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

MILJØMERKET
241 TRYKKSAK 731

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 UUVs. *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

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

$\boldsymbol{x}^{\underset{\rightarrow}{A}}$ 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 $x_{\underline{AB}}$.

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 $\underset{\rightarrow}{A}$, $\boldsymbol{p}_{\underset{\cdot}{A}\underset{\cdot}{B}}^{\underset{\rightarrow}{A}}$ will simply express the coordinates of the point $\underset{\cdot}{B}$ relative to frame $\underline{A}$. From the notation $\boldsymbol{p}_{\underset{\cdot}{A}\underset{\cdot}{B}}^{\underset{\rightarrow}{A}}$ 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*,

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

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

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

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*,

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

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

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

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

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

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

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.

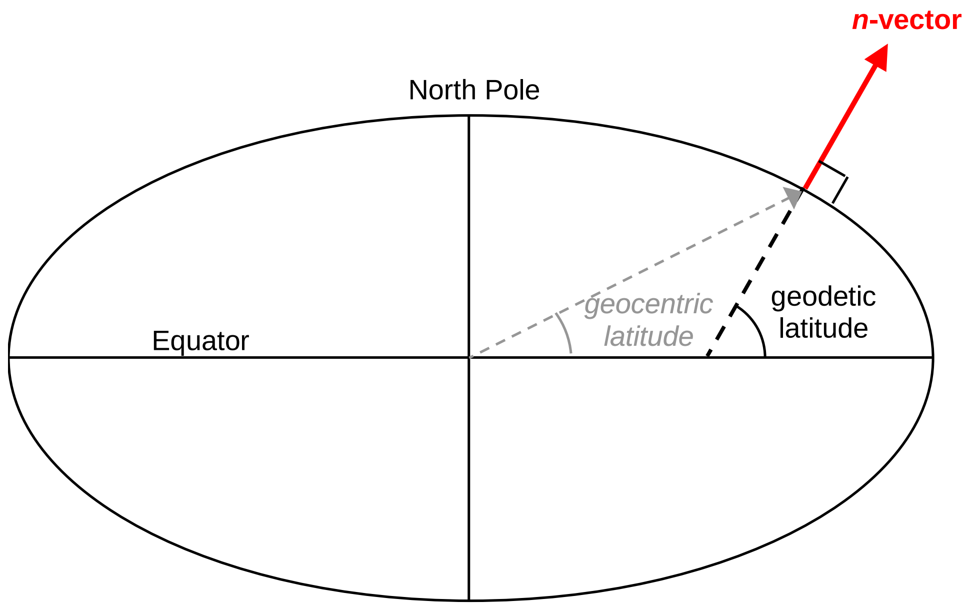


Figure 3.1. Earth reference ellipsoid with n-vector, standard (geodetic) latitude and geocentric latitude.

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.

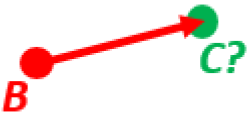
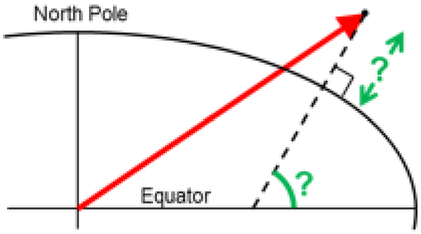
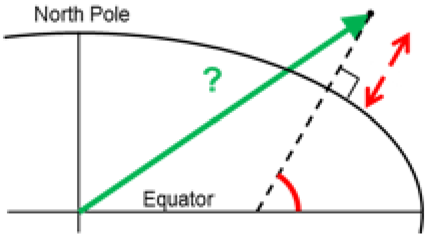
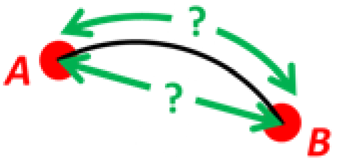
#	Simple description	Simple figure
1.	***A* and *B* to delta** Given two positions A and B. Find the exact vector from A to B in meters north, east and down, and find the direction (azimuth/bearing) to B, relative to north. Use WGS-84 ellipsoid.	
2.	***B* and delta to *C*** Given the position of vehicle 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.

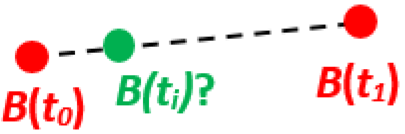
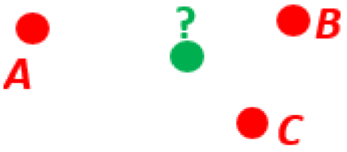
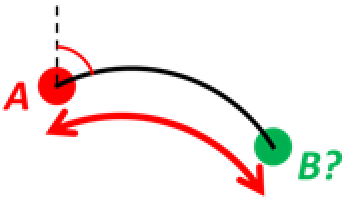
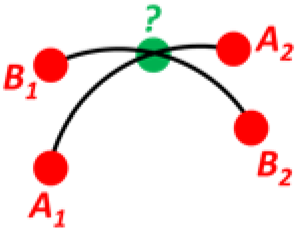
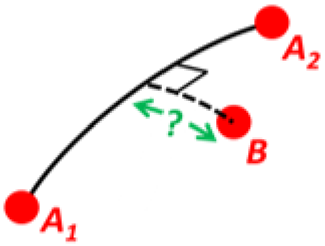
#	Simple description	Simple figure
6.	**Interpolated position** Given the position of B at time t_0 and t_1. Find an interpolated position at time t_i.	
7.	**Mean position (center/midpoint)** Given three positions A, B, and C. Find the mean position (center/midpoint).	
8.	***A* and azimuth/distance to *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

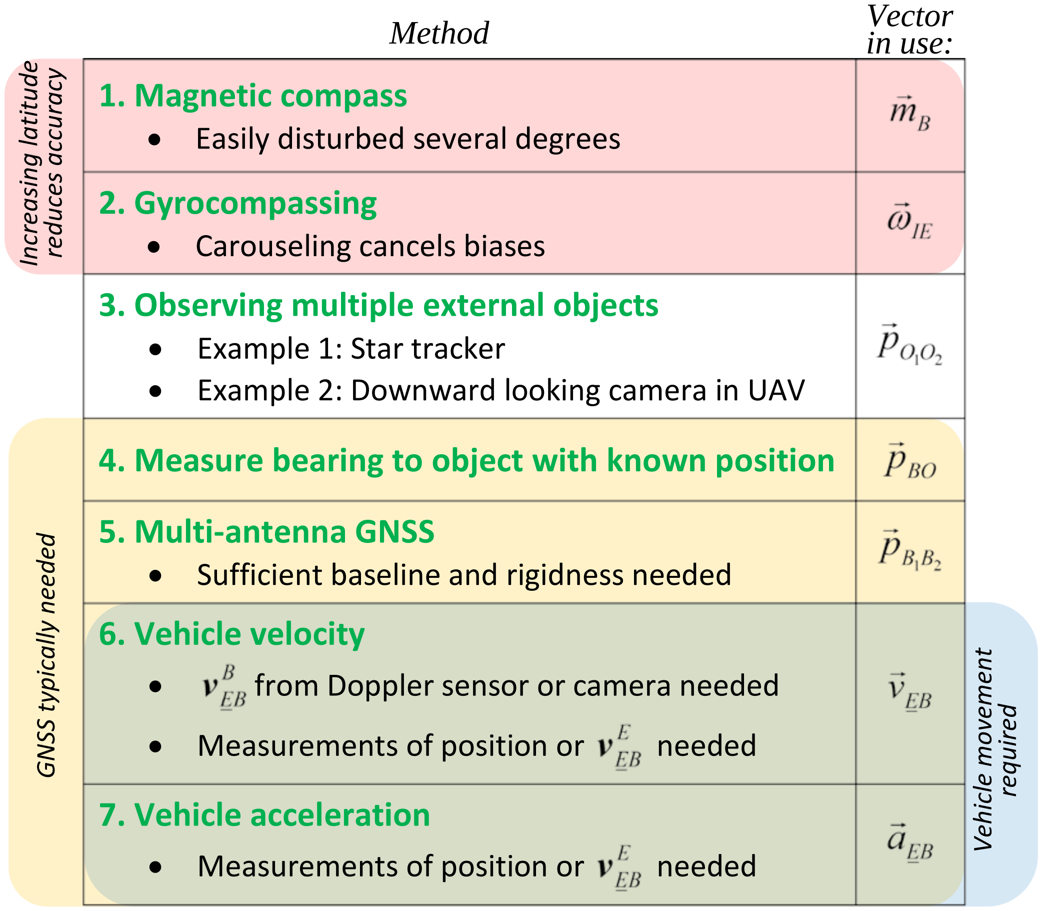
Table 3.4. The four categories (A1, A2, B1, and B2) of inertial navigation systems, broken down by the availability of GNSS and the accuracy of gyros (the table is from ***Paper II****).*

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 • $\boldsymbol{v}_{\underline{EB}}^{B}$ from Doppler sensor or camera needed • Measurements of position or $\boldsymbol{v}_{\underline{EB}}^{E}$ needed	$\vec{v}_{\underline{EB}}$
7. Vehicle acceleration • Measurements of position or $\boldsymbol{v}_{\underline{EB}}^{E}$ needed	$\vec{a}_{\underline{EB}}$

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.

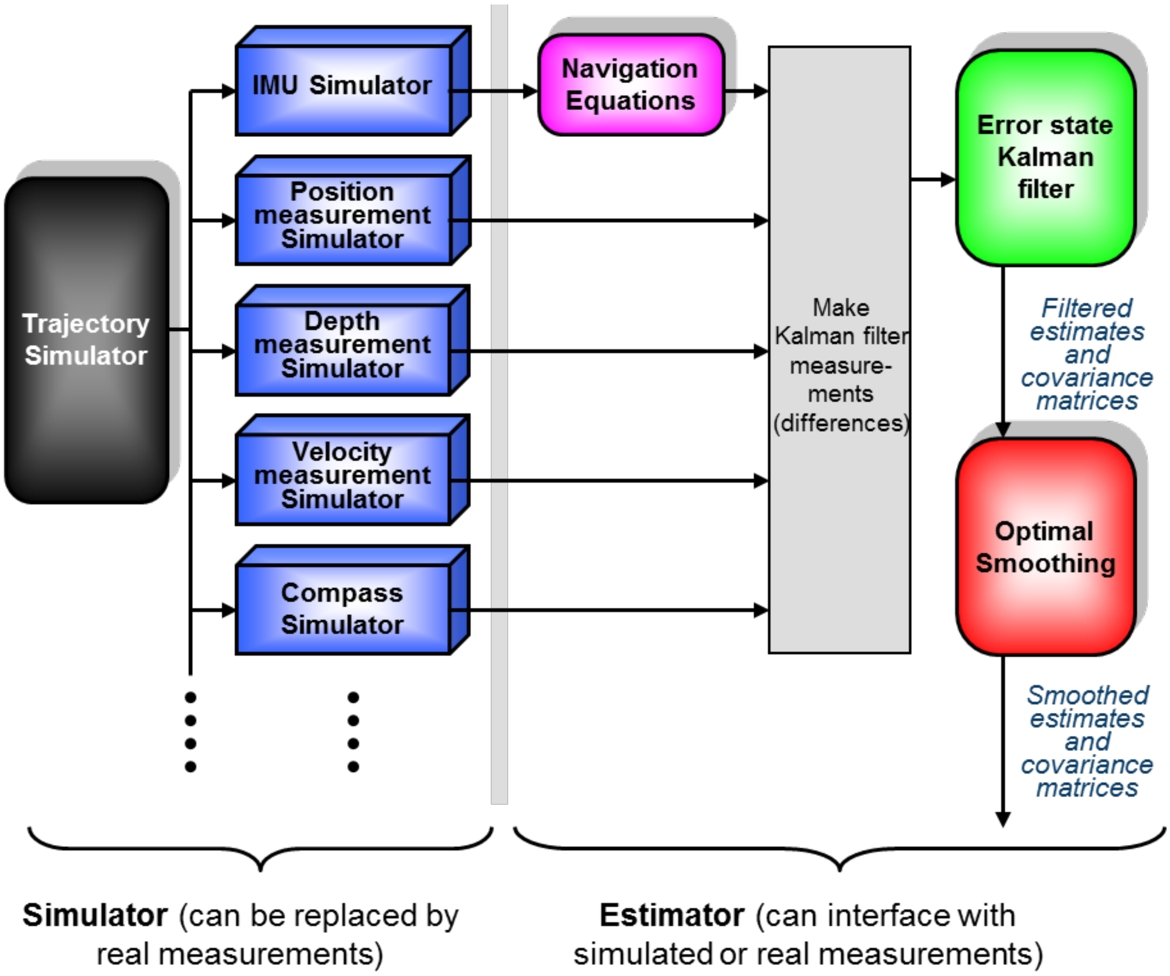

IMU Simulator
Position measurement Simulator
Depth measurement Simulator
Velocity measurement Simulator
Compass Simulator
Trajectory Simulator
Navigation Equations
Make Kalman filter measure-ments (differences)
Error state Kalman filter
Filtered estimates and covariance matrices
Optimal Smoothing
Smoothed 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.

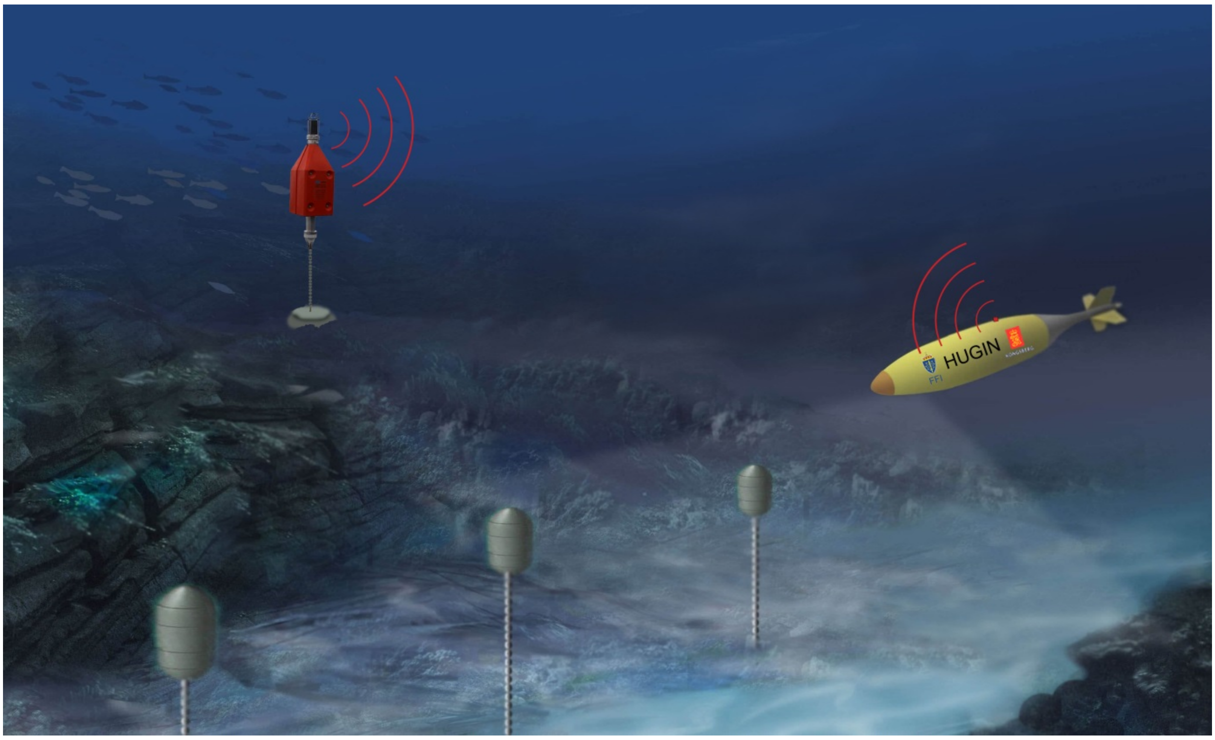


Figure 5.1. AUV measuring range to an underwater transponder.

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 UUVs. *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

Available at:

http://www.navlab.net/Publications/A_Nonsingular_Horizontal_Position_Representation.pdf

Paper II

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

Available at:

http://www.navlab.net/Publications/The_Seven_Ways_to_Find_Heading.pdf

Paper III

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

Available at:

http://www.navlab.net/Publications/An_Aided_Navigation_Post_Processing_Filter_for_Detailed_Seabed_Mapping_UUVs.pdf

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

Available at:

http://www.navlab.net/Publications/NAVLAB_a_Generic_Simulation_and_Post_processing_Tool_for_Navigation.pdf

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

Available at:

http://www.navlab.net/Publications/A_Toolbox_of_Aiding_Techniques_for_the_HUGIN_AUV_Integrated_Inertial_Navigation_System.pdf

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

Available at:

http://www.navlab.net/Publications/DVL_Velocity_Aiding_in_the_HUGIN_1000_Integrated_Inertial_Navigation_System.pdf

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

Available at:

http://www.navlab.net/Publications/Towards_Model_Aided_Navigation_of_Underwater_Vehicles.pdf

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

Available at:

http://www.navlab.net/Publications/Underwater_Transponder_Positioning_and_Navigation_of_Autonomous_Underwater_Vehicles.pdf