doppler sensor, such as an underwater acoustic Doppler velocity log or a Doppler radar. One or more cameras can also be used, where the optical flow of Earth-fixed features is tracked (Zinner et al., 1989; Sivalingam and Hagen, 2012). Sensors that measure velocity relative to water or air may also be used if sea current or wind is known (or small relative to $\vec{v}_{\underline{EB}}$). Finally, $\boldsymbol{v}^B_{\underline{EB}}$ can also be found from knowledge of the vehicle movement, e.g. a vehicle on rails or wheels may have a restricted movement such that

$$\boldsymbol{v}^B_{\underline{EB}} \approx \begin{bmatrix} x \\ 0 \\ 0 \end{bmatrix}, \tag{7}$$

where x is the forward speed (and hence the course equals the heading). For vehicles in air/water, an aerodynamic/hydrodynamic model may be used to calculate velocity relative to the surrounding air/water.

For Method 6 to work, we also need $\boldsymbol{v}^E_{\underline{EB}}$, which can be obtained from GNSS (utilising Doppler shift or carrier phase (van Graas and Soloviev, 2004). If position measurements ($\boldsymbol{p}^E_{\underline{EB}}$) are available, $\boldsymbol{v}^E_{\underline{EB}}$ can in theory be found by direct differentiation. In practice, a correctly designed integrated navigation system will use Method 6 to estimate heading when position measurements and $\boldsymbol{v}^B_{\underline{EB}}$ are available. An example is the navigation system of the HUGIN Autonomous Underwater Vehicle (AUV), where Method 6 is necessary to find heading with the required accuracy for vehicles that are not equipped with high-accuracy gyros (Gade and Jalving, 1998). The method gives a heading accuracy of about 0·5° when $\boldsymbol{p}^E_{\underline{EB}}$ is regularly transmitted to the AUV from a surface ship (that has found $\boldsymbol{p}^E_{\underline{EB}}$ by combining acoustic relative positioning and GNSS), since the AUV has a Doppler velocity log and under normal operation has a horizontal velocity component.

The accuracy of method 6 will clearly increase with higher horizontal speed, while the two errors $\overrightarrow{\delta x_E}$ and $\overrightarrow{\delta x_B}$ typically will originate from two different sensors, and which of them is dominating will depend on the given scenario. Some key properties of Method 6 are summarised in Table 9.

Table 9. Method 6 (velocity vector), some key properties.

Advantages	Disadvantages
May give accurate heading from sensors that are already present	Sufficient horizontal vehicle velocity is required Sufficient knowledge about $\boldsymbol{v}_{\underline{EB}}^{B}$ is needed, e.g. from a Doppler sensor or a camera Position measurements or measurements of $\boldsymbol{v}_{\underline{EB}}^{E}$ must be obtained

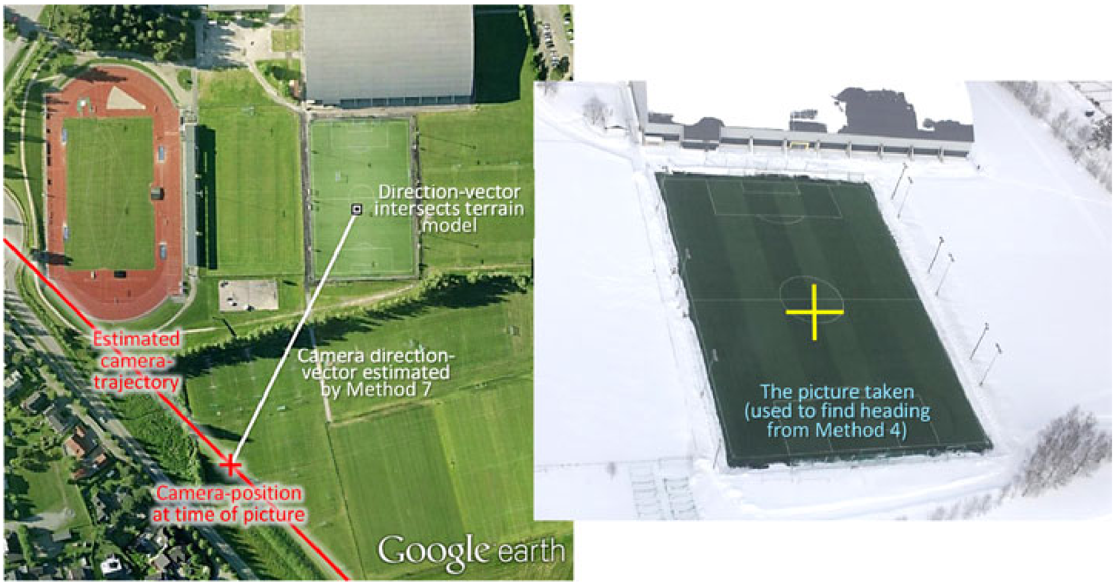

Direction-vector intersects terrain model
Estimated camera-trajectory
Camera direction-vector estimated by Method 7
Camera-position at time of picture
Google earth
The picture taken (used to find heading from Method 4)

Figure 2. Verifying Method 7 by Method 4, from helicopter. NavLab has calculated the orientation and position of the camera when the picture was taken, giving a direction vector. The position where the direction vector intersects an available terrain model is compared with the centre of the picture taken.

4.7. *Method 7: Vehicle acceleration vector.* Method 6 required a sensor measuring $\boldsymbol{v}_{\underline{EB}}^{B}$ (or sufficient knowledge about this vector), which for many navigation systems will not be available. However, almost every navigation system will have accelerometers, and thus we can instead find the vector $\boldsymbol{a}_{\underline{EB}}^{B}$ (after subtracting the contribution from the gravity and Coriolis force). Hence, if the vehicle has a horizontal acceleration component, $\vec{a}_{\underline{EB}}$ can be used to find heading. Obtaining $\boldsymbol{a}_{\underline{EB}}^{E}$ can be done similarly as in Method 6, i.e. by getting measurements of $\boldsymbol{v}_{\underline{EB}}^{E}$ or $\boldsymbol{p}_{\underline{EB}}^{E}$, typically from GNSS.

FFI has investigated the accuracy of this method when using MEMS IMUs, by comparing the heading calculated from Method 7 with the heading from two rigidly attached references. The first reference is Honeywell HG9900 IMU (gyro biases of only 0·003°/h), from which the estimator in the navigation software NavLab (Gade, 2005) will find heading accurately using Method 2. The second reference is a camera, where heading is found by Method 4 each time a picture is taken, as shown in Figure 2.

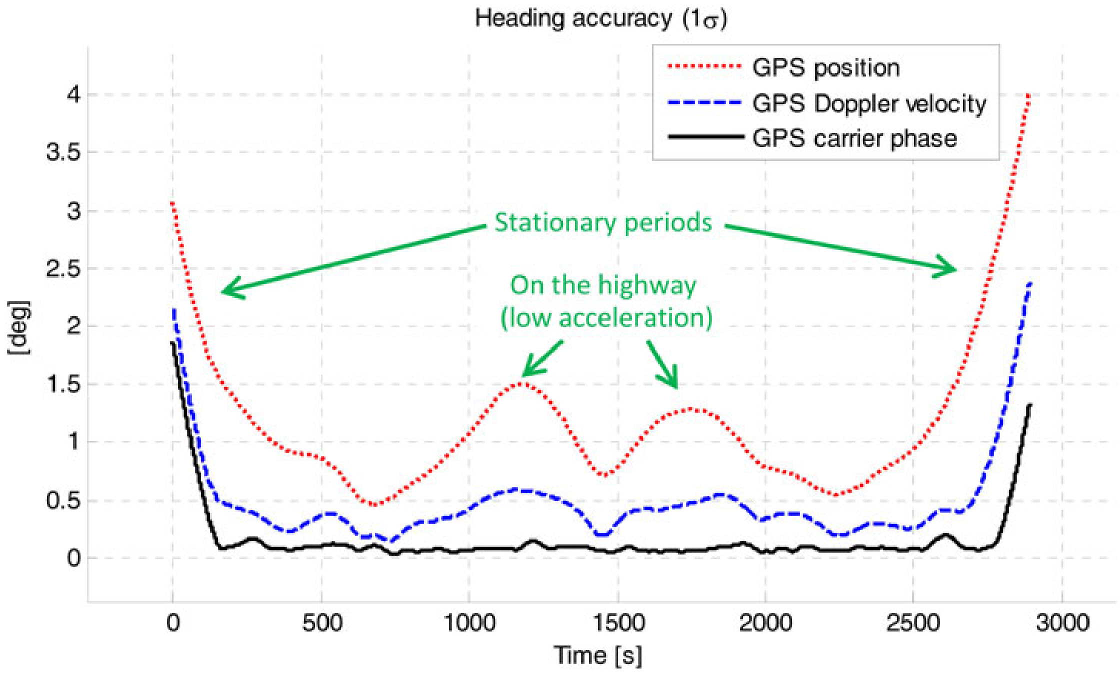


Figure 3. The heading accuracy from Method 7 in the car test varied with acceleration and with different methods to find $\boldsymbol{a}_{\underline{EB}}^{E}$. The car had a stationary period before driving on smaller roads, and then a period on a highway, before driving back the same way. The three graphs show theoretical uncertainty in smoothed heading when NavLab is aided with GPS position ($\boldsymbol{p}_{EB}^{E}$) or GPS velocity ($\boldsymbol{v}_{\underline{EB}}^{E}$) from Doppler shift or carrier phase. Only the uncertainty with GPS position aiding is verified with an external heading reference, the two others are simulated with accuracies based on u-blox (2015) and van Graas and Soloviev (2004).

Table 10. Method 7 (acceleration vector), some key properties.

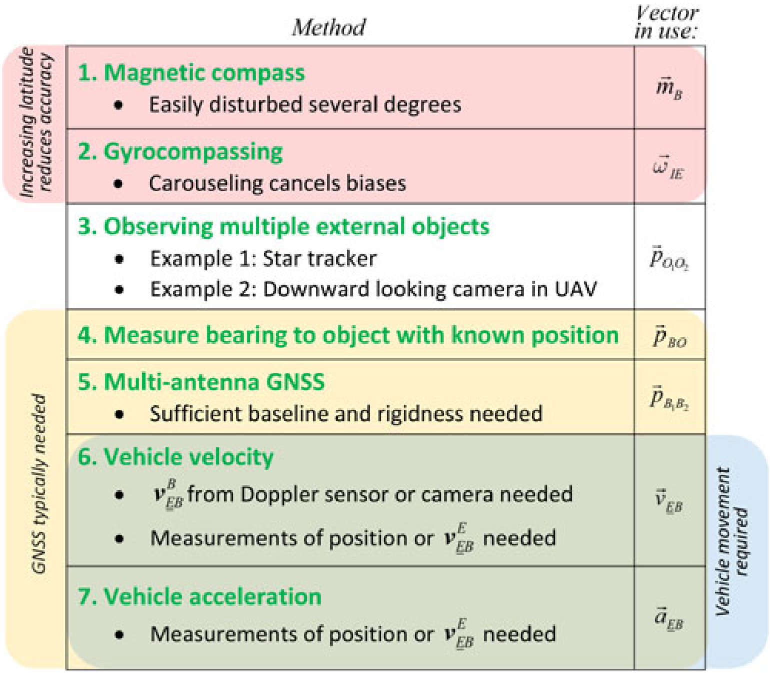
Advantages	Disadvantages
Method is often available, since it uses accelerometers, and typically GNSS, and it is thus an important method for many MEMS-based navigation systems	Sufficient horizontal vehicle acceleration is required Position measurements or measurements of $\boldsymbol{v}_{\underline{EB}}^{E}$ with sufficient accuracy and rate must be obtained

In a helicopter doing turns, $\vec{a}_{\underline{EB}}$ is large, and heading is found accurately from Method 7. By using Sensonor STIM300 IMU and GPS position aiding, NavLab (with smoothing) obtained a heading accuracy of 0·16° (1σ). This accuracy was found by measuring the deviation in 26 pictures (and it is in accordance with both the deviation from the HG9900-reference and the theoretical uncertainty reported by the NavLab estimator).

A lower acceleration will obviously reduce the accuracy (due to shorter $\vec{x}_{horizontal}$), but by aiding NavLab with $\boldsymbol{v}_{\underline{EB}}^{E}$ instead of $\boldsymbol{p}_{EB}^{E}$, less acceleration, with shorter duration is needed to find heading. This is because the integrated acceleration can be distinguished more quickly from the noise in $\boldsymbol{v}_{\underline{EB}}^{E}$ than that in $\boldsymbol{p}_{EB}^{E}$, since the latter requires a second integration.

FFI has looked at the accuracy available from Method 7 in a car, and Figure 3 shows the result from a test where $\boldsymbol{a}_{\underline{EB}}^{E}$ is found from $\boldsymbol{p}_{EB}^{E}$ and from two variants of $\boldsymbol{v}_{\underline{EB}}^{E}$.

Method 7 is summarised in Table 10.



Method	Vector in use:
1. Magnetic compass • Easily disturbed several degrees	$\vec{m}_B$
2. Gyrocompassing • Carouseling cancels biases	$\vec{\omega}_{IE}$
3. Observing multiple external objects • Example 1: Star tracker • Example 2: Downward looking camera in UAV	$\vec{p}_{O_1O_2}$
4. Measure bearing to object with known position	$\vec{p}_{BO}$
5. Multi-antenna GNSS • Sufficient baseline and rigidness needed	$\vec{p}_{B_1B_2}$
6. Vehicle velocity • v_{EB}^{B} from Doppler sensor or camera needed • Measurements of position or v_{EB}^{E} needed	$\vec{v}_{EB}$
7. Vehicle acceleration • Measurements of position or v_{EB}^{E} needed	$\vec{a}_{EB}$

Increasing latitude reduces accuracy
GNSS typically needed
Vehicle movement required

Figure 4. A simplified summary of the seven methods, and some key features/examples of each method.

4.8. *Other methods.* Seven different vectors that are all widely used to find heading have been presented. However, given all the different applications around the globe that find orientation, we cannot guarantee that all methods in use are covered among the seven, but we think the most important ones are included. Appendix B will briefly discuss some possible methods not covered by the seven, but to our knowledge, they are in little or no use in practical navigation.

5. CONCLUSIONS.

There are several different techniques in use for heading estimation in various applications, and it may be difficult to get an overview and see how they relate to each other. However, by studying the vector utilised to find the heading, different heading estimation methods can be defined based on which vector is in use. This categorisation has given seven different methods to find heading that are all in common use. Figure 4 shows a simplified summary of these for quick reference.

For Method 2 it takes time to find the initial heading, while for the other six methods it is in many cases possible to find heading almost instantly. We have found the list of the available methods to be very useful when designing navigation systems, in particular those belonging to Category B1 or B2 of Table 1. The list helps to ensure that all feasible methods are considered for a given scenario and it works as a common reference in discussions about heading estimation. Since the theory is quite fundamental, it has also turned out to be valuable for teaching.

ACKNOWLEDGEMENTS

The author is grateful for all feedback and suggested improvements of this paper received from researchers, professors, and engineers at FFI, the Norwegian University of Science and Technology (NTNU), the University Graduate Centre (UNIK), and Kongsberg Maritime.

REFERENCES

Allan, D.W. (1966). Statistics of atomic frequency standards. *Proceedings of the IEEE*, **54**, 221–230.

Britting, K.R. (1971). *Inertial Navigation Systems Analysis.* Wiley Interscience.

Dorrie, H. (1965). *100 Great problems of elementary mathematics.* Dover Publications.

Dzamba, T., Enright, J., Sinclair, D., Amankwah, K., Votel, R., Jovanovic, I. and McVittie, G. (2014). Success by 1000 Improvements: Flight Qualification of the ST-16 Star Tracker. *Proceedings from 28th Annual AIAA/USU Conference on Small Satellites*, Logan, Utah.

Gade, K. and Jalving, B. (1998). An Aided Navigation Post Processing Filter for Detailed Seabed Mapping UUVs. *Proceedings of the IEEE 1998 Workshop on Autonomous Underwater Vehicles*, Cambridge, Massachusetts.

Gade, K. (2005). NavLab, a Generic Simulation and Post-processing Tool for Navigation. *Modeling, Identification and Control*, **26**, 135–150.

Gade, K. (2010). A Non-singular Horizontal Position Representation. *Journal of Navigation*, **63**, 395–417.

Godha, S., Petovello, M.G., and Lachapelle, G. (2005). Performance Analysis of MEMS IMU/HSGPS/Magnetic Sensor Integrated System in Urban Canyons. *Proceedings of ION-GNSS Conference*, Long Beach, California.

Hauschild, A., Steigenberger, P. and Rodriguez-Solano, C. (2012). QZS-1 Yaw Attitude Estimation Based on Measurements from the CONGO Network. *Journal of the Institute of Navigation*, **59**, 237–248.

Healey, A.J., An, E.P. and Marco, D.B. (1998). Online Compensation of Heading Sensor Bias for Low Cost AUVs. *Proceedings of the IEEE 1998 Workshop on Autonomous Underwater Vehicles*, Cambridge, Massachusetts.

Hemisphere GNSS (2015). Vector V103 and V113 GNSS Compasses, Data Sheet. http://hemispheregnss.com. Accessed 7 September 2015.

International Association of Geomagnetism and Aeronomy, IAGA (2010). International Geomagnetic Reference Field: the eleventh generation, *Geophysical Journal International*, **183**, 1216–1230.

Iozan, L.I., Kirkko-Jaakkola, M., Collin, J., Takala, J. and Rusu, C. (2010). North Finding System Using a MEMS Gyroscope. *Proceedings of European Navigation Conference on Global Navigation Satellite Systems*, Braunschweig, Germany.

Lalonde, J.F., Narasimhan, S.G., and Efros, A.A. (2010). What do the sun and the sky tell us about the camera? *International Journal of Computer Vision*, **88**, 24–51.

Langley, R.B. (2003). The Magnetic Compass and GPS. *GPS World*, September 2003.

Leaman, D.E. (1997). Magnetic rocks – their effect on compass use and navigation in Tasmania. *Papers and Proceedings of the Royal Society of Tasmania.*

Leica Geosystems (2008). Leica TPS1200+ (Total Station), Applications, Field Manual, Version 6.0

Lucido, L., Pesquet-Popescu, B., Opderbecke, J., Rigaud, V., Deriche, R., Zhang, Z., Costa, P. and Larzabal, P. (1998). Segmentation of Bathymetric Profiles and Terrain Matching for Underwater Vehicle Navigation. *International Journal of Systems Science*, **29**, 1157–1176.

McGill, D.J. and King, W.W. (1995). *Engineering Mechanics.* PWS-KENT, Boston, 3rd edn.

Renkoski, B.M. (2008). The Effect of Carouseling on MEMS IMU Performance for Gyrocompassing Applications. Master of Science thesis. Massachusetts Institute of Technology.

Rockwell Collins (2015). Defence Advanced GPS Receiver (DAGR) – brochure. http://www.rockwellcollins.com. Accessed 7 September 2015.

Samaan, M.A., Mortari, D. and Junkins, J.L. (2008). Compass Star Tracker for GPS-like Applications. *IEEE Transactions on Aerospace and Electronic Systems*, **44**, 1629–1634.

Sivalingam, B. and Hagen, O.K. (2012). Image-aided Inertial Navigation System based on Image Tokens, *Proceedings of NATO/RET/SET-168 Symposium*, Izmir, Turkey.

Thomson, A.W., McKay, A.J., Clarke, E. and Reay, S.J. (2005). Surface Electric Fields and Geomagnetically Induced Currents in the Scottish Power Grid During the 30 October 2003 Geomagnetic Storm. *Space Weather*, **3**.

u-blox (2015). NEO-7, u-blox 7 GNSS Modules, Data Sheet. http://www.u-blox.com. Accessed 7 September 2015.

van Graas, F. and Soloviev, A. (2004). Precise Velocity Estimation using a Stand-alone GPS Receiver. *Journal of the Institute of Navigation*, **51**, 283–292.

Zinner, H., Schmidt, R. and Wolf, D. (1989). Navigation of Autonomous Air Vehicles by Passive Imaging Sensors. NATO AGARD (Advisory Group for Aerospace Research and Development) *Conference Proceedings No. 436*, Guidance and Control of Unmanned Air Vehicles.

APPENDIX A: STUDYING HEADING USING THE LIST OF METHODS – AN EXAMPLE

The list of methods can be used to study how to find the heading of a given vehicle, and this can be illustrated by a simple example. Consider a wheeled vehicle with MEMS IMU, GPS, and camera (belonging to Category B1 of Table 1). If we consider the feasibility of each heading method for this vehicle, the result may look like this:

Method 1: No; too much electromagnetic interference from the vehicle
Method 2: No; accurate gyros unavailable (too expensive for this application)
Method 3: No; multiple external recognisable objects in general not visible at the same time
Method 4: Sometimes; can be used when a recognisable object, such as another vehicle with known position, can be seen by the camera
Method 5: No; not enough space for the navigation unit to get the required baseline
Method 6: Yes; wheel-slip is common, i.e. Equation (7) cannot be assumed, however $\boldsymbol{v}_{\underline{EB}}^{B}$ can be found from the camera
Method 7: Yes

Thus, this vehicle can sometimes find heading from Method 4, and when in movement Method 6 and 7 can be used. An integrated navigation system will utilise both Method 6 and 7 at the same time, and when the vehicle velocity is high, and there is no or low acceleration, Method 6 will contribute the most (due to the $|\vec{x}_{horizontal}|$-term of Equation (5)).

APPENDIX B: OTHER METHODS TO FIND HEADING

When looking for other possible methods, we can use the seven methods as a basis.

B.1. *Vector fields with known direction relative to the Earth.* Looking at Method 1 (magnetic vector field), it is natural to ask if other Earth-fixed vector fields can be utilised. E.g., the gravity vector has a small horizontal component (deflection) at some locations due to non-homogenous mass distribution, but it is probably too small to be of practical use in heading estimation.

If the horizontal gradient of a scalar field, such as temperature or particle concentration (in air or water) is known in *E*, heading could be found from distributed scalar sensors (in a similar manner as a snake can detect the particle gradient with its forked tongue).

Another example is wind or sea current, whose direction can be found accurately relative to the vehicle (*B*) for several applications, i.e. $\overrightarrow{\delta x_B}$ is small. However, for both this and the previous examples, the direction in *E* is usually not known with sufficient accuracy to make the vector useful to find heading in practical navigation.

Man-made vector fields (with a known direction in *E*) may be in use for heading estimation in some applications, but to our knowledge, such applications are not common.

B.2. *Methods 2-5.* The use of $\vec{\omega}_{IE}$ (Method 2) is possible since *E* is a non-inertial frame, but there are probably no other non-inertial features of the Earth that can be used to find heading.

The three next methods related to positions of objects (Methods 3–5) utilise all the three possible combinations of external objects and sensors mounted on the vehicle, and thus it is difficult to see how to extend this list.

B.3. *Methods utilising vehicle motion.* Methods 6 and 7 can be naturally extended to the use of the jerk vector (the derivative of acceleration). However, the jerk vector changes quickly for most vehicles, and it is difficult to find in E. In practice, a system that would be able to find heading from the jerk vector would probably find heading with much higher accuracy from the acceleration vector (Method 7).

For a rotating vehicle, the vector $\vec{\omega}_{EB}$can be used to find heading. E.g. for a spinning vehicle with simple gyros, $\boldsymbol{\omega}_{EB}^{B}$ would be measured (Earth rotation is below the noise level). One or more Earth-fixed cameras filming the vehicle could calculate $\boldsymbol{\omega}_{EB}^{E}$ from the rotating movement of the pattern on the vehicle's surface and hence heading can be found. If such cameras were available, in most cases it would probably be better to paint recognisable markers on the vehicle and find its heading with Method 5, avoiding the need for gyros and the need for rotation.

For practical applications, we are not aware of vehicles that use $\vec{\omega}_{EB}$ or its derivatives to find heading.

Paper III

Gade, K. and Jalving, B. (1999). An Aided Navigation Post Processing Filter for Detailed Seabed Mapping. *Modeling, Identification and Control*, vol. 20, no. 3, pp. 165-176

MODELING, IDENTIFICATION AND CONTROL, 1999, VOL. 20, NO. 3, 165-175
doi: 10.4173/mic.1999.3.2

AN AIDED NAVIGATION POST PROCESSING FILTER FOR DETAILED SEABED MAPPING UUVS

Kenneth Gade and Bjørn Jalving

Norwegian Defence Research Establishment (FFI)
P. O. Box 25, 2007 Kjeller, Norway

Keywords: *Aided navigation, Kalman filter, Smoothing, Untethered Underwater Vehicle (UUV), Sea trials*

Abstract - **HUGIN is an untethered underwater vehicle (UUV) intended for bathymetric data collection for detailed seabed surveying. The HUGIN sensor suite, consisting of standard commercially available navigation sensors and a multibeam echosounder, is briefly presented. A Kalman filter based post processing integration of UUV sensors and survey vessel sensors is discussed. Resulting UUV position and heading accuracy and important characteristics of the post processing filter is shown with simulation results and results from a commercial survey operation. Finally, we briefly show how the claimed position and heading accuracy has been verified.**

1 Introduction

In the HUGIN development program two untethered underwater vehicles have been produced. The vehicles are fitted with a Kongsberg Simrad EM 3000 multibeam echosounder for underwater surveys to depths of 600 m. HUGIN I had its first sea trial in summer 1996 and has been used as a test and demonstration platform. HUGIN II was in spring 1998 put into commercial operation, offering services to the survey market. The HUGIN development program is a co-operation between Norwegian Defence Research Establishment (FFI), Kongsberg Simrad AS, Norwegian Underwater Intervention AS (NUI) and Statoil, Størkersen et. al. [1].

The aided post processing navigation system presented in this paper, was used in a commercial survey operation (Åsgard Transport) with HUGIN I on the Norwegian continental shelf in autumn 1997. The claimed positioning accuracy has been verified and documented in Jalving & Gade [2]. The aided navigation system is currently being integrated in the Neptune/Merlin commercial post processing package from Kongsberg Simrad AS.

2 UUV positioning

The objective of the HUGIN system is to collect data for detailed seabed mapping. Fig. 1 shows the navigation systems and sensors necessary for positioning of multibeam echosounder data in global coordinates. A commercial survey vessel will typically have its position provided by Differential Global Positioning System (DGPS). The position of the HUGIN vehicle relative to the surface vessel is measured by means of the High Precision Acoustic Positioning system (HiPAP) from Kongsberg Simrad AS. In order to determine the orientation of the EM 3000 transducer, which is necessary for positioning of the EM 3000 footprint relative to the UUV, HUGIN is equipped with a Seatex Motion Reference Unit (MRU), which among several data, outputs the vehicle's roll and pitch angle. MRU has an inertial sensor assembly of three gyros and three accelerometers. Heading is measured by a Leica Digital Magnetic Compass and depth is measured with a Digiquartz 9001K-101 pressure transmitter. An EDO 3050 Doppler Velocity Log (DVL) provides a velocity measurement.

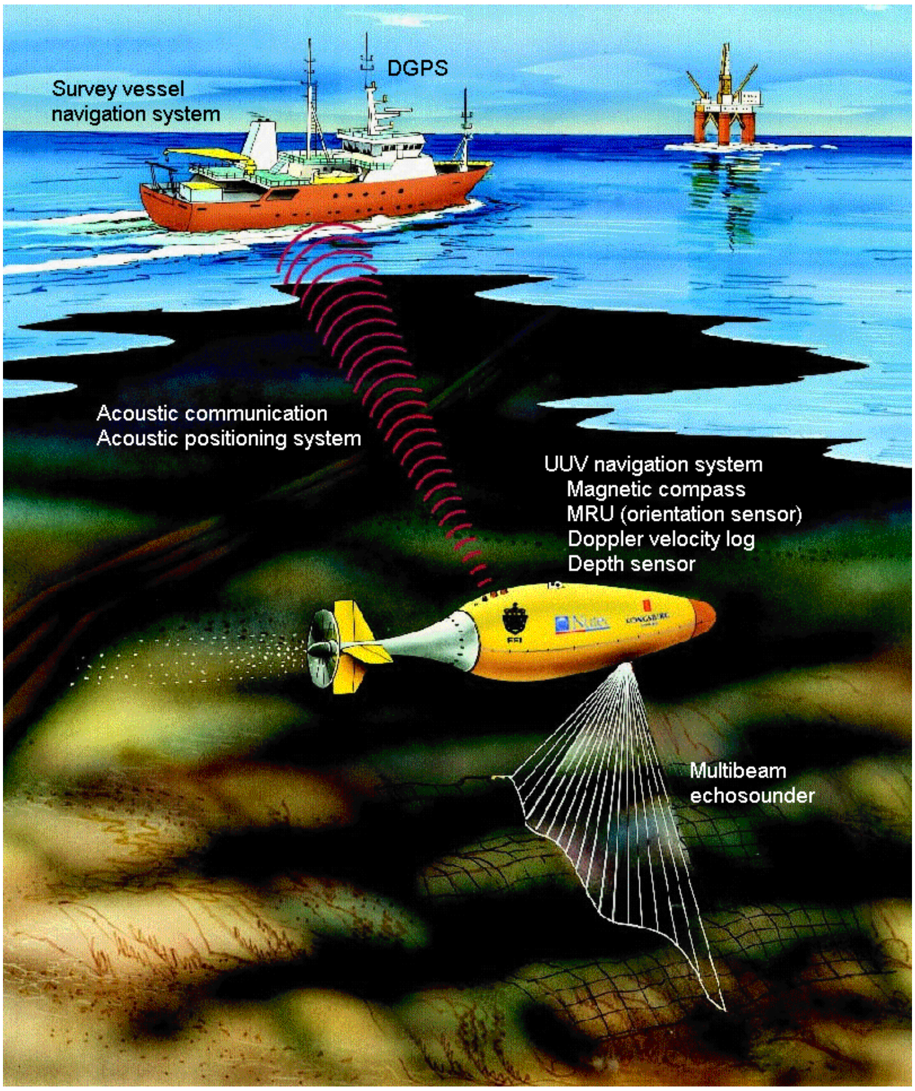


Fig. 1. A seabed mapping scenario with the HUGIN system.

During a survey mission, EM 3000 multibeam echosounder data and HUGIN sensor data are stored locally in the UUV on a hard disk. After a mission, these data are merged with DGPS/HiPAP position data stored aboard the survey vessel, in the post processing filter described below.

From a complete EM 3000 footprint positioning error budget presented in Jalving & Gade [3], it is seen that the *horizontal* UUV position measurement (combined DGPS and HiPAP) and the UUV magnetic heading measurement are candidates for substantially improved accuracy.

3 Kalman filter design

Estimating horizontal position and heading, a possible basis includes:

- Sensor measurements
- System knowledge (i.e. models of the UUV, its sensors and the environment)
- Control variables (i.e. rudder deflection, stern plane deflection and propeller revolution)

By combining measured control variables with a hydrodynamic UUV model and a sea current model, it is possible to calculate estimates of for instance linear and angular velocity. However, due to considerable model uncertainty, these estimates are far less accurate than the measurements from the

Doppler velocity log and the MRU gyros, and thus this strategy offers no significant aid to the estimates.

Consequently, the position and heading estimates should be based on sensor measurements and knowledge of their error models. The optimal way to combine this information is by means of a Kalman filter. Since we have measurements of the wanted quantities (position and heading), it is convenient to use an *error-state* Kalman filter. Rather than estimating the position and heading directly, this filter estimates *errors* in measured and computed quantities.

Table 1. Available sensor measurements for the integrated navigation system

Sensor	**Measurement**	**Typical accuracy (1σ)**
HiPAP + DGPS	UUV position (relative earth)	2 m - 4 m
MRU	Roll and pitch UUV angular velocity (relative the inertial frame) projected into the body coordinate system	0.07° > 10°/h
Compass	Heading	2° - 3°
DVL	UUV velocity (relative the seabed) projected into the body coordinate system	0.015 m/s
Pressure sensor	Depth (after calculations)	0.1 m

In order to estimate any errors, we need some kind of redundant information, which in case of an error-state Kalman filter should be realized by providing more than one measurement of each state. As seen in Table 1, no such measurements are available, and hence we need external computations, i.e. some combination of measurements calculating the desired quantity:

- An alternative position can be calculated by integrating the body fixed velocity vector in the direction given by the measured roll, pitch and heading (dead reckoning).
- Integration of the angular rates with roll and pitch can give an alternative heading (compensating for earth's angular rate).

In this manner we get two independent positions and headings available. The independent positions and headings also have complementary characteristics. Whereas the measured quantities may have significant high-frequency errors, the computed quantities will be very accurate in the high-frequency band, as they are based on measurements of the derivative. On the other hand the computed quantities have very poor low-frequency properties, drifting off the true value due to sensor errors. Hence, the limited errors of the measured position and heading are vital to ensure low-frequency stability of the Kalman filter estimates. Altogether a combined solution offers increased low and high frequency accuracy.

Measurements to the error-state Kalman filter are the difference between measured and computed quantities, as shown in Fig. 2.

Based on the measurements and sensor error models, the Kalman filter estimates all the colored sensor errors and the errors in the computed quantities.

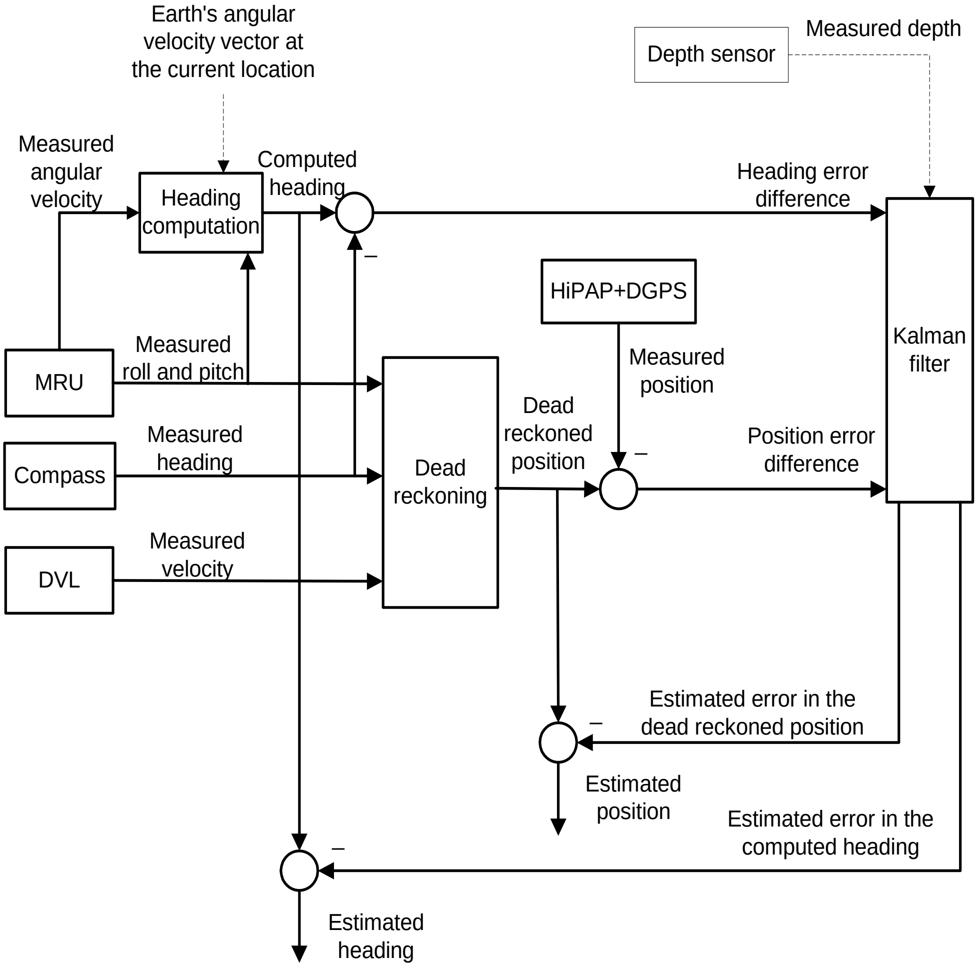


Fig. 2. Kalman filter structure

The sensor error models were found by established system identification methods. Sensor data from both sea trials and static conditions (fixed HiPAP transponder, fixed DGPS receiver and fixed UUV orientation) was used. The errors were modelled as combinations of white and colored noise as shown in Table 2. The colored parts are well represented by first order Markov processes.

The colored sensor errors thus sum up to four Kalman filter states (the position measurement error has both a north and east component). Further, one single integration gives a new state, leading to three states from the estimation of errors in the computed quantities. Thus, the Kalman filter has a total of seven states.

Table 2. Summary of the sensor error models

Sensor	**Colored part**	**White-noise part**
HiPAP + DGPS	**X**	**X**
MRU, roll and pitch		**X**
MRU, angular rate	**X**	
Compass	**X**	**X**
DVL		**X**

The final position and heading estimates can be calculated by subtracting the corresponding error estimates from either the measured or the computed quantities. As shown in Fig. 2, the latter is preferred, motivated by the following:

- White-noise is not possible to estimate, hence only the colored parts of the errors in the position and heading measurements are estimated. Consequently, a measurement based estimate would contain white-noise. As for the computed quantities, the integration process has eliminated the white noise component from the MRU, compass and Doppler velocity log, and the entire error may be estimated.
- The computed quantities are higher order processes, and are much more correlated in time than the first order Markov processes. Hence, errors in the dead reckoned position and the computed heading are far more predictable, giving more accurate a priori estimates and thus reduced a posteriori estimation uncertainty. During measurement drop-outs, the Kalman filter can only predict the errors, and the predictability is particularly important.
- Due to occasional measurement drop-outs of the DGPS or HiPAP, the more reliable dead reckoned position is a preferred basis.

4 Smoothing

A Kalman filter is recursive and its estimates at time t_k are based on all measurements prior to and including t_k. Since there is no real-time requirement in the post processing, measurements after t_k should also be utilized. The matter of finding an optimal estimate based on both previous and future measurements is referred to as *smoothing*.

Smoothing has several advantages compared to just a conventional Kalman filter:

- Since all measurements are known a priori, there is no delay in the estimates.
- The smoothed estimates are in accordance with the process model. This is different from a conventional Kalman filter, where the process model is used only in the prediction part. Thus, when updating the filter, unexpected measurements lead to steps in the a posteriori estimate.
- In a conventional Kalman filter, estimating the current state, most weight is put on the latest measurements (due to the states' correlation in time). Thus making smoothed estimates, the number of relevant measurements is doubled.
- During measurement drop-outs, the estimation uncertainty of an ordinary Kalman filter increases in accordance with the process noise until the measurement is back. Knowledge of the next measurement reduces the uncertainty increase-rate of the smoothed estimate and causes its maximum to occur in the middle of the drop-out time interval.

To find the smoothed estimates, first the ordinary Kalman filter is run through the whole time series, saving all estimates and covariance matrices. The saved data is then processed recursively backwards in time using an optimal *smoothing algorithm* (Minkler & Minkler [4] or Gelb [5]) adjusting the filtered estimates.

5 Filter characteristics and performance

This section includes results from tests where both simulated measurements and measurements from sea trials were applied to the designed filter

5.1 Observability

All the Kalman filter states are observable. However, errors in the position measurement that are more low-frequent than the drift in the dead reckoned position, are not possible to estimate. In the compass measurement though, both high-frequency and low-frequency errors are estimated (assuming UUV velocity not zero). The high-frequency error is found by means of the gyros (computed heading), and the low-frequency part is estimated by observing the drift in the dead reckoned position (with the aid of the DGPS/HiPAP position measurement).

5.2 Simulation results

Simulations are very useful for demonstrating typical filter characteristics. A set of simulated measurements was derived from dynamical models of the UUV, the environment and sensor errors. The subset of sensor error models which was also implemented in the Kalman filter, was based on the same error-modelling. The simulated UUV trajectory is shown in Fig. 3, the UUV forward velocity was 2.1 m/s.

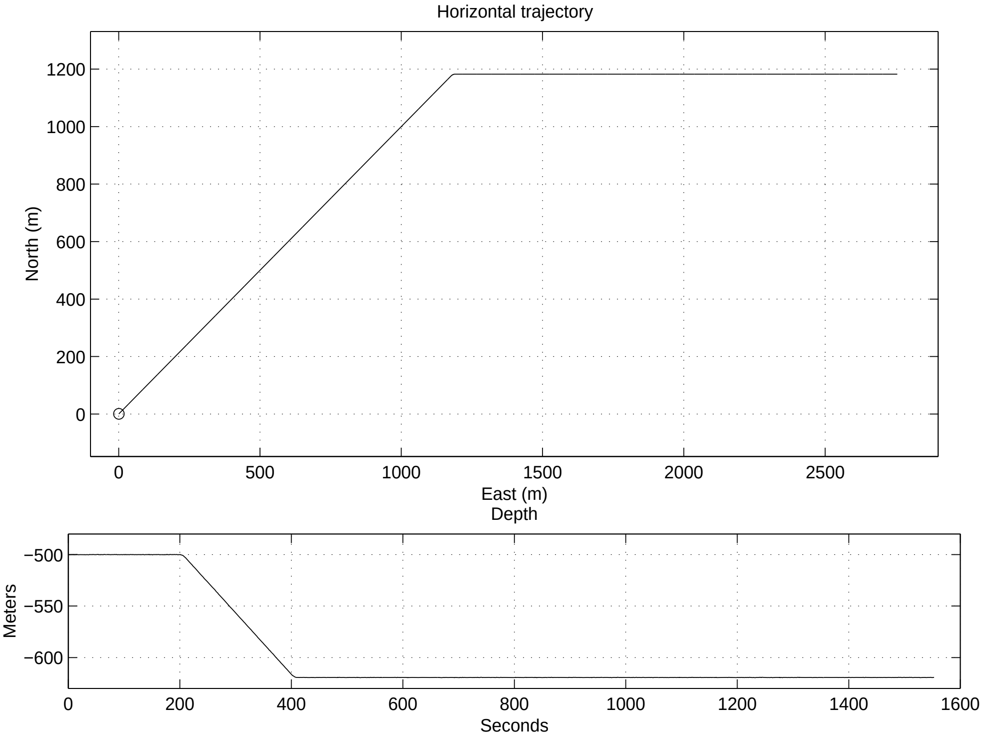


Fig. 3. Simulated UUV trajectory. Circle: starting point.

5.2.1 Position estimation

Fig. 4 shows the results from the position estimation. The UUV has moved straight eastwards, and clearly the position measurement contains both high- and low-frequency noise. The dead reckoned position is drifting, but is very smooth, which in this case means small high-frequency errors. As the

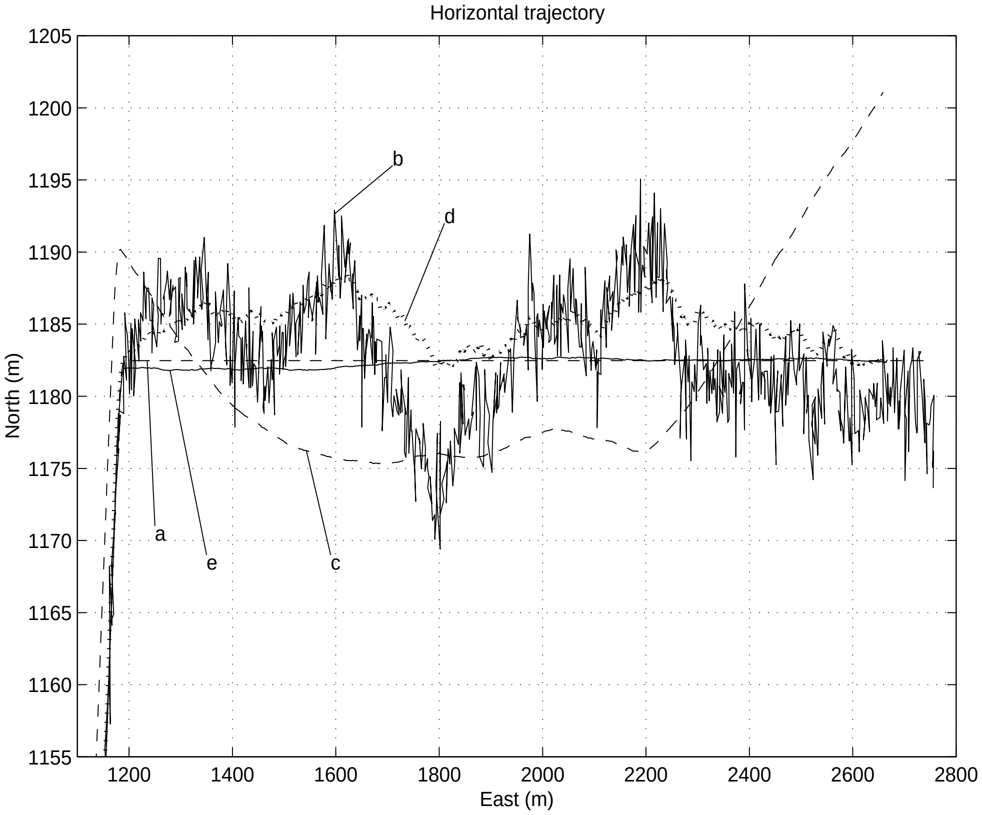


Fig. 4. Horizontal UUV trajectory. True (a), measured (b), dead-reckoned (c), filtered estimate (d), smoothed estimate (e)

Kalman filtered estimate is susceptible to measurement noise, it is not as smooth as the dead-reckoned position. However, the smoothed estimate is both smooth and very close to the truth.

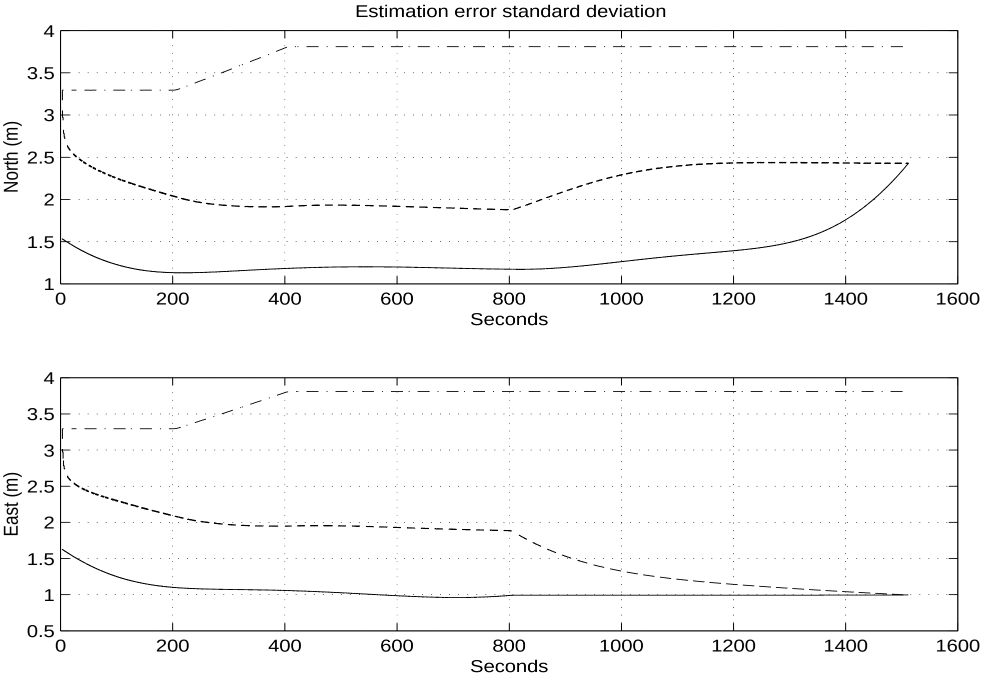


Fig. 5. Position estimation uncertainty (standard deviation). Filtered estimate (dashed), smoothed estimate (solid), measurement uncertainty (dash-dot)

The accuracy of the north and east position estimates is shown in Fig. 5. For comparison, the position measurement uncertainty is also indicated. The change in this quantity is due to the depth increase, leading to lower HiPAP accuracy.

Because the filter is initialized by the position measurement, the accuracy of the filtered estimate equals the measurement accuracy in the first time step. As the number of relevant measurements increases, the accuracy converges to below 2 meters. At time = 800 seconds, the UUV starts a 45° turn, leading to increased uncertainty in the north direction and decreased uncertainty in the east direction. This demonstrates the difference in accuracy along-track and cross-track, which is due to a similar characteristic of the dead-reckon drift. The main contributor to the dead-reckon drift is the compass-error, whose drift contribution is of first order in the cross-track direction, but only of second order along-track.

From the figure it is evident that the smoothed estimate is generally better, but at the last time step there are no more future measurements available, and the smoothed estimate equals the filtered, both in accuracy and value.

5.2.2 Heading estimation

Fig. 6 shows the heading estimation. The graphs are very similar to the corresponding graphs of the position estimation. In addition it is apparent that both the computed heading and the filtered heading estimate are initialized by the compass measurement.

The heading estimation error standard deviation is shown in Fig. 7. The smoothed solution offers more than a tenfold improvement in accuracy over the compass measurement.

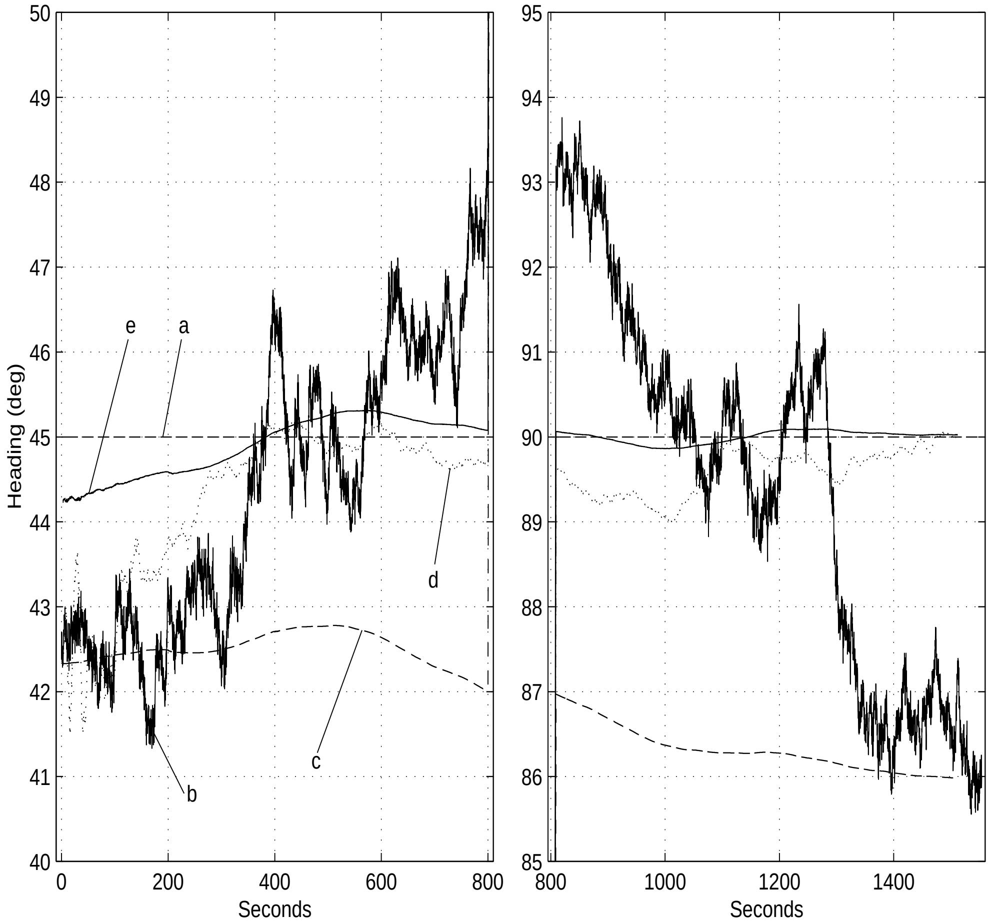


Fig. 6. UUV heading. True (a), measured (b), computed (c), filtered estimate (d), smoothed estimate (e)

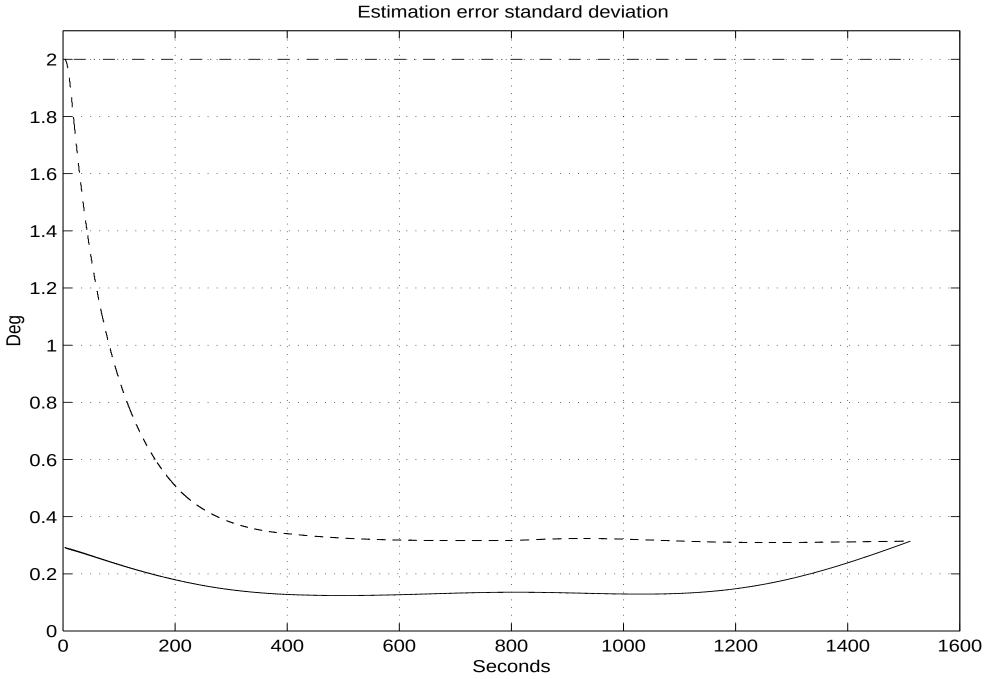


Fig. 7. Heading estimation uncertainty (standard deviation). Filtered estimate (dashed), smoothed estimate (solid), measurement uncertainty (dash-dot)

5.3 Results from real surveys

Fig. 8 shows the horizontal trajectory from a real survey in Boknafjorden in Norway, at a depth of approximately 320 m. The Kalman filtered position estimate is clearly more susceptible to measurement errors than the smoothed. The considerable drift in the dead-reckoned position is due to a significant steady compass error.

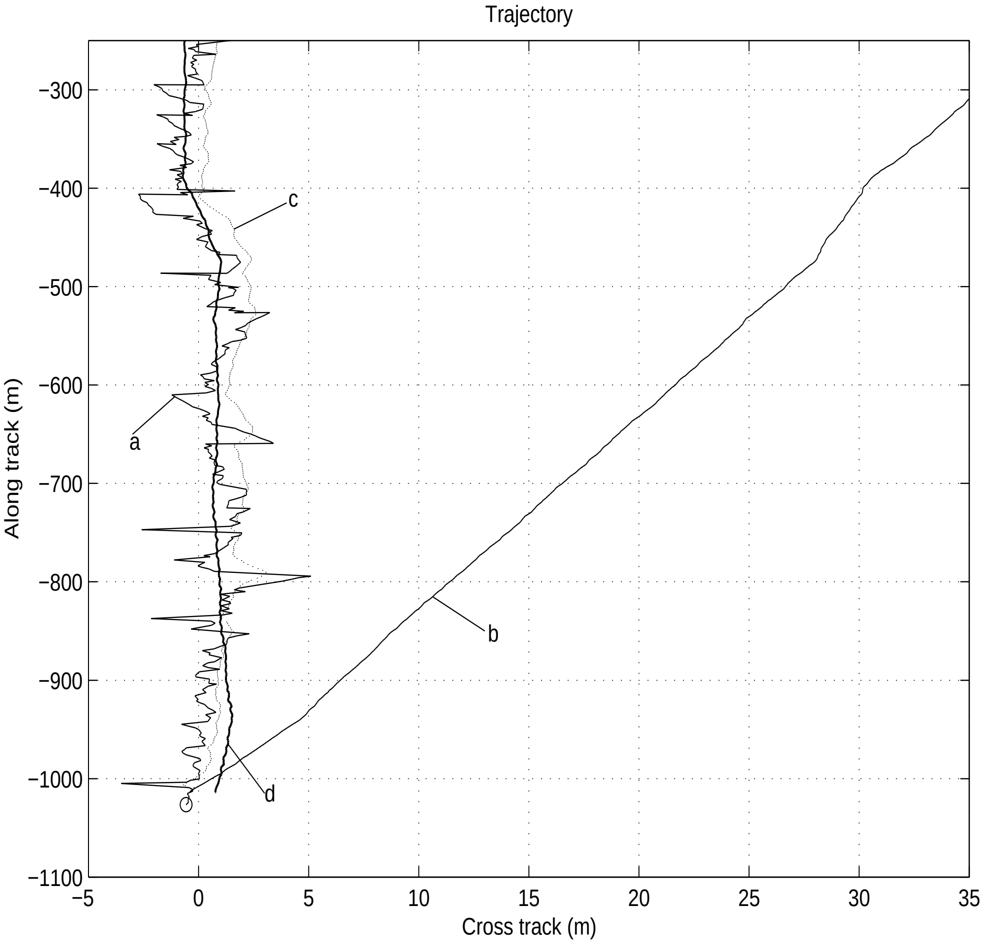


Fig. 8. Horizontal UUV trajectory. Measured (a), dead-reckoned (b), filtered estimate (c), smoothed estimate (d)

In Fig. 9 measured and estimated heading are shown. According to the filter there is a compass error in the order of 3°. At the time of this survey, declination and the UUV's magnetic signature was not yet compensated for.

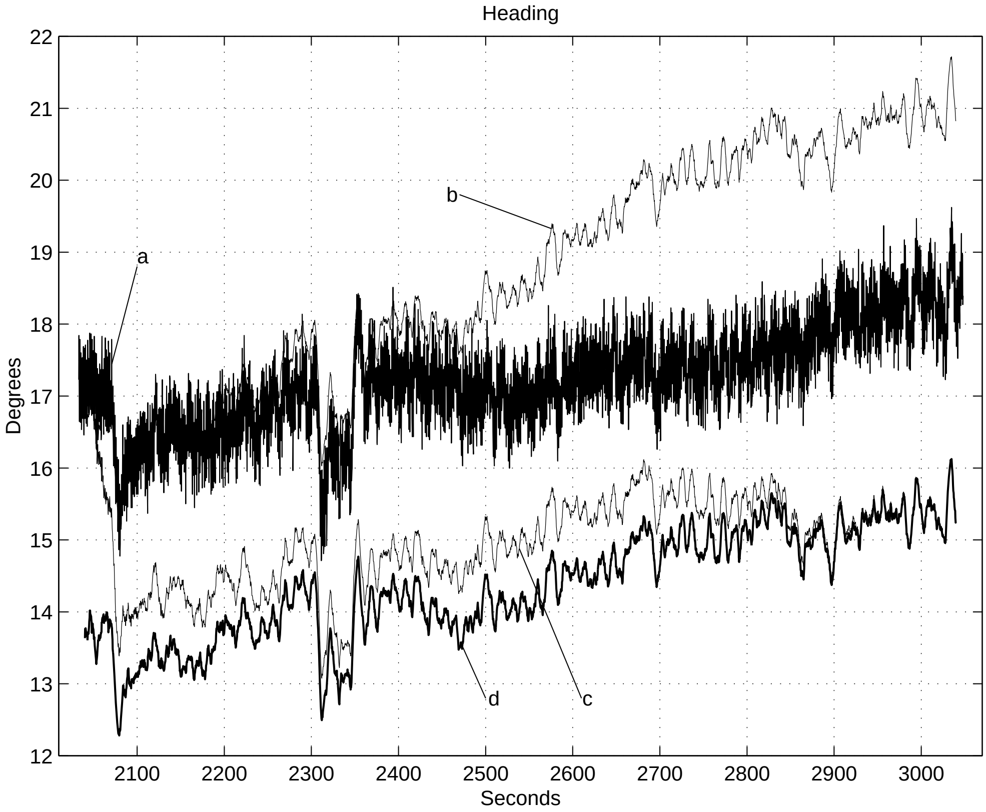


Fig. 9. UUV heading. Measured (a), computed (b), filtered estimate (c), smoothed estimate (d)

6 Verification of performance

There are several established methods for determining the quality of the produced Digital Terrain Model (DTM). An obvious method would be to map a marker placed on the seabed in a known location. At present, no such data is available, but natural features, for instance rocks, are visible on the sonar data and can be classified as objects. In cases where we have overlapping sonar data and can identify the same object on two footprints, an offset between the two observations indicates DTM position error(s). According to Jalving and Gade [3], the UUV heading and position uncertainties are the main contributors to the DTM position uncertainty. Comparing the offset between the observations prior to[1] and after the filtering thus gives an idea of the improvement achieved in the post processing.

The observed offsets can also be compared with theoretical values calculated from the uncertainty in the position and heading. Prior to the filtering, the position and heading uncertainties are given by the DGPS/HiPAP and compass accuracies (listed in Table 1). For the observations in the filtered data set, the theoretical value is based on the Kalman filter standard deviation of the position and heading error estimate. Due to a temporarily invidious installation of a magnetic valve and a few other non-ideal circumstances, we assumed a heading uncertainty of 0.8° instead of the much better Kalman filter standard deviation. Table 3 summarizes comparisons for all the objects we found in the runs from Åsgard Transport and Boknafjorden. Each object is mapped two times separated with a time interval of 30 minutes or more, passing the object from opposite directions. Most of the objects from the Åsgard Transport were at a depth about 350 meters.

For some objects there was no measurable position offset between the two observations, which is indicated in Table 3 by using the “less than” sign (<). The value after the sign is dependent upon the accuracy of the observation. When computing average, this value is divided by two.

[1] Prior to the filtering, the measured position and heading are used directly.

Table 3 Comparison of object observation position offset in filtered data set (smoothed position and heading) and unfiltered data set (combined DGPS/HiPAP position and compass heading). Theoretical uncertainties of the two data sets are also calculated.

	Observation position offset prior to filtering (m)		Observation position offset after smoothing (m)		Theoretical uncertainty prior to filtering (1σ) (m)		Theoretical uncertainty after smoothing (1σ) (m)	
Object no.	North	East	North	East	North	East	North	East
1	**7.5**	2.0	< 0.5	< 0.5	5.12	3.40	1.17	1.00
2	4.6	2.3	< 0.5	< 0.5	5.06	3.40	1.16	1.00
3	1.8	**4.5**	< 0.5	0.8	4.92	3.39	1.15	1.00
4	3.0	1.2	< 0.5	**2.0**	4.92	3.38	1.15	1.00
5	**6.2**	2.0	< 0.5	< 0.5	5.43	3.42	1.21	1.00
6	4.2	1.3	**1.6**	< 0.5	5.59	3.41	1.22	1.00
7	**7.0**	**5.0**	**1.5**	**1.2**	5.48	3.32	1.21	0.99
8	**8.0**	< 0.5	**1.7**	0.6	4.64	3.32	1.12	0.99
9	**8.5**	0.7	**2.0**	0.5	4.26	3.32	1.08	0.99
10	3.6	2.2	**1.7**	0.5	4.82	3.31	1.14	0.99
11	0.7	2.0	**1.3**	**2.0**	3.28	3.24	1.13	1.27
12	2.9	1.1	0.3	< 0.2	3.24	3.24	1.13	1.27
13	0.1	0.8	0.1	0.3	3.30	3.24	1.14	1.27
14	3.3	2.3	< 0.5	< 0.5	4.17	3.76	0.84	0.79
Average:	4.39	1.98	0.84	0.66	4.59	3.37	1.13	1.04

In Table 3 we notice a significant improvement in the filtered data. Furthermore, we can compare the observation position offset after filtering with its theoretical standard deviation. Assuming normally distributed errors, 68% of the observed position offsets should be within its standard deviation. The bold figures indicate an offset exceeding the standard deviation, and we have 19 of 28 inside, which is exactly 68%! However, this test only compares each value with a boundary, not taking into account how far from the boundary they are. Investigating the average actually indicates a better performance than anticipated. This may suggest that the filtered heading uncertainty of 0.8° used in the theoretical standard deviation calculations is too conservative. In Table 3 we also notice that the observation position offsets prior to filtering are slightly better than the theoretical values. This is probably due to a counteractive effect of the UUV static magnetic signature.

7 Conclusions

The accuracy of seabed maps based on UUV data can be considerably improved by an aided navigation post processing filter.

It has been demonstrated that a combination of all relevant sensors in an error state Kalman filter offers a far more accurate position and heading, than direct use of the position and heading measurements. Further, the Kalman filtered estimates may be considerably enhanced through a smoothing algorithm. At 300 m depth, a UUV position accuracy of 1 m (1σ) and a heading accuracy of 0.5° (1σ) has been achieved.

References

[1] N. Størkersen, J. Kristensen, A. Indreeide, J. Seim and T. Glancy, “HUGIN - UUV for seabed surveying”, *Sea Technology*, February 1998.

[2] B. Jalving and K. Gade, Positioning Accuracy for Seabed Surveying Untethered Underwater Vehicles (UUVs), submitted to IEEE Journal of Oceanic Engineering.

[3] B. Jalving and K. Gade, Positioning Accuracy for the HUGIN Detailed Seabed Mapping UUV, Proceedings from Oceans '98, 28. Sept. - 1. Oct. 1998, Nice, France

[4] G. Minkler and J. Minkler, *Theory and Application of Kalman Filtering*, Magellan Book Company, 1993.

[5] A. Gelb, *Applied Optimal Estimation*, The MIT Press, 1974

Paper IV

Gade, K. (2005). NavLab, a Generic Simulation and Post-processing Tool for Navigation. *Modeling, Identification and Control*, vol. 26, no. 3, pp. 135-150

MODELING, IDENTIFICATION AND CONTROL, 2005, VOL. 26, No. 3, 135–150
doi: 10.4173/mic.2005.3.2

NavLab, a Generic Simulation and Post-processing Tool for Navigation

KENNETH GADE
Norwegian Defence Research Establishment (FFI)

Keywords: *Navigation Software, Simulation, Estimation, Analysis, Post-processing, Aided Inertial Navigation System, Kalman Filtering, Optimal Smoothing*

Abstract

The ambition of getting one common tool for a great variety of navigation tasks was the background for the development of NavLab (Navigation Laboratory). The main emphasis during the development has been a solid theoretical foundation with a stringent mathematical representation to ensure that statistical optimality is maintained throughout the entire system. NavLab is implemented in Matlab, and consists of a simulator and an estimator.

- Simulations are carried out by specifying a trajectory for the vehicle, and the available types of sensors. The output is a set of simulated sensor measurements.
- The estimator is a flexible aided inertial navigation system, which makes optimal Kalman filtered and smoothed estimates of position, attitude and velocity based on the available set of measurements. The measurements can be either from the simulator or from real sensors of a vehicle.

This structure makes NavLab useful for a wide range of navigation applications, including research and development, analysis, real data post-processing and as a decision basis for sensor purchase and mission planning. NavLab has been used extensively for mass-production of accurate navigation results (having post-processed more than 5000 hours of real data in four continents). Vehicles navigated by NavLab include autonomous underwater vehicles (AUVs), remote operated vehicles (ROVs), ships and aircraft.

1 Introduction

For many navigation related activities it is very useful to have one common software tool. The tool should cover applications such as navigation system research and development, analysis and real data post-processing. With a long tradition of developing navigation systems, The Norwegian Defence Research Establishment (FFI) started development of such a tool in 1998. The result is NavLab (Navigation Laboratory), a powerful and versatile tool that serves a variety of navigation purposes. For the long-term success of this tool, a strong focus on a solid theoretical foundation and a flexible structure has been crucial.

1.1 NavLab’s theoretical foundation

The most significant feature of NavLab is its solid theoretical foundation. NavLab is a result of an innovative research process to establish a completely general theoretical basis for navigation and for implementation of navigation systems. The development has led to the following contributions:

- A new stringent and unified system for notation and mathematical representation
- A unified design and implementation of algorithms and aiding techniques for the Kalman filter, where statistical optimality is maintained throughout the entire system
- Elimination of numerical problems by
    - Deducing and implementing exact formulas (rather than approximations)

- Using only nonsingular representations
- Controlling accumulation of the computer's inherent round-off errors

Articles reporting the above work will be published, but currently the most relevant report available is [1].

1.2 A flexible structure

The main structure of NavLab is shown in Figure 1. NavLab's different components can be used alone or together, allowing a variety of applications. A list of usages is given in Section 4.

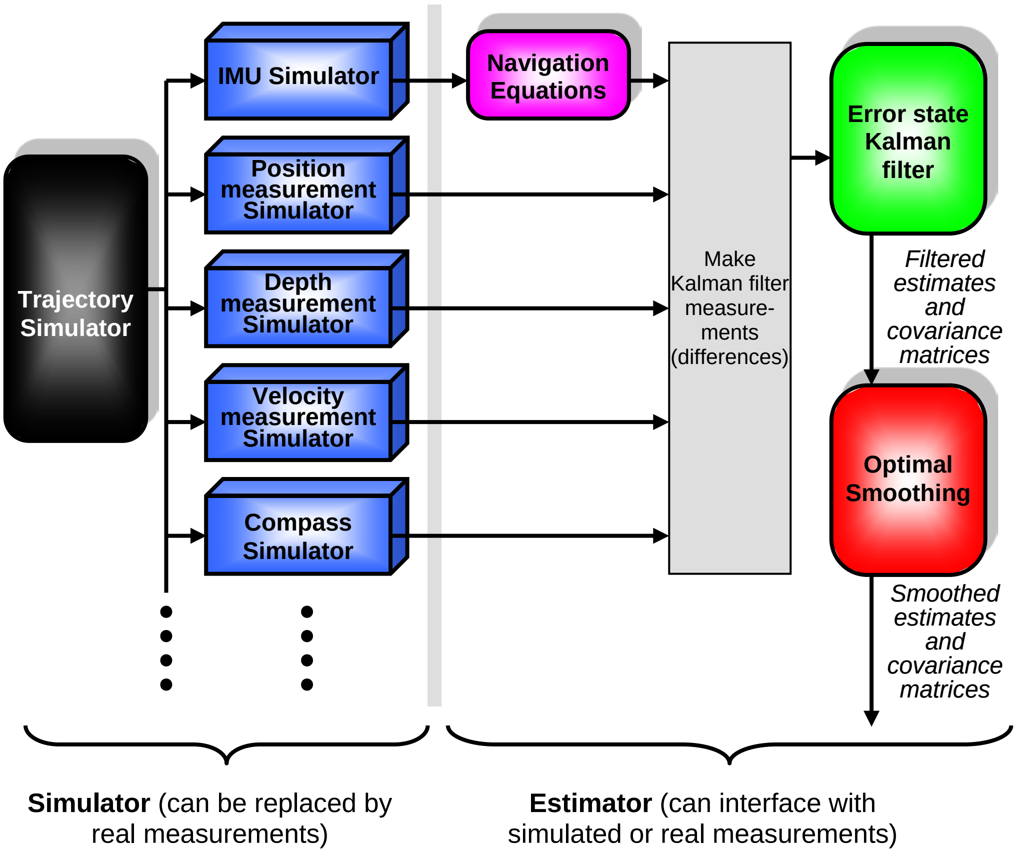

Trajectory Simulator
IMU Simulator
Position measurement Simulator
Depth measurement Simulator
Velocity measurement Simulator
Compass Simulator
Navigation Equations
Make Kalman filter measure-ments (differences)
Error state Kalman filter
Filtered estimates and covariance matrices
Optimal Smoothing
Smoothed estimates and covariance matrices
Simulator (can be replaced by real measurements)
Estimator (can interface with simulated or real measurements)

Figure 1. NavLab main structure. Note: The colors used in the figure correspond to the colors of the graphs generated by the different parts of NavLab (blue is the measurement, red is the smoothed estimate etc).

The simulator can simulate artificial measurements from a chosen scenario. The estimator will, based on the available set of measurements from either the simulator or from sensors of a real vehicle, make the best possible estimates of position, attitude, velocity and sensor errors. The simulator and estimator are described in more detail in Sections 2 and 3.

In addition to the simulator and estimator, NavLab includes:

- A pre-processing tool (Preproc), which is used to handle real measurements (by removing outliers, compensating for lever arms and misaligned sensors, converting measurements to the correct format etc).

- An export tool, which creates files for exporting to other programs (containing the estimated position, attitude and velocity).

Figure 2 shows the NavLab program modules. Different modules are used in different cases. Typical examples are:

- *Simulations*: Simulator → Estimator
- *Post-processing of real data*: Preproc → Estimator → Export

The modules interface each other via files of a specified format (see [2]), or via memory to save time.

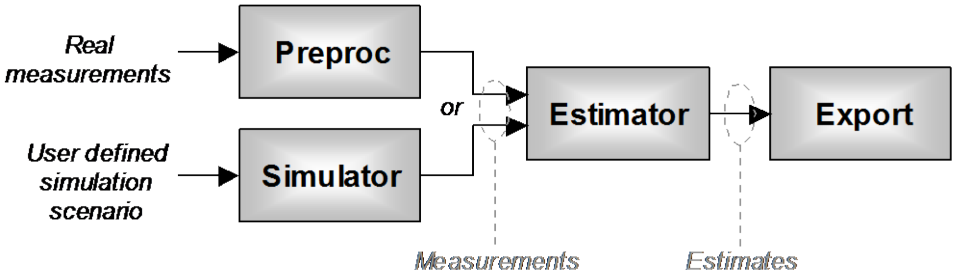


Figure 2. NavLab program modules

2 Simulator

The trajectory simulator can simulate any vehicle trajectory specified by the user. In addition, the user specifies a set of available sensors and their characteristics. Based on the specified trajectory and sensor characteristics, the sensor simulators calculate a set of artificial sensor measurements.

2.1 Trajectory simulator

The coordinate systems *I* (Inertial), *E* (Earth), *L* (Local) and *B* (Body) are simulated (see [2] for definitions). All relevant positions, orientations, linear and angular velocities, accelerations and forces describing the trajectory are calculated.

Features:

- Any trajectory in the vicinity of the Earth can be simulated (with unlimited complexity).
- All vehicle attitudes can be simulated without singularities.
- All possible vehicle positions relative to the Earth can be simulated without singularities.
- Includes all Coriolis and centripetal effects due to the rotating Earth and own movement over the Earth curvature.
- Includes WGS-84 gravity model and elliptic Earth model.

Trajectories are specified in the trajectory simulator by first giving the initial position, attitude and velocity, and then specifying changes in attitude and velocity as a function of time. When developing a trajectory simulator, the actual mathematical quantities that are used to describe these changes must be selected carefully, to ensure that it is simple for the user to express a trajectory that follows the Earth ellipsoid in both position and attitude. Selecting the mathematical quantities[1] $\boldsymbol{\omega}_{LB}^{B}$ and $\dot{\boldsymbol{v}}_{EB}^{B}$ actually makes this just as simple for the user as it would have been if the surface of the Earth were planar. Thus if no changes in these quantities are specified, the vehicle will travel around the Earth at constant depth/height if the initial velocity was horizontal.

Figure 3 shows an example trajectory from the simulator. This trajectory is simply specified by two periods of constant change in attitude (angular velocity about z, $\boldsymbol{\omega}_{LB}^{B} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{T}$ deg/s) and two periods of constant change in velocity (deceleration/acceleration in z, $\dot{\boldsymbol{v}}_{EB}^{B} = \begin{bmatrix} 0 & 0 & 20 \end{bmatrix}^{T}$ m/s^2).

Using a plugin for NavLab, it is also possible to specify the trajectory by giving a dynamical model of the vehicle and then marking 3D waypoints in a map, see [3].

[1] $\boldsymbol{\omega}_{LB}^{B}$ is the angular velocity of the body, *B*, relative to *L*, where *L* is a local system with zero angular velocity relative to Earth about its vertical axis (see [1] or [2] for more details). $\dot{\boldsymbol{v}}_{EB}^{B}$ is the velocity of *B* relative to Earth, differentiated in the *B* system.

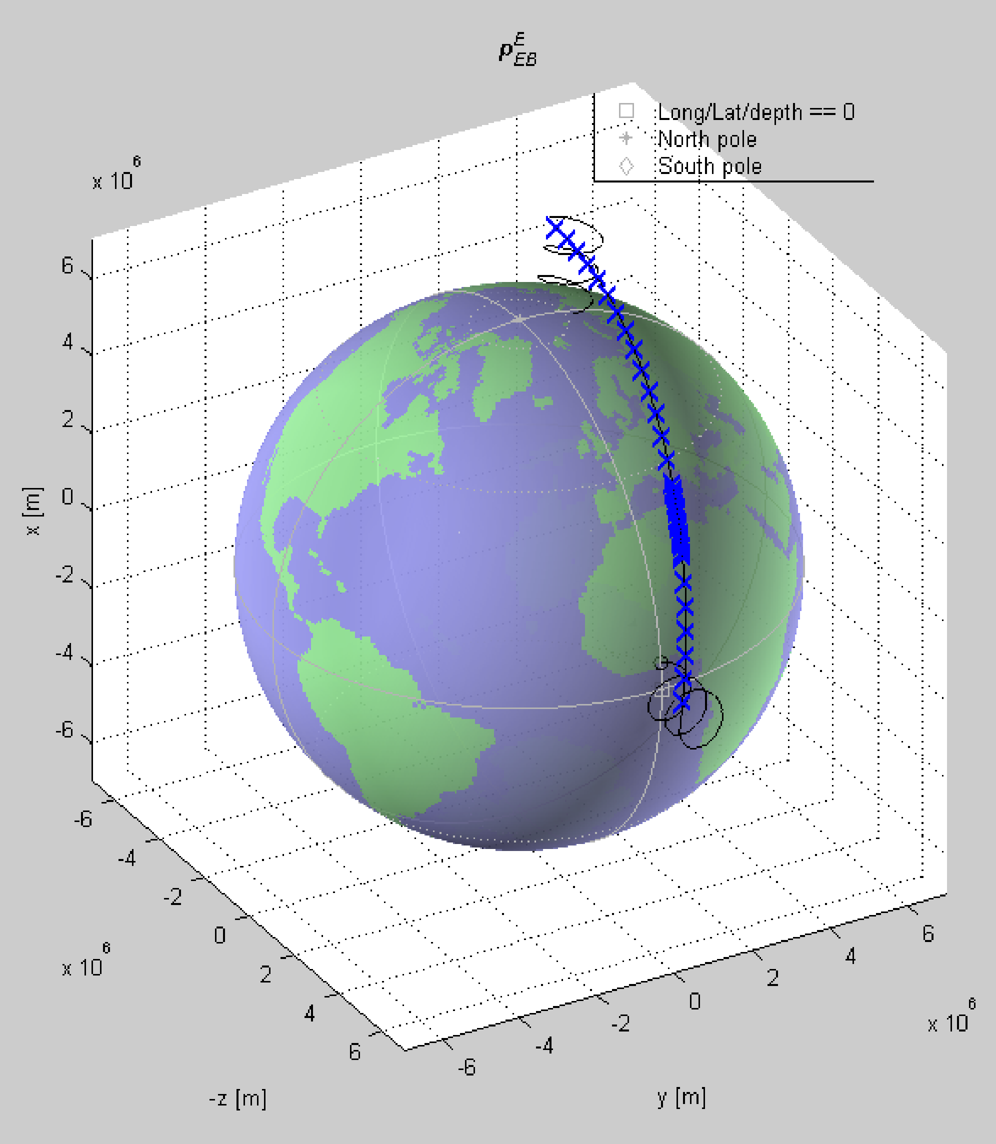


Figure 3. Earth plot from NavLab. Black circle: Starting point. Black line: True trajectory (from the trajectory simulator). Blue crosses: Simulated position measurements.

2.2 Sensor simulators

The most significant error types, such as white-noise, colored noise and scale factor error are included in the sensor simulators, and any other types can also be added. The magnitude, time-constants and other parameters that describe the different errors are user selectable, and can be given as fixed values or as functions of time.

The sensor simulators can produce measurements at any user-specified time. This can be specified as a constant rate during the entire simulation, different rates in different intervals, or each single time of measurement can be specified in a time-series. Figure 3 shows position measurements with one period of high rate, and also periods of low and zero rate.

3 Estimator

The main purpose of the estimator is to estimate a vehicle’s position, attitude and velocity. This is done by combining all available knowledge such as sensor measurements and mathematical models of the sensor errors. The optimal (given certain assumptions) method of combining this knowledge is by means of a Kalman filter[1] (see [4] for details). Thus, if the model used in the Kalman filter is correct, all information is used optimally, and no better estimates can be made. An example illustrating this is the concept of gyrocompassing, i.e. finding north by inspecting the direction of the Earth's angular velocity, measured by the gyros. Gyrocompasses are manufactured containing gyros, accelerometers and dedicated algorithms for this purpose. When the same sensors are available for the estimator, it will gyrocompass optimally as a natural part of its estimation procedure.

The main structure of the estimator is given in Figure 4. Measurements from the IMU (Inertial Measurement Unit) are integrated by the navigation equations (see Section 3.1) to calculate position, attitude and velocity. Each time-step where a measurement from any of the aiding sensors is available, it will be compared to the corresponding quantity from the navigation equations, and the difference is sent as a measurement to the Kalman filter.

Note that each of the sensors shown in Figure 4 are general and can represent different types, e.g. NavLab has used different types of position measurements, including range measurements to a known position (see [5] or [6] for examples of different sensor types that have been integrated).

The navigation equations and optimal smoothing are described in Sections 3.1 and 3.2.

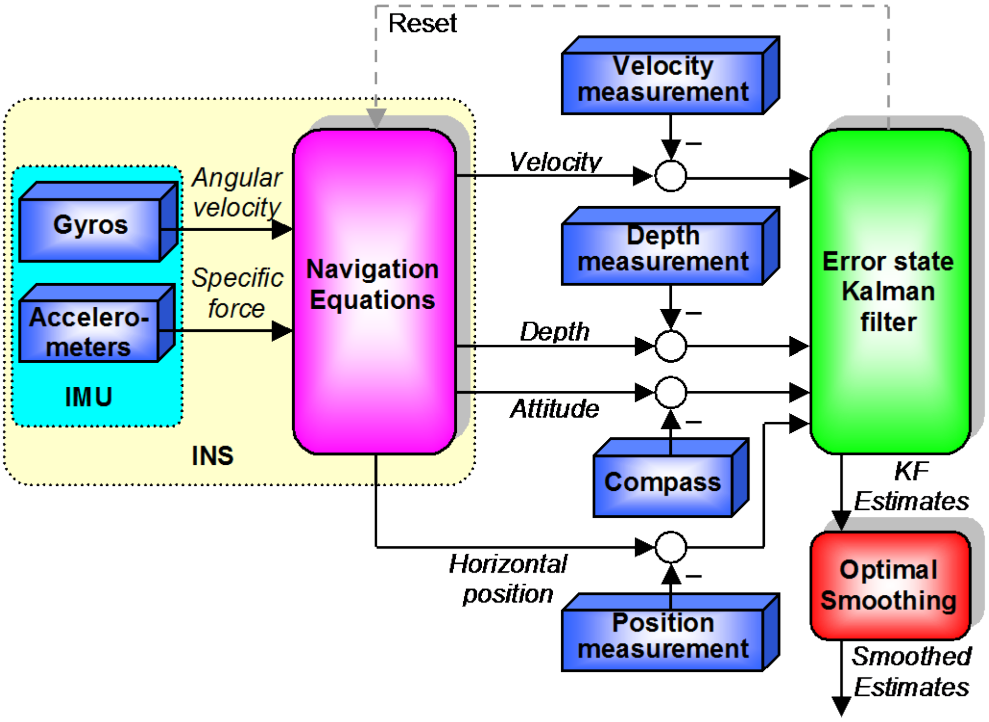


Figure 4. Estimator main structure (simplified). The sensors shown can be either simulated or real. (INS: Inertial Navigation System)

[1] If future measurements are available, a better estimator exists, see Section 3.2.

Features:

- The estimator accepts arbitrary time-series of measurements from all sensors.
- Along with each single sensor measurement, new sensor parameters can be specified, describing that particular measurement, hence describing a varying quality.
- Zero velocity update (ZUPT) and depth/height measurements are included in the same Kalman filter in an optimal manner.
- The horizontal position measurements are nonsingular (i.e. with maximum accuracy also near/at the poles).
- Iterated Extended Kalman filter is used to improve the performance in cases of significant nonlinearities.

3.1 Navigation Equations

The navigation equations calculate position, attitude and velocity based on the IMU measurements, as shown in Figure 4.

Features:

- Nonsingular for all positions and attitudes
- Foucault wander azimuth
- Direction cosine matrix attitude update
- Numeric drift control
- WGS-84 gravity model and elliptic Earth model
- Trapezoid updates to prevent systematic errors from the forward or backward Euler methods

3.2 Optimal Smoothing

The Kalman filter is the optimal estimator at time *t*, when measurements before and including *t* are used, thus it is well suited for real-time estimation. However, if measurements after *t* are also available (which is the case for post-processing, see Section 4.1), it is possible to make a better estimator at time *t*, by using these additional measurements. The best possible algorithm, utilizing all measurements both before and after *t*, is called *optimal smoothing* (see [4] for details).

- This algorithm is effectively doubling the set of relevant measurements for each estimate, since the *next x* seconds of measurements are normally just as important as the *previous x* seconds.
- A symmetrical interval of past and future measurements prevents a systematical delay in the estimates, which is unavoidable in real-time estimators.

- Another limitation of an optimal real-time estimator (Kalman filter) is its inability to deliver estimates that are in accordance with the process model. At each time-step such estimators make a prediction (that is in accordance with the process model), but when a new measurement arrives, it is weighed against the prediction to give a new updated estimate. Unexpected[1] measurements thus lead to jumps in the estimates that are not in accordance with the process model (e.g. an unexpected velocity measurement leads to a jump in the velocity estimate that corresponds to an acceleration that is too large according to the process model). Since no measurements are unexpected for the smoothing algorithm, this problem is eliminated, and the smoothed estimate is always in accordance with the process model (hence the name “smoothing”).

Figure 5 shows an example of position estimation uncertainty (1σ) in the Kalman filter and in the optimal smoothing. Position measurements are unavailable in an interval of 2 hours, and in this period the Kalman filter estimation uncertainty grows, before dropping instantly when position measurements become available at the end. The smoothing algorithm on the other hand, utilizes the position measurements at the end during the whole interval, and thus has a maximum uncertainty in the middle of the interval. At the last time-step, no future measurements are available and the two algorithms give equal estimates.

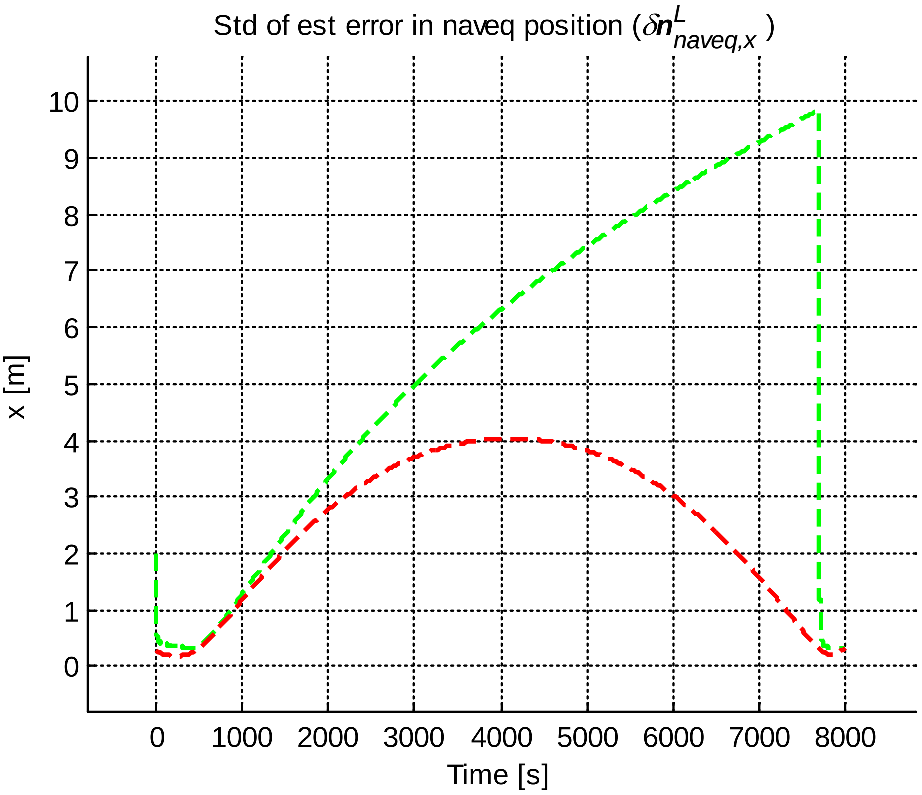


Figure 5. Estimation uncertainty in north-position by Kalman filter (green) and optimal smoothing (red). (A straight-line trajectory to the east, at latitude 45° is simulated. Sensors: 1 nmi/h class IMU, 600 kHz DVL. Position measurements are available the first 500 seconds and the last 300 seconds.)

[1] All measurements that are not exactly equal to the predicted value are unexpected, which in practice means every measurement.

3.2.1 Performance in cases with large modeling errors (robustness)

Another property of the smoothing algorithm, that is often even more important than the improved accuracy, is its robustness. As mentioned above, smoothed estimates are always in accordance with the process model, and this quality is crucial in cases with wrong models or faulty measurements. If a measurement has an error that is significantly larger than what was modeled in the Kalman filter, a large jump in the estimates from the real-time filter is inevitable. A real-data example of such a jump is shown in Figure 6, where a position outlier (wild-point) with an error of about 41 meters is present[1]. Since the Kalman filter expects a total position measurement uncertainty of 2.4 m (1σ), the error of this measurement is above 17 sigma, and hence extremely unlikely according to the model. In the example, the outlier is followed by a period of position measurement dropout (which is typical), and thus the filtering error remains until the sound[2] measurements bring the estimate back on track. The smoothing algorithm however, also seeing the measurements from all sensors *after* the outlier, is barely affected, even though it uses the same sensor model as the Kalman filter.

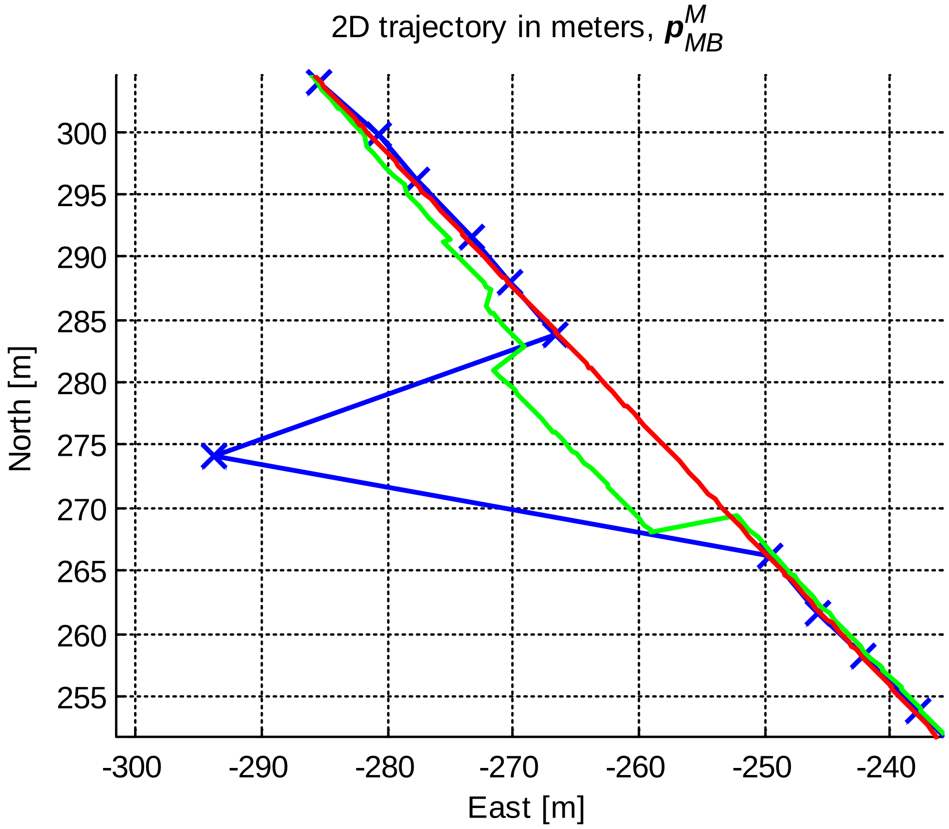


Figure 6. Trajectory from the HUGIN 1000 AUV. The track shows the vehicle going northwest. Blue: position measurement from DGPS+USBL (differential GPS + Ultra Short Base Line acoustic positioning). Green: Kalman filtered estimate. Red: Smoothed estimate.

The optimal smoothing algorithm is also robust against systematic sensor errors. In a HUGIN 3000 navigation accuracy verification sea trial in October 2000 (described in Section 5.2.2), there was a constant error in the DVL (Doppler Velocity Log) measurements that was above

[1] Outliers of this magnitude will by default be automatically removed in NavLab by a wild-point detection algorithm, but is left here for demonstration.

[2] I.e. in accordance with the Kalman filter model.

8.3 sigma (due to an incorrect DVL configuration in this particular trial). This huge[1] unmodeled velocity error led to a position error in the order of 10-15 m for the real-time estimates, while the smoothing, using the same model, proved a performance of 1.2 and 1.7 meters (1σ north and east), see Figure 8 in Section 5.2.2.

4 NavLab usage

NavLab has been extensively used by numerous different users since 1999, including several international research groups, universities and commercial survey companies. The flexible structure of NavLab makes it useful for a wide range of applications. Some users are only working with simulated data, whereas others use the estimator alone to post-process real data. Finally, there are many cases where both simulations and real data processing are of interest. A summary of current NavLab usage is given below.

Navigation system research and development (using simulations and real data)

- Development, testing and comparison of new navigation concepts and algorithms, including new aiding sensors and aiding techniques.
- Development of real-time navigation systems, where the algorithms are implemented and tested in NavLab, and then ported to the real-time system. A typical development process is:
  - Implement algorithms in NavLab
  - Test in simulations (NavLab)
  - Test with real data (NavLab)
  - Port algorithms to the real-time navigation system (C++ or similar program language)
  - Test real-time system

  The real-time navigation system in the HUGIN vehicles was developed using NavLab (see [5] for a description of the real-time navigation system and [7] or [8] for an overview of the HUGIN AUV Programme).

Analysis of a given navigation system (using simulations and real data)

- Analysis of navigation system behavior under different maneuvers/trajectories and sensor configurations.
- Robustness analysis. The performance of the estimator is studied for the cases of:
  - Wrong sensor models used in the Kalman filter
  - Sensor dropouts
  - Sensor errors

Teaching navigation theory (using simulations)

By specifying appropriate simulations, everything from basic principles to complex mechanisms can be demonstrated and visualized.

Decision basis for navigation sensor selection/purchase (using simulations)

Simulations of the relevant scenarios are carried out to investigate how varying quality of the different sensors will affect the obtainable navigation performance. Parameters for different sensors available in the market are usually entered for comparison. The goal is to achieve a well-balanced and economical sensor suite.

[1] According to the model, the probability of an error of this magnitude in one measurement is only about 10^{-16}, and in this trial all measurements had this error!

Decision basis for mission planning (using simulations)

Even if the set of sensors is given, the navigation accuracy can vary significantly with the mission type. Important mission parameters include:

- Activation/deactivation of sensors or change of measurement rate (reasons to deactivate might be to stay covert, avoid interference with other systems or just to save power)
- Going to areas where certain measurements are available or are more accurate (e.g. go close to bottom to get DVL bottom track, go close to a transponder or go to surface for GPS measurements)
- Running maneuvers to increase the observability in the estimator
- Running in patterns that cancel out error growth

When setting up complex mission plans, simulations are helpful to ensure effective missions that meet the navigation accuracy requirements for all parts of the mission (transit phase, mapping phase etc).

Post-processing of real navigation data (using real data)

Post-processing of real data improves the navigation accuracy, robustness and integrity compared to a real-time navigation solution. See 4.1 for more details.

Tuning of real-time and post-processing navigation systems (using real data)

Proper Kalman filter tuning is essential for optimal estimation accuracy. Tuning is often based on the sensor specifications, but the actual sensor performance can differ from these numbers, and in such cases the tuning should be based on empirical data. Finding the correct tuning based on a recorded data set is best done by means of the error estimates from the smoothing algorithm.

Sensor evaluation (using real data)

After purchasing a new sensor, an evaluation of the sensor is usually desired. Large sensor errors might be detected by inspecting the measurements from this sensor alone, but for a more thorough sensor evaluation, the measurements should be compared with other sensors (with uncorrelated errors) or a known reference. Running a relevant mission or lab test and analyzing the result in NavLab will usually reveal errors above the specification and often also the characteristics of such errors.

Improving sensor calibration (using real data)

Even if a sensor is approved in an evaluation, it can exhibit systematic errors, typically due to imperfect calibration or misaligned mounting. Such (deterministic) errors should be removed before sending the measurements to the estimator, otherwise the performance will be reduced (in particular for the real-time Kalman filter). To find these systematic errors, the smoothing algorithm should be used, as it is significantly better than the real-time filter at estimating such errors. When systematic errors are known, they can be compensated for in future missions.

4.1 Using NavLab for real data post-processing

For vehicles storing their navigation sensor measurements during missions, it is possible to make post-processed estimates of position, attitude, velocity and sensor errors. There are many situations where these estimates are of great interest after the mission is finished, for instance if the vehicle has recorded payload data that require accurate geo-referencing (e.g. bathymetric data for terrain maps or image data for object detection). NavLab is well suited and extensively used to produce optimal post-processed navigation results. These results are valuable also when the vehicle has calculated and stored real-time navigation estimates. When the time constraints allow, post-processed estimates are preferred to the real-time estimation results, since important properties such as estimation accuracy, robustness and integrity are improved:

- *Increased accuracy* is mainly due to the use of the optimal smoothing (see Section 3.2). In addition, real-time issues like delayed measurements and incomplete data sets from remote sensors[1] are eliminated. Finally, the absence of a real-time computing requirement makes it possible to use iterations to improve estimation performance.
- *Improved robustness* is partly due to the smoothing algorithm, which in general is more robust against degraded sensor performance than the real-time Kalman filter (see Figure 6 and Figure 8). In addition, the possibility of rerunning the estimation increases the ability to recover a faulty data set. To do so, one can modify either the degraded sensor measurements or the filter tuning (or both) to get the best possible navigation for the faulty data set.
- The *Integrity* of the estimator, i.e. the ability to detect degraded sensor performance and degraded total navigation performance, is critical for the users of the navigation data. The optimal smoothing algorithm has a very high capability of detecting reduced sensor quality. In addition it can often tell which sensor is having problems. When deviations are detected, the data can usually be rerun as described above, and the final estimates will be reliable (i.e. more accurate and associated with a trustworthy accuracy estimate). In practice, the ability to recover the navigation data in the case of degraded sensor performance means that the need for a new mission is avoided.

Also, the smoothing might allow purchasing less expensive sensors or using them less frequently, and still obtaining the required accuracy. For instance, a submerged vehicle might need to surface to get position measurements. In Figure 5, we see that with a position accuracy requirement of 5 meters, the real-time filter would require position measurements after a period of 2500 seconds, while with smoothing a position accuracy better than 5 meters is obtained even with a 2 hours dropout interval.

Post-processing of real data has become one of the most important NavLab applications, and through mass-production of accurate navigation results more than 5000 hours of recorded payload data has been positioned. Any vehicle with recorded sensor data can be navigated, and currently AUVs, ROVs, ships and aircraft have been navigated with NavLab.

4.2 Practical usage

NavLab is written in the mathematical programming language Matlab [9], but it can also be compiled to a Windows application (exe file). Post-processing of a recorded data set with 3-5 Hz Kalman filter update rate and 100 Hz IMU data, is approximately 15 times faster than real-time, when using a 3 GHz Pentium 4 processor.

The user interface can vary from “Scientific”, where all parameters and steps are fully controllable, to “One-click” [10] where all processes are automated. In Scientific mode, a general multi-menu based plot function is used after a simulation or estimation. This function plots a range of figures containing numerical summaries and many different 2D and 3D plots with a total of more than 500 graphs, for results analysis. The plot function is also programmable to show only a predefined subset of plots for users wanting just a simplified summary of the results. The very simplest output is used in the One-click mode, where a green/red light at the end of the estimation indicates if the data was OK or not.

[1] For instance a surface vehicle measuring the AUV position by means of DGPS+USBL. A full set of measurements is not transmitted to the AUV in real time, but is available for use in NavLab after the mission.

5 Verification of estimator performance

Verification of the estimator performance has been a crucial part of the NavLab development. Both the Kalman filter and the optimal smoothing calculate an expected uncertainty for their estimates, which is the theoretically optimal accuracy obtainable for the given scenario. When using a correct model in the estimator, the actual estimation error should be as small as the theoretical uncertainty limit. A correct model can be used when the measurements are from the simulator, but since the real world has infinite complexity, it is impossible to use a completely correct model in the estimator when using real data. In cases where the model used by the estimator differs from the model generating the measurements, the actual estimation error will be larger than the theoretical limit. The most challenging part of the estimator development is to keep its error as close as possible to the theoretical limit in cases of modeling errors (and nonlinearities). To minimize the loss of accuracy, a very careful design and implementation of all parts of the estimator is vital. In this section it is demonstrated that it is possible to achieve a performance close to the optimal under a range of different non-ideal conditions.

5.1 Verifying performance using the simulator

The simulator, having a more complex nonlinear system model than the estimator, is an effective tool for verifying the estimator performance. Any scenario can be tested and different modeling errors can be used. After running the estimator, the plot function will calculate and plot the true estimation error and compare it with the theoretical estimation uncertainty (also Monte Carlo simulations can be run to determine the statistics of the error). Thorough and extensive testing of the estimator since 1999 by different research groups, testing a variety of scenarios, has proven the estimator to be very robust and to give close to optimal performance in all scenarios.

5.2 Verifying performance using real data

The ultimate test of the estimator is to use real data from a representative mission, where the trajectory and all sensor errors are (by definition) totally realistic. The challenge with real runs is that it is more difficult to investigate the estimation errors, since the true trajectory is unknown. However, some possibilities do exist, and these are discussed in the following.

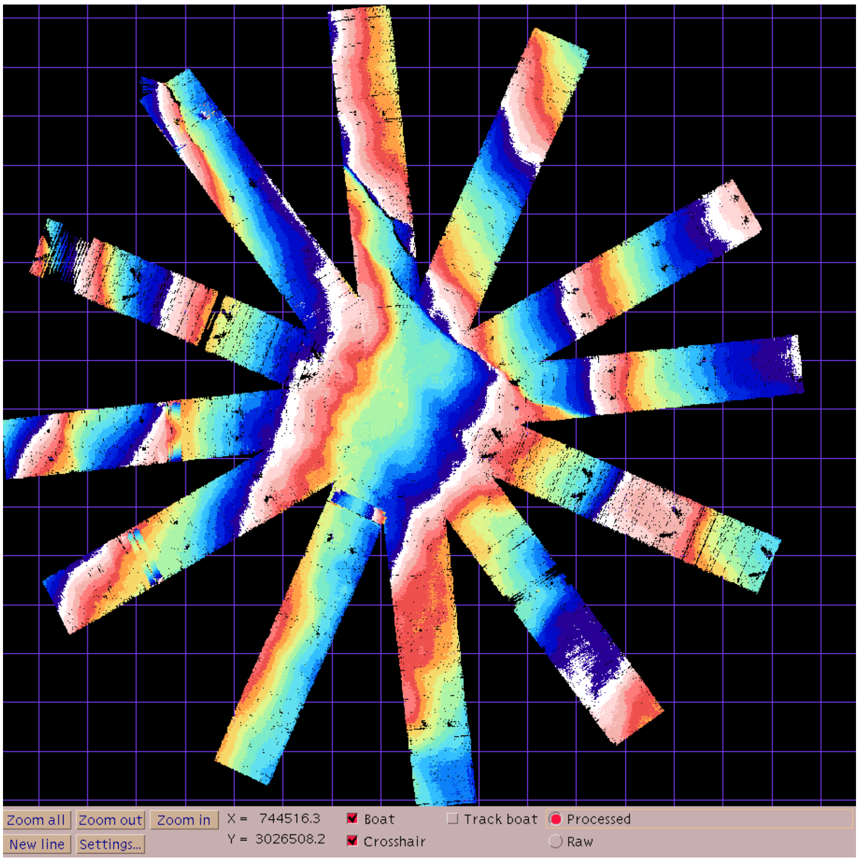
5.2.1 Redundant sensors

A significant sensor measurement can be made unavailable for the navigation system, and later be used as a reference. For instance, a surface ship might follow a submerged AUV, continually measuring its position using DGPS+USBL, but not sending the measurements to the AUV. The AUV, typically using an IMU, a depth sensor, a DVL and in some cases a compass, will have a drift in position that after a while will be significantly larger than the uncertainty in the DGPS+USBL position measurements. Hence the estimation error is observable and is compared with the theoretical uncertainty. All such tests have documented a very high estimator performance, that was in accordance with the theoretical uncertainty, see [11] and [5].

5.2.2 Verifying the positioning by means of mapped objects

For a seabed mapping vehicle, an accurate positioning of the final map is essential, and estimates of the vehicle's 6 degrees of freedom (position and attitude) are used to position the bathymetric data. Estimation errors in vehicle position will be directly translated to errors in the map position, while the effect of attitude errors will depend on the geometry between the vehicle and a given patch of the seafloor. A crucial test of the entire navigation system is to verify the position accuracy in the final maps. In such tests, all available aiding sensors are used so that the maximum accuracy is evaluated.

Customers buying HUGIN and NavLab for detailed seabed mapping have had a strong focus on position accuracy of the maps and have thus run navigation performance trials as part of the customer acceptance tests. These trials determine if the real-life performance of the estimator match the accuracy that was predicted in NavLab simulations before the vehicle was built. The standard method is to map the same object at the seafloor several times, comparing the position estimate of each individual object observation. Errors that are uncorrelated between each passing will be visible, as the object will be positioned differently in each observation. Correlated errors are typically following the AUV or a ship giving DGPS+USBL measurements (e.g. timing problems, systematic velocity error and misaligned acoustic positioning transducer). Hence, to also reveal these errors, different headings are used for the AUV and ship for each passing ("wagon wheel pattern", as shown in Figure 7). Figure 7 shows maps from HUGIN 3000 in an accuracy test carried out by the HUGIN customer C&C Technologies at 1300 m water depth in the Gulf of Mexico in October 2000. 11 different headings were used (5 of the lines were mapped in opposite directions) when mapping the object (a wellhead), to maximize the visibility of any correlated errors following the AUV or ship. The positions of the wellhead observations when using NavLab smoothing are shown in Figure 8, obtaining an accuracy of 1.2 m and 1.7 m (1σ) north and east (even with a large unmodeled DVL error present, see Section 3.2.1). The theoretical estimation uncertainty in the smoothed position was about 1.7 m (1σ, north and east) during the passings. 60 m from the wellhead, but within the swath width, another object (natural feature) was also visible in the data. Since the object is 60 m off the center of the maps, a somewhat higher uncertainty is expected due to the AUV heading uncertainty (and also due to the increased mapping sonar uncertainty), and indeed this object had a distribution of 1.3 m north and 1.9 m east.



X = 744516.3
Y = 3026508.2
☒ Boat
☒ Crosshair
□ Track boat
● Processed
○ Raw

Figure 7. A wellhead is mapped repeatedly with different headings to evaluate the positioning accuracy of the final map

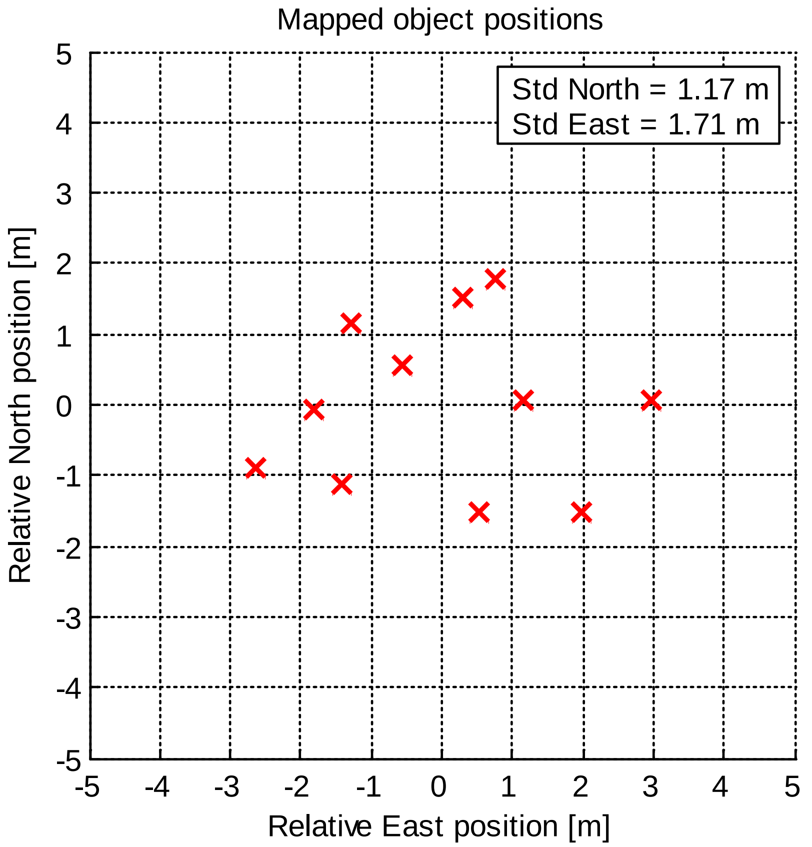


Figure 8. The mapped positions of the wellhead (at 1300 m depth) using NavLab smoothing

The test shown is the only test where a large unmodeled sensor error was present. After this test many similar navigation accuracy evaluations have been carried out by different HUGIN customers, with other vehicles and navigation sensors. The accuracy has been tested down to a depth of 2200 m (obtaining 2.3 m and 3.3 m accuracy in north and east), and in general the tests have proven exceptionally good estimator accuracy, even slightly better than the anticipated theoretical uncertainty limit. The reason for this is a combination of the navigation sensors performing somewhat better than their specifications and the estimator producing close to optimal estimates.

6 Conclusions

NavLab is a powerful and versatile tool with usage ranging from research and development by scientists and academics, to mass production of high-accuracy maps by commercial companies (having post-processed more than 5000 hours of data from around the world).

Even when a real-time navigation system is available, it is often beneficial to post-process the data with NavLab:

- The navigation results, i.e. estimates of position, attitude and velocity, will be more accurate and smooth (no jumps in the data).
- The navigation results will be more reliable (any critical sensor errors are detected).

- Even in cases of sensor degradation or failure, accurate navigation can often be obtained (no need for a new mission). This is due to the increased robustness of smoothing and the possibility to rerun the data.
- Lower quality navigation sensors might be used, while still obtaining satisfactory navigational accuracy.

The most significant feature of NavLab is its theoretical foundation, where statistical optimality is maintained throughout the entire system. This has been repeatedly demonstrated through extensive performance verifications, both with simulations and real missions. These tests have proven very high estimator performance, close to the theoretical optimum.

7 Acknowledgements

The author wishes to thank the rest of the Navigation Group at FFI, Bjorn Jalving in particular, for significant contributions to the development of NavLab. Also thanks to Bjorn Jalving, Kongsberg Maritime and HUGIN customers for organizing, carrying out and reporting a range of different tests of the navigation system accuracy.

8 References

[1] Gade, K (1997): Integrering av treghetsnavigasjon i en autonom undervannsfarkost (in Norwegian), FFI/RAPPORT-97/03179

[2] Gade, K (2003): NavLab - Overview and User Guide November 2003, FFI/RAPPORT-2003/02128

[3] Svartveit, K and Berglund, E (2003): NavLab Plugin: Waypoint editor, FFI (to be published)

[4] Minkler, G and Minkler, J (1993): Theory and Application of Kalman Filtering, Magellan Book Company

[5] Jalving, B, Gade, K, Hagen, O K and Vestgard, K (2003): A Toolbox of Aiding Techniques for the HUGIN AUV Integrated Inertial Navigation System. Proceedings from Oceans 2003, September 22-26, San Diego, USA

[6] Jalving, B, Bovio, E and Gade, K (2003): Integrated inertial navigation systems for AUVs for REA applications. Proceedings from MREP 2003, NATO SACLANT Undersea Research Centre, May 2003, Italy

[7] Hagen, P E, Størkersen, N and Vestgård, K (2003): The HUGIN AUVs - multi-role capability for challenging underwater survey operations. EEZ International, Summer 2003

[8] The HUGIN AUV Programme homepage: www.ffi.no/hugin

[9] Mathworks homepage: www.mathworks.com

[10] Svartveit, K (2004): NavLab One-Click, FFI (to be published)

[11] Jalving, B, Gade, K, Svartveit, K, Willumsen, A and Sorhagen, R (2004): DVL Velocity Aiding in the HUGIN 1000 Integrated Inertial Navigation System. Proceedings from ADCPs in Action 2004, June 3-4, Nice, France

Paper V

Jalving, B., Gade, K., Hagen, O. K. and Vestgård, K. (2004). A Toolbox of Aiding Techniques for the HUGIN AUV Integrated Inertial Navigation System. *Modeling, Identification and Control*, vol. 25, no. 3, pp. 173-190

MODELING, IDENTIFICATION AND CONTROL, 2004, VOL. 25, No. 3, 173–190
doi: 10.4173/mic.2004.3.3

A Toolbox of Aiding Techniques for the HUGIN AUV Integrated Inertial Navigation System

Bjørn Jalving, FFI   Kenneth Gade, FFI   Ove Kent Hagen, FFI
FFI, Norwegian Defence Research Establishment
P. O. Box 25, 2027 Kjeller, Norway
Bjorn.Jalving@ffi.no

Karstein Vestgård, Kongsberg Simrad
Kongsberg Simrad
P. O. Box 111, 3191 Horten, Norway
Karstein.Vestgard@kongsberg-simrad.com

Keywords: *Autonomous underwater vehicle, aided inertial navigation, Kalman filter, Doppler velocity log, synthetic aperture sonar, GPS, USBL, transponder navigation, terrain navigation, smoothing*

Modern AUV designs must handle submerged autonomous operation for long periods of time. The state of the art solution embedded in the HUGIN AUVs is a Doppler Velocity Log (DVL) aided Inertial Navigation System (INS) that can integrate various forms of position measurement updates. In autonomous operations, position updates are only available in limited periods of time or space, thus the core velocity aided inertial navigation system must exhibit high accuracy. However, position uncertainty of a DVL aided inertial navigation system will eventually drift off, compromising either mission operation or requirements for accurate positioning of payload data. To meet the requirements for a range of military and civilian AUV applications, the HUGIN vehicles come with a flexible and powerful set of navigation techniques. Methods for position updates include GPS surface fix, DGPS-USBL, Underwater Transponder Positioning (UTP) and bathymetric terrain navigation. Based on synthetic aperture sonar technology, a potentially revolutionary accurate velocity measurement is under development. HUGIN also comes with a navigation post-processing system (NavLab), which can be applied to increase navigational integrity and maximize position accuracy.

1 INTRODUCTION

Autonomous Underwater Vehicles (AUVs) have in recent years convincingly demonstrated their capabilities in real applications. Civilian applications include detailed seabed mapping, environmental monitoring and research and inspection work for offshore industry. Short time frame military applications include Mine Counter Measures (MCM) and Rapid Environmental Assessment (REA). In a longer time frame, AUVs will play an important part in the general robotization of modern warfare.

Kongsberg Simrad and FFI have cooperated in developing the HUGIN family of autonomous underwater vehicles. HUGIN 3000 was the world’s first AUV used in commercial survey operations, [1], [2]. The four HUGIN AUVs currently in service have been used in areas as diverse as the Gulf of Mexico, the Mediterranean, Brazil, West Africa, the North Sea and the Norwegian Sea. Building on more than 5 years of field experience with commercial AUV use, the HUGIN 1000 vehicle is now under development (first delivery end 2003), targeting the military market and civilian environmental monitoring and research.

Compared to HUGIN 3000, HUGIN 1000 is smaller, easier to handle, has lower depth rating and shorter endurance, but software, electronics and system design are almost identical, [3]. This paper discusses the design of the integrated inertial navigation system for the HUGIN family and the development of a toolbox of navigation techniques to meet the requirements for a range of AUV applications.

2 HUGIN INTEGRATED INERTIAL NAVIGATION SYSTEM

2.1 Integrated Inertial Navigation System Structure

In Fig. 1, the structure of the HUGIN integrated inertial navigation system is shown. The Inertial Navigation System (INS) calculates position, velocity and attitude using high frequency data from an Inertial Measurement Unit (IMU). An IMU consists of three accelerometers measuring specific force and three gyros measuring angular rate. A Kalman filter will, in a mathematically optimal manner, utilize a wide variety of navigation sensors for aiding the INS. The Kalman filter is based on an error-state model and provides a much higher total navigation performance than is obtained from the independent navigation sensors.

2.2 DVL aided INS - Core Navigation System

2.2.1 DVL Aided INS

Autonomous operation in deep water or covert military operations requires the AUV to handle submerged operation for long periods of time. The solution for modern AUVs is a low drift Doppler Velocity Log (DVL) aided inertial navigation system that can integrate various forms of position measurement updates. In Fig. 1, the core DVL aided INS system consists of the IMU and the navigation equations, the error state Kalman filter and DVL, compass (optional) and pressure aiding sensors.

Inertial navigation systems are usually classified by the standard deviation of the positional error growth of their free inertial (unaided) performance (see Table 1). A free inertial INS will, after a short period of time, have unacceptable position errors. The HUGIN navigation system can in principle interface any IMU, but for most applications the IMU will be in the 1 nmi/h class.

DVL accuracy is dependent on frequency. Higher frequency yields better accuracy at the sacrifice of decreased range as illustrated in Table 2. Prioritization between range and accuracy is dependent on the application.

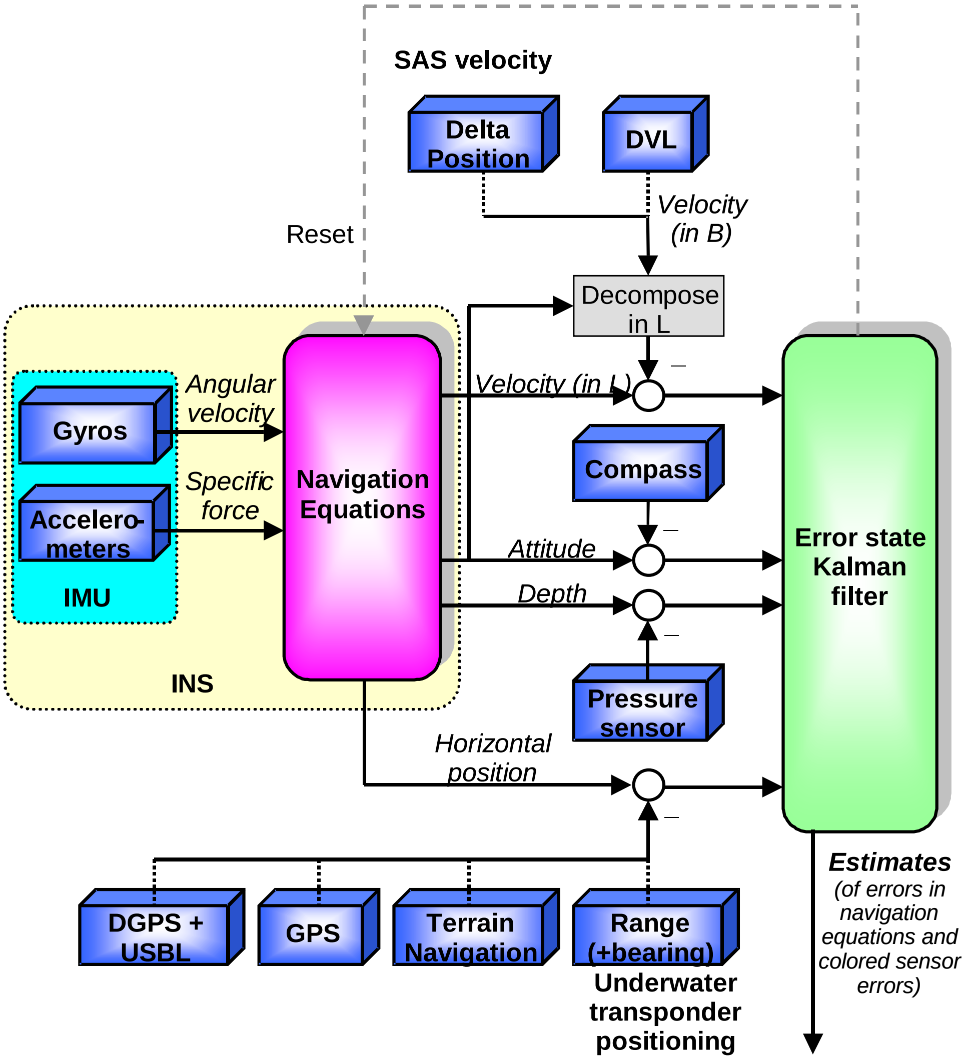


Fig. 1. HUGIN integrated inertial navigation system structure

Table 1. INS classes. Notes 1: RLG – Ring Laser Gyro, FOG – Fiber Optic Gyro

Class	Gyro technology	Gyro bias	Accelerometer bias
>10 nmi/h	RLG, FOG[1]	1°/h	1 milli g
1 nmi/h	RLG, FOG	0.005°/h	30 micro g

Table 2. RDI Workhorse Navigator Doppler Velocity Log accuracy and range specifications, [4]. o.s. – of speed.

Frequency	Long term accuracy	Range
150 kHz	±0.5% o.s. ± 2 mm/s	425 – 500 m
300 kHz	±0.4% o.s. ± 2 mm/s	200 m
600 kHz	±0.2% o.s. ± 1 mm/s	90 m
1200 kHz	±0.2% o.s. ± 1 mm/s	30 m

2.2.2 Simplified Error Analysis Straight Trajectories

The simplified error analysis presented in this section is useful for understanding the basic mechanisms of a DVL aided INS and assessing how IMU and DVL sensor accuracy is determining the overall position accuracy.

The horizontal position drift in a DVL-aided INS is determined by the error in the estimated Earth-fixed velocity (i.e. North and East velocity). The main contributors to this error are:

- Error in the body-fixed velocity
- Error in heading.

The error in estimated body fixed velocity, is mainly determined by the low-frequency error in the DVL itself (without position aiding this error is not observable when going at a straight line). High frequency velocity errors are estimated by means of the accelerometers. Even the most accurate INS will without aiding after a short period of time have a velocity uncertainty larger than the DVL accuracy. Referring to Table 2, a 300 kHz DVL typically have a scale factor type of error of 0.4% of speed, contributing to an along track error drift of 0.4% of traveled distance, or 28.8 m/hour for an AUV traveling at 2 m/s (4 knots). However, there are ways to improve the DVL accuracy. Sacrificing range, the 1200 kHz version from the same vendor has an accuracy specification of 0.2% of speed, corresponding to 0.2% of traveled distance, or 14.4 m/hour (AUV speed 2 m/s). The scale factor error is observable by the Kalman filter when position measurements are available or when the AUV is turning. Thus, the Kalman filter can compensate for part of the scale factor error when running more complex missions than a straight line. This is discussed in Section 2.2.3.

The error in heading is determined by the gyrocompassing capability of the integrated system. The heading estimation error will typically be of low frequency, corresponding to non-observable gyro bias dynamics. Referring to Table 1, a 1 nmi/h navigation class IMU typically gyrocompass to an accuracy of $\sigma(\delta\psi)$ = 0.02 deg sec latitude. This corresponds to an error drift of $\sigma(\delta\psi)\cdot 100$ % of traveled distance ($\sigma(\delta\psi)$ in radians). At 45° latitude, this equals 0.05% of traveled distance, or 3.4 m/hour at 2 m/s AUV speed.

In [5] position accuracy for an INS with 1 nmi/h IMU and 1200 kHz DVL following a straight line was simulated. Along track position error drift was in the order of 8 m/hour while cross track position error drift was in the order of 2.5 m/hour. This is a somewhat smaller drift than predicted by the simplified error analysis. There are two main reasons; the Kalman filter compensates for a scale factor error estimated when position measurements were available and the actual scale factor error is modeled as a first order Markov process and not a constant error. Choosing time constants that realistically reflect the physical error process is very important when estimating DVL aided INS error drift and when tuning the Kalman filter for real applications.

Since 1 nmi/h navigation class IMUs are relatively easily obtainable in the marketplace and the DVL induced position error is close to an order of magnitude larger than the IMU induced position error for straight line trajectories, most focus should actually be on how to improve the velocity accuracy. This explains the importance of the work presented in Section 2.3.

2.2.3 Countering DVL Aided INS Position Error Growth

For a submerged AUV without position updates, the position error growth of a DVL aided INS can be countered by:

1. Mission pattern for canceling of error growth
2. Kalman filter estimation and compensation of DVL error

The accuracy estimates in Section 2.2.2 are valid for straight-line trajectories. Since the main error contributors of DVL aided INS is body fixed velocity and heading, a canceling effect of the error growth is obtained when for instance running a lawn mower pattern. The canceling effect increases with the stability of the body fixed velocity error and heading error. Also the canceling effect increases with shorter line lengths.

A second important effect of maneuvering is that the velocity error actually becomes observable by means of comparing expected centripetal acceleration with measured acceleration from the IMU. If the velocity error is the same during the maneuver (i.e. when it is observed) as it is in the following line, this estimation will significantly reduce the drift. However real DVL-data from RDI Workhorse Navigator 300 kHz shows that during the maneuver the error might be different, and in such cases this effect will have limited importance for the overall position drift. This real data problem can be countered by a sophisticated compensation method, but preferably, other sensors or frequencies might not exhibit this error characteristic. When the mechanism works, the error growth when running long straight lines can be significantly reduced by adding 360° turns at regular intervals.

The two effects combined are very effective, as seen in Table 3, which contains results from NavLab simulations (see Section 2.9 for NavLab description).

Table 3. Typical reduction in position error drift for a DVL aided INS when comparing a straight-line trajectory with a lawn mower pattern, [5]. The numbers apply for a 1200 kHz DVL and a 1 nmi/h IMU at 45° latitude.

Position error drift (% of traveled distance)	**Straight line**	**Lawn mower pattern with 1 km lines**
Along track	0.11%	0.01%
Across track	0.03%	0.001%

2.3 SAS Velocity Aiding

In Section 2.2.2 it was shown that for an AUV equipped with a 1 nmi/h type of IMU or better, the DVL accuracy is the limiting factor to the position accuracy during submerged navigation with no position updates.

Modern MCM and REA AUVs are likely to be equipped with Synthetic Aperture Sonar (SAS) due to the improved resolution and image quality offered compared to Side Scan Sonar (SSS). SAS requires very good relative navigation to obtain focused images. Relative navigation in SAS over a synthetic aperture is often referred to as micro-navigation. One method of micro-navigation, called Displaced Phase Center Array (DPCA), generates, as a by-product, a revolutionary good velocity (or more precisely, *displacement*) measurement. This complex displacement measurement needs to be integrated in the Kalman filter in a non-traditional way, which is an on-going research effort.

The DPCA velocity measurement technique, based on expensive and sophisticated sonar hardware and advanced signal processing, is in fact very similar to the technique used in a Correlation Velocity Log (CVL). If expectations are proved true and the DPCA velocity measurement is an order of magnitude more accurate than DVL, along track error contribution will be in the same order as across track error contribution. Consequently a leap in performance of velocity aided inertial navigation systems has been achieved, allowing longer time intervals between position updates.

2.4 GPS Surface Fix

As seen in Fig. 1, there are several alternatives for providing the integrated inertial navigation systems with position updates. GPS surface fix is the most intuitive method and should be applied when possible. The following GPS services can be used:

- GPS Standard Positioning Service (SPS)
- GPS Precise Positioning Service (PPS)
- Differential GPS (DGPS)
- Real-Time Kinematic GPS (RTKGPS)

GPS SPS is available to all users worldwide. GPS PPS is available only to authorized users and primarily intended for military purposes. GPS PPS receivers should be the choice for military AUVs, at least for operations in denied areas. Compared to GPS SPS, GPS PPS is more resistant to jamming and deception. GPS SPS and GPS PPS have comparable accuracy. AUVs for detailed seabed mapping will typically be equipped with DGPS, or in some cases even RTKGPS.

2.5 Combined DGPS-USBL (Ultra Short Base Line)

In deep water seabed mapping, deploying and following the AUV with a survey vessel is the preferred method for obtaining maximum position accuracy. The survey vessel is equipped with differential GPS and tracks the AUV with an USBL system. Combined DGPS-USBL position measurements are transmitted to the AUV at regular intervals to bind the position error drift. See Section 4.1 for operational results.

2.6 LBL (Long Base Line)

LBL systems provide accurate AUV position measurements once a network of four LBL transponders has been deployed and calibrated. In principle, the HUGIN inertial navigation system can easily be integrated with a LBL system. However, the operational efforts involved in deployment and calibration is drastically reduced with underwater transponder positioning (Section 2.7), where only one underwater transponder is necessary to bind the INS position drift. LBL systems in AUV applications will probably become obsolete with the advent of this new navigation technique.

2.7 Underwater Transponder Positioning (UTP)

2.7.1 Old Principle – Revolutionary Solution

Pinging a transponder on the seafloor and measuring range and bearing is the traditional approach to acoustic navigation. From range and bearing measurements, position has been computed in commercial Ultra Short Base Line (USBL) and Short Base Line (SBL) systems for decades. Instead of integrating a complex USBL system in an AUV, the AUV can be fitted with two transducers separated by as long baseline as possible (this is basically a SBL system).

This principle is called Underwater Transponder Positioning (UTP) and is the result of a joint development effort by FFI and Kongsberg Simrad. Kongsberg Simrad has delivered UTP to the American survey company C&C Technologies on a commercial basis. The range and bearing measurements are tightly integrated as position measurements in the Kalman filter of the inertial navigation system (actually position measurements can be produced with only range measurements available as well). The system works with only one underwater transponder, but can utilize any number of transponders in an optimal way. Compared to a traditional LBL system, UTP has improved accuracy due to tight coupling with the INS, increased operating area and significantly less deployment costs, since only one transponder is necessary to bind the position drift.

2.7.2 Concurrent Deployment and Navigation (CDN)

Current version of UTP requires that a survey vessel equipped with USBL box in the position of the underwater transponders. The transponder position coordinates must be sent to HUGIN prior to UTP navigation. In the next version, the HUGIN navigation system will be able to estimate the position of an underwater transponder while navigating with another. In this way, the AUV will be able to deploy a trail of underwater acoustic buoys for UTP navigation and acoustic communication. This concept can be denoted Concurrent Deployment and Navigation (CDN) or UTP CDN.

2.8 Bathymetric Terrain Navigation

2.8.1 Correlation Methods

Terrain correlation may be done for one measurement, or on a sequence of measurements. The measured water depths are shifted around an offset area around current position estimate, and a correlation between the measurements and the depth data in the Digital Terrain Model (DTM) is calculated in this area. The calculated correlation is called the correlation surface. The correlation surface is analysed to determine convergence, calculating a position offset, the error covariance and a position fix confidence.

Terrain correlation runs on any sensor providing bathymetric data, for instance multibeam echosounder (MBE), altimeter, DVL or interferometric sonar. Terrain navigation accuracy depends on sensor accuracy, map accuracy, map resolution and not least terrain suitability.

In Fig. 2 the HUGIN terrain correlation system is illustrated. The *Geographic data producer* converts AUV depth + bathymetric sensor data in AUV body-fixed coordinates to geographical referenced data, using the current navigation solution. The *Terrain Correlator* runs the terrain correlation algorithm on one measurement or iteratively on a sequence of measurements. *Map Database* readies the DTMs for random access by the Terrain Correlator. Position updates are sent to the integrated inertial navigation system Kalman filter to bind the INS position drift.

The actual correlation can be done selecting different algorithms:

- *Terrain Contour Matching (TERCOM), [6]*

  A well-proven and robust algorithm that uses the mean absolute distance as a correlation measurement. Models for sensor and map noise may be included. The covariance matrix of the position fix is found through the correlation surface.

- *Point Mass Filter (PMF), [7]*

  A more sophisticated algorithm that actually calculates the position Probability Density Function (PDF) using Bayesian estimation. PMF enables the use of advanced models for sensor and map noise and enables a statistically sound use of the navigation system accuracy as an input. The covariance matrix of the position fix is found directly from the PDF.

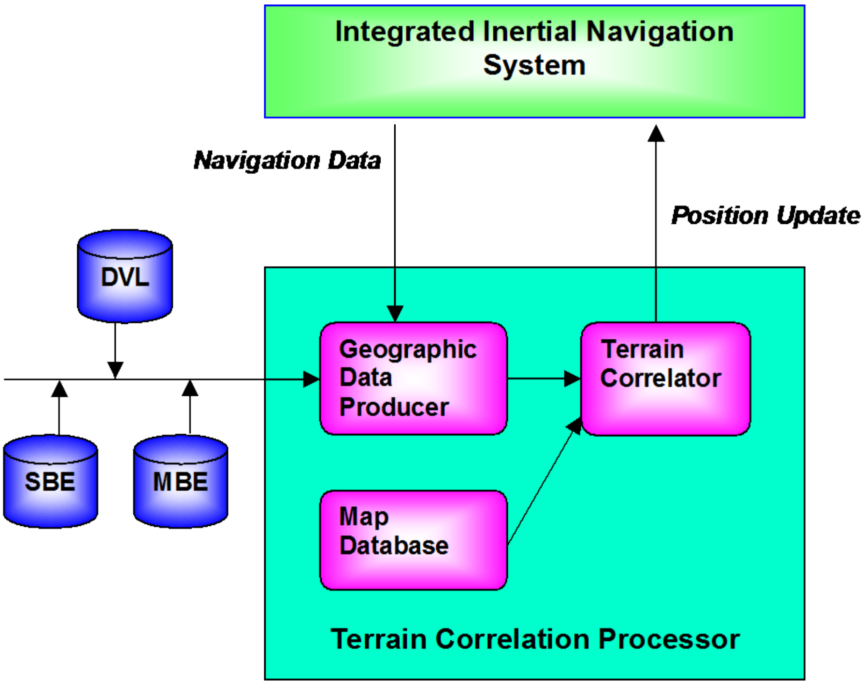


Fig. 2. Structure of the HUGIN terrain correlation system

2.8.2 Tightly Integrated Terrain-Tracking Algorithms

Terrain navigation algorithms can conceptually be divided into correlation based global search algorithms (described in Section 2.8.1) and tightly integrated terrain tracking algorithms. The latter are characterized by integration of range measurements and the bathymetric map into the Kalman filter. Thus, all available information in the integrated navigation system is utilized. Compared to correlation methods, the algorithms have less robust behavior in highly non-linear terrain. FFI has invested a considerable effort in developing a terrain-tracking algorithm called TRIN [8]. This is planned for integration in HUGIN, following the completion of the work on correlation-based methods.

2.8.3 Concurrent Mapping and Navigation (CMN)

An attractive feature of tightly integrated terrain-tracking algorithms is that a solution for Concurrent Mapping and Navigation (CMN) follows inherently.

Similar to UTP CDN, CMN is important to missions in unknown or denied areas. Solutions to CMN, considering both tightly integrated terrain-tracking algorithms and correlation algorithms, is an ongoing research effort.

2.9 NavLab (Navigation Laboratory)

NavLab (Navigation Laboratory), [9], [10], is a powerful and versatile tool intended for:

- Navigation system research and development
- Navigation system accuracy analysis
- Navigation data post-processing

NavLab consists of a Simulator and an Estimator, see Fig. 3. The Simulator can simulate any vehicle trajectory and a selected set of sensor measurements. The Estimator will, based on the available measurements, produce filtered and smoothed optimal estimates of position, velocity, attitude and sensor errors.

Prior to implementation in the HUGIN real-time navigation system, NavLab is used for algorithm research and development, simulation and testing. NavLab is also used for navigation system accuracy analysis and mission planning (even by HUGIN customers).

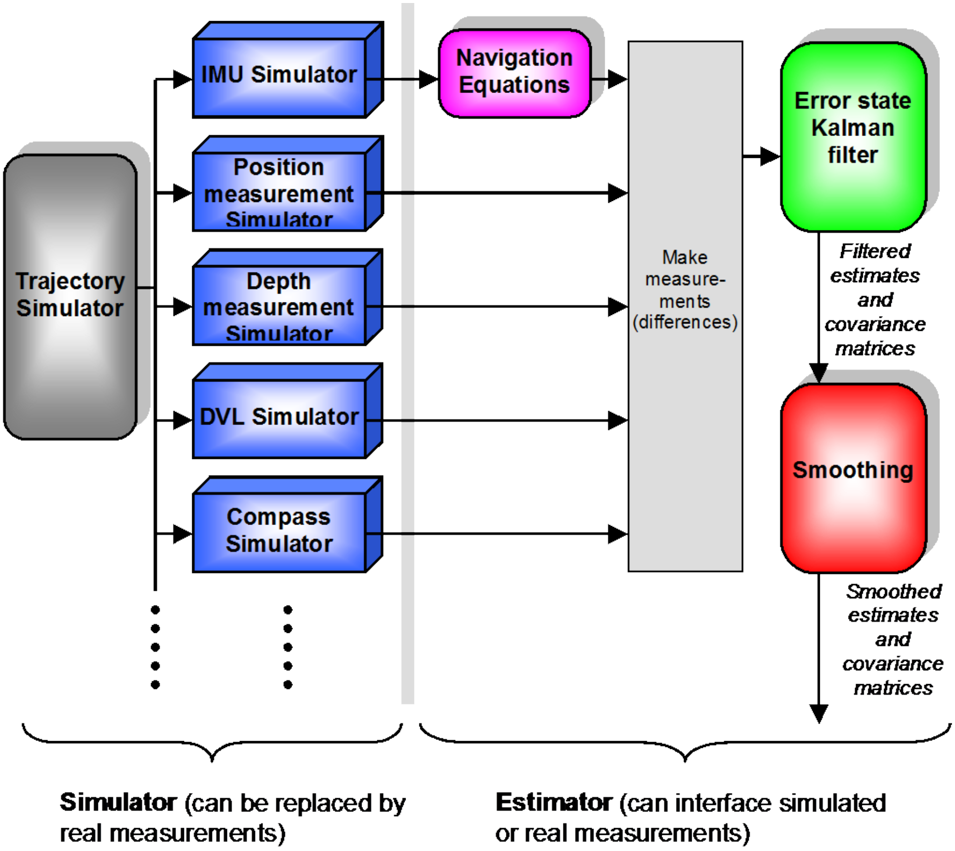

IMU Simulator
Position measurement Simulator
Depth measurement Simulator
DVL Simulator
Compass Simulator
Trajectory Simulator
Navigation Equations
Make measure-ments (differences)
Error state Kalman filter
Filtered estimates and covariance matrices
Smoothing
Smoothed estimates and covariance matrices
Simulator (can be replaced by real measurements)
Estimator (can interface simulated or real measurements)

Fig. 3. NavLab structure

Since the Estimator works equally well with simulated and real measurements, NavLab is well suited and extensively used to produce optimal post-processed navigation results from HUGIN missions. When time and cost constraints allow, post-processed results are preferred to the real-time estimation results, since both the estimation accuracy and the integrity are improved. The increased accuracy is due to the use of smoothing, which is an optimal estimation technique that utilizes both past and future measurements. Smoothing is especially effective when position updates are scarce, which is the case with GPS surface fixes, terrain navigation with few reference areas and scattered underwater transponders. In Fig. 4 the effect of navigation post-processing when running a 15 km straight-line trajectory with GPS fix at the end is shown. The effect is less, but still substantial when running a lawn mower pattern, [5].

The HUGIN real-time integrated inertial navigation system comes with extensive systems for integrity check. This is of crucial importance to safeguard against jamming, multipath effects, internal sensor failures etc. However, if the integrity mechanisms should fail to detect a navigation sensor wild point or degraded sensor performance, the real-time navigation estimates can be seriously affected. An important feature of navigation post-processing is increased navigational integrity and increased ability to recover faulty data sets. The smoothing algorithm is in general more robust against degraded sensor performance than the real-time Kalman filter and degraded sensor data sets can be filtered and improved.

NavLab has been extensively used by numerous international research groups and commercial mapping companies since 1999.

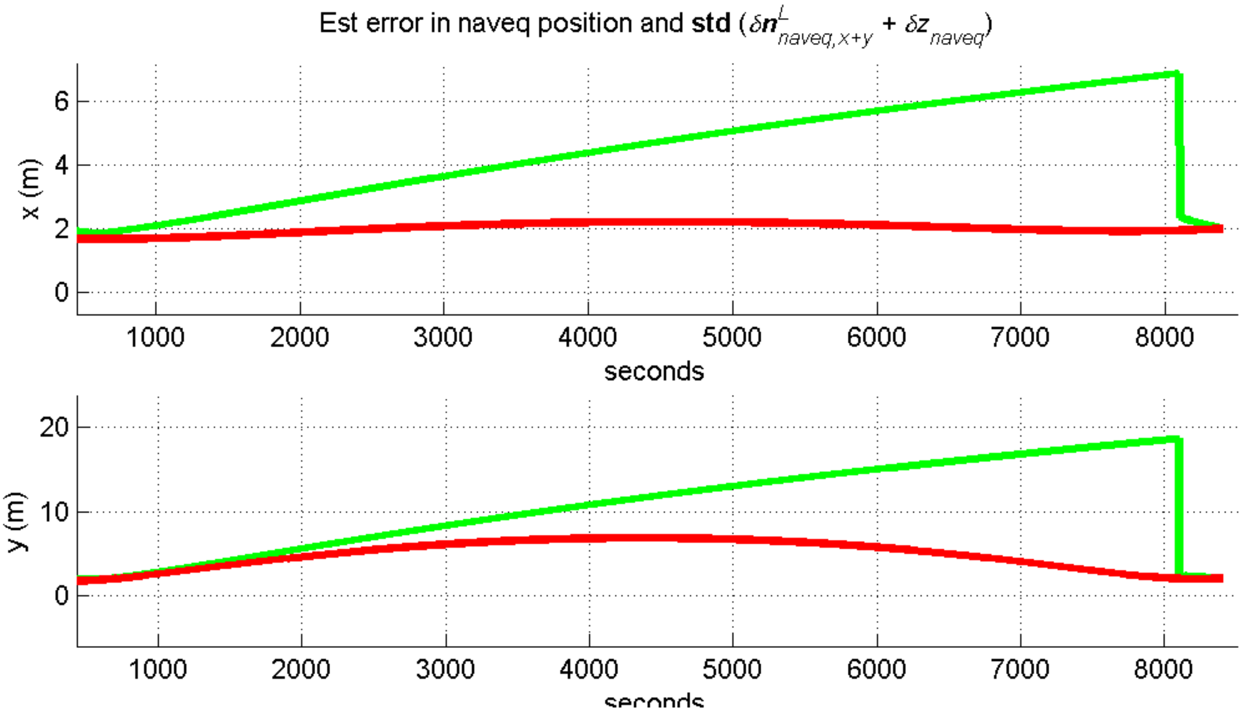


Fig. 4. The effect of navigation post-processing when running a straight trajectory with GPS fix every 15 km. Green graph: real-time position accuracy (1σ). Red graph: post-processed position accuracy (1σ). x and y in local level (L) corresponds roughly to North and East direction.

3 USE OF NAVIGATION TOOLBOX IN DIFFERENT APPLICATIONS

An AUV operator will tailor the use of the integrated inertial navigation system to his specific needs and requirements. However, to illustrate the versatility of the HUGIN toolbox of navigation techniques, Table 4 suggests typical use of the navigation system in different applications.

Table 4. Typical use of the HUGIN navigation system in different AUV applications.

Application	Navigation System Use
Detailed seabed mapping	DVL aided INS DGPS-USBL position aiding NavLab post-processing
Environmental monitoring and research	DVL aided INS GPS surface fix
Inspection work for offshore industry	DVL aided INS UTP
MCM home areas - shallow water	DVL aided INS GPS surface fix
MCM home areas- deep water	DVL aided INS UTP Terrain navigation
REA - low visibility	DVL aided INS GPS surface fix NavLab post-processing
REA – covert	DVL aided INS UTP CDN Terrain navigation with CMN NavLab post-processing
MCM denied areas (REA)	Same as REA – covert

GPS surface fix is the obvious, easy and accurate method for position update when water depth and covertness requirements allow. In deep water the actual AUV traveling time makes GPS fixes undesirable. Furthermore, loss of DVL bottom track will reduce the effect of the position fix due to the INS drift when diving.

Not mentioned in Table 4, UTP in concert with DGPS-USBL can potentially increase position accuracy for detailed seabed mapping in deep waters.

For MCM in home areas, accurate digital terrain models will be available for terrain navigation. In home areas, underwater transponders can also be pre-deployed in strategic locations.

Navigation strategies for REA operations are thoroughly analyzed in [5]. Covert REA operations typically involves advanced concepts such as UTP CDN (Section 2.7.2) and CMN (Section 2.8.3).

4 OPERATIONAL RESULTS

4.1 Detailed Seabed Mapping for Offshore Industry

In detailed seabed mapping for the offshore industry DGPS-USBL position updates is the preferred method to obtain maximum position accuracy. In [11] position accuracies in the final digital terrain models in water depths down to 3000 m have been thoroughly analyzed.

HUGIN 3000 position accuracy was verified in commercial mapping operations in varying water depths in the Gulf of Mexico in March 2001. The method used was mapping a known object, typically a wellhead, multiple times with reciprocal lines in different directions (”wagon wheel” pattern) and observe the position variance of the wellhead observations in the final DTMs. Applying NavLab post-processing a position accuracy of 2 m (1σ) in 1300 m water depth and 4 m (1σ) in 2100 m water depth was demonstrated. See Fig. 5 for results in 1300 m water depth.

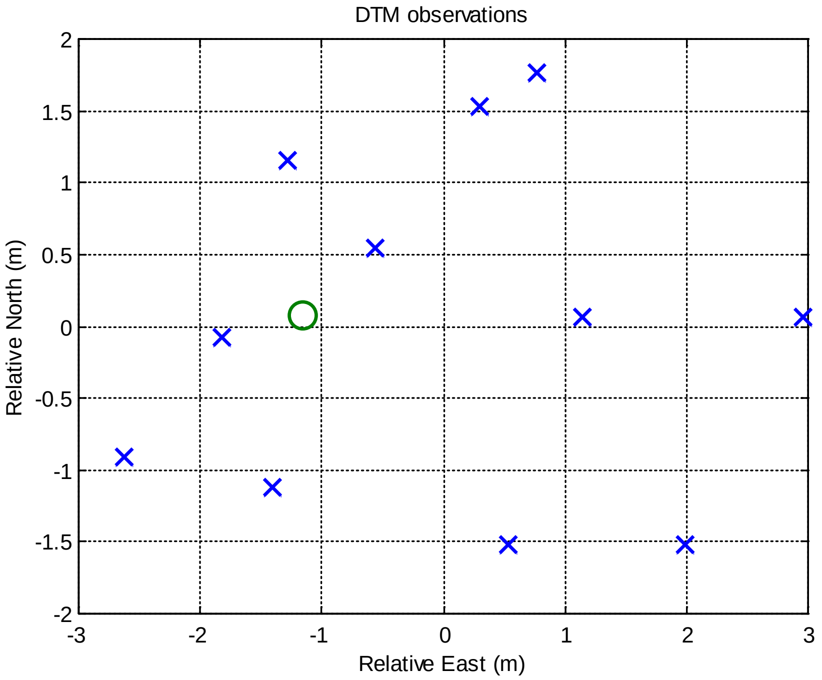


Fig. 5. HUGIN 3000 position accuracy results. Blue crosses: Position estimates of different DTM wellhead observations in 1300 m water depth. Standard deviation in North is 1.2 m, in East 1.7 m.

4.2 REA Mission with Norwegian Navy

In August 2002, the Royal Norwegian Navy completed upgrade of a permanent HUGIN infrastructure on its KNM Karmøy mine countermeasures vessel. KNM Karmøy and HUGIN I are regularly used in operations of actual military worth [12]. In Fig. 6, the HUGIN I trajectory from a mission with KNM Karmøy in May 2003 is shown. HUGIN I was running an autonomous REA type of mission navigating with DVL aided INS and GPS surface fixes at regular intervals. HUGIN I was equipped with a 1 nmi/h type IMU (Table 1), a 300 kHz DVL (Table 2) and a GPS SPS receiver. In Fig. 7 difference between GPS and HUGIN INS is shown. When getting position fixes, the HUGIN INS position converges towards the GPS position. Considering the accuracy of a 300 kHz DVL and GPS SPS (not differential), the results are very good. Navigation accuracy in-between the GPS fixes can be further improved with NavLab post-processing, as explained in Section 2.9.

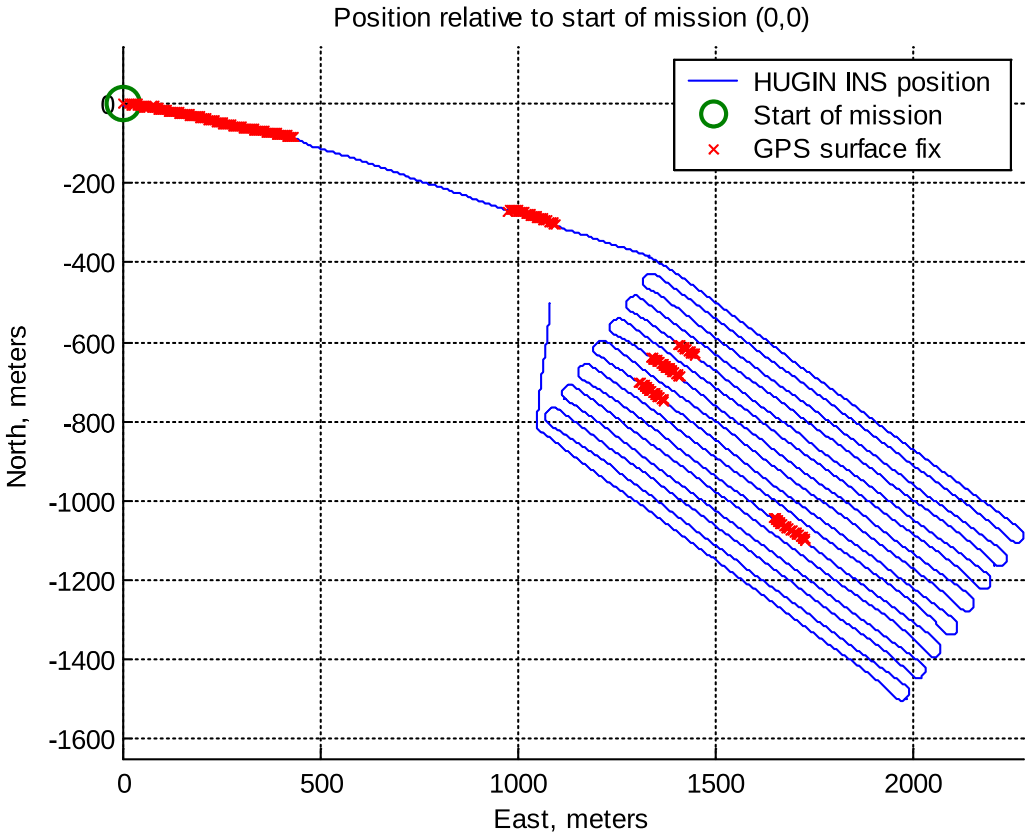


Fig. 6. HUGIN I trajectory in autonomous mission from KNM Karmøy May 2003.

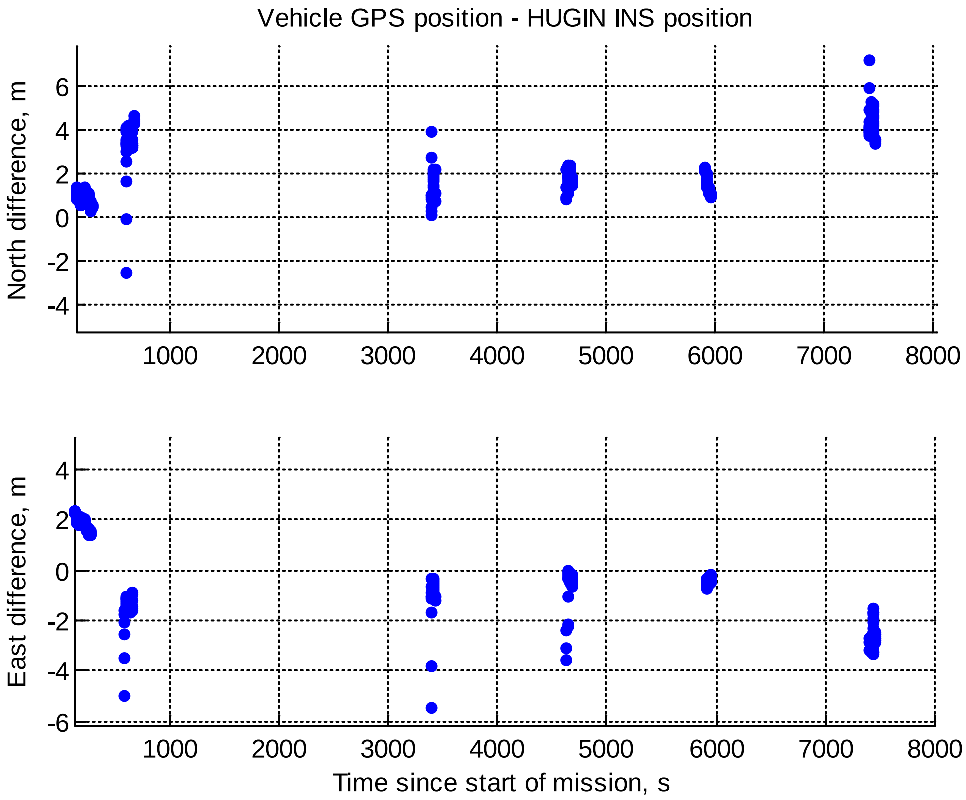
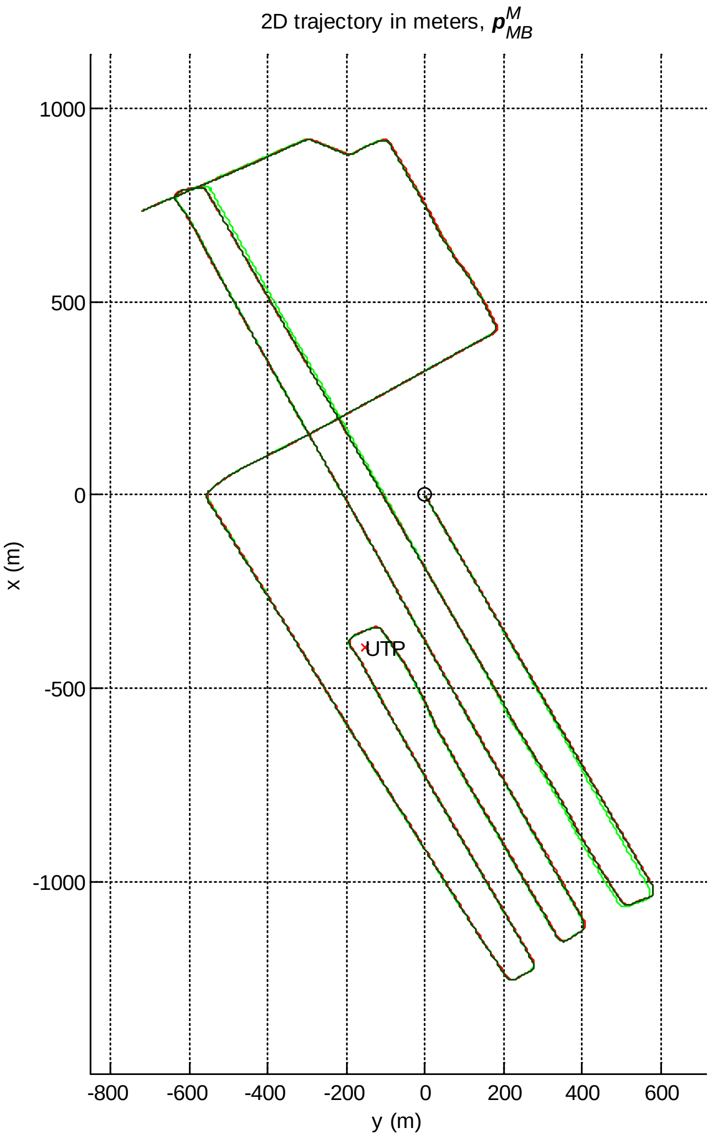


Fig. 7. Difference between vehicle GPS and HUGIN integrated navigation system position estimate. When getting position fixes, the HUGIN INS position converges towards the GPS position.

4.3 Underwater Transponder Positioning

A number of UTP sea trials were performed outside Horten, Norway, March 2003, with very good results. In Fig. 8 the HUGIN trajectory and a picture of the deployed underwater transponder is shown. The relatively large size of the transponder is mainly due to a large battery pack and buoyancy material needed for retrieving the transponder.

HUGIN navigated at 180 m water depth with UTP as the only source for position updates. Post-mission, the navigation data was compared to independent DGPS-USBL data stored on the survey vessel. The average difference between the two data sets in North and East was 2.2 m and 2.6 m (1σ, RMS). When NavLab post-processing (smoothing) was applied, the difference reduced to 1.2 m in North and 1.5 m in East (1σ, RMS). This is very close to the accuracy of the DGPS-USBL system. Fig. 9 shows the difference between DGPS-USBL position estimate and the UTP post-processed navigation solution.



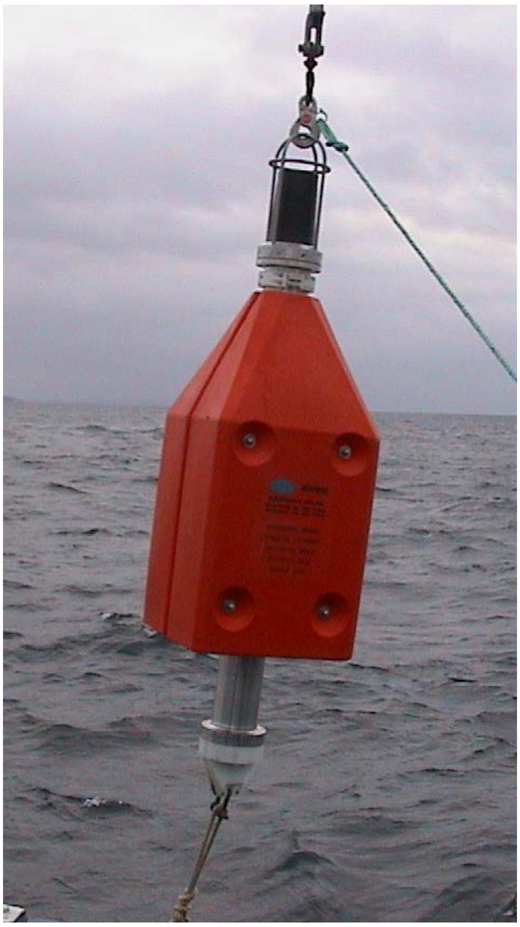


Fig. 8. Left: HUGIN 2D trajectory in UTP sea trial. UTP was deployed at x = -396 m, y =-151 m (relative coordinates).
Right: Underwater transponder used in sea trial.

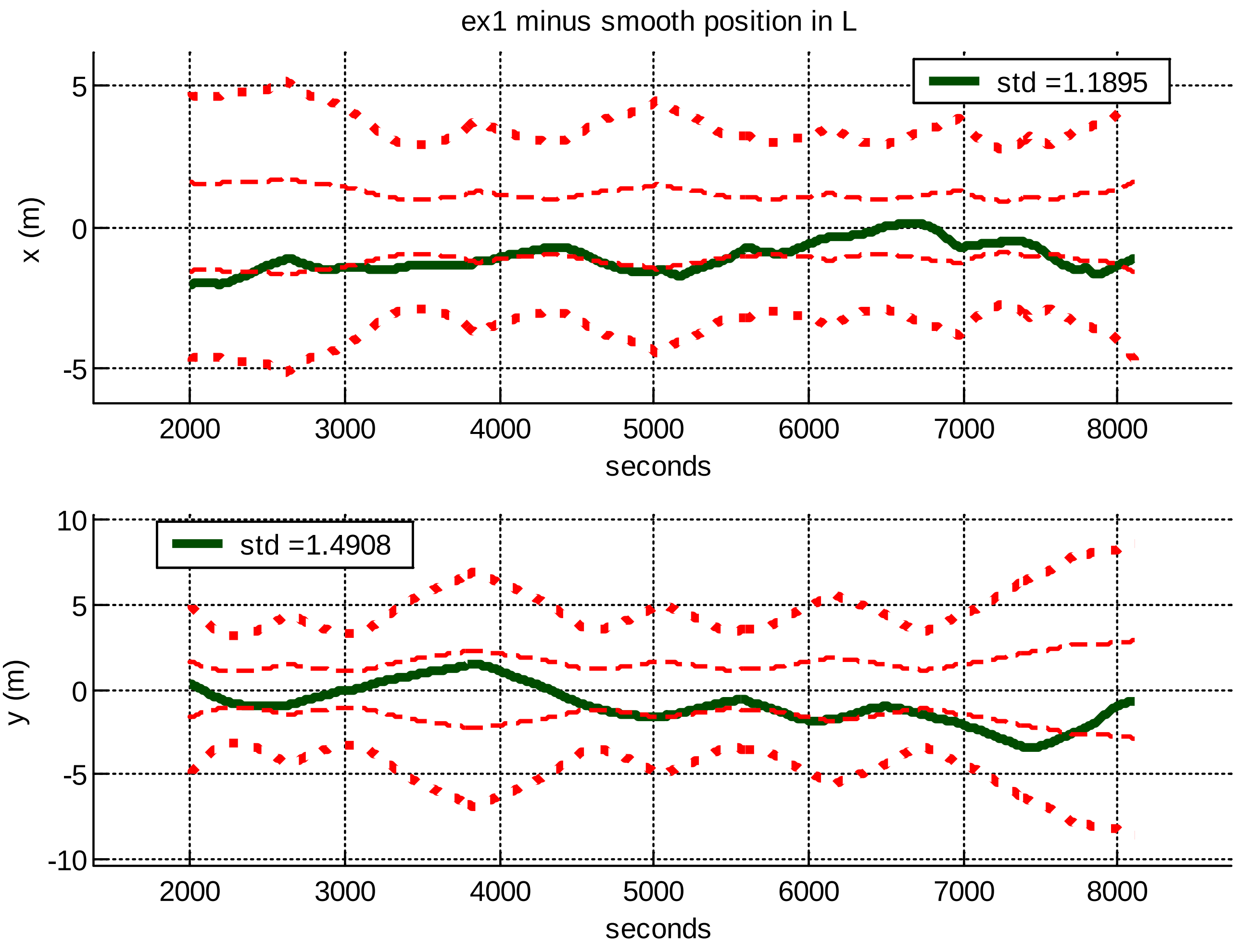


Fig. 9. Results from UTP sea trial. Green graph: difference between UTP post-processed navigation solution and independent DGPS-USBL position estimates. Red thin and red bold graphs are ±1σ and ±3σ estimated uncertainty of the difference.

4.4 Terrain Navigation

The HUGIN terrain correlation system described in Section 2.8.1, is currently tested on recorded data from HUGIN missions conducted in a test area outside Horten in the Oslo fjord. The test area was surveyed by FFI's research vessel HU Sverdrup II in January 2001. A high quality DTM of 10 m resolution was produced. This DTM is statistically independent of the bathymetric data collected by HUGIN I, which is very important with respect to realistic testing of terrain navigation algorithms.

A data player plays the recorded real-time navigation solution and MBE and DVL bathymetric data. Except for the data player, the system is identical to the real-time version, which is due for the first sea trials in August 2003.

Fig. 10 shows the contour lines of the inverse of the resulting correlation surface of the TERCOM algorithm for a position fix. Each fix is rated by a confidence value 0 (low) to 1 (high). This value indicates stability of the fix and the presence of possible multiple solutions. For each fix an estimate of position standard deviation in northern and southern direction, along with the position covariance, are calculated using the correlation surface.

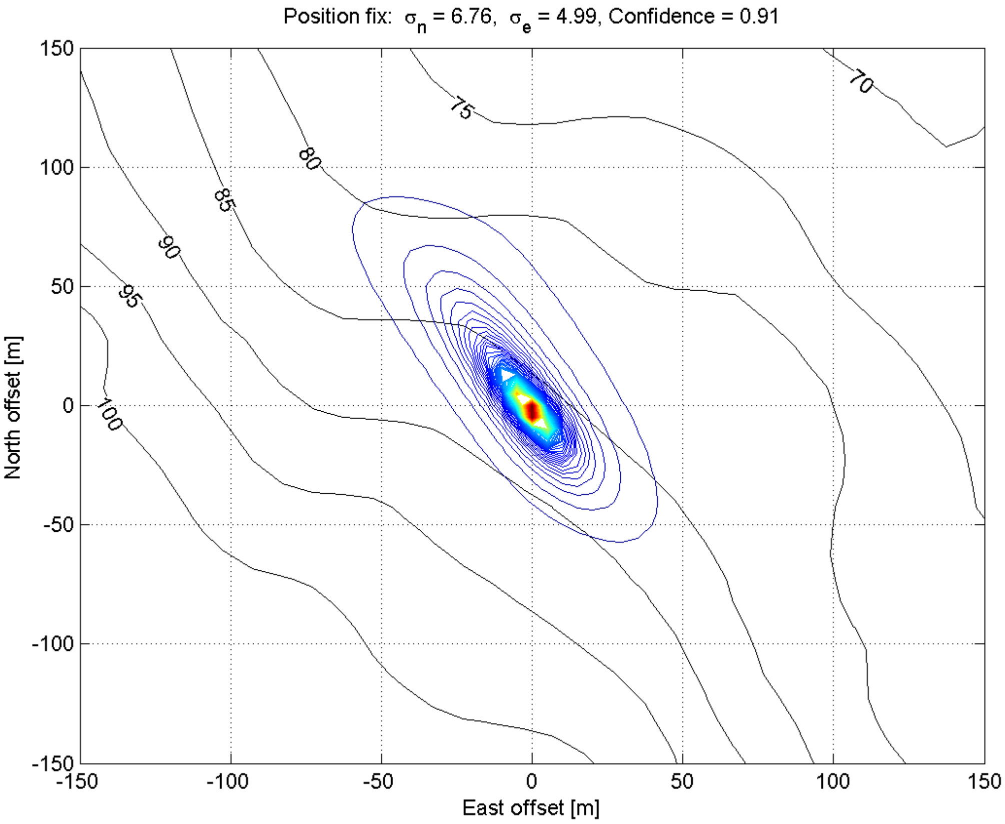


Fig. 10. The correlation surface contour lines overlaid the DTM contour lines for a 300m x 300m area. HUGIN I's position estimate (considered true position) is in the origin of this grid. Notice that the uncertainty of the fix is greater in the direction along the contour lines than across, indicating the importance of the position fix covariance.

5 SUMMARY

The main purpose of this paper has been to present the HUGIN integrated inertial navigation system and the extensive toolbox of navigation techniques, which has been designed to meet the navigation requirements for a broad variety of civilian and military AUV applications.

HUGIN Navigation Toolbox
DVL aided INS ▪ Mission pattern for cancelling of error growth ▪ Estimation and compensation of DVL error
SAS velocity aiding
GPS surface fix
DGPS-USBL
Underwater transponder positioning (UTP) ▪ Navigation with pre-deployed transponders ▪ Concurrent deployment and navigation (CDN)
Terrain navigation ▪ Navigation with known DTM ▪ Concurrent mapping and navigation (CMN)
NavLab ▪ Navigation post-processing ▪ Navigation system simulation and accuracy analysis

The core navigation system consists of a low drift velocity aided inertial navigation system based on a 1 nmi/h class IMU and an accurate DVL. There are several ways to counter the position error growth of a DVL aided INS. Cancelling of error growth with a lawn mower pattern is a very useful technique (Section 2.2.3).

If development work succeeds, a velocity measurement based on SAS technology can provide a leap in performance of velocity aided inertial navigation, allowing longer time intervals between position updates.

GPS surface fixes is the obvious, easy and accurate method for position updates when moderate water depths and covertness requirements allow.

For detailed seabed mapping operations in deep water, DGPS-USBL is the preferred method for obtaining maximum position accuracy. HUGIN 3000 demonstrated in March 2001 2 m (1σ) position accuracy in 1300 m water depth and 4 m (1σ) position accuracy in 2100 m water depth (with NavLab post-processing). To our knowledge, this accuracy has not yet been matched by any other survey AUV.

Underwater transponder positioning and terrain navigation allow for submerged position updates in autonomous missions. With only one transponder necessary for operation, UTP provides larger operational area and reduced deployment cost compared to LBL. UTP has in sea trials demonstrated very good accuracy (Section 4.3). Next development step is to facilitate concurrent deployment and navigation (UTP CDN, see Section 2.7.2).

Bathymetric terrain navigation is an appealing method for submerged position updates since bathymetric data from a standard AUV sensor suite is utilized: DVL, MBE, altimeter or interferometric sonar. In many scenarios, a digital terrain model will be available and actually used in mission planning. Next development step involves techniques for concurrent mapping and navigation (CMN).

Navigation post-processing maximizes the position accuracy and provides increased integrity check to a collected data set, features of crucial importance for deepwater detailed seabed mapping. Post-processing is especially effective when position fixes are scarce, making it very attractive for covert REA applications. The NavLab Simulator can be used for navigation system accuracy analysis and can thus been an important tool in mission planning.

With the exception of SAS velocity aiding, UTP CDN and CMN, all the navigation techniques described in this paper is working commercially available technology (bathymetric terrain navigation is being tested in sea trials at time of writing). Furthermore, the HUGIN navigation system has in real applications onboard civilian survey vessels and on a navy mine countermeasures vessel demonstrated very good performance.

REFERENCES

[1] T. C. Chance, A. A. Kleiner and J. G. Northcutt, “The HUGIN 3000 AUV,” *Sea Technology,* vol. 41, no 12, December 2000, pp. 10-14.

[2] K. Vestgård, R. Hansen, B. Jalving and O. A. Pedersen, “The HUGIN 3000 Survey AUV”, ISOPE-2001, Stavanger, Norway, June 2001.

[3] P. E. Hagen, N. Størkersen, K. Vestgård and P. Kartvedt: "The HUGIN 1000 Autonomous Underwater Vehicle for Military Applications", Proceedings from Oceans 2003, San Diego, CA, USA, September 2003.

[4] RD Instruments, “Workhorse Navigator Doppler Velocity Log (DVL) ”, http://www.dvlnav.com/pdfs/navbro.pdf, June 2003.

[5] B. Jalving, E. Bovio and K. Gade, “Integrated inertial navigation systems for AUVs for REA applications”, SACLANTCEN conference proceedings from MREP 2003, NATO SACLANT Undersea Research Centre, May 2003.

[6] J. P. Golden, “Terrain contour matching (TERCOM) : a cruise missile guidance aid”, Image Processing for Missile Guidance, 238, 1980, pp. 10-18

[7] N. Bergman, L. Jung and F. Gustafsson, “Terrain navigation using Bayesian statistics”, IEEE Control Systems Magazine, 19(3), 1999, pp. 33-40

[8] O. K. Hagen and P. E. Hagen, “Terrain referenced integrated navigation systems for underwater vehicles”, SACLANTCEN conference proceedings CP-46, NATO SACLANT Undersea Research Centre, August 2000.

[9] K. Gade, ”NavLab User Guide”, FFI/Report 2003/02128, Norwegian Defence Research Establishment, 2003.

[10] K. Gade, “NavLab, a Generic Simulation and Post-processing Tool for Navigation”, paper to be published in 2004/2005.

[11] B. Jalving, K. Vestgård, N. Størkersen, “Detailed seabed surveys with AUVs” in *Technology and Applications of Autonomous Underwater Vehicles*. Edited by Gwyn Griffiths. Taylor & Francis, London and New York, 2003, pp. 179-201.

[12] P. E. Hagen, N. Størkersen, K. Vestgård, P. Kartvedt and G. Sten, ”Operational military use of the HUGIN AUV in Norway”, Proceedings from UDT Europe 2003, Malmö, Sweden, June 2003.

Paper VI

Jalving, B., Gade, K., Svartveit, K., Willumsen, A. and Sørhagen, R. (2004). DVL Velocity Aiding in the HUGIN 1000 Integrated Inertial Navigation System. *Modeling, Identification and Control*, vol. 25, no. 4, pp. 223-236

MODELING, IDENTIFICATION AND CONTROL, 2004, VOL. 25, No. 4, 223–235
doi: 10.4173/mic.2004.4.2

DVL Velocity Aiding in the HUGIN 1000 Integrated Inertial Navigation System

Bjørn Jalving, Kenneth Gade, Kristian Svartveit
Norwegian Defence Research Establishment (FFI)
P O Box 25, NO-2027 Kjeller, Norway
bjorn.jalving@ffi.no, kenneth.gade@ffi.no, kristian.svartveit@ffi.no

Are Willumsen
UniK - University Graduate Center
P O Box 70, NO-2027 Kjeller, Norway
are.b.willumsen@unik.no

Robert Sørhagen
Kongsberg Maritime
P O Box 111, NO-3191 Horten, Norway
robert.sorhagen@kongsberg.com

Keywords: *Autonomous underwater vehicle, aided inertial navigation, Kalman filter, Inertial measurement unit, Doppler velocity log*

Abstract

The RDI WHN-600 Doppler Velocity Log (DVL) is a key navigation sensor for the HUGIN 1000 Autonomous Underwater Vehicle (AUV). HUGIN 1000 is designed for autonomous submerged operation for long periods of time. This is facilitated by a low drift velocity aided Inertial Navigation System (INS). Major factors determining the position error growth are the IMU and DVL error characteristics and the mission plan pattern. For instance, low frequency DVL errors cause an approximately linear drift in a straight-line trajectory, while these errors tend to be cancelled out by a lawn mower pattern. The paper focuses on the accuracy offered by the DVL. HUGIN 1000 is a permanent organic mine countermeasure (MCM) capacity on the Royal Norwegian Navy MCM vessel KNM Karmøy. HUGIN 1000 will be part of the NATO force MCMFORNORTH in fall 2004.

1 Introduction

Kongsberg Maritime and FFI have cooperated in developing the HUGIN family of autonomous underwater vehicles. HUGIN 3000 was the world’s first AUV used in commercial survey operations, [1], [2], [3]. The three HUGIN 3000 class AUVs have been used in areas as diverse as the Gulf of Mexico, the Mediterranean, Brazil, West Africa, the North Sea and the Norwegian Sea. Building on more than 5 years of field experience with commercial AUV use, the HUGIN 1000 vehicle was developed, targeting the military market and civilian research, mapping and monitoring applications. Compared to HUGIN 3000, HUGIN 1000 is smaller, easier to handle, has lower depth rating and shorter endurance, but software, electronics and system design are almost identical, [4]. The first HUGIN 1000 was delivered to the Royal Norwegian Navy in

January 2004. The vehicle is permanently installed on the KNM Karmøy mine countermeasures (MCM) vessel. Organic AUV MCM operations as a concept is continuously developed, refined and evaluated while HUGIN 1000 is contributing to military worth in military exercises and operations. In fall 2004 KNM Karmøy and HUGIN 1000 will be part of the standing NATO force MCMFORNORTH.

Autonomous operation in deep water or covert military operations requires the AUV to handle submerged operation for long periods of time. The state of the art solution embedded in the HUGIN AUVs is a Doppler Velocity Log (DVL) aided Inertial Navigation System (INS) that can integrate various forms of position measurement updates. The HUGIN vehicles are using the RDI WHN-300 and WHN-600 DVLs. This paper presents the HUGIN integrated inertial navigation system in general and discusses DVL velocity aiding in particular. The data examples are taken from the new HUGIN 1000, still undergoing operational evaluation.

Figure 1. Left: HUGIN 1000 on aft deck of KNM Karmøy. Right: KNM Karmøy mine countermeasures vessel.

2 HUGIN 1000 Navigation System Philosophy

2.1 Integrated Inertial Navigation System Structure

In Figure 2 the structure of the HUGIN integrated inertial navigation system and a summary of the HUGIN navigation toolbox is shown. The Inertial Navigation System (INS) calculates position, velocity and attitude using high frequency data from an Inertial Measurement Unit (IMU). An IMU consists of three accelerometers measuring specific force and three gyros measuring angular rate, relative to inertial space. An extended Kalman filter will, in a mathematically optimal manner, utilize a wide variety of navigation sensors for aiding the INS. The Kalman filter is based on an error-state model and provides a much higher total navigation performance than would be obtained from the independent navigation sensors.

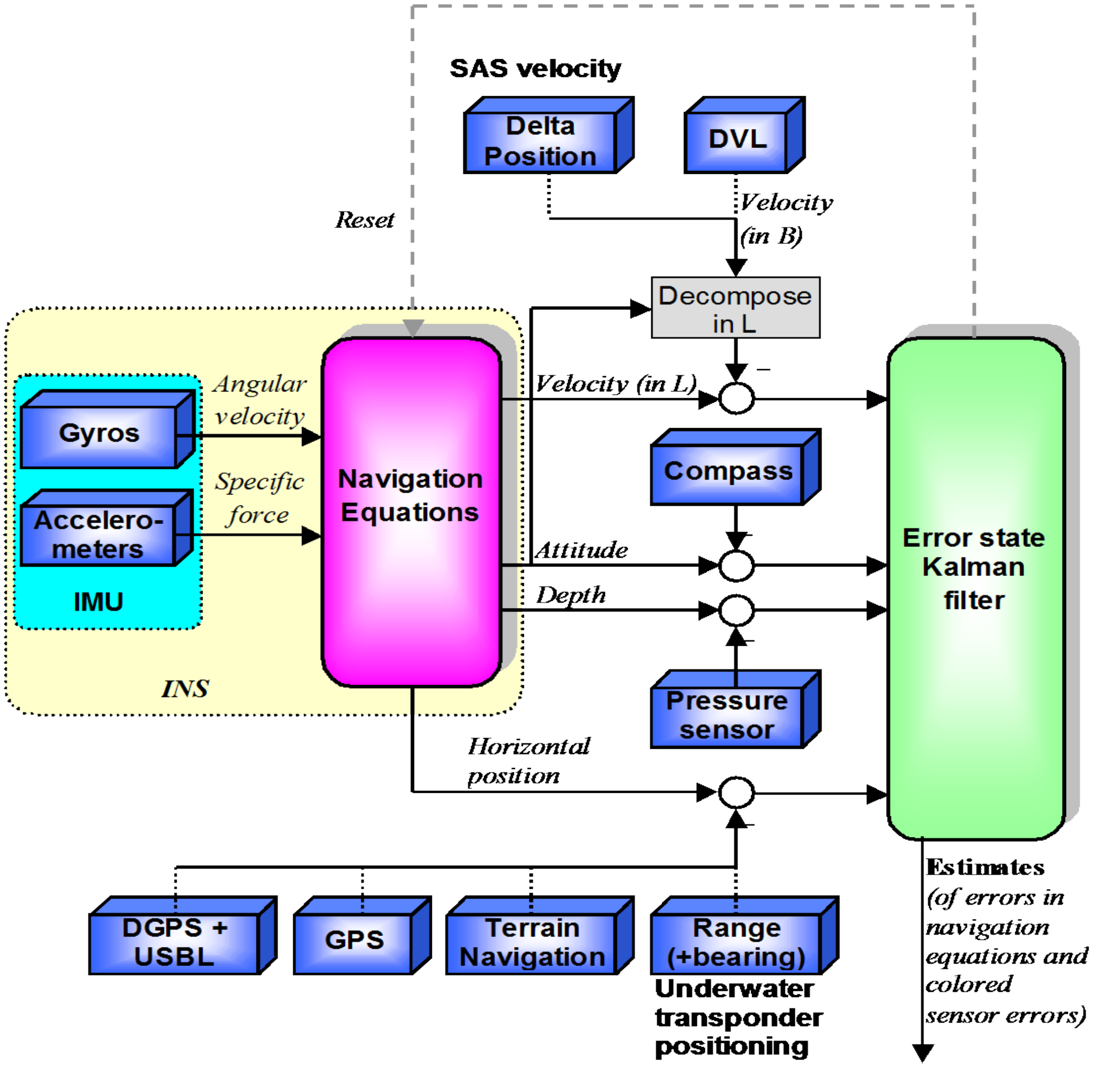
A core low drift DVL aided inertial navigation system is capable of handling submerged autonomous operation for long periods of time. In Figure 2, the core DVL aided INS system consists of the IMU and the navigation equations, the error state Kalman filter and the following aiding sensors: DVL, compass (optional) and pressure sensor.

Depending on position accuracy requirements, the navigation system must get occasional position measurement updates. GPS surface fixes is the preferred method for position updates when moderate water depths, mission efficiency and covertness requirements allow. For submerged position updates, the HUGIN vehicles come with bathymetric terrain navigation and acoustic ranging to underwater transponders (UTP). If HUGIN is in acoustic vicinity of its mother ship, DPGS-USBL (ultra short baseline) aiding can be used. NavLab post-processing allow for maximum position accuracy in demanding applications, [5]. SAS velocity aiding is a potentially rewarding method, however, not yet demonstrated on real AUV data. More information on the toolbox and use of the toolbox in different applications can be found in [6].

Inertial navigation systems are usually classified by the standard deviation of the positional error growth of their free inertial (unaided) performance (see Table 1). A free unaided INS will, after a short period of time, have unacceptable position errors. The HUGIN navigation system can interface any IMU, but for most applications the IMU will be in the navigation grade (1 nmi/h) class.

Table 1. INS classes. RLG – Ring Laser Gyro, FOG – Fiber Optic Gyro

Class	**Gyro technology**	**Gyro bias**	**Accelerometer bias**
>10 nmi/h	RLG, FOG	1°/h	1 milli g
1 nmi/h	RLG, FOG	0.005°/h	30 micro g



HUGIN Navigation Toolbox
DVL aided INS ▪ Mission pattern for cancelling of error growth ▪ Estimation and compensation of DVL error
Synthetic aperture sonar (SAS) velocity aiding
GPS surface fix
DGPS-USBL
Underwater transponder positioning (UTP) ▪ Navigation with pre-deployed transponders ▪ Concurrent deployment and navigation (CDN)
Terrain navigation ▪ Navigation with known DTM ▪ Concurrent mapping and navigation (CMN)
NavLab ▪ Navigation post-processing ▪ Navigation system simulation and accuracy analysis

Figure 2. Top: HUGIN integrated inertial navigation system structure. Bottom: HUGIN navigation toolbox summary. Refer to [6].

3 DVL Velocity Aiding

3.1 DVL Alternatives

DVL accuracy is dependent on frequency. Higher frequency yields better accuracy at the sacrifice of decreased range as illustrated in Table 2. Prioritization between range and accuracy is dependent on the application. RDI WHN-300 is standard outfit for the HUGIN 3000 AUVs that operate in waters down to 3000 m and is typically followed by a mother ship equipped with DGPS and USBL for continuous position updates. HUGIN 1000 on the other hand normally operates in autonomous mode. Prioritizing autonomous submerged navigation accuracy, standard HUGIN 1000 is equipped with RDI WHN-600.

Table 2. RDI Workhorse Navigator Doppler Velocity Log accuracy and range specifications, ref [7]. o.s. – of speed. The accuracy figures can be interpreted as 1σ values. The error consists of two independent varying components; scale factor error and bias. The total error is the root mean square (rms) of the two components.

Frequency	**Long term accuracy**	**Range**
150 kHz	±0.5% o.s. ± 2 mm/s	425 – 500 m
300 kHz	±0.4% o.s. ± 2 mm/s	200 m
600 kHz	±0.2% o.s. ± 1 mm/s	90 m
1200 kHz	±0.2% o.s. ± 1 mm/s	30 m

3.2 Simplified Error Analysis Straight Trajectories

The simplified error analysis presented in this section is useful for understanding the basic mechanisms of a DVL aided INS and assessing how IMU and DVL sensor accuracy is determining the overall position accuracy.

The horizontal position drift in a DVL-aided INS is determined by the error in the estimated Earth-fixed velocity (i.e. North and East velocity). The main contributors to this error are:

- Error in the body-fixed velocity
- Error in heading.

The error in estimated body fixed velocity is mainly determined by the low-frequency error in the DVL itself (without position aiding this error is not observable when going at a straight line). High frequency velocity errors are estimated by means of the accelerometers. Even the most accurate INS will without aiding after a short period of time have a velocity uncertainty larger than the DVL accuracy. Referring to Table 2, a 300 kHz DVL typically have a scale factor type of error of 0.4% of speed, contributing to an along track error drift of 0.4% of traveled distance, or 28.8 m/hour for an AUV traveling at 2 m/s (4 knots). However, there are ways to improve the DVL accuracy: Sacrificing range, the 600 kHz and 1200 kHz versions have an accuracy specification of 0.2% of speed, corresponding to 0.2% of traveled distance, or 14.4 m/hour (AUV speed 2 m/s). The scale factor error is observable by the Kalman filter when position measurements are available or when the AUV is turning. Thus, the Kalman filter can compensate for part of the scale factor error when running more complex missions than a straight line. This is discussed in Section 3.3.

The error in heading is determined by the gyrocompassing capability of the integrated system. The heading estimation error will typically be of low frequency, corresponding to non-observable gyro bias dynamics. Referring to Table 1, a 1 nmi/h navigation class IMU typically gyrocompass to an accuracy of $\sigma(\delta\psi)$ = 0.02° / cos(latitude). This corresponds to an across track error drift of $\sigma(\delta\psi)\cdot 100$ % of traveled distance ($\sigma(\delta\psi)$ in radians). At 45° latitude, this equals 0.05% of traveled distance, or 3.4 m/hour at 2 m/s AUV speed. DVL error in AUV body y-direction also contributes to across track drift. If there is no current, the scale factor error can be ignored leaving the constant error defined in Table 2. The WHN-600 accuracy specification of 1 mm/s contributes with 3.6 m/hour 1σ. Assuming uncorrelated error processes, across track error drift amounts to 5.0 m/hour at 2 m/s AUV speed.

In [8] position accuracy for an INS with 1 nmi/h IMU and 1200 kHz DVL following a straight line was simulated. Along track position error drift was in the order of 8 m/hour while cross track position error drift was in the order of 2.5 m/hour. This is a somewhat smaller drift than predicted by the simplified error analysis. There are two main reasons for this: the Kalman filter compensates for a scale factor error estimated when position measurements were available and the actual scale factor error is modeled as a first order Markov process and not a constant error. Choosing time constants that realistically reflect the physical error processes is very important when estimating DVL aided INS error drift and when tuning the Kalman filter for real applications. This explains the importance of characterization of the DVL errors discussed in Section 5.1.

3.3 Countering DVL Aided INS Position Error Growth

For a submerged AUV without position updates, the position error growth of a DVL aided INS can be countered by:

1. Mission pattern for canceling of error growth
2. Kalman filter estimation and compensation of DVL error

The accuracy estimates in Section 3.2 are valid for straight-line trajectories. Since the main error contributors of DVL aided INS is body fixed velocity and heading, a canceling effect of the error growth is obtained when for instance running a lawn mower pattern. The canceling effect increases with the stability of the body fixed velocity error and heading error. Also the canceling effect increases with shorter line lengths.

A second important effect of maneuvering is that the velocity error actually becomes observable by means of comparing expected centripetal acceleration with measured acceleration from the IMU. If the velocity error is the same during the maneuver (i.e. when it is observed) as it is in the following line, the error is estimated and compensated for. The error growth when running long straight lines can thus be reduced by adding 360° turns at regular intervals. However this mechanism requires accurate DVL time stamping and unchanged DVL error characteristics in the turn (refer to Sections 5.3 and 5.4), and is thus challenging to demonstrate on real data in real-time.

The two effects combined are very effective, as seen in Table 3, which contains results from NavLab simulations (see [5] and [6] for NavLab description).

Table 3. Typical reduction in position error drift for a DVL aided INS when comparing a straight-line trajectory with a lawn mower pattern, [7]. The numbers apply for a 1200 kHz DVL and a 1 nmi/h IMU at 45° latitude.

Position error drift (% of traveled distance)	**Straight line**	**Lawn mower pattern with 1 km lines**
Along track	0.11%	0.01%
Across track	0.03%	0.001%

4 Heading Estimation Accuracy and Importance of DVL Mounting Accuracy

Gyrocompassing (heading estimation by observing Earth rotation) is an inherent part of an optimal estimator as long as it gets position or velocity measurement updates. The IMU gyro bias limits the accuracy of the INS heading estimate. Ignoring gyro angular random walk and acceleration uncertainty, INS gyro compassing accuracy is approximately given by

$$\sigma(\delta\psi_{IMU}) = \frac{\Delta\omega_{gyrobias}}{\omega_{IE}\cos\mu} \tag{1}$$

where $\sigma(\delta\psi_{IMU})$ is heading accuracy due to IMU gyro bias, $\Delta\omega_{gyrobias}$ is gyro bias and $\omega_{IE}\cos\mu$ is the horizontal component of Earth's rotation rate (μ is latitude).

The DVL measures the velocity vector in AUV body coordinates. The INS heading, pitch and roll estimates are used to transform this velocity into an Earth fixed coordinate system. If position measurements are available, the Kalman filter estimates errors in velocity and heading. Ignoring white measurement noise, this mechanism contributes with a heading accuracy of

$$\sigma(\delta\psi_{DVL}) = \frac{\Delta v_{DVLerror,y}}{v_{EB,x}^{B}} \tag{2}$$

where $\sigma(\delta\psi_{DVL})$ is heading accuracy due to DVL, $\Delta v_{DVLerror,y}$ is low frequent DVL error in AUV body y and $v_{EB,x}^{B}$ is AUV forward velocity.

Insertion of typical figures for a navigation grade IMU and a RDI WHN-600 results in $\sigma(\delta\psi_{IMU})$ = 0.05° at 45° latitude and $\sigma(\delta\psi_{DVL})$ = 0.1°. In theory, these mechanisms will work in concert improving the overall heading accuracy. However, if there is a mounting misalignment between the IMU and the DVL, which is not accounted for in the Kalman filter, they can actually be counter productive. Thus, DVL mounting accuracy is very important to achieve the accuracy offered by the IMU and the DVL sensors themselves.

In HUGIN 1000, special care is taken during production to mount all sensors, including IMU and DVL, as accurately as possible. After assembling, NavLab is used to estimate any DVL misalignment. The IMU and DVL are mounted with steering pins. Thus, if the sensors are dismounted for service or inspection, the steering pins will assure the same orientation when remounted.

5 DVL Error Analysis

5.1 DVL Error Budget

As discussed in Section 3.2 the magnitude and the frequency characteristics of the DVL error is important to the position drift in the integrated inertial navigation system. White measurement noise causes a position uncertainty that increases with square root of time. A constant error causes a linear position error drift, which potentially can be cancelled out by a lawn mower pattern. From a Kalman filter point of view, the physical error process should be modeled truthfully to get as accurate position estimate as possible.

Table 4 summarizes a scale factor error budget for a 300 kHz broadband DVL, [9]. The two constant components are candidates for estimation and compensation. However, the major part of the error budget is time varying. This is in accordance with Kalman filter estimates in HUGIN 1000, see Figure 6. Further investigation, identification and understanding of the DVL error processes are of priority because of the prospects of improving overall navigation accuracy by more accurate error modeling.

Table 4. 300 kHz DVL scale factor error budget. Copied from [9].

Error source	**Scale factor error (%)**	**Time varying**
Absorption bias. 2 sigma over 0 to 80% of range.	0.081	Yes
Terrain bias. 2 sigma about centre of typical range of backscatter slopes.	0.144	Yes
Sound speed temperature dependence. Assuming 0.5º rms uncertainty.	0.100	Yes
Sound speed salinity dependence. Assuming 0.5 ppt rms uncertainty.	0.040	Yes
Sidelobes (4 dB rms)	0.040	Yes
Beam angle	0.086	No
Other minor sources (clock drift, etc.)	0.080	No
"Total" (rms of values above)	0.233	

5.2 Importance of Sound Speed Accuracy

The velocity scale factor is proportional to the sound speed at the transducer. The WHN DVLs can be set to compute sound speed based on internal sensors or sent data of salinity, temperature and depth. If there is an error, correct sound speed can be post-processed using the following equation (see [11]):

$$v_{corrected} = v_{uncorrected} \frac{C_{real}}{C_{ADCP}} \tag{3}$$

where C_{real} is the true sound speed at the transducer and C_{ADCP} is the sound speed recorded by the ADCP. Using this formula the DVL error due to wrong sound velocity estimate can be computed as:

$$\delta v_{DVL} = \frac{v_{uncorrected}}{C_{ADCP}} \delta C \qquad (4)$$

where δv_{DVL} is DVL error and δC is sound velocity error. Assuming C_{ADCP}=1500 m/s and $v_{uncorrected}$ = 2 m/s, a sound velocity error, δC = 3 m/s, causes a DVL error of 0.2% of speed. The sound speed induced velocity error should be considerably less than the DVL scale factor error (refer to Table 2). A sound speed accuracy of 0.5 m/s is a reasonable specification.

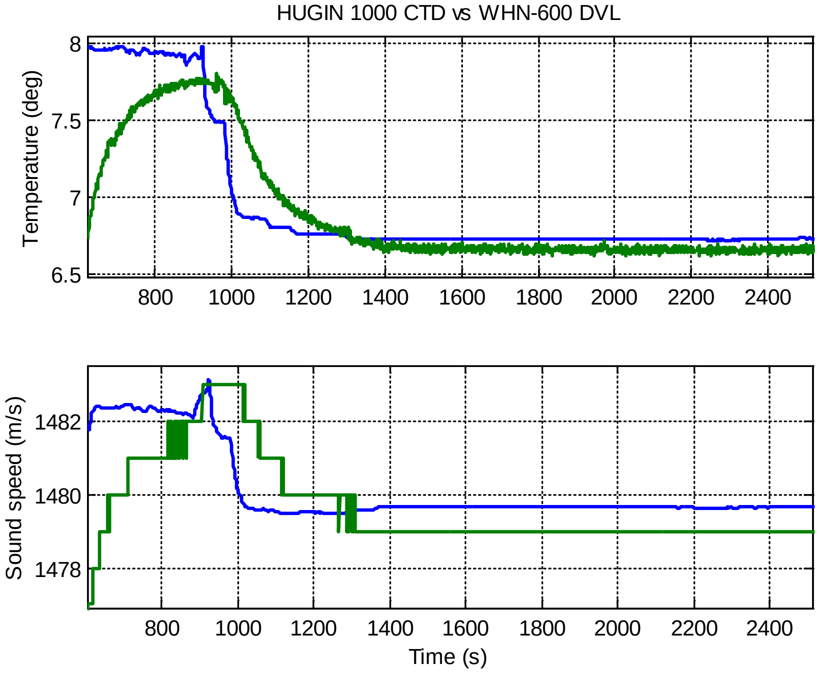


Figure 3. Comparison of temperature and sound speed between the WHN-600 (green graphs) and the HUGIN 1000 CTD sensor (blue graphs). Note that least significant digit for sound speed in DVL datagram is 1 m/s.

In the sea trial referred to in Section 6, the DVL calculated sound speed using the default value for salinity (35 parts per thousand), its internal temperature sensor and depth set by the HUGIN Control Processor. In, CTD temperature and sound velocity calculation are compared with DVL temperature and sound velocity calculation. HUGIN 1000 is equipped with the oceanographic grade Falmouth Scientific 2” Micro CTD, [10]. The DVL temperature sensor has a slower response time than the CTD temperature sensor, which affects the sound speed calculation. When stabilized, the difference between CTD sound speed and DVL sound speed is less than 1 m/s. AUVs experiencing rapid changes in temperature, typically due to depth changes, should preferably send a CTD computed sound speed to the WHN.

5.3 Time Stamp Accuracy Requirements for DVL Aiding

At 2 m/s WHN-600 has a velocity accuracy of 0.004 m/s. In practice an underwater vehicle experience some dynamics. An ROV due to cable effects, an AUV during turns and when surfacing for GPS fixes. Thus IMU time stamp accuracy, DVL time stamp accuracy and DVL lever arm compensation are important to utilize the velocity accuracy offered by the DVL sensor itself.

Time stamp sensitivity to vehicle acceleration

If the vehicle is accelerating, the velocity error, δv, due to incorrect time stamps when comparing DVL measurement with INS velocity in the Kalman filter, is given by:

$$\delta v = a \cdot \delta t \tag{5}$$

where *a* is vehicle acceleration and δt is time stamp error.

Time stamp sensitivity to vehicle rotation rate change

If the AUV is rotating and there is an arm between the IMU and the DVL, the DVL velocity must be lever arm compensated before being compared to the INS velocity in the INS body frame.

Only considering one axis (for simplicity) DVL lever arm compensation, v_{comp}, is given by:

$$v_{comp} = l \cdot \omega \tag{6}$$

where *l* is lever arm and ω is angular rotation rate ($\left|\omega_{EB}^{B}\right|$). Error in DVL lever arm compensation, $\delta\ v_{comp}$, due to time stamp error is given by:

$$\delta v_{comp} = l \cdot \dot{\omega} \cdot \delta t \tag{7}$$

where $\dot{\omega}$ is change in angular rate and δt is time stamp error. The error is proportional to the length of the lever arm. Preferably, lever arms should by vehicle design be as small as possible.

Time stamp requirements

An obvious requirement is that time stamp induced velocity error due to vehicle acceleration and vehicle rotation rate change is less than DVL scale factor error. Analysis of typical HUGIN 1000 dynamics during GPS surface fixes, show that DVL time stamp accuracy should be better than 10 ms. Such accuracy can be achieved by running the WHN-600 in external sync, time stamp the sync pulse and calculate correct time stamp from this value. This is a rather cumbersome method. Preferably the DVL output datagrams should include a DVL latency defined as midpulse-on-bottom to first character of output transmission.

5.4 Geometric Effect of Turn Rate on DVL Accuracy

In order to examine the geometric effect of turning the following assumptions are made:

1. The vehicle has no heave, roll or pitch motion.
2. The four DVL beams are (from a horizontal point of view) directed 90° to each other.

The DVL is rotated such that the first beam is an angle α (usually 45º) away from the forward direction. As there are no heave, roll or pitch motion, all Doppler effects come from sway, surge and yaw. Hence, only the horizontal part is considered. The vehicle's speed in each of the four DVL directions at transmission time can in terms of forward speed $v_{EB,x}^{B}$ and starboard speed $v_{EB,y}^{B}$, be expressed as

$$\boldsymbol{v}_{D,trans} = \begin{bmatrix} v_{D1,trans} \\ v_{D2,trans} \\ v_{D3,trans} \\ v_{D4,trans} \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \\ -\cos\alpha & -\sin\alpha \\ \sin\alpha & -\cos\alpha \end{bmatrix} \begin{bmatrix} v^B_{EB,x} \\ v^B_{EB,y} \end{bmatrix} = \boldsymbol{A}(\alpha) \begin{bmatrix} v^B_{EB,x} \\ v^B_{EB,y} \end{bmatrix} = \boldsymbol{A}_{trans} \begin{bmatrix} v^B_{EB,x} \\ v^B_{EB,y} \end{bmatrix} \tag{8}$$

Studying the case when the vehicle is turning, the vehicle has rotated an angle $\Delta\psi$, when the reflected beams return. The speed at reception in the four DVL directions is given by

$$\boldsymbol{v}_{D,rec} = \boldsymbol{A}(\alpha - \Delta\psi) \begin{bmatrix} v^B_{EB,x} \\ v^B_{EB,y} \end{bmatrix} = \boldsymbol{A}_{rec} \begin{bmatrix} v^B_{EB,x} \\ v^B_{EB,y} \end{bmatrix} \tag{9}$$

The relative Doppler shifts can, since $\mathrm{v}_{\mathrm{trans}} << c$, be expressed as

$$\Delta\boldsymbol{f} \approx \frac{\boldsymbol{v}_{D,trans} + \boldsymbol{v}_{D,rec}}{c} = \frac{1}{c}\left(\boldsymbol{A}_{trans} + \boldsymbol{A}_{rec}\right) \begin{bmatrix} v^B_{EB,x} \\ v^B_{EB,y} \end{bmatrix} \tag{10}$$

The DVL measurement is given by

$$\begin{bmatrix} \tilde{v}^B_{EB,x} \\ \tilde{v}^B_{EB,y} \end{bmatrix} = \frac{c}{2}\left(\boldsymbol{A}^T_{trans}\boldsymbol{A}_{trans}\right)^{-1} \boldsymbol{A}^T_{trans}\Delta\boldsymbol{f} = \frac{1}{2}\left(\boldsymbol{I} + \left(\boldsymbol{A}^T_{trans}\boldsymbol{A}_{trans}\right)^{-1} \boldsymbol{A}^T_{trans}\boldsymbol{A}_{rec}\right) \begin{bmatrix} v^B_{EB,x} \\ v^B_{EB,y} \end{bmatrix} \tag{11}$$

This yields a scale factor type of error. It is though negligible for most AUV missions because the angle $\Delta\psi$ is small. For instance with 50 m AUV altitude and $\dot{\psi}$ = 6°/s turn rate, results in $\Delta\psi$ ≈ 0.5°.

6 Results with RDI WHN-600

The results shown in this section are based on a HUGIN 1000 test run in Horten January 20th 2004. HUGIN 1000 ran with DVL aiding only, but a surface ship followed HUGIN 1000 and made independent DGPS-USBL (ultra short baseline) measurements of the AUV position. These position measurements were later used in a NavLab post-processing to provide an accurate position reference, with which the DVL aided INS results were compared. Please note that navigation systems are complex statistical processes and a number of separate missions should be run and compared for proper statistical performance characterization.

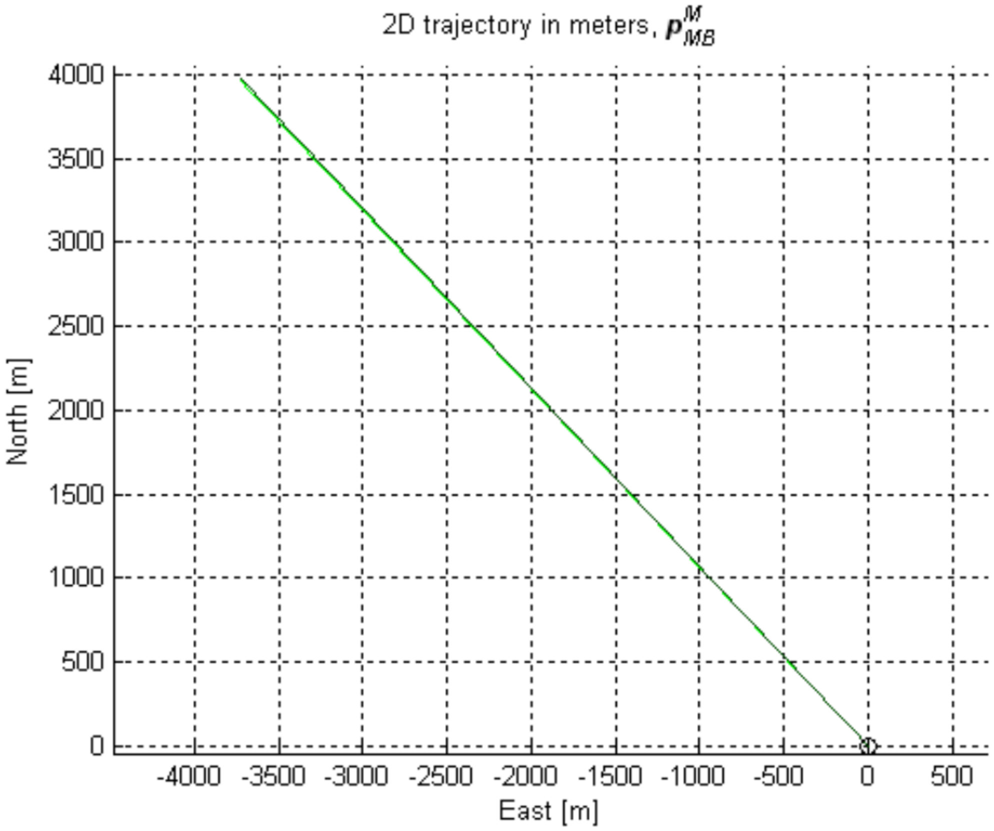


Figure 4. 2D trajectory. Black circle: Starting point. Black line: Post-processed position reference. Green line: Estimated (Kalman filtered) position using only DVL aiding.

Figure 4 shows the 5.5 km straight-line trajectory run by HUGIN 1000. Prior to this line, the navigation system was aligned with position measurements. With DVL aiding only, the position estimate drifts slowly off as explained in Section 3.2. The drift is hardly visible in Figure 4, but is clearly shown in Figure 5, which plots the difference between the DVL aided INS position estimate and the independent position reference. The drift is shown relative to the AUV body (B) system.

Along track error drift is in the order of 4.5 m. Theoretically the WHN-600 0.2% scale factor error (refer to Table 2) should contribute with 11 m (1σ). The good result indicates that the WHN-600 performed better than specification. This is confirmed in Figure 6 where the DVL error has been estimated in the post-processed NavLab run with position measurements. The post-processed Kalman filter estimated the DVL scale factor error to be less than 0.1% (< 2 mm/s at 2 m/s AUV speed).

Across track error drift is approximately 11 m. HUGIN 1000 was equipped with an IMU with a gyro bias specification of 0.01°/h (refer to Table 1). According to Equation (1) and Section 3.2, this corresponds to an across track drift of 7.3 m (1σ) at 60° latitude. There was virtually no current, and hence marginal sideways speed. The DVL across track error contribution is thus left with the constant y-velocity specification of 1 mm/s (Table 2), which contributes with 2.7 m (1σ). The importance of DVL misalignment estimation was discussed in Section 4. In this run no misalignment of the DVL relative to the IMU was compensated for. Mechanically a production uncertainty of 0.1° is expected, theoretically contributing to 9.6 m drift in this case (1σ). Combining these error sources, the across track error drift is reasonable.

Uncompensated DVL misalignment about the z-axis behaves like a bias in DVL y-direction when the AUV has a constant forward velocity. This can be seen in the second graph in Figure 6, which shows the estimated DVL bias from the NavLab post-processing with position measurements. The estimated bias in y-direction has an average of about -4.5 mm/s, exceeding the constant DVL specification of 1 mm/s, but corresponding to a misalignment of 0.13° (neglecting DVL y bias), not far from the expected accuracy of the mechanical mounting. Based on post-processing of a few missions, a DVL misalignment can be estimated. DVL misalignment is an ini-file parameter, enabling the navigation system to compensate for the misalignment in real-time. As mentioned in Section 4, the IMU and DVL are mounted with steering pins, allowing service and inspection without necessitating renewed estimation of DVL misalignment.

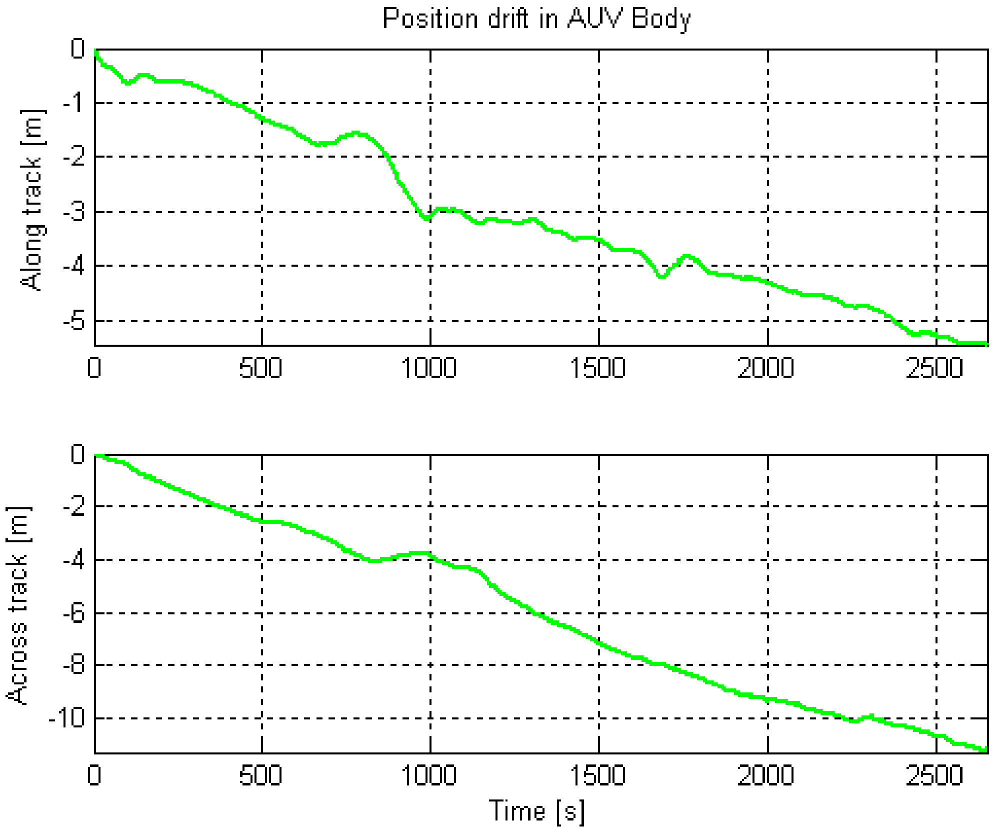


Figure 5. DVL aided INS position drift.

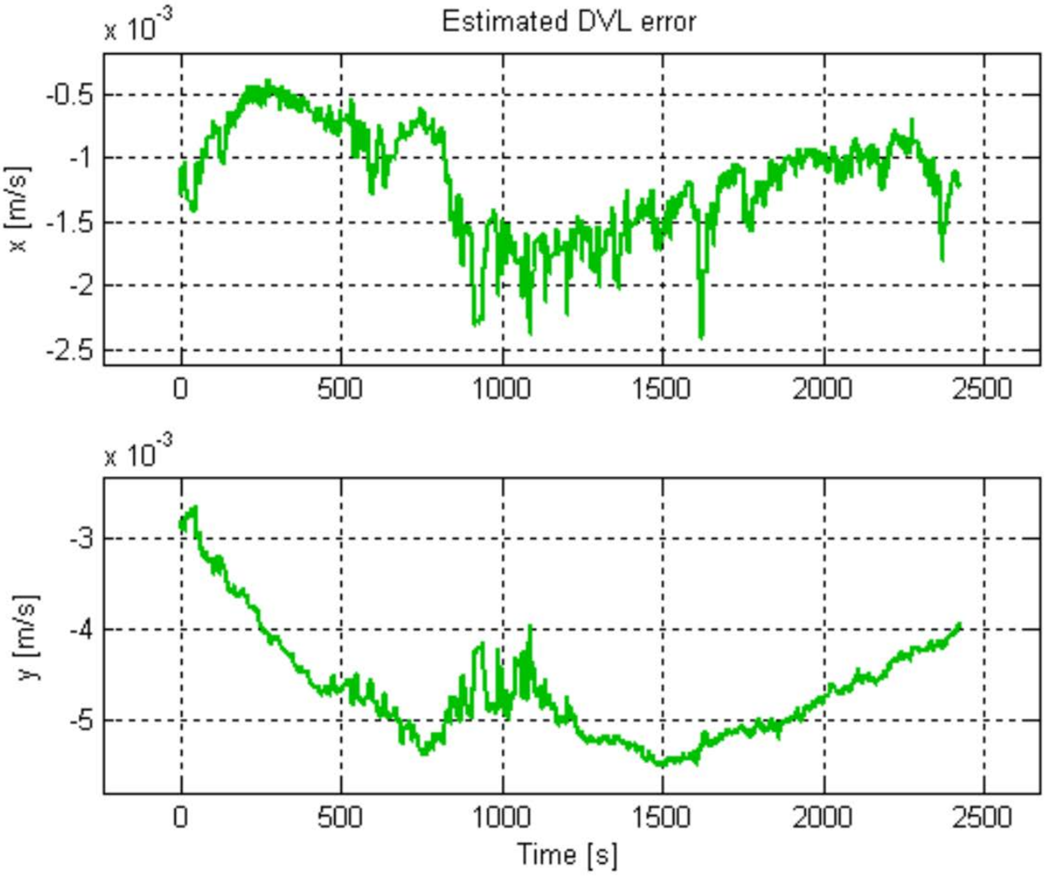


Figure 6. Post-processed DVL error estimate of the same run as shown in Figure 4 and Figure 5. The post-processing was done with DGPS-USBL position updates to improve DVL error observability. The DVL measures velocity in three dimensions, but error estimates are only shown in body x and y for clarity.

7 Summary

The Doppler velocity log and the inertial measurement unit are the key AUV navigation sensors enabling submerged operation for long periods of time. To utilize the velocity accuracy offered by the DVL, mounting misalignment between the IMU and the DVL must be minimized, sound speed must be accurately calculated and the sensor data properly time tagged.

HUGIN 1000 sea trials indicate that performance of the RDI WHN-600 is well within specification.

References

[1] T. C. Chance, A. A. Kleiner and J. G. Northcutt, "The HUGIN 3000 AUV," *Sea Technology*, vol. 41, no. 12, December 2000, pp. 10-14.

[2] R. A. George, J. Shuy and E. Cauquil, “Deepwater AUV Logs 25,000 Kilometers Under the Sea”, *Sea Technology,* vol. 44, no. 12, December 2003, pp.10-15.

[3] K. Vestgård, R. Hansen, B. Jalving and O. A. Pedersen, “The HUGIN 3000 Survey AUV”, ISOPE-2001, Stavanger, Norway, June 2001.

[4] P. E. Hagen, N. Størkersen, K. Vestgård and P. Kartvedt: "The HUGIN 1000 Autonomous Underwater Vehicle for Military Applications", Proceedings from Oceans 2003, San Diego, CA, USA, September 2003.

[5] K. Gade, ”NavLab – Overview and User Guide”, FFI/Report 2003/02128, Norwegian Defence Research Establishment, November 2003.

[6] B. Jalving, K. Gade, O. K. Hagen and K. Vestgård, ”A Toolbox of Aiding Techniques for the HUGIN AUV Integrated Inertial Navigation System”, Proceedings from Oceans 2003, September 23 – 25, San Diego, CA, USA.

[7] RD Instruments, “Workhorse Navigator Doppler Velocity Log (DVL) ”, http://www. dvlnav.com/pdfs/navbro. pdf, June 2003.

[8] B. Jalving, E. Bovio, K. Gade, “Integrated Inertial Navigation Systems for AUVs for REA Applications”, NATO Underwater Research Center Conference Proceedings from MREP 2003, NATO Underwater Research Center May 12 – 15, 2003, La Spezia, Italy.

[9] Graham Lester, “Email to Bjørn Jalving on RDI DVL error budget”, June 21st 2001.

[10] Falmouth Scientific, “2” Micro CTD Specification Sheet”, www.falmouth.com, April 2004.

[11] RD Instruments, Acoustic Doppler Current Profiler, Principles of operation, A practical primer, San Diego, CA, USA, January 1996.

Paper VII

Hegrenæs, Ø., Hallingstad, O. and Gade, K. (2007). Towards Model-Aided Navigation of Underwater Vehicles. *Modeling, Identification and Control,* vol. 28, no. 4, pp. 113-123

Modeling, Identification and Control, Vol. 28, No. 4, 2007, pp. 113–123

Towards Model-Aided Navigation of Underwater Vehicles*

Øyvind Hegrenæs[1, 2] Oddvar Hallingstad[1, 2] Kenneth Gade[3]

[1] *University Graduate Center at Kjeller (UNIK), NO-2027 Kjeller, Norway. E-mail:* {hegrenas,oh}@unik.no

[2] *Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), NO-7491 Trondheim, Norway.*

[3] *Norwegian Defence Research Establishment (FFI), NO-2027 Kjeller, Norway. E-mail:* kenneth.gade@ffi.no

Abstract

This paper reports the development and preliminary experimental evaluation of a model-aided inertial navigation system (INS) for underwater vehicles. The implemented navigation system exploits accurate knowledge of the vehicle dynamics through an experimentally validated mathematical model, relating the water-relative velocity of the vehicle to the forces and moments acting upon it. Together with online current estimation, the model output is integrated in the navigation system. The proposed approach is of practical interest both for underwater navigation when lacking disparate velocity measurements, typically from a Doppler velocity log (DVL), and for systems where the need for redundancy and integrity is important, e.g. during sensor dropouts or failures, or in case of emergency navigation. The presented results verify the concept that with merely an addition of software and no added instrumentation, it is possible to considerably improve the accuracy and robustness of an INS by utilizing the output from a kinetic vehicle model. To the best of our knowledge, this paper is the first report on the implementation and experimental evaluation of model-aided INS for underwater vehicle navigation.

Keywords: Inertial navigation; Kalman filtering; Model aiding; State estimation; Underwater vehicles.

1 Introduction

Deciding which sensor outfit to include in an underwater navigation system is important both from a performance and cost perspective. A typical sensor outfit may consist of standard components such as compass, pressure sensor, and some class of inertial navigation system (INS). In addition, various sources of position aiding may be available, for instance long baseline (LBL) or ultra short baseline (USBL) acoustics, terrain-based techniques, and surface GPS. For an extensive survey on sensor systems and underwater navigation the reader should refer to Kinsey et al. (2006) and references therein.

In practice, a submersible does not have continuous position updates, hence a navigation solution based solely on INS, and in particular low-cost INS, will have an unacceptable position error drift without sufficient aiding. While most high-end systems also incorporate a Doppler velocity log (DVL) in their sensor suite in order to limit the drift, this additional expense is not always feasible for low-cost systems. Even when a DVL unit is included, situations may also occur where it fails to work or measurements are discarded due to decreased quality. In either case, in the absence of DVL measurements, alternative velocity information is required to achieve an acceptable low drift navigation solution between position updates. One possibility is to utilize

*Published in Proceedings of the 15th International Symposium on Unmanned Untethered Submersible Technology (UUST'07), Durham, NH, USA, August 19-22, 2007.

ISSN 1890–1328

doi:10.4173/mic.2007.4.3

mathematical models describing the vehicle dynamics, in conjunction with online sea current estimation.

The purpose of this paper is twofold. First, with the aim of providing model-based velocity measurements, an experimentally validated kinetic vehicle model is presented. Second, the potential use of such a model as a mean for aiding the INS of an underwater vehicle is investigated, and the effectiveness of the integrated navigation system is evaluated on experimental data.

To date, the use of model-based state estimators for underwater navigation has primarily focused on applying purely kinematic models, i.e. models describing the vehicle motion without the consideration of the masses or forces that bring it about. State estimators based on kinetic underwater vehicle models are rare. Model-based nonlinear deterministic observers utilizing the knowledge of the vehicle dynamics together with disparate measurements are proposed in Kinsey and Whitcomb (2007); Refsnes et al. (2007). Both papers evaluate their observer using experimental data. As for model-aided INS, some simulation studies have been reported for aerial vehicles (Bryson and Sukkarieh, 2004; Koifman and Bar-Itzhack, 1999; Vasconcelos et al., 2006). To the best of our knowledge however, no results have been reported through simulations or experiments, where the output from a kinetic vehicle model is used to aid the INS of an underwater vehicle.

Note that as studied herein, the integration of vehicle models in underwater navigation systems is of particular interest for systems without a DVL unit. Other important implications involve systems (also having a DVL) where the need for redundancy and integrity is important, e.g. during sensor dropouts or sensor failures, or in case of emergency navigation.

The remainder of this paper is organized as follows. Section 2 presents the mathematical vehicle model utilized in this paper. The integrated navigation system with model aiding included is described in Section 3, including a brief discussion on assumptions applied during development. Section 4 and 5 describe the experimental setup and experimental results, where in particular, the solutions from the navigation systems with and without model aiding in place are compared.

2 Modeling

The steps involving development and validation of the finite-dimensional mathematical vehicle model utilized in this paper have been rigorously treated in Hegrenæs et al. (2007a). For an extended review and historical recap of work related to modeling of underwater vehicles the reader should refer to the same paper and references therein. The main results are presented in the following.

2.1 Preliminaries

In cases where a vehicle operates in a limited geographical area, it is common to apply a flat Earth approximation when describing its location. Let $\{m\}$ denote a local Earth-fixed coordinate frame where the origin is fixed at the surface of the WGS-84 Earth ellipsoid, and the orientation is north-east-down (NED). Similarly, let $\{w\}$ denote a reference frame where the origin is fixed to, and translates with the water (due to current). The current is assumed irrotational, hence $\{w\}$ does not rotate relative to $\{m\}$. The frame $\{b\}$ is a body-fixed frame where the axes coincide with the principal axes of the vehicle. The origin is located at the vehicle center of buoyancy. A general expression of the vehicle position can now be written as

$$\begin{aligned} \boldsymbol{p}_{mb}^{m} &= \boldsymbol{p}_{mw}^{m} + \boldsymbol{p}_{wb}^{m} \\ &= \boldsymbol{p}_{mw}^{m} + \boldsymbol{R}_{w}^{m}\boldsymbol{p}_{wb}^{w}, \end{aligned} \tag{1}$$

where $\boldsymbol{p}_{wb}^{m} \in \mathbb{R}^3$ is the vector from the origin of $\{w\}$ to the origin $\{b\}$, decomposed in $\{m\}$, and $\boldsymbol{R}_{w}^{m} \in SO(3)$ is the coordinate transformation matrix from $\{w\}$ to $\{m\}$. The velocity of $\{b\}$ relative to $\{m\}$, represented in $\{m\}$, is given as $\boldsymbol{v}_{mb}^{m} := \dot{\boldsymbol{p}}_{mb}^{m}$, or decomposed in $\{b\}$ as $\boldsymbol{v}_{mb}^{b} := \boldsymbol{R}_{m}^{b}\boldsymbol{v}_{mb}^{m}$. The interpretation of the other variables follows directly. Taking the time derivative of both sides of (1) yields

$$\begin{aligned} \dot{\boldsymbol{p}}_{mb}^{m} &= \dot{\boldsymbol{p}}_{mw}^{m} + \dot{\boldsymbol{R}}_{w}^{m}\boldsymbol{p}_{wb}^{w} + \boldsymbol{R}_{w}^{m}\dot{\boldsymbol{p}}_{wb}^{w} \\ &= \dot{\boldsymbol{p}}_{mw}^{m} + \boldsymbol{R}_{w}^{m}\dot{\boldsymbol{p}}_{wb}^{w}, \end{aligned} \tag{2}$$

where $\dot{\boldsymbol{R}}_{w}^{m}$ equals zero due to the assumption of irrotational current. Multiplying both sides of (2) with $\boldsymbol{R}_{m}^{b}$ finally gives the velocity relationship

$$\boldsymbol{v}_{mb}^{b} = \boldsymbol{R}_{m}^{b}\boldsymbol{v}_{mw}^{m} + \boldsymbol{v}_{wb}^{b}. \tag{3}$$

Analogous to the linear velocities, their angular counterparts are given as $\boldsymbol{\omega}_{mb}^{m}$ and $\boldsymbol{\omega}_{mb}^{b} := \boldsymbol{R}_{m}^{b}\boldsymbol{\omega}_{mb}^{m}$.

For navigation purposes, two additional reference frames are common. The Earth-centered Earth-fixed (ECEF) coordinate frame is denoted $\{e\}$. The frame $\{l\}$ denotes a wander azimuth frame, defined such that it has zero angular velocity relative to the Earth about its z-axis. The initial orientation is NED and its origin is directly above the vehicle at the surface of the Earth ellipsoid. Note that $\{m\}$ is fixed relative to $\{e\}$, and that $\boldsymbol{R}_{l}^{b} \approx \boldsymbol{R}_{m}^{b}$ for operations in limited geographical areas, far from the poles. In light of the new frames, (3) may be restated as

$$\boldsymbol{v}_{eb}^{b} = \boldsymbol{R}_{l}^{b}\boldsymbol{v}_{ew}^{l} + \boldsymbol{v}_{wb}^{b}. \tag{4}$$

The correspondence between the variables above and the SNAME (1950) notation is shown in Table 1.

Table 1: Nomenclature

Description	Variable	Entries*
Local vehicle position	$\boldsymbol{p}^m_{mb}$	(x, y, z)
Earth-relative linear velocity	$\boldsymbol{v}^b_{mb} = \boldsymbol{v}^b_{eb}$	(u, v, w)
Water-relative linear velocity	$\boldsymbol{v}^b_{wb}$	(u_r, v_r, w_r)
Current velocity	$\boldsymbol{v}^l_{mw} = \boldsymbol{v}^l_{ew}$	(u^l_c, v^l_c, w^l_c)
Vehicle angular velocity	$\boldsymbol{\omega}^b_{mb} = \boldsymbol{\omega}^b_{eb}$	(p, q, r)
External forces on vehicle	$\boldsymbol{f}^b$	(X, Y, Z)
External moments on vehicle	$\boldsymbol{m}^b$	(K, M, N)
Attitude (roll, pitch, yaw)	$\boldsymbol{\Theta}$	(ϕ, θ, ψ)

* Based on SNAME notation.

2.2 Kinetic Vehicle Model

As shown in Fossen (2002) a general expression of the rigid body equations of motion can be written as

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau}_{RB}, \tag{5}$$

where $\boldsymbol{M}_{RB}$ is the rigid body inertia matrix, $\boldsymbol{C}_{RB}$ is the corresponding matrix of Coriolis and centripetal terms, and $\boldsymbol{\tau}_{RB}$ is a generalized force vector of external forces and moments. For 3 DOF motion in the horizontal plane (surge, sway, and yaw), the generalized force and velocity vectors are $\boldsymbol{\tau}_{RB} = [X, Y, N]^\top$ and $\boldsymbol{\nu} = [u, v, r]^\top$.

The difficulty in modeling an underwater vehicle arises when expressing the right hand-side of (5). One possibility is to linearly decompose $\boldsymbol{\tau}_{RB}$ as

$$\boldsymbol{\tau}_{RB} = \boldsymbol{\tau}_S + \boldsymbol{\tau}_H + \boldsymbol{\tau}, \tag{6}$$

where the generalized hydrostatic force $\boldsymbol{\tau}_S$ is known in its exact form. The generalized hydrodynamic force $\boldsymbol{\tau}_H$ arises from the reaction between the surrounding fluid and the submerged vehicle in motion. The last generalized force component $\boldsymbol{\tau}$ consists of forces and moments from propulsion and control surfaces.

The HUGIN 4500 autonomous underwater vehicle (AUV) is used as a case study in this paper. Its bare hull is a body of revolution, and it has a cruciform tail fin configuration that is top-bottom, port-starboard symmetric. A 3 DOF kinetic model for this vehicle can, after adding up the contributions in (6), be written as

$$\begin{aligned}\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau} - \boldsymbol{M}_A\dot{\boldsymbol{\nu}}_r - \boldsymbol{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r - \\ \boldsymbol{d}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r - \boldsymbol{l}(\boldsymbol{\nu}_r) - \boldsymbol{g}(\boldsymbol{\Theta}).\end{aligned} \tag{7}$$

A description and complete expressions for the various terms are given in Hegrenæs et al. (2007a). Note the difference between $\boldsymbol{\nu}$ and $\boldsymbol{\nu}_r = [u_r, v_r, r]^\top$, denoting generalized Earth-relative (inertial) and generalized water-relative velocity, respectively.

For (7) one must decide upon using either $\boldsymbol{\nu}$ or $\boldsymbol{\nu}_r$ as the velocity state. As discussed in Hegrenæs et al. (2007a), a reasonable assumption at low vehicle angular rates or small current amplitudes is that $\dot{\boldsymbol{\nu}} \approx \dot{\boldsymbol{\nu}}_r$. This yields the final model

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_r = \boldsymbol{\tau} - \boldsymbol{c}(\boldsymbol{\nu}, \boldsymbol{\nu}_r) - \boldsymbol{d}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r - \boldsymbol{l}(\boldsymbol{\nu}_r) - \boldsymbol{g}(\boldsymbol{\Theta}), \tag{8}$$

where for simplicity we used

$$\begin{aligned}\boldsymbol{M} &:= \boldsymbol{M}_{RB} + \boldsymbol{M}_A \\ \boldsymbol{c}(\boldsymbol{\nu}, \boldsymbol{\nu}_r) &:= \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r.\end{aligned}$$

As seen from (8), the term $\boldsymbol{c}(\boldsymbol{\nu}, \boldsymbol{\nu}_r)$ depends on both $\boldsymbol{\nu}$ and $\boldsymbol{\nu}_r$. If there is no current then $\boldsymbol{\nu} = \boldsymbol{\nu}_r$. Also, only the translational part of $\boldsymbol{\nu}$ and $\boldsymbol{\nu}_r$ differ since the current is irrotational by assumption. The inertial velocity can be calculated from (4), once the current velocity and the water-relative velocity are known. This implies that the current must be measured or estimated. In the integrated navigation system studied in this paper, the current is included as a state in the Kalman filter (KF).

The equation in (8) can be solved using a standard numerical integration routine in order to recover the state. That is, model-based measurements of the water-relative velocity in surge and sway, as well as the yaw rate, can be attained from control inputs, attitude and current. Both control inputs and the vehicle attitude are usually measured. Also note that (8) was derived assuming negligible coupling from heave, and roll and pitch rate. This is a reasonable assumption for normal operations with the HUGIN 4500 AUV.

The model in (8) is a typical grey-box model where the vehicle behavior is described by a set of nonlinear differential equations with unknown parameters. For the model considered herein, the parameters were found from semi-empirical relationships, open-water test, and from navigation data collected by the HUGIN 4500. More information on the steps involved for identifying the parameters is found in Hegrenæs (2006); Hegrenæs et al. (2007a,b).

3 Model-Aided Underwater Navigation

Navigation systems built upon inertial principles, time of flight acoustics, velocity logs, and global positioning systems are all common. As pointed out in Kinsey et al. (2006), none of these techniques are perfect however, and in practice a combination of them is usually employed. This section reports the concept and development of an integrated model-aided INS for underwater navigation.

3.1 Traditional INS

The key components of any INS consist of an inertial measurement unit (IMU) and a set of equations imple-

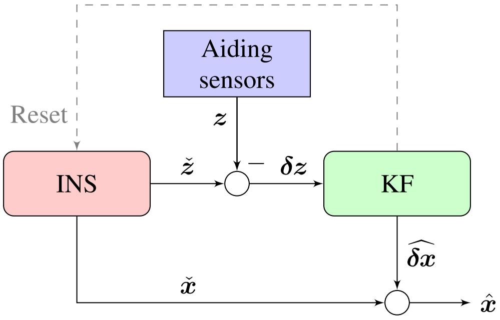


Figure 1: High-level outline if traditional aided INS.

mented in software. The navigation equations take the gyro and accelerometer measurements from the IMU as inputs and integrate them to velocity, position and orientation. The general solution of the navigation equations does not require any information on the dynamics of the vehicle in which the IMU is installed.

Since an INS is a diverging system, it requires an aiding system to limit the growth of its errors. Classically, aiding is accomplished using external measurements, e.g. position from acoustics and velocity from a DVL. A coarse schematic diagram of a traditional aided INS is shown in Figure 1, where the input to the KF is the difference between the aiding sensor output and that of the INS. The KF output includes estimates of the accumulated errors in the navigation equations, which are used for resetting the INS and for obtaining the best possible estimate of the true vehicle state (position, velocity and orientation). Besides modeling the INS errors, additional states may also be included in the KF, for instance colored noise in the aiding sensors.

3.2 Model-Aided INS

As mentioned above, a necessity to restrain the INS drift is the integration of external aiding sensors. Standard components such as compass and pressure sensor are almost always included, where the latter effectively binds the vertical geographical drift, i.e. drift along the z-axis of $\{m\}$, or more precisely $\{l\}$ (recall Section 2). For navigation in the geographical horizontal plane the situation is more complicated, and to date, the main aiding methods involve time of flight acoustic positioning and Doppler sonar velocity measurements.

A DVL may or may not be part of the sensor suite, and even when it is, situations will occur where it fails to work or measurements are discarded due to decreased quality. In either case, in the absence of DVL measurements, alternative velocity information is required to achieve an acceptable low drift navigation solution between position updates. As for the acoustics, measurements may be available often or only sporadically.

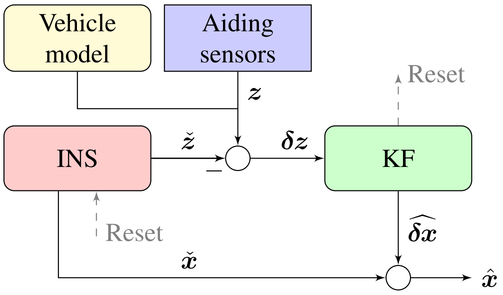


Figure 2: High-level outline of model-aided INS.

Both measurements are crucial for the INS performance. As is experimentally validated in Section 5, the output from an INS with neither position nor velocity measurements in place, rapidly becomes useless. This leads back to the question addressed in this paper – can the output from a kinetic vehicle model improve the accuracy and robustness of an INS?

The basic idea and concept of using a vehicle model for aiding an INS is illustrated in Figure 2, where the output from the kinetic model is treated analogously to that of an external aiding sensor. The model-aided INS clearly resembles the traditional INS in Figure 1, and both systems may share many of the same aiding sensors. As implemented herein, the DVL unit in the traditional INS is merely replaced by the vehicle model, after doing necessary modifications in the KF. A model-aided INS utilizing both external velocity measurements and model output is subject to ongoing research. Note that the integration of a vehicle model in the navigation system does not require any additional instrumentation. A more detailed outline of the navigation systems is shown in Figure 3, differing only in the velocity aiding. This is illustrated with a switch. The traditional INS with DVL serves as the basis when later evaluating the traditional and model-aided INS in Section 5.

3.3 Measurement and Process Equation

A DVL measures the vehicle velocity relative to the bottom, hence it is unaffected by the current. In contrast, the translational velocity calculated by the vehicle model is relative to the water. Consequently, in order to better make use of this velocity estimate for navigation purposes, the current must be accounted for.

In accordance to Figure 2 and conventional KF notation, the general input to the KF is given as

$$\delta z_k = z_k - \check{z}_k, \tag{9}$$

where the accent $\check{(\cdot)}$ denotes a calculated variable, in this case from the INS. For the velocity we then get

$$\delta z_{vel} = z_{vel} - \check{z}_{vel}, \tag{10}$$

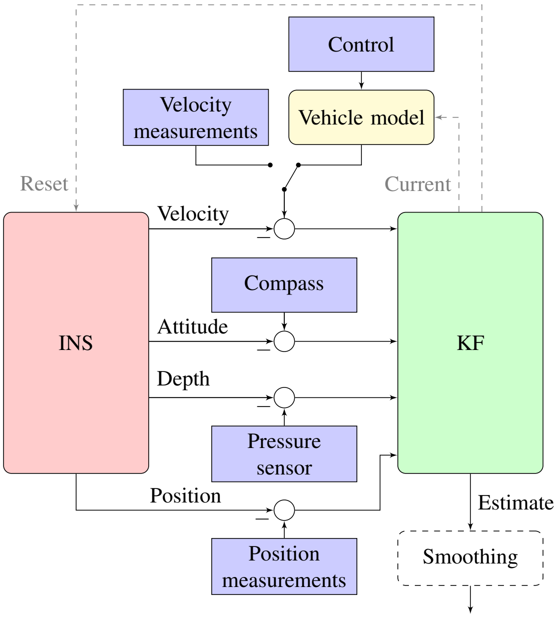


Figure 3: Block diagram of model-aided (and traditional) INS. Additional velocity measurements are not included in this paper when utilizing the output from the vehicle model, and the other way around when using velocity measurements. This is illustrated with a switch/selector. The position measurements may be available often or only sporadically.

where the discrete time index k is dropped for simplicity. As is standard for INS, the calculated velocity is $\check{\boldsymbol{z}}_{vel} = \check{\boldsymbol{v}}^l_{eb}$, which ideally implies that $\boldsymbol{z}_{vel} = \tilde{\boldsymbol{v}}^l_{eb}$, where the accent $\tilde{(\cdot)}$ denotes a measured quantity. In case of using the output from the vehicle model this is not the case, and the best we can to is to let

$$\boldsymbol{z}_{vel} := \check{\boldsymbol{R}}^l_b \check{\boldsymbol{v}}^b_{wb} + \check{\boldsymbol{v}}^l_{ew}, \tag{11}$$

which after substitution in (10) yields the expression

$$\boldsymbol{\delta z} = \check{\boldsymbol{R}}^l_b \check{\boldsymbol{v}}^b_{wb} + \check{\boldsymbol{v}}^l_{ew} - \check{\boldsymbol{v}}^l_{eb}. \tag{12}$$

The variables $\check{\boldsymbol{v}}^l_{eb}$ and $\check{\boldsymbol{R}}^l_b$ stem from the INS, $\check{\boldsymbol{v}}^b_{wb}$ is given by the vehicle model, and $\check{\boldsymbol{v}}^l_{ew}$ can, for instance, be calculated from empirical tide or current tables. If the current was measured it could be used in place of $\check{\boldsymbol{v}}^l_{ew}$. In this paper we assume that $\check{\boldsymbol{v}}^l_{ew} = \boldsymbol{0}$, which is to say that our best a priori guess of the current is zero. It does not mean that the true current is zero. Since the model does not include the water-relative velocity in heave as a state, this model output will be assumed to be zero. The inclusion of a depth sensor in the navigation system is presumed to compensate for this simplification.

A true variable is given as the sum of its calculated value and a corresponding error (similarly for a measured quantity), that is,

$$(\cdot) = \check{(\cdot)} + \delta(\cdot) \quad \text{or} \quad (\cdot) = \tilde{(\cdot)} + \delta(\cdot). \tag{13}$$

Replacing the current velocity and the vehicle model velocity in (12) with their errors and true values yields

$$\boldsymbol{\delta z} = \check{\boldsymbol{R}}^l_b(\boldsymbol{v}^b_{wb} - \boldsymbol{\delta v}^b_{wb}) + (\boldsymbol{v}^l_{ew} - \boldsymbol{\delta v}^l_{ew}) - \check{\boldsymbol{v}}^l_{eb}, \tag{14}$$

which after some manipulation and first order approximations leads to the final expression of the measurement equation associated with the vehicle model

$$\boldsymbol{\delta z}_{vel} = \boldsymbol{\delta v}^l_{eb} - \boldsymbol{S}(\check{\boldsymbol{v}}^l_{eb})\boldsymbol{e}^l_{lb} - \check{\boldsymbol{R}}^l_b \boldsymbol{\delta v}^b_{wb} - \boldsymbol{\delta v}^l_{ew}, \tag{15}$$

where the variable $\boldsymbol{e}^l_{lb}$ is a measure of the calculation error in $\check{\boldsymbol{R}}^l_b$ (Gade, 1997). The variables in (15) are all calculated by the INS or included in the KF process equation. In this work, we assume that the vehicle model output error $\boldsymbol{\delta v}^b_{wb}$ can be modeled as white noise. A more advanced error description is to be implemented in further work. As for the current $\boldsymbol{\delta v}^l_{ew}$, it is modeled as the sum of colored noise and white noise. The colored noise is implemented as a 1. order Markov process driven by white noise Gelb (1974). The vector entries of $\boldsymbol{\delta v}^b_{wb}$ are assumed uncorrelated. Similarly for $\boldsymbol{\delta v}^l_{ew}$. Finally note that the KF estimate of $\boldsymbol{\delta v}^l_{ew}$ is also an estimate of the true current, since $\check{\boldsymbol{v}}^l_{ew} = \boldsymbol{0}$ by assumption, and consequently, $\boldsymbol{v}^l_{ew} = \boldsymbol{\delta v}^l_{ew}$.

4 Experimental Setup

Navigation data collected by the field-deployed HUGIN 4500 AUV are used for evaluating the performance of the model-aided INS proposed in Section 3. An overview of vehicle particulars is given subsequently, followed by a description of the conducted experiments.

4.1 Vehicle Specifications

The Kongsberg Maritime HUGIN 4500 is the latest member of the HUGIN AUV family. Figure 4 shows a picture from one of the sea-trials in September 2006.

The length of the vehicle is approximately 6.5 m and the maximum diameter is 1 m. This gives a nominal dry mass of 1950 kg. Designed for large depths and long endurance, the vehicle can operate for 60-70 hours at depths down to 4500 meters. The cruising speed of the vehicle is about 3.7 knots or 1.9 m/s. The vehicle is passively stable in roll and close to neutrally buoyant.

For propulsion, the vehicle is fitted with a single three-bladed propeller. A cruciform tail configuration with four identical control surfaces is used for maneuvering. The vehicle can operate in either UUV (unmanned

Figure 4: The HUGIN 4500 AUV during sea-trial.

underwater vehicle) or AUV mode. In AUV mode the vehicle maneuvers without supervision, and independently of the mother ship. In UUV mode the vehicle operates near the mother ship, hence enabling real-time supervision. The data used in this paper were collected while operating in UUV mode.

HUGIN 4500 is equipped with a traditional aided INS. Some IMU specifications are listed in Table 2. In UUV mode the surface ship tracks the submersible with an ultra short baseline acoustic position system (USBL). By combining DGPS with USBL, a global position estimate can be obtained, which is then transmitted to the AUV. Additional navigation sensors include compass, pressure sensor, and Doppler velocity log (DVL). Primary aiding sensors and some of their specifications are listed in Table 3. A schematic outline of the integrated navigation system is shown in Figure 3. Readers are referred to Gade (2004); Jalving et al. (2003a,b) for additional information on the navigation system and navigation system accuracy.

4.2 Experimental Description

During September and October 2006, several sea-trials were conducted with the HUGIN 4500 in the vicinity of 59° 29’ N, 10° 28’ E, in the Oslo-fjord, Norway. More than 60 hours of data were collected, of which roughly 3 hours are utilized in this paper. The test area and the horizontal vehicle trajectory are shown in Figure 5. The vehicle followed a standard lawn-mover pattern, typical for a survey AUV like the HUGIN 4500. During the entire run the vehicle was kept at a close to constant depth at 140 meters. Note that no parts of the experimental data used herein were used during the development process of the vehicle model.

4.3 Data Post-Processing

The raw data collected by the HUGIN 4500 were post-processed before being utilized in this paper. The first steps involved wild-point filtering of the position measurements. The HUGIN navigation system then reprocessed the data to get real-time estimates from the KF (this is done using a true copy of the at-sea navigation system). The data were finally smoothed to enhance accuracy. All these steps were done using NavLab (Gade, 2004) and without generating any artificial data. The accuracy of the smoothed vehicle position was estimated to be 0.75 meters (1σ) north and east. The experimentally proven accuracy of the navigation system is thoroughly discussed in Section 5.2.2 of Gade (2004). The smoothed data collected with the vehicle configuration described in Section 4.1 serve as the basis for evaluating the performance of the traditional and model-aided INS. NavLab is also used during the evaluation process in Section 5.

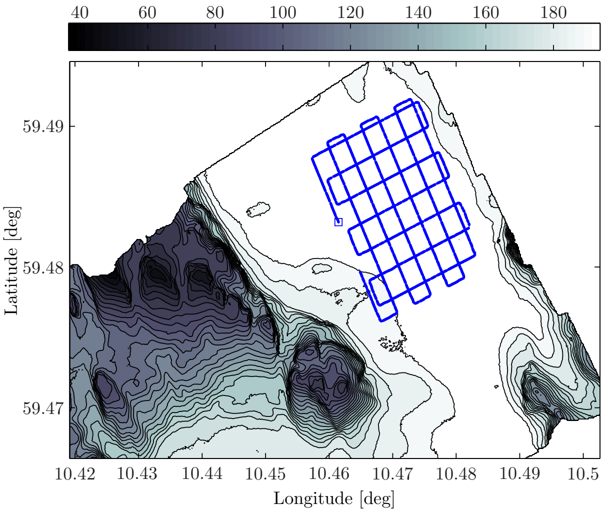


Figure 5: Test area and outlier-filtered HUGIN position measurements, logged topside at 1/3 Hz. The square shows the start position.

5 Experimental Evaluation

This section evaluates the performance of the model-aided INS discussed in Section 3. The performance is compared to the traditional aided INS. With exception of the tuning parameters associated with the vehicle model, all the KF parameters are identical. Depth sensor and compass are always included as aiding sensors. The compass is however given a large covariance and is consequently weighted insignificantly in the KF. The position measurements are available either as logged topside at about 1/3 Hz, or as received onboard the AUV at about 1/30 Hz. External velocity measurements are absent. The position error is taken as the difference between the local position in the basis data

Table 2: IMU specifications

Model	Gyro Technology	Gyro Bias	Accelerometer Bias
IXSEA IMU90	Fiber optic	±0.05°/h	±500 μg

Table 3: Primary navigation aiding sensors

Variable	Sensor	Precision	Rate
Position	Kongsberg HiPAP	Range, Angle: < 20 cm, 0.12°	Varying*
Velocity	RDI WHN 300	±0.4 % ± 0.2 cm/s	1 Hz
Pressure	Paroscientific	0.01 % full scale	1 Hz

* Approximately 1/3 Hz. In real-time position updates are received at about 1/30 Hz, from the surface vessel via an acoustic link.

and the local position estimated by the navigation system under consideration. The navigation systems are evaluated according to the following two cases:

5.0.1 Topside position fix with dropout

The scenario is best illustrated in Figure 6(a), where the vehicle starts at the same initial position as the basis data. The real-time KF receives position measurements at topside rate for about 83 minutes. The position aiding is then disabled for 30 minutes, before again being enabled for the remaining of the survey. This experiment was done in order to evaluate the performance of the two systems in the case position measurements become unavailable.

5.0.2 Onboard position fix

Similar scenario as before, but with position measurements being available at onboard (AUV-side) rate, and with no extraordinary dropouts. The position fix update rates for the entire run are shown in Figure 7(a). This experiment was done in order to evaluate the performance of the two navigation systems in the case were position measurements are available at a reduced frequency.

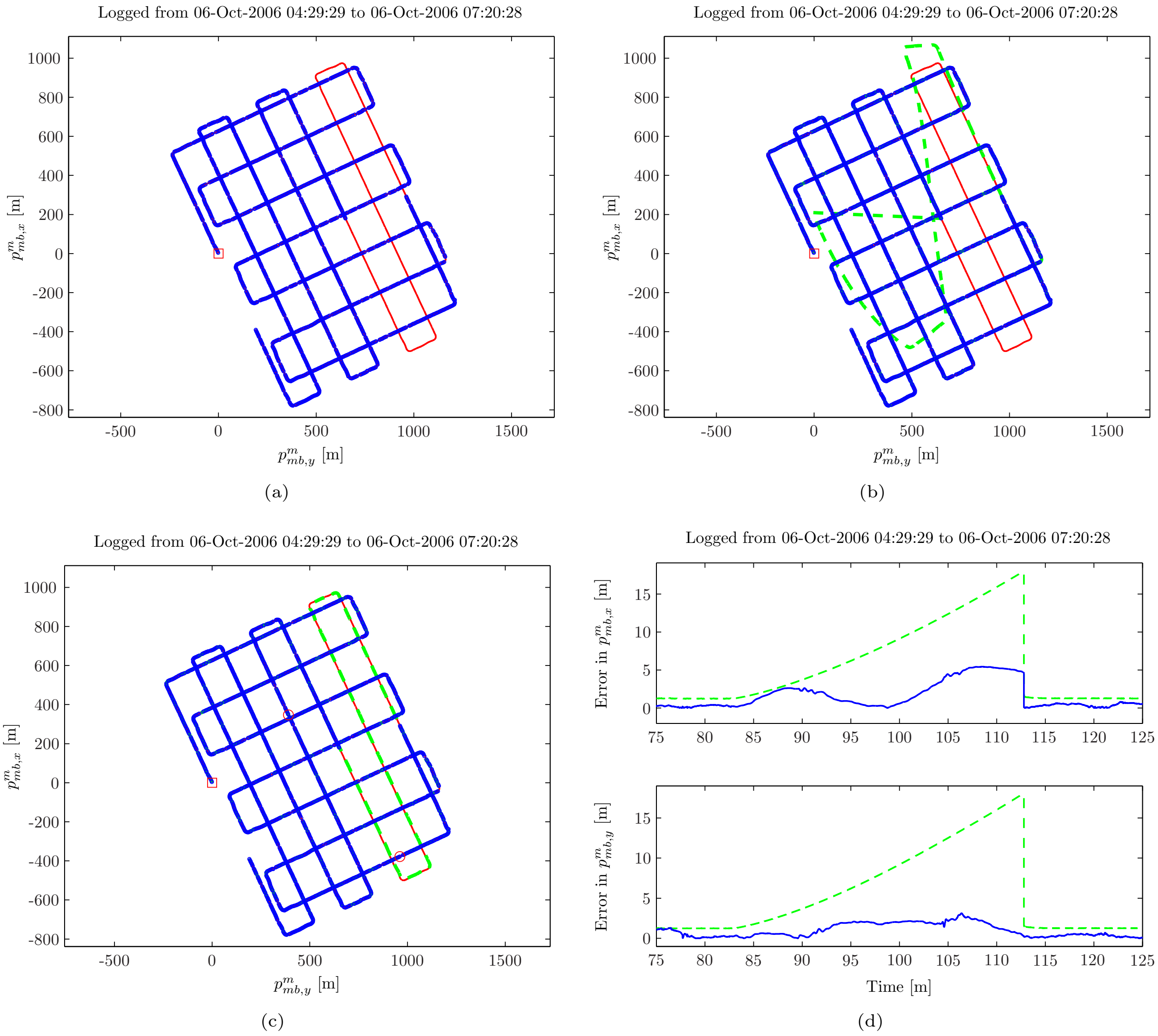
5.1 Navigation Performance - Case 1

During the first and last part of the survey, the model-aided INS and the traditional INS are found to perform comparably in terms of calculated position errors. The position uncertainties estimated by the model-aided INS are slightly lower however, and less jagged. For the part without position aiding, the traditional INS breaks down quickly, as can be seen in Figure 6(b) where the maximum Euclidian norm of the position error is close to 700 meters. The model-aided INS continues to perform excellent, and the maximum norm of the position error is 6 meters. From Figure 6(d) this can be seen to be well within the estimated one standard deviation (1σ). The median of the estimated north and east position uncertainties are 1.2 meters. The estimated trajectory is shown in Figure 6(c), closely following the basis data. Overall the model-aided INS performs excellent, and superior to the traditional INS. Note that the navigation accuracy obtained during time slots without position aiding is limited to the accuracy of the KF estimated current. If the current does not vary significantly throughout the time period where position measurements are absent, the navigation accuracy will remain good.

5.2 Navigation Performance - Case 2

As can be seen in Figure 7(b), the two navigation systems provide very different estimates of the position uncertainties. A similar behavior was also observed when using position measurements at topside rate. The position uncertainties estimated by the model-aided INS are clearly lower, and they appear more reliable. The estimates are also much smoother. The beneficial effect of including the vehicle model for aiding the INS is apparent in Figure 7(c), where the tallest spikes for the traditional INS correspond to approximately 60 seconds since receiving the preceding position measurement.

In terms of position errors the model-aided INS again performs excellent, and well within one standard deviation (1σ) as seen in Figure 7(d). The traditional INS also performs acceptable in terms of position errors, and comparable to the model-aided INS when the position measurements appear frequently. As mentioned earlier, the drift without external position or velocity aiding is not linear, and the performance of the traditional INS worsens when the position update frequency changes from 1/30 Hz to 1/60 Hz. We conclude that the model-aided INS performs better than the traditional INS during time periods without position aiding, and it provides better error covariance estimates throughout.



(a)

(b)

(c)

(d)

Figure 6: Traditional and model-aided INS evaluated according to case 1: (a) The red (solid) trajectory serves as basis for evaluating the navigation systems. The red square shows the initial position used in the KF. The blue (o) data show wild-point filtered position measurements logged topside. The segment without position measurements corresponds to 30 minutes. (b) Real-time navigation solution obtained with traditional INS shown in green (dashed). Other data as before. The system shows poor performance without position measurements. (c) Real-time navigation solution obtained with model-aided INS shown in green (dashed). Other data as before. The system shows excellent performance, also without position measurements. The circles (red) indicate 75 and 125 minutes into the run. (d) The true position errors (assuming basis is correct) for the model-aided INS in north and east are shown in blue (solid). The corresponding estimated real-time KF position uncertainties (1σ) are shown in green (dashed).

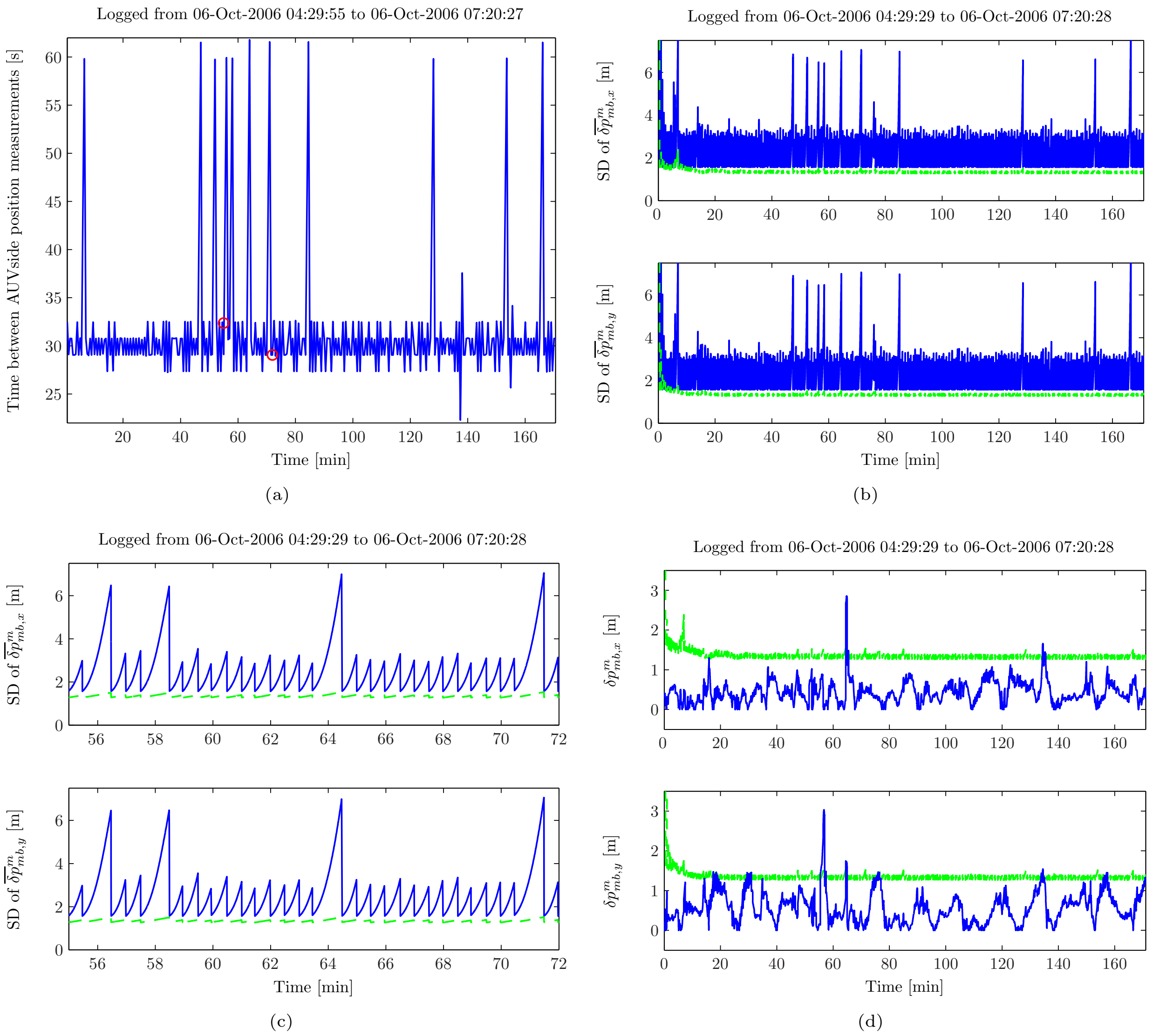


Figure 7: Traditional and model-aided INS evaluated according to case 2: (a) The blue (solid) line shows the time between position measurements, as received by the AUV. The basis trajectory is the same as in Figure 6(a). The circles indicate 54 and 72 minutes into the run. (b) The estimated real-time position uncertainties (1σ) for the model-aided INS are shown in green (dashed). The estimated real-time position uncertainties (1σ) for the traditional INS are shown in blue (solid). (c) Magnified version of Figure 7(b). The model-aided INS provides smoother estimates than the traditional INS. (d) The estimated real-time position uncertainties (1σ) for the model-aided INS are shown in green (dashed). The true position errors (assuming basis is correct) for the model-aided INS in north and east shown in blue (solid).

6 Conclusions and Further Work

This paper reports the development of a model-aided INS for underwater vehicle navigation. The navigation system is novel in that accurate knowledge of the vehicle dynamics is utilized for aiding the INS, and the navigation performance is experimentally evaluated using real AUV data. It is found that the error in the model-aided INS position estimate is significantly lower than that of the traditional INS throughout time segments where position and velocity measurements are absent. The model-aided INS also performs equally good or better than the traditional INS in cases with regular position updates, and the difference in performance increases with decreasing position update rate. The experimental results demonstrate that with merely an addition of software and no added instrumentation, it is possible to considerably improve the accuracy and robustness of an INS by utilizing the output from a kinetic vehicle model. To the best of our knowledge, the presented results are the first report on the implementation and experimental evaluation of model-aided INS for underwater vehicle navigation. The conclusion has an important practical consequence, and the proposed approach shows promise to improve underwater navigation capabilities both for systems lacking disparate velocity measurements, and for systems where the need for redundancy and integrity is important.

6.1 Further Work

A more advanced error description of the vehicle model output may be implemented, and observability conditions for the vehicle model error and the sea current should be investigated. A model-aided INS utilizing both external velocity measurements and vehicle model output is of great practical interest, and should be implemented. This is subject to ongoing research.

Acknowledgment

The first author acknowledges the financial support from the Research Council of Norway, Kongsberg Maritime, and Kongsberg Defence & Aerospace, through the UNaMap program. Thanks also the Kongsberg Maritime HUGIN engineers and the crew of M/S Simrad Echo for all their help during the experiments.

References

Bryson, M. and Sukkarieh, S. Vehicle model aided inertial navigation for a UAV using low-cost sensors. In *Australasian Conference on Robotics and Automation.* Canberra, Australia, 2004 .

Fossen, T. I. *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles.* Marine Cybernetics, 2002.

Gade, K. Integrering av treghetsnavigasjon i en autonom undervannsfarkost (in Norwegian). Technical Report FFI/RAPPORT-97/03179, Norwegian Defence Research Establishment (FFI), 1997.

Gade, K. NavLab, a generic simulation and post-processing tool for navigation. *European Journal of Navigation*, 2004. 2(4):21–59. (see also http://www.navlab.net/).

Gelb, A. *Applied Optimal Estimation.* The MIT Press, 1974.

Hegrenæs, Ø. Open-water propulsion test using full-scale propellers for the HUGIN AUV family. Technical report, University Graduate Center, Kjeller, Norway, 2006.

Hegrenæs, Ø., Hallingstad, O., and Jalving, B. A comparison of mathematical models for the HUGIN 4500 AUV based on experimental data. In *Proceedings of the IEEE International Symposium on Underwater Technology (UT).* Tokyo, Japan, 2007a .

Hegrenæs, Ø., Hallingstad, O., and Jalving, B. A framework for obtaining steady-state maneuvering characteristics of underwater vehicles using sea-trial data. In *Proceedings of the 15th IEEE Mediterranean Conference on Control and Automation (MED).* Athens, Greece, 2007b .

Jalving, B., Gade, K., Hagen, O., and Vestgård, K. A toolbox of aiding techniques for the HUGIN AUV integrated inertial navigation system. In *Proceedings of the MTS/IEEE Oceans Conference and Exhibition.* San Diego, CA, 2003a .

Jalving, B., Vestgård, K., and Størkersen, N. Detailed seabed surveys with AUVs. In G. Griffiths, editor, *Technology and Applications of Autonomous Underwater Vehicles*, volume 2, pages 179–201. Taylor & Francis, 2003b.

Kinsey, J. C., Eustice, R. M., and Whitcomb, L. L. A survey of underwater vehicle navigation: Recent advances and new challenges. In *Proceedings of the 7th IFAC Conference of Manoeuvring and Control of Marine Craft (MCMC).* Lisbon, Portugal, 2006 .

Kinsey, J. C. and Whitcomb, L. L. Model-based nonlinear observers for underwater vehicle navigation: Theory and preliminary experiments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).* Rome, Italy, 2007 .

Koifman, M. and Bar-Itzhack, I. Inertial navigation system aided by aircraft dynamics. *IEEE Transactions on Control Systems Technology*, 1999. 7(4):487–493.

Refsnes, J. E., Sørensen, A. J., and Pettersen, K. Y. A 6 DOF nonlinear observer for AUVs with experimental results. In *Proceedings of the 15th IEEE Mediterranean Conference on Control and Automation (MED)*. Athens, Greece, 2007 .

SNAME. Nomenclature for treating the motion of a submerged body through a fluid. Technical Report Technical and Research Bulletin No. 1-5, The Society of Naval Architects and Marine Engineers, 1950.

Vasconcelos, J. F., Silvestre, C., and Oliveira, P. Embedded vehicle dynamics and LASER aiding techniques for inertial navigation systems. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*. Keystone, CO, 2006 .

Paper VIII

Hegrenæs, Ø., Gade, K., Hagen, O. K. and Hagen, P. E. (2009). Underwater Transponder Positioning and Navigation of Autonomous Underwater Vehicles. *Proceedings of the IEEE Oceans Conference*, Biloxi, MS, USA

Underwater Transponder Positioning and Navigation of Autonomous Underwater Vehicles

Øyvind Hegrenæs*, Kenneth Gade†, Ove Kent Hagen†, and Per Espen Hagen*

*Kongsberg Maritime Subsea, AUV Department, NO-3191 Horten, Norway
†Norwegian Defence Research Establishment, NO-2027 Kjeller, Norway

Abstract— **Navigation of underwater vehicles has been and remains a substantial challenge to all platforms. The need for improved accuracy and robustness, sustainability, and de-risking develops with the emergence of new applications, and with the growing acceptance of autonomous underwater vehicles (AUVs) in both military and civilian institutions. One of the main driving factors is the ability to carry out long-duration missions fully autonomous and without supervision from a surface ship. Combined with inertial navigation, the use of one or several transponders on the seabed is an accurate and cost-effective approach toward solving several of these challenges. The principle discussed in this paper is called underwater transponder positioning (UTP), and requires only one transponder due to tight coupling with the inertial navigation system (INS). For many scenarios UTP may be a better alternative than using a long baseline (LBL) system.**

This paper reports in-situ and post-processed navigation results obtained with a state-of-the-art UTP aided INS, onboard a HUGIN 1000 AUV. The results demonstrate the feasibility of UTP in large-scale autonomous operations. Excellent real-time navigation is achieved, and the accuracy obtained in post-processing is shown to be close to that obtained when aiding the INS with an ultra-short baseline (USBL) positioning system.

I. Introduction

After two decades of dedicated research and development, autonomous underwater vehicles (AUVs) are today becoming accepted by an increasing number of users in both military and civilian institutions. The number of AUV systems sold worldwide is well into triple digits. The bulk of these systems have been manufactured within the last five years, so the sector is in rapid growth. Next to improved payload quality and endurance, much of this success is due to recent advances in navigation sensor technologies and fusion algorithms [1], [2], [3]. Despite these achievements, navigation remains a substantial challenge to all submersibles. The actual autonomy of the vehicles in existence today is also limited. Further advances in both areas will enable new operations which earlier have been considered infeasible, or at best difficult.

This paper is concerned with inertial navigation of AUVs, with particular focus on tight integration of range measurements from one or several transponders. The concept, called underwater transponder positioning (UTP), is available to all the Kongsberg Maritime HUGIN AUVs. The first at-sea demonstration of single transponder UTP aided inertial navigation was carried out in 2003, as described in [2]. This paper is a continuation to this work, incorporating multiple transponders, deployed far apart. An overview of additional inertial navigation system (INS) aiding tools available to HUGIN and other AUVs is found in [3]. While the use of transponders is not completely self-governing (requires deployment), it allows for truly autonomous operations once deployed on the seabed (upcoming transponders have a battery capacity of five years). Typical applications of UTP include:

- ◇ Pipeline inspection and intervention.
- ◇ Under ice surveys (transponders may be deployed from an ice breaker or along the ice ridge).
- ◇ Scenarios where repeated dives in an area are required.
- ◇ Autonomous surveys where surfacing or support from a surface ship are infeasible due to e.g. heave traffic.

UTP is also complementary to traditional ultra-short baseline (USBL) and long baseline (LBL) positioning. Compared to an LBL system, UTP has improved accuracy due to tight coupling with the INS, increased operating area and significantly less deployment cost, since only one transponder is required.

Data from a HUGIN 1000 AUV mission prove the in-situ real-time navigation performance of the UTP aided INS. HUGIN navigated autonomously with UTP as the only position aiding for roughly 8 h, much of the time without. The data analysis also include a comparison with conventional USBL aided INS, as well as results showing the accuracy enhancement obtained by using NavLab [4] in post-processing.

For the remainder, UTP-INS is used for short when discussing UTP aided INS without distinguishing on accompanying sensors. Additional notation is appended when discussing the integration of specific aiding sensors such as USBL and Doppler velocity log (DVL). Pressure sensor data are always present. This paper is furthermore organized as follows. The remainder of this section reviews inertial navigation of underwater vehicles. Section II describes some of the principles of UTP, as well as operational procedures. The experimental setup is described in Section III, followed by an experimental evaluation of the proposed navigation system in Section IV.

A. *Underwater Vehicle Inertial Navigation*

An INS calculates position, velocity and attitude using high frequency data from an inertial measurement unit (IMU) which typically consists of three accelerometers measuring specific force and three gyros measuring angular rate, all relative to the inertial space. Due to inherent errors in the gyros and accelerometers, the solution of the navigation equations

0-933957-38-1/09/$20.00 ©2009 MTS

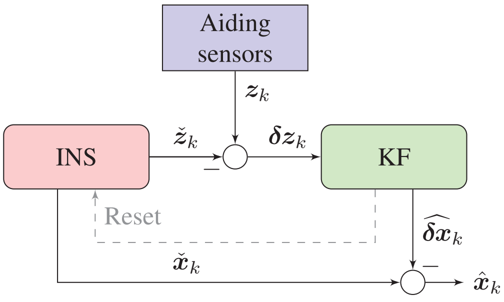


Fig. 1. Outline of conventional aided INS.

embedded in the INS will have an unbounded drift unless counteracted. A performance measure for an INS is given by its pure inertial drift in position, where the divergence rate depends on the IMU quality. A navigation grade INS drifts in the order of one nautical mile per hour. Since an INS is a diverging system, an aiding framework is needed to limit or reduce the error growth. An overview of INS aiding tools is given in [2], [3]. For autonomous missions it may be important to retain good navigation accuracy between position updates, which will usually be sparse. The use of bottom-track data from a DVL is today the most common approach. See Section I-B for additional information regarding DVL-INS.

In order to fuse the data from the INS and the aiding sensors, some form of filtering must be implemented. This is typically accomplished using a Kalman filter (KF). An outline of a conventional aided INS is shown in Fig. 1, where the KF input is taken as the difference between the output from the appropriate aiding sensors and the INS. A perturbation method is used in this paper for deriving the INS error states. The states are included in the KF with the assumptions of small errors, i.e. first order approximation. The KF also estimates the colored errors of the navigation sensors. The INS and aiding sensors considered in this paper are shown in Fig. 2.

B. DVL-INS

In many practical situations position measurements will be unavailable for extended periods of time and the INS will then chiefly depend on external velocity aiding. While alternatives exist (see e.g. [5], [6], [7]), the application of DVL with bottom-track is predominant. If within sensor range, the DVL measures the vehicle linear velocity relative to the seabed along four acoustic beams. Data obtained from a minimum of three beams are combined in order to calculate the velocity.

The amount of literature on error sources in DVL based navigation is extensive. For DVL-INS the horizontal position drift is determined by the error in the estimated Earth-fixed velocity. The main contributors are body-fixed velocity error, and heading error. The error in estimated body-fixed velocity is mainly determined by the low-frequency errors of the DVL itself (e.g. alignment and speed of sound scaling). These errors are not observable if the vehicle is traveling along a straight line and without position aiding. High frequency velocity errors are on the other hand estimated by means of the IMU. As for the error in heading, it is determined by the gyrocompassing capability of the integrated system. The heading estimation error will typically be of low frequency, corresponding to non-observable gyro bias dynamics.

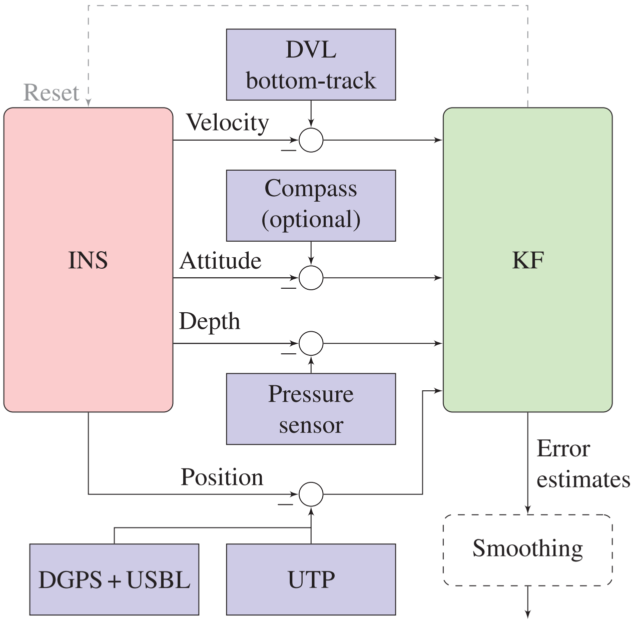


Fig. 2. The integrated INS and aiding sensors considered in this paper. A large toolbox of additional aiding sources is available to the HUGIN AUVs.

In order to minimize the drift in the DVL-INS, the heading estimate should be properly initialized prior to launch or before carrying out an autonomous mission with sparse position aiding. It is also vital that the misalignment between the DVL and the IMU is compensated for. Sound speed scaling may be reduced by using an external CTD sensor. As mentioned, a KF can also compensate for part of the DVL errors when running more complex survey patterns, or when position updates are available. The reader is referred to [2] for a further discussion.

Examples of in-situ DVL-INS accuracy obtained by the HUGIN 1000 AUVs are shown in Fig. 3. The real-time navigation system onboard HUGIN is called NavP (navigation processor). In each of the dives HUGIN ran along two straight lines in opposite direction, each roughly 7 km in length. Similar verification trials prove a NavP navigation accuracy in the order of 0.1% of distance traveled (or better) when running without position aiding along a straight line. Depending on the application, a low in-situ DVL-INS drift may be imperative to mission success. A low real-time drift is also important when utilizing the UTP range measurements and when traveling between the transponders, as discussed in Section II. Note also that the accuracy and robustness may be further enhanced in NavLab post-processing (see Section III-C for further details).

II. Underwater Transponder Positioning

LBL and USBL acoustic positioning are both well known principles which today are used routinely in a number of applications, including underwater vehicle INS aiding. When operating within an LBL network the vehicle interrogates

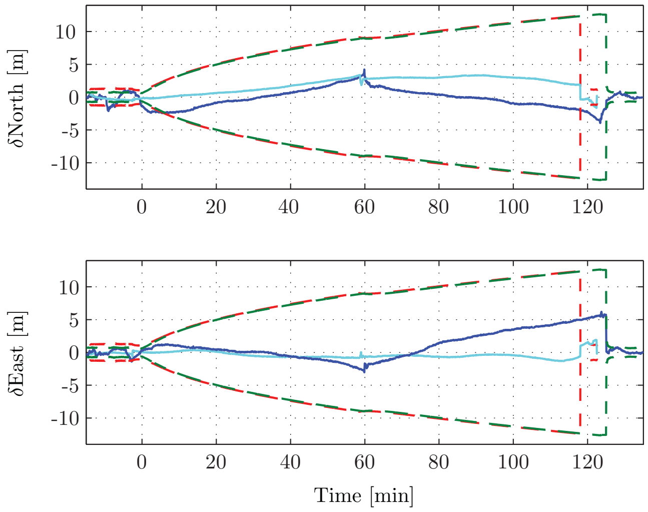


Fig. 3. Proven in-situ navigation performance of DVL-INS on HUGIN 1000 AUVs when traveling along two reciprocal lines, each about 7 km. The blue and cyan (solid) data show the north and east position errors from two dives. The KF real-time uncertainties (1σ) are shown in red and green (dashed). The positioning aiding was turned off at time 0 and remained off for about 2 h (topside logging of DGPS-USBL position measurements continued at 1/3 Hz).

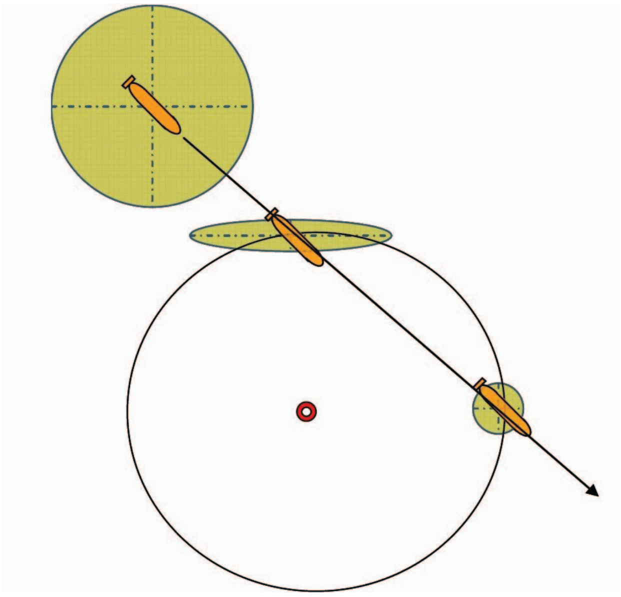


Fig. 4. Principle and effectiveness of UTP-INS. The shaded areas are examples of horizontal covariance ellipses, initially compressed radially.

the transponders, and the replies are used for calculating the range to each them. If the geographical position of each transponder is known, a unique position can be computed by triangulation (three or more transponders are needed if AUV and transponder depths are known). As for USBL, a typical approach is to measure the range and bearing of a transponder on the underwater vehicle relative to a transducer mounted on a surface vessel. This is e.g. the case when using Kongsberg Maritime HiPAP together with the HUGIN AUVs. A global position measurement, which may be transmitted to the submersible using an acoustic link, can be obtained by combining surface ship GPS and USBL measurements.

Within the last decade, an increasing number of single transponder systems have been proposed as alternatives to LBL and USBL. The growing interest is in large part due to the significant logistics and calibration involved when establishing an LBL network. Also, following an AUV with surface ship USBL may not always be feasible. On the technical side, the usage of single transponder navigation (range aiding) has been made possible due to improved dead-reckoning capabilities, e.g. navigation accuracy of DVL-INS. Further details on single transponder range aiding and UTP is given subsequently.

Common to the systems above is the dependency on two-way travel time (TWTT). An alternative in single-range navigation is to explore synchronous-clocks for direct measurement of one-way travel time (OWWT). See e.g. [8] for details.

A. UTP-INS

From a navigation point of view a single range transponder may be thought of as an underwater lighthouse providing the AUV with ranges relative to its fixed geographical location. By fusing this information, a global position is obtained which may be used for aiding the INS. In UTP this is done in tight integration with the INS. In contrast, most LBL aiding schemes are loosely coupled. In-depth simulations (including full acoustic modeling, ray-tracing, line-of-sight considerations) were carried out in [9] for pipeline surveying and touchdown monitoring on the Ormen Lange gas-field in the Norwegian Sea. For the terrain and environment considered in the report, UTP aided INS was found to be better than LBL aided INS in all aspects since each individual range measurement in UTP is optimally utilized to provide increased accuracy and robustness toward loss of transponder coverage.

Single transponder navigation is not a new concept. As mentioned, the first at-sea demonstration of single transponder UTP aided inertial navigation was carried out in 2003, as described in [2]. This paper is a continuation to this work, incorporating multiple transponders, deployed far apart. Work by other authors on single range navigation include [10], [11], [12], [13], [14], [15], [16], [17]. The majority of the work rely on either least-squares (LS) or Kalman filtering in order to calculate the vehicle position based on one or several ranges.

In principle a range measurement only tells that the vehicle is located somewhere on a circle with the transponder in its center. An innovative algorithm is implemented in UTP for determining the best possible location on the circle and for tightly integrating the range measurements with the INS. As the vehicle passes through the transponder area the algorithm takes advantage of the slow error drift of the DVL-INS (see Section I-B) and the geometry change due to physical vehicle movement. The principle behind UTP is illustrated in Fig. 4.

1) Transponder Deployment and Georeferencing: An example showing two Kongsberg Maritime UTP transponders is shown in Fig. 5. The transponders are equipped with a release mechanism for easy recovery. As for deploying the UTP transponders, standard network design parameters such as

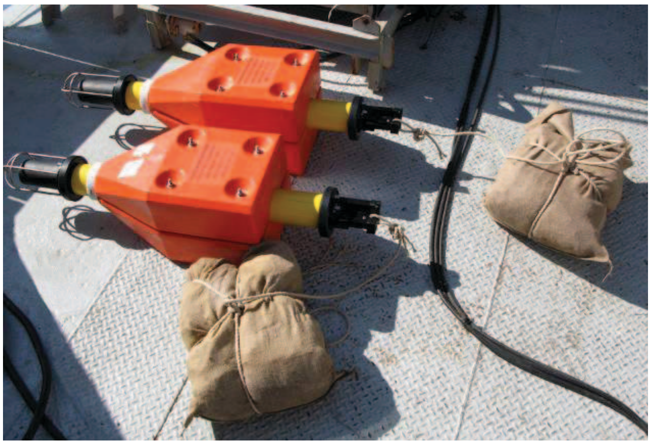


Fig. 5. Two Kongsberg Maritime UTP transponders ready for deployment. Dissolvable sandbags used as weights in this particular mission. The new generation transponders have a battery life capacity up to five years.

maximum range and distance between consecutive transponders (depending on DVL-INS performance and required survey accuracy) must be considered. Also, in order to minimize errors related to speed of sound inaccuracy, the transponder depth should be roughly similar to the operational depth of the AUV. In practice the speed of sound can be measured by a CTD on the vehicle or on the UTP transponder (or both).

For ranging techniques like UTP to work, the geographical location of the transponders must be known or estimated (box-in). The preferred method is to measure the position directly using USBL on a surface ship. Using Kongsberg Maritime HiPAP and assuming a GPS north and east accuracy of 0.4 m (1σ), the Earth-fixed location of the transponder can determined to within 0.6, 1, 2 and 3 m at 500, 1000, 2000 and 3000 m depth, respectively. As the depth decreases the GPS accuracy becomes the dominant source in the error budget. An important observation is that the box-in process will usually be done at the same time as deploying the transponders, hence the additional operational time needed is small. Other box-in methods also exist. A simultaneous localization and mapping (SLAM) approach is e.g. applied in [17] in order to obtain a successively improving estimate as the AUV interrogates the transponder. A limitation of this approach is that the box-in accuracy is lower bounded by the accuracy of the aided INS at the time the box-in process starts, hence making it less suitable to deep water operations (e.g. involving operations above the DVL range) or when transponders are deployed far apart.

2) Operational Procedures: The usage of UTP transponders is not completely self-governing since it requires deployment and box-in. It does however allow for truly autonomous operations once on the seabed, and for a long time due to high battery capacity. The AUV may navigate autonomously with bounded error by visiting the network occasionally.

When running UTP with the HUGIN AUVs, the interrogation of the transponder may be started and stopped both manually and automatically. In Auto mode the interrogation is initiated when operating inside the transponder range, also taking the DVL-INS navigation uncertainty into account.

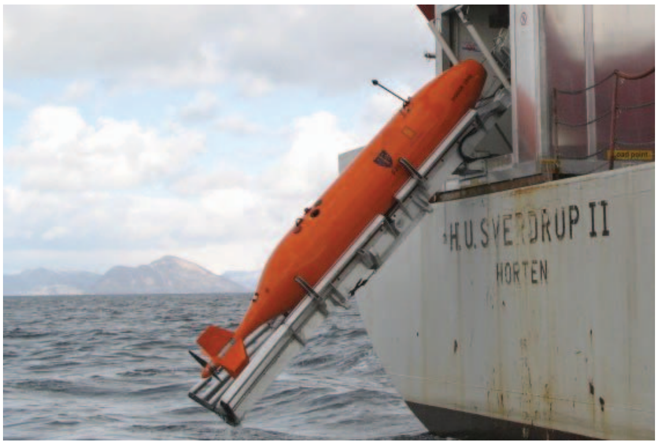


Fig. 6. HUGIN 1000 during recovery onboard H.U. Sverdrup II.

TABLE I
IMU SPECIFICATIONS

	Bias		Scale Factor		
Model	Gyro	Acc	Gyro	Acc	Rate
Honeywell HG9900	0.003 deg/h	25 μg	5 PPM	100 PPM	300 Hz

TABLE II
PRIMARY NAVIGATION AIDING SENSORS

Variable	Sensor	Precision	Rate
Position	Kongsberg HiPAP USBL* Kongsberg UTP	$<$20 cm, 0.12 deg $<$10 cm	Varying† Varying‡
Velocity	RDI DVL 300kHz	$\pm$0.4 %$\pm$0.2 cm/s	$>$1 Hz
Pressure	Paroscientific	0.01 % full scale	1 Hz

* Surface ship GPS and USBL are combined to give a global vehicle position. The accuracy of the final position also depends on the ship GPS precision.

† Depends on the slant range. While submerged, the AUV receives position updates at about 1/30 Hz, from the surface via an acoustic commando link.

‡ Depends on the range from the AUV to the transponder. Usually $>$1/2 Hz.

III. EXPERIMENTAL SETUP

An overview of the experimental setup, including vehicle particulars, employed navigation sensors, mission trajectory, and the processing of raw navigation data, is given in this section. The experimental results are discussed in Section IV.

A. Vehicle Description

The performance and comparison of the integrated INS with and without UTP aiding is evaluated using in-situ (real-time) NavP navigation data and raw sensor data. The data were collected by a HUGIN 1000 AUV with 3000 m depth rating, owned and operated by the Norwegian Defence Research Establishment (FFI). The launch and recovery system and the AUV are shown in Fig. 6. The diameter and length (base version) of the vehicle are 0.75 and 5.3 m. It can operate with full payload for 25 h at a cruising speed of about 2 m/s.

HUGIN 1000 is equipped with an aided INS, as outlined in Fig. 2. Additional aiding tools are also available, but are not discussed any further in this paper. Some IMU specifications are shown in Table I. The primary navigation aiding sensors relevant to the data utilized in this work are listed in Table II.

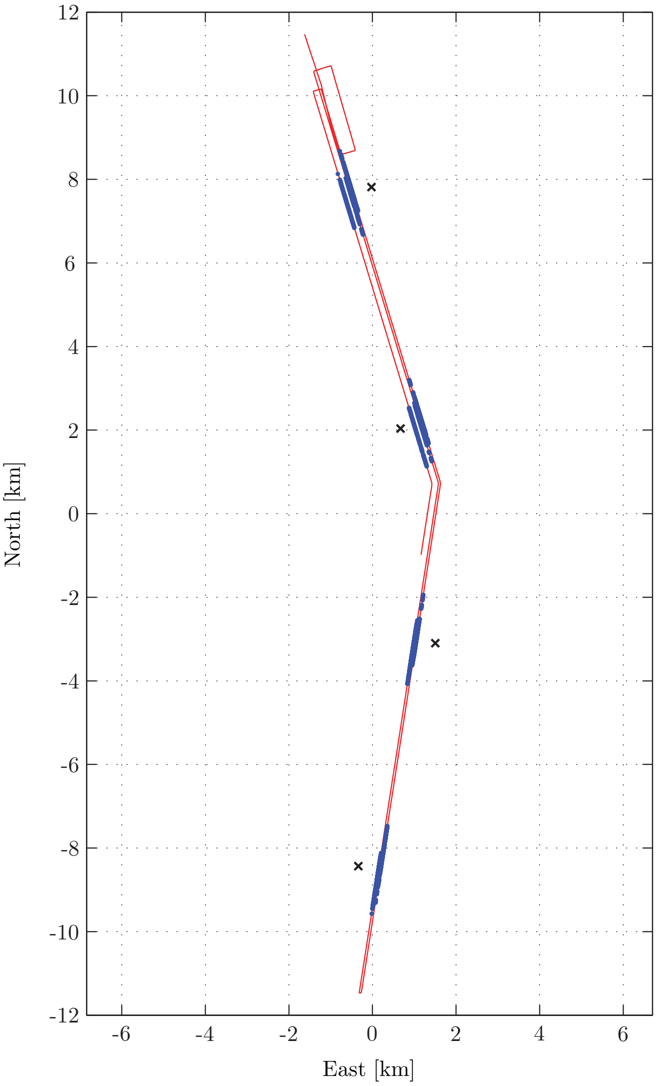


Fig. 7. UTP-INS navigation - August 2008. The true AUV trajectory is shown in red (solid), the positions derived from the range measurements are shown in blue (●), and the UTP transponder locations are shown in black (x).

B. Experiment Description

The data in this paper were collected August 2008 in the northern parts of Norway, close to Tromsø. The vehicle trajectory and transponder locations are shown in Fig. 7. The mission is representative for a pipeline inspection survey.

HUGIN navigated in real-time (NavP) with UTP as the only position aiding tool for roughly 8 h, much of the time without. The AUV occasionally revisited the four UTP transponders, which were deployed about 6 km apart. The transponder depths ranged from 270 to 325 m, and the vehicle height above the seabed was about 15 m throughout the mission. A 30 kHz system was used, which in this mission gave up to 1.2 km practical range for each transponder, as seen in Fig. 8. Lower frequency yields longer range (at the cost of precision). Due to shadow from the AUV hull, the first received ranges appeared at about 600-700 m when approaching the transponders.

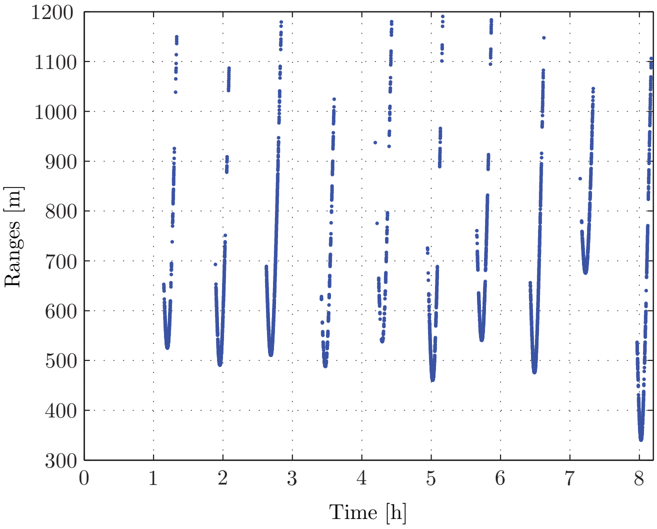


Fig. 8. Practical ranges obtained for the survey trajectory and transponder locations in Fig. 7. A total of 10 passes were done, and ranges up to 1200 m were obtained. When approaching the transponders the first ranges were received later than desirable, and is due to the shadow from the hull. An alternative location of the vehicle mounted transducer is being considered.

C. Data Post-Processing

The real-time solution from NavP is the original navigation data collected at sea. As for the post-processed results, raw navigation sensor values are used throughout. The re-navigation routines are implemented in NavLab [4], a tool which has been used extensively with all the HUGIN AUVs since the late 1990's. In addition to re-navigating the real-time navigation system, NavLab also contains offline smoothing functionality, based on a Rauch-Tung-Striebel (RTS) implementation. The RTS smoother utilizes both past and future sensor measurements and KF covariances, hence efficiently improving the integrity and accuracy of the final navigation solution [18]. In this paper the smoothed USBL-DVL-INS solution with the highest navigation sensor update rates available serves as the reference when evaluating the performance of the UTP-INS. DVL data at 3 Hz and USBL data at about 1/2 Hz were utilized. The accuracy of the RTS smoothed reference position was estimated to be 0.7 m (1σ) in north and east.

IV. Experimental Results

This section evaluates the performance of the UTP-INS described in Section II-A. As mentioned above, the RTS smoothed reference solution serves as the ground truth during comparison. All the results have also been verified by comparing the AUV multi-beam echosounder (MBE) and side-scan (SSS) data with data collected using the surface ship Kongsberg Maritime EM710. Both in-situ real-time results and post-processed results are investigated. An estimation error is taken as the difference between the RTS reference solution and the navigation solution being evaluated.

The in-situ navigation accuracy of NavP is shown in Fig. 9. The maximum horizontal error when running UTP-DVL-INS (completely autonomous) is about 8 m, but for most parts

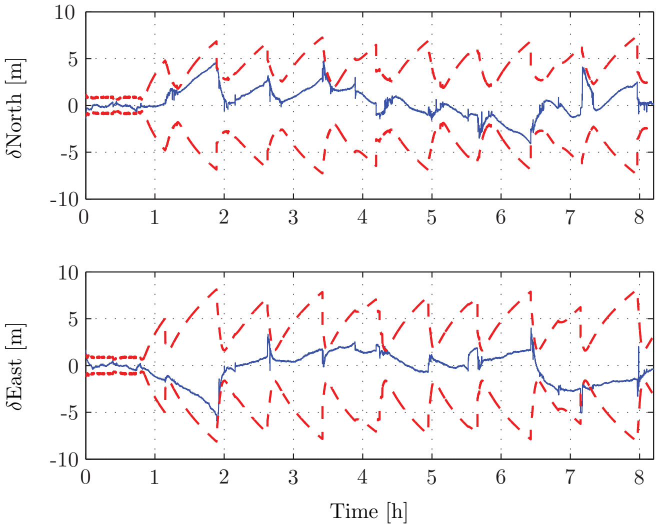


Fig. 9. The blue (solid) data show the north and east position errors of the in-situ UTP-DVL-INS. Kongsberg HiPAP USBL was used during the first 45 min. The real-time uncertainties (1σ) are shown in red (dashed).

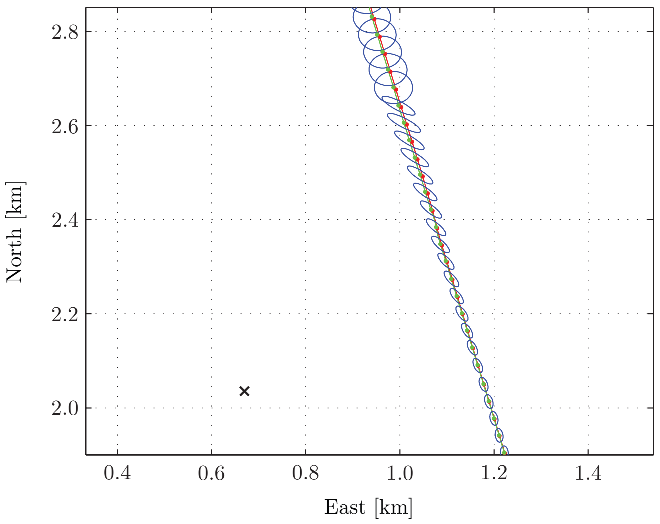


Fig. 10. The true AUV trajectory is shown in red (solid), the trajectory estimated by NavP in real-time is shown in green (solid), and the UTP transponder location is shown in black (x). The axes are the same as in Fig. 7. The blue circles and ellipses are the horizontal covariance matrices, scaled to 5σ for easier display. As seen in Fig. 9, the errors are within 1σ throughout.

within 5 m. The sample mean and standard deviation of the errors are 0.5±1.6 m in north and -0.3±1.6 m in east. During the first 45 min in Fig. 9 USBL aiding was used regularly, hence lowering the uncertainty and error. The use of USBL was mainly done for control purposes and for avoiding any initial position error prior to running autonomously. As for INS initialization, this was done while on deck using GPS and during the very start with USBL and DVL. The misson was carried out north of the polar circle, hence requiring longer time for initialization. The real-time heading uncertainty in NavP was in the order of 0.07 deg at launch and about 0.05 deg when starting the 8 h part running with UTP.

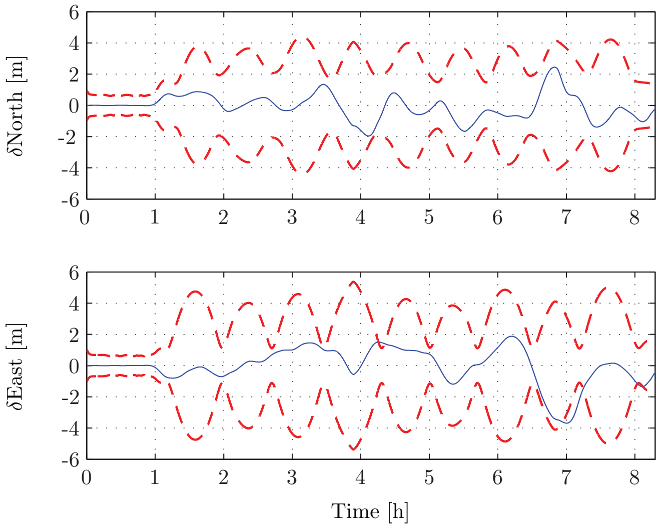


Fig. 11. The blue (solid) data show the north and east position errors of the NavLab post-processed UTP-DVL-INS. Kongsberg HiPAP USBL was used during the first 45 min. The uncertainties (1σ) are shown in red (dashed).

The NavP performance is also shown in Fig. 10 for a small subset of the data. The NavP position covariances are visualized as error ellipses (5σ is used for easier display). As seen in Fig. 9, the north and east errors are well within 1σ most of the time. The same ellipses illustrate the principle and effectiveness of UTP-INS (see also Fig. 4). The error covariance is initially compressed radially, but is also compressed tangentially as the vehicle travels through the UTP zone.

As mentioned in Section III-C it is also possible to further enhance the navigation accuracy by using NavLab. NavLab is particularly effective when position measurements are sparse, as is the case in this mission. The post-processed navigation accuracy is shown in Fig. 11. The maximum horizontal error when running UTP-DVL-INS is now about 5 m, but for most parts within 3 m. The sample mean and standard deviation of the errors are 0.08±0.8 m in north and 0.04±1.14 m in east. The errors are again consistent and within 1σ most of the time. The accuracy obtained in NavLab post-processing is close to the accuracy of the reference solution where USBL was used.

V. Conclusion

This paper has reported the usage of underwater transponder positioning (UTP) and the tight integration with INS. Both in-situ and post-processed navigation results verify that UTP aiding is a feasible and accurate approach for large-scale operations, improving underwater navigation capabilities for systems where the need for flexibility, redundancy and autonomy is important. A strength of UTP is that only one transponder is needed to effectively bind the INS error drift.

References

[1] J. C. Kinsey, R. M. Eustice, and L. L. Whitcomb, "A survey of underwater vehicle navigation: Recent advances and new challenges," in *Proceedings of the 7th IFAC Conference of Manoeuvring and Control of Marine Craft (MCMC)*, Lisbon, Portugal, 2006.

[2] B. Jalving, K. Gade, O. Hagen, and K. Vestgård, “A toolbox of aiding techniques for the HUGIN AUV integrated inertial navigation system,” in *Proceedings of the MTS/IEEE Oceans Conference and Exhibition*, San Diego, CA, 2003, pp. 1146–1153.

[3] P. E. Hagen, Ø. Hegrenæs, B. Jalving, Ø. Midtgaard, M. Wiig, and O. K. Hagen, “Making AUVs truly autonomous,” in *Underwater Vehicles*. Vienna: I-Tech Education and Publishing, January 2009.

[4] K. Gade, “NavLab, a generic simulation and post-processing tool for navigation,” *European Journal of Navigation*, vol. 2, no. 4, pp. 21–59, Nov. 2004, (see also http://www.navlab.net/).

[5] R. E. Hansen, T. O. Sæbø, K. Gade, and S. Chapman, “Signal processing for AUV based interferometric synthetic aperture sonar,” in *Proceedings of the MTS/IEEE Oceans Conference and Exhibition*, vol. 5, Sep. 22–26, 2003, pp. 2438–2444.

[6] Ø. Hegrenæs, E. Berglund, and O. Hallingstad, “Model-aided inertial navigation for underwater vehicles,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, May 2008, pp. 1069–1076.

[7] Ø. Hegrenæs and E. Berglund, “DVL water-track aided inertial navigation for autonomous underwater vehicles,” in *Proceedings of the IEEE Oceans Conference and Exhibition*, Bremen, Germany, May 2009.

[8] R. M. Eustice, L. L. Whitcomb, H. Singh, and M. Grund, “Experimental results in synchronous-clock one-way-travel-time acoustic navigation for autonomous underwater vehicles,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Rome, Italy, Apr. 2007, pp. 4257–4264.

[9] O. K. Hagen and B. Jalving, “LBL and UTP navigation in Ormen Lange scenario,” Norwegian Defence Research Establishment, Tech. Rep. FFI/RAPPORT-2006/00314, 2006.

[10] A. P. Scherbatyuk, “The AUV positioning using ranges from one transponder LBL,” in *Proceedings of the MTS/IEEE Oceans Conference and Exhibition*, vol. 3, San Diego, CA, USA, Oct. 9–12, 1995, pp. 1620–1623.

[11] M. B. Larsen, “Synthetic long baseline navigation of underwater vehicles,” in *Proceedings of the MTS/IEEE Oceans Conference and Exhibition*, vol. 3, Providence, RI, USA, 2000, pp. 2043–2050.

[12] J. Vaganay, P. Baccou, and B. Jouvencel, “Homing by acoustic ranging to a single beacon,” in *Proceedings of the MTS/IEEE Conference and Exhibition*, vol. 2, Providence, RI, USA, 2000, pp. 1457–1462.

[13] P. Baccou and B. Jouvencel, “Homing and navigation using one transponder for AUV, postprocessing comparisons results with long base-line navigation,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, 2002, pp. 4004–4009.

[14] A. Gadre and D. Stilwell, “A complete solution to underwater navigation in the presence of unknown currents based on range measurements from a single location,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada, 2005, pp. 1420–1425.

[15] C. E. G. LaPointe, “Virtual long baseline (VLBL) autonomous underwater vehicle navigation using a single transponder,” Master’s thesis, Massachusetts Institute of Technology, 2006.

[16] P. M. Lee, B. H. Jun, K. Kim, J. Lee, T. Aoki, and T. Hyakudome, “Simulation of an inertial acoustic navigation system with range aiding for an autonomous underwater vehicle,” *IEEE Journal of Oceanic Engineering*, vol. 32, no. 2, pp. 327–345, Apr. 2007.

[17] E. Willemenot, P.-Y. Morvan, H. Pelletier, and A. Hoof, “Subsea positioning by merging inertial and acoustic technologies,” in *Proceedings of the IEEE Oceans Conference and Exhibition*, Bremen, Germany, 2008.

[18] A. B. Willumsen and Ø. Hegrenæs, “The joys of smoothing,” in *Proceedings of the IEEE Oceans Conference and Exhibition*, Bremen, Germany, 2009.