# Deep-Sea Model-Aided Navigation Accuracy for Autonomous Underwater Vehicles Using Online Calibrated Dynamic Models

Zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften
(Dr.-Ing.)

bei der Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte
DISSERTATION

von
Dipl.-Math. Dipl.-Inform. David Oertel
geb. in Berlin

| | |
|---|---|
| Datum der mündlichen Prüfung: | 5.2.2018 |
| Referent: | Prof. Dr.-Ing. Dr. h.c. Heinz Wörn |
| Korreferent: | Prof. Dr.-Ing. Uwe Hanebeck |

*Wahrlich es ist nicht das Wissen, sondern das Lernen, nicht das Besitzen, sondern das Erwerben, nicht das Da-Sein, sondern das Hinkommen, was den größten Genuss gewährt.*

Carl Friedrich Gauß

# Credits

First of all, I would like to thank my supervisor, Prof. Wörn, for his support and confidence in me as well as giving me the chance for gathering plenty of valuable experience - in science, engineering, project work and personally. Also, I would like to thank my co-supervisor, Prof. Hanebeck, for his interest, super-vision and valuable advice for my work. Thanks to Torsten Kröger for his support and advice during the final dissertation and initial post-dissertation time.

Thanks to all my colleagues at the IPR for the friendly and enjoyable atmosphere during all the mostly awesome years as an employee. Thanks especially to magnificent Mirksi, superb Scalez, jolly Jazz and the elite coffee-club: lively Lucile and munificent Mende who greatly supported me during the final year. As well as Laserman for introducing me to the IPR and his occasional "Worms Armageddon" collaborations; and Jan for his epic GIMP skills.
I would like to thank my SMIS colleague Sergej for his big efforts within the SMIS project as well as the very pleasant atmosphere, discussions, collaborations at the IPR and his good taste in humour.
Thanks to the BMWi for funding the SMIS project and thanks to all colleagues who worked on it: from TUB, URO, IMPaC, Enitech and IOW, as well as the crews and captains of the Briese RVs for their support during our research missions at sea.

I would like to thank my proof readers for their valuable time and advice: the wrong but also quite adequately competent and funny Dave; Dennis, Hissi, Jonas, Luzie, Sam, Scalez and Sergh.

Thanks to all my friends for their general support and understanding of possibly feeling somewhat left-out during the metaphorical submarine mission of conjuring a PhD thesis. Thanks to Sibby for her moral support especially during emergencies and to the Fiesels for their consistency during these years.

Lots of gratitude goes to my entire family to which I'm fortunately belonging. Especially to my parents, sister, brother and grandpa for patiently accompanying me during my dissertation and entire life as well as literally and figuratively providing me with the opportunity to take this path in the first place.

An additional personal gratitude goes to Lena. Your unexpected appearance close to the final dissertation mayhem helped me a lot to put this work into perspective. You inspire, motivate and fascinate me by just being yourself and you've surprisingly countered the end of the exhausting final PhD year with some of the most beautiful days of my life. For these and many more things I am very grateful beyond your impact on this thesis.

# Abstract

In this work, the accuracy of inertial-based navigation systems for autonomous underwater vehicles (AUVs) in typical mapping and exploration missions up to $5000\,m$ depth is examined. The benefit of using an additional AUV motion model in the navigation is surveyed.

Underwater navigation requires acoustic positioning sensors. In this work, so-called Ultra-Short-Baseline (USBL) devices were used allowing the AUV to localize itself relative to an opposite device attached to a (surface) vehicle. Despite their easy use, the devices' absolute positioning accuracy decreases proportional to range. This makes underwater navigation a sophisticated estimation task requiring integration of multiple sensors for inertial, orientation, velocity and position measurements.
First, error models for the necessary sensors are derived. The emphasis is on the USBL devices due to their key role in navigation - besides a velocity sensor based on the Doppler effect. The USBL model is based on theoretical considerations and conclusions from experimental data.
The error models and the navigation algorithms are evaluated on real-world data collected during field experiments in shallow sea. The results of this evaluation are used to parametrize an AUV motion model. Usually, such a model is used only for model-based motion control and planning. In this work, however, besides serving as a simulation reference model, it is used as a tool to improve navigation accuracy by providing virtual measurements to the navigation algorithm (model-aided navigation). The benefit of model-aided navigation is evaluated through Monte Carlo simulation in a deep-sea exploration mission.

The final and main contributions of this work are twofold. First, the basic expected navigation accuracy for a typical deep-sea mission with USBL and an ensemble of high-quality navigation sensors is evaluated. Secondly, the same setting is examined using model-aided navigation.
The model-aiding is activated after the AUV gets close to sea-bottom. This reflects the case where the motion model is identified online which is only feasible if the velocity sensor is close to the ground (e.g. $100\,m$ or closer).

The results indicate that, ideally, deep-sea navigation via USBL can be achieved with an accuracy in range of $3 - 15\,m$ w.r.t. the expected root-mean-square error. This also depends on the reference vehicle's position at the surface.
In case the actual estimation certainty is already below a certain threshold (ca. $< 4\,m$), the simulations reveal that the model-aided scheme can improve the navigation accuracy w.r.t. position by $3 - 12\%$.

# Zusammenfassung

In dieser Arbeit wird die Genauigkeit von inertialen Navigationssystemen für autonome Unterwasser-Vehikel (AUV) für typische Kartographierungsmissionen in der Tiefsee bis zu $5000\,m$ Tiefe untersucht. Zusätzlich wird für diesen Fall geprüft, welche Vorteile die Nutzung eines AUV-Bewegungsmodells in der Navigationslösung aufzeigen.

## Einleitung und Motivation

Die Ozeane spielen eine zentrale Rollen für alles Leben auf der Erde. Um diese zu erforschen und die Balance ihrer teilweise fragilen und trägen Ökosysteme nicht zu stören, ist es von Vorteil, automatisierte Messfahrten durchzuführen. Dies ist besonders dann wichtig, wenn der Mensch durch Untersee-Strukturen wie Kabel, Pipelines oder Windparkfundamente direkt in die Natur eingreift.
Hierfür bieten sich AUV an, für die jedoch durch die Abwesenheit von GPS u.ä. unter Wasser die Navigation und Lokalisierung eine anspruchsvolle Aufgabe darstellt, wobei es nötig ist, akustische Lokalisierung einzusetzen. Besonders die Verwendung von kompakten Geräten (USBL) zur relativen Lokalisierung des AUVs durch ein Oberflächenfahrzeug über große Distanzen (von Oberfläche zum Seeboden) ist schwierig, da die absolute Positionsgenauigkeit der Geräte proportional zur Distanz abnimmt.
Daher ist es wichtig, die Navigationsgenauigkeit für AUV-Missionen in der Tiefsee vorher abzuschätzen - abhängig von den gegebenen Unterwassersensoren zur Navigation. Zusätzlich besteht Bedarf, diese Genauigkeit weiter zu verbessern. Dazu stehen verschiedene Möglichkeiten bereit, wobei in dieser Arbeit der Fokus darauf liegt, ein AUV-Bewegungsmodell als virtuelle Messung in den Navigationsansatz einfließen zu lassen.

## Stand der Technik und Verwandte Arbeiten

Unterwasser-Navigation ist ein aufwendiges Zustandsschätzungsproblem. Üblicherweise wird hierfür das Erweiterte Kalman Filter (EKF) eingesetzt. Das Systemtransitionsmodell wird dabei allgemein auf Basis der Inertialdaten (lineare Beschleunigungen und Drehraten) formuliert und zur eigentlichen Schätzaufgabe mit diversen Messungen gespeist. Dies betrifft neben Kompass und Drucksensoren vor allem akustischen Positionsmessungen (USBL) und akustischen Geschwindigkeitsmessungen auf Basis des Dopplereffekts.
Diese Navigationslösung kann zusätzlich mittels eines AUV-Bewegungsmodells gestützt werden. Dabei handelt es sich um ein Systemmodell aus Differentialgleichungen, die

den Zusammenhang von Aktorik (Propellerdrehzahl und Ruderwinkel) zur aktuellen Beschleunigung, Geschwindigkeit und AUV-Position herstellen, wobei zahlreiche hydrodynamische Effekte eine Rolle spielen.
Sofern eine genaue Parametrisierung des Bewegungsmodells vorliegt, die zum realen Fahrverhalten des AUV passt, kann dieses als virtueller Sensor in die Navigationslösung einfließen. Hierfür wird in dieser Arbeit die Ausgabe der linearen Geschwindigkeiten aus dem Bewegungsmodell genutzt - ähnlich wie ein zusätzlicher akustischer Geschwindigkeitssensor.

## Sensorfehlermodelle, Navigationsdetails und Umsetzung

Die einzelnen Sensorfehlermodelle werden vorgestellt, inklusive der Behandlung von systematischen Abweichungen, dem Bias. Der Fokus liegt hierbei auf einer genauen Untersuchung der USBL-Sensoren mittels Planarwellen-Approximation. Durch diese Annäherung kann die USBL-Messung als zwei separate Messungen aufgefasst werden: die relative Richtung eines USBL-Gerätes zu einem sendenden Gerät über örtliche Signalkorrelation sowie die Distanz zwischen beiden Geräten über die Signallaufzeit. Dadurch kann die USBL-Messung insgesamt als Messung mit weißem Gaußschen Rauschen behandelt werden und in das EKF einfließen.
Um die Navigationsgleichungen algorithmisch ausführen zu können werden die Filtergleichungen diskretisiert, wobei Zeitschritte der Größe $0.1s$ verwendet werden, wie sie in einem realen Messdatensatz vorliegen.

Das AUV-Bewegungsmodell in Simulink wird vorgestellt und ein detailliertes Schema, mit dem dieses für die Navigationslösung genutzt werden kann. Dazu wird das aktuelle Ergebnis des Navigationsalgorithmus nach Empfang einer akustischen Geschwindigkeitsmessung in das Simulink-Modell eingespeist und für eine kurze Zeitspanne simuliert ($0.3\,s$). Das Ergebnis dieser Kurzzeitsimulation wird anschließend dem Navigationsalgorithmus als virtuelle Messung übergeben. Hierbei werden nur die simulierten linearen Geschwindigkeiten verwendet. Alternativ könnten auch weitere Daten aus dem Modell in die Navigation einfließen. Jedoch birgt dies die Gefahr, diese virtuelle Messquelle zu überschätzen. Zudem bietet die Konzentration auf die Geschwindigkeiten den Vorteil, dass dieses Schema direkt zur modularen Erweiterung eines bereits bestehenden Navigationssystems geeignet ist.
In dieser Arbeit liegt der Fokus auf dem folgenden Anwendungsfall: alle Sensoren voll verfügbar und das Bewegungsmodell wird darauf aufbauend zusätzlich eingespeist, statt etwa einen Sensorausfall aufzufangen.

Zur Evaluation der erreichbaren Genauigkeit einer AUV-Tiefseemission werden Cramér-Rao-Schranken für das vorliegende nichtlineare diskrete Filterproblem ermittelt. Diese Schranken geben den zu erwartenden mittleren Fehler an, mit dem unabhängig vom eigentlich eingesetzten Schätzer gerechnet werden muss. Da diese Schranken i.A. nicht analytisch lösbar sind, werden diese approximativ mittels Monte-Carlo-Simulationen berechnet.
Zusätzlich wird das Einspeisen der virtuellen Messungen aus dem Bewegungsmodell in die Navigationslösung per Monte-Carlo-Simulation mit EKF als Schätzern getestet,

um das Potential des Verfahrens in einem typischen Navigationssystem in der Tiefsee zu evaluieren.

## Evaluation

Auf Basis eines realen Messdatensatzes für ein prototypisches AUV im Flachwasser ($100\,m$ Tiefe) werden zunächst die Sensorfehlermodelle und der Navigationsalgorithmus evaluiert. Die Ergebnisse werden genutzt, um Bewegungsmodell des AUV zu parametrieren, welches in der Folge als Referenz für die Simulationen genutzt wird.
In der Simulation fährt das AUV eine Referenztrajektorie ab; bestehend aus einer spiralförmigen Tauchbewegung bis kurz vor dem Grund ($5000\,m$ Tiefe) und einer daraufhin gefolgten typischen Mäanderbewegung über den Meeresgrund, um diesen z.B. zu kartographieren. Dabei referenziert sich das AUV via USBL zu einem Oberflächenfahrzeug.

Zunächst wird die Navigationsgenauigkeit der Standardnavigation evauliert, wobei Fehlerkonstanten von für diese Anwendung kommerziell erwerbbaren bestmöglichen Sensoren angenommen werden. Anschließend wird das gleiche Szenario mit dem Schema zum Einspeisen des Bewegungsmodells evaluiert.
Die Resultate den Monte-Carlo-Simulationen lassen sich folgendermaßen zusammenfassen: die Genauigkeit der grundlegenden Tiefseenavigation via USBL kann idealerweise in Bezug auf den erwarteten Fehler mit $3 - 15\,m$ (quadratisches Mittel) angegeben werden. Falls die aktuelle Schätzunsicherheit der Standardnavigation unterhalb eines Schwellwerts liegen (ca. $< 4\,m$) zeigen die Simulationen, dass das Einspeisen des Bewegungsmodells die Navigationsgenauigkeit um $3 - 12\%$ verbessern kann (bzgl. der Position).

## Fazit, Diskussion und Ausblick

USBL-Messungen sollten zum Einspeisen in ein EKF wegen der nötigen additiven weißen Gaußschen Rauschannahme stets getrennt als Richtungs- und Distanzmessungen einfließen.
Beim Einmessen einer Bodenstation durch ein Oberflächenfahrzeug ist nach ca. $20\,min$ mit einer horizontalen Genauigkeit von $7\,m$ (oder schlechter) zu rechnen.
Tiefseenavigation, bei der ein AUV sich an einem Oberflächenfahrzeug via USBL referenziert, ist nur mit sehr guter Navigationssensorik möglich. In den Fällen, wo der Navigationsalgorithmus bereits aktuell eine gute Schätzung mit geringer Kovarianz vorliegen hat, kann das Einspeisen eines Bewegungsmodells die Genauigkeit idealerweise nochmals um einige Prozente verbessern.

Die theoretisch erreichbare Navigationsgenauigkeit ist erstaunlich, wenn man bedenkt, dass eine Einzelmessung eines USBL-Gerätes über $5000\,m$ eine mittlere quadratische Abweichung der Größenordnung $100\,m$ aufweist.
Trotz der Genauigkeit könnte es jedoch in der Praxis von Vorteil sein, weitere AUVs als Zwischenstationen in der Wassersäule oder Bodenstationen zur Referenzierung in die

Mission einzuschließen.
Die ermittelten Ergebnisse in Bezug auf die vom Bewegungsmodell gestützte Navigation sind konsistent mit praktischen Versuchen aus der Literatur im Flachwasser. Zugleich ist der größte Vorteil dieses Konzepts in der Praxis aber womöglich, die Navigationslösung robuster zu machen, wenn akustische Sensoren (Position und Geschwindigkeit) vorübergehend ausfallen.
Zudem ist es möglich, dass selbst wesentlich simplere Bewegungsmodelle mit guten Ergebnissen für die Navigationsverbesserung eingesetzt werden können.

Bezüglich des Dynamik-Modells wäre es wissenschaftlich interessant, ob die Parameterschätzung - durch Korrelationsmethoden - auch auf versteckte zusätzliche Parameter des Modells anwendbar ist, die sonst nur über aufwendige Spezialtests und Rechnungen ermittelbar sind. Wenngleich dies praktisch weniger relevant ist.

Sehr vielversprechend könnte sich Maschinelles Lernen für Schätzung allgemeinerer Bewegungsmodelle einsetzen lassen. Diese können zwar nur sehr wenige Bewegungen abbilden, diese aber dafür sehr genau erlernen, um daraus eine Stützung der Navigation zu generieren. Denn in vielen AUV-Missionen ist die Bewegungsvielfalt auf Geradeausfahren und simple Kurven beschränkt.

# Contents

# 1. Introduction

The world's Oceans are playing a key role for living beings on earth. This makes it essential to research and observe the sea for better understanding of the entire life on the planet.
Moreover, it is important to keep the fragile Ocean's ecosystem at balance. Every interference by humans requires precise preparation, careful design and accurate monitoring. This holds especially in deep-sea as there, the general "clock" of life is much much slower than e.g. on land [DRS02]. This includes recording precise maps of the sea-floor, see figure 1.1 as an example, as well as examining the impact of human-made structures on the sea-floor such as sub-sea cables, pipelines and foundations of off-shore structures such as windmills.

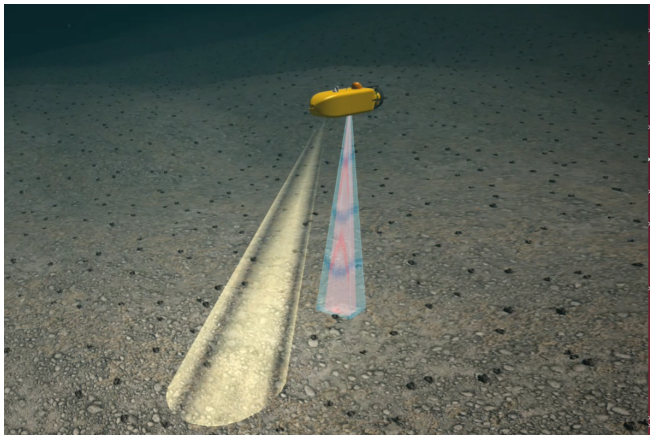

Figure 1.1.: Rendered illustration of an AUV mapping the sea-floor or examining minerals on the sea-floor through, e.g. a combination of a camera (yellow beam) and a sonar device (blue beam). (Picture courtesy of IMPaC GmbH, Hamburg)

Due to the big effort required in all of these tasks, it is convenient to use some kind of automation. Among several other options, the use of *Autonomous Underwater Vehicles* (AUVs) is a promising one having received a lot of attention since the 1990's - in science as well as in industrial, geological and biological applications.
To achieve these tasks, it is crucial for any automated system to have a precise estimate of its own location at all times. In contrast to terrestrial applications, accurate localization underwater is a quite sophisticated, elaborate and costly matter. In fact, the majority

of an AUV's equipment on board besides energy storage and buoyancy equipment is uniquely dedicated to make the AUV capable of estimating its own position - both in hardware/sensors and software.
On the one hand, it is desirable to know the navigation/localization accuracy of an operating AUV given a certain set of sensors. This goes as far as estimating if operation is even feasible in the first place or if the expected accuracy is possibly too bad for the desired operation. On the other hand, there is a large demand for techniques to improve localization and navigation through both high- and low-level schemes since AUV operation is a very costly endeavour.

This work is based on experiences from the research and development project "SMIS", which is introduced in section 1.1.1.

## 1.1. Motivation

The availability of the high-quality Global Positioning System (GPS) has revolutionized terrestrial navigation for humans in general and specifically for all types of traffic systems in air, on the ground and on water over the past decades. This also applies to autonomous robotic systems in research and technology. Unfortunately, this does not hold at all for underwater systems. Due to the attenuation of electro-magnetic signals underwater, especially for GPS, implies the necessity of specialized techniques and sensors required for sub-sea navigation. Due the general sound propagation features of water, the role of GPS systems underwater is in most cases replaced by acoustic positioning devices. These share a key role in underwater navigation - together with acoustic sensors for velocity measurements using the Doppler effect.
Especially for AUV missions in deep-sea up to $5000m$ depth, the accuracy of the aiding position and velocity sensors is absolutely crucial to maintain a reliable position estimate. Classically, localization is achieved through acoustic positioning based on preinstalled transponders which require a lot of time to be calibrated. Recently, compact devices for relative acoustic positioning have gained popularity.
Since the localization accuracy of these compact devices decreases proportional to range, long-range deep-sea localization is sophisticated with these devices. This raises the question of how accurate navigation can be in this case - given a certain set of sensors for inertial, positioning and velocity measurements.

There is quite an amount of possibilities to further improve navigation accuracy, such as simultaneous localization and mapping (SLAM) as well as cooperative localization, just to name a few "hot topics". In this work, it was tried to use results from AUV control theory to improve the navigation solution from virtual measurements by an AUV dynamic model describing its motion in water.
Besides deriving general estimates of deep-sea navigation accuracy, the key motivation of this work was to let the navigation system benefit from the results of control theory for AUV motion - since deriving an AUV motion model is necessary for AUV operation anyway, i.e. for motion control and for trajectory planning.

### 1.1.1. The SMIS Project

This work is based on results of the research and development project "Subsea Monitoring via Intelligent Swarms" (SMIS) which was running from January 2013 to September 2016. It was funded by the German federal ministry of economic affairs and energy - "Bundesministerium für Wirtschaft und Energie" (BMWi) [bmw].

The goal of this project was to develop a team of underwater robots for deep-sea exploration tasks, e.g. mapping of the sea-floor. The set-up of the basic team can be seen in figure 1.2. The photographs in figure 1.2 were taken during sea-trials at lake Constance

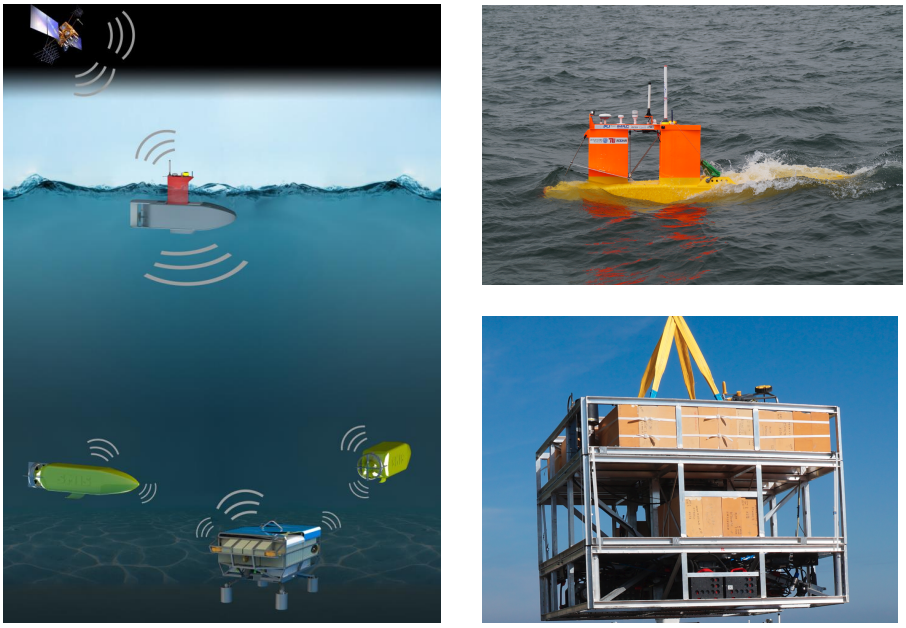

Figure 1.2.: Left: Overview of the basic SMIS vehicle team; AUVs in yellow, SBS at the bottom in blue, USV with red top on the surface. (Picture courtesy of TUB, Berlin). Right: SMIS USV (top) during sea-trials and the SMIS SBS (bottom) being lifted (Pictures courtesy of IMPaC GmbH, Hamburg, TUB, Berlin, and University of Rostock).

in May 2016. The basic SMIS team consists of two (semi-)autonomous underwater vehicles (AUVs), one sea-bed station (SBS) and an unmanned surface vehicle (USV).
The AUVs perform the core exploration tasks. The SBS is moored on the sea-bed on mission start to support the AUVs in their tasks by recharging them, storing measurement data and serving as a geo-reference. The USV also serves as a geo-reference through sharing its GPS position. Moreover, it serves as a communication node from sub-sea to ship- or land-based operators through a radio or satellite link. Through this setting, the mission times can be increased up to several days and the ship carrying the vehicles

does not need to stay in the vicinity of the AUV operation. A picture of the AUV can be found in the next chapter, see section 2.1.1.

The project was accomplished by a group of research institutions and companies consisting of:

- Technical University Berlin, Department of Surface Transport Design and Operation of Maritime Systems,
- University of Rostock (URO), Center for Marine Information Systems,
- Karlsruhe Institute of Technology (KIT), Institute for Anthropomatics and Robotics (IAR) - Intelligent Process Automation and Robotics Lab (IPR),
- IMPaC Offshore Engineering GmbH, Hamburg,
- Enitech GmbH, Rostock, Bentwisch,
- Leibniz Institute for Baltic Sea Research Warnemünde (IOW).

The manifold work packages of the project can be very roughly summarized as follows. The main tasks of TUB were to construct, build and test the SBS [BGRH14], [GBRH16], [BGRH16] as well as construct the USV [RTB+15]. URO was responsible for several automation and control tasks [RKD+15] of the AUV as well as design, installation and testing of the USV [RDK+15, KRD+15]. The work of KIT regarded simulation, see figure 1.3, modelling and high-level control of the AUVs, including navigation and local-

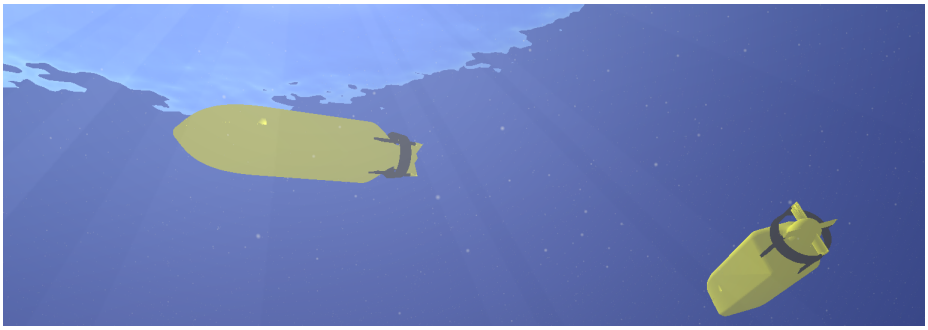

Figure 1.3.: Screenshot of a physical and graphical simulation containing $3d$-models of the SMIS AUVs. Simulation is based on ROS [QCG+09] and UWSim [PPFS12].

ization [NOW+15, ONW+16]. The IOW was responsible for marine research applications which could be solved by the SMIS vehicles as well as organizing multiple ship-based sea trials for the vehicles in the Baltic Sea and Atlantic Ocean.
For further information, please refer to the work of J.J. Waniek et al, e.g. [MPD+14] and her work-groupâĂŹs homepage [IOW].
Enitech was responsible for the mechanical and electrical construction of the AUV(s), using the experience from several previous projects.

Last but not least, IMPaC was responsible for coordinating the project with its partners scattered over Germany. Additionally, they provided several FEM analyses for the partners.
The sea trials were hosted by the Briese Schifffahrt GmbH on the Research Vessel (RV) Elisabeth-Mann-Borgese (EMB) and RV Poseidon.

Knowledge collected and data sets recorded from multiple sea trials during the project run-time were used in this work for evaluation, analysis and generating the core ideas of this work in the first place.

More information on the SMIS project is publicly available at the project homepage [smi].

## 1.2. Aim and Contribution of this Work

The central aim of this work is to examine deep-sea AUV navigation accuracy using specialized acoustic position technologies (USBL, see section 2.4) and to thus provide extended autonomy features to AUVs.

The two main contribution for the underwater community is an extension of the works of Hegrenaes et al. and Morgado et al., see section 2.8:

- Estimate the accuracy of basic deep-sea navigation with USBL,
- Examine the benefit of re-using an AUV motion model for navigation in the deep-sea domain.

Besides that, there are several secondary scientific results and technical details in this work useful for the community: hands-on description and details on sensor modelling, navigation, dynamic modelling and identification evaluated for a real-world AUV. Especially, this work contains details on USBL modelling, theoretical error analysis and evaluation of its practical accuracy.

In more detail, the extensions of the aforementioned works of Hegrenaes et al. and Morgado et al. can be identified as:

- Accuracy estimates for deep-sea instead of shallow-water applications,
- Including Velocity measurements (Doppler Velocity Log),
- Use of full 6 degrees of freedom (DoF) AUV motion model instead of 3DoF,
- Benefit of using the AUV motion model for navigation in case of full sensor availability.

In a side chapter, the the accuracy of a surface vehicle localizing a fixed transponder (SBS) on the sea-floor via USBL is evaluated in order to perform a pre-test on USBL localization before examining the entire sensor ensemble needed for full underwater navigation.

## 1.3. Outline

This work is organized as follows.
In chapter 2, the State of the Art and all necessary fundamentals of underwater technology, filtering and dynamic modelling is reviewed. It is also provides the fundamentals for underwater navigation, including the sensors used therein.
Additionally, the concept of Posterior Cramér-Rao Bounds (PCRB) is introduced. which is useful for estimating navigation accuracy in the absence of ground-truths.
After providing the necessary tools for AUV dynamic modelling, the fundamentals of Model-Aided navigation are given.

Chapter 3 is dedicated to details of sensor error models, Strapdown navigation and implementation outlines of these topics including the Dynamic Model in Simulink [MAT17], Model-Aided navigation and the framework to compute the approximate PCRBs.
The emphasis in the error models is on the special type of acoustic positing sensors used in this work (USBL) since it plays such an important role in underwater navigation.
Moreover, the mathematical details of underwater navigation are provided and implementations of the Model-Aided navigation scheme are given. Additionally, a AUV dynamic model implementation in Simulink is provided.

In chapter 4 the evaluation is presented. This includes: evaluating the error models and navigation algorithms on a real-world data set, from which the parameters for the AUV dynamic model are derived. This model is then used in the main simulations for deriving accuracy bounds in deep-sea navigation - the main theoretical results of this work.

Finally, chapter 5 contains a conclusion and discussion providing an outlook on possible future work in the field.

Before moving on to chapter 2, this chapter is concluded by a few mathematical prerequisites in section 1.4 which are needed in the subsequent chapters.

## 1.4. Mathematical Prerequisites

In this section, a few basic mathematical tools are subsumed. Due to their relevance w.r.t. the following chapters, they are briefly given here for completeness.

Further notes and explanations on mathematical background are given in the appendix, see section A.1. This contains a quick review of several basics of Stochastic, including explanations of the related notations used in this work for clarification.

**Theorem 1** (Multiplication Theorem)**.** *For events $A_1, \ldots, A_n$ with $P(A_1 \cap \ldots \cap A_n) > 0$ it holds:*

$$\begin{aligned} P(A_1 \cap \ldots \cap A_n) = P(A_1) \cdot P(A_2 \mid A_1) \cdot P(A_3 \mid A_2 \cap A_1) \cdot \\ \ldots \cdot P(A_n \mid A_{n-1} \cap \ldots \cap A_1) \end{aligned} \tag{1.1}$$

For proof, refer to some standard text book like [Geo13]. The analogue also holds in the continuous case.

### 1.4.1. Cramér-Rao Bounds

The famous concept of Cramér-Rao bounds is relevant for the simulations on navigation accuracy. It's original form is shortly introduced in this section. Further details and proves can be found in [Hen13] or [Geo13].

Fisher information $I_f(\theta)$, for discrete Random variables

$$0 < I_f(\theta) := \sum_{x \in M_f} \left[ \frac{\partial}{\partial \theta} \log f(x, \theta) \right]^2 \cdot f(x, \theta) < \infty \quad \forall \theta \in \Theta \tag{1.2}$$

Fisher-Information, general case

$$I_f(\theta) := \mathbb{E}_\theta \left[ \left( \frac{\partial}{\partial \theta} \log f(x; \theta) \right)^2 \right] \tag{1.3}$$

$$= \int \left( \frac{\partial}{\partial \theta} \log f(x; \theta) \right)^2 \cdot f(x; \theta)\, dx \quad \forall \theta \in \Theta \tag{1.4}$$

The famous Cramér-Rao bound

$$\mathrm{Var}_\theta(T) \geq \frac{\left[ \frac{\partial}{\partial \theta} \mathbb{E}_\theta(T) \right]^2}{I_f(\theta)} \tag{1.5}$$

for an estimator $T$ for $\theta$ based on (observed) data $x$

For unbiased estimators, Cramér-Rao simplifies to

$$\mathrm{Var}_\theta(T) \geq \frac{1}{I_f(\theta)} \tag{1.6}$$

Multivariate case of Cramér-Rao ($\theta \in \mathbb{R}^r$), given the proper assumptions:

$$I_f(\theta) = \mathbb{E}_\theta \left[ \left( \frac{\partial}{\partial \theta} \log f(x; \theta) \right) \cdot \left( \frac{\partial}{\partial \theta} \log f(x; \theta) \right)^T \right] \tag{1.7}$$

$$\mathrm{cov}_\theta(T) \geq \left[ \frac{\partial \mathbb{E}_\theta(T)}{\partial \theta} \right] \cdot I(\theta)^{-1} \cdot \left[ \frac{\partial \mathbb{E}_\theta(T)}{\partial \theta} \right]^T \tag{1.8}$$

which, again simplifies to:

$$\mathrm{cov}_\theta(T) \geq I(\theta)^{-1} \tag{1.9}$$

for unbiased estimators $T$, where the matrix-inequality "$A \geq B$" denotes that $A - B$ is positive semi-definite.

### 1.4.2. Miscellaneous Identities

**Lemma 1** (Woodbury Matrix Identity)**.** *For $A \in \mathbb{R}^{n\times n}, U \in \mathbb{R}^{n\times k}, C \in \mathbb{R}^{k\times k}$ and $V \in \mathbb{R}^{k\times n}$ with invertible $A$ and $C$, it holds:*

$$(A+UCV)^{-1} = A^{-1} - A^{-1}U\left(C^{-1}+VA^{-1}U\right)^{-1}VA^{-1} \tag{1.10}$$

This equation is known as the famous *Woodbury matrix identity* playing, e.g., an important role in Kalman filtering applications, see section 2.2. For proof and explanation, see Woodbury's original report [Woo50].

**Lemma 2** (Schur Complement)**.** *Given a matrix*

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

*with $A \in \mathbb{R}^{p\times p}, B \in \mathbb{R}^{p\times q}, C \in \mathbb{R}^{q\times p}, D \in \mathbb{R}^{q\times q}$ and invertible $D$. Then the* Schur complement *of block matrix $D$ is given by*

$$M/D = A - BD^{-1}C \tag{1.11}$$

*and the Schur complement of matrix $A$ is given by*

$$M/A = D - CA^{-1}B \tag{1.12}$$

The Schur complement has multiple applications in matrix operations. Most notably, it delivers the following conditional covariance.

**Corollary 1** (Conditional Covariance)**.** *Given two Random Variables $X \in R^n$, $Y \in R^m$ with Random Variables $Z = (X, Y) \in R^{m+n}$ having covariance matrix*

$$cov(Z) = \Sigma = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}$$

*which means that $cov(X) = A$, $cov(Y) = C$ and $cov(X, Y) = B$, then the* conditional covariance *$cov(X \mid Y)$ is provided by the Schur complement:*

$$cov(X \mid Y) = A - BC^{-1}B^T$$

See [Zha06] for further explanation on the Schur complement.

#### Cross product and Quaternion representation

In order to unify cross products and quaternion multiplications to matrix multiplication operations, the following equations are introduced. (Unit) quaternions are used in this work in order to represent orientations, see for example [Roo77] as explanation. In this work, the orientation quaternions' first entry represents the scalar part, see section 3.3 for details.

**Definition 1.** *For a given vector* $\vec{w} = (w_1, w_2, w_3)^T \in \mathbb{R}^3$*, let the* skew symmetric matrix $[\vec{w}\times]$ *be defined as:*

$$[\vec{w}\times] := \begin{pmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{pmatrix}$$

Using this definition, the cross product with $\vec{a} \in \mathbb{R}^3$ can be rewritten as:

$$\vec{w} \times \vec{a} = [\vec{w}\times] \cdot \vec{a} = -[\vec{a}\times] \cdot \vec{w} = -\vec{a} \times \vec{w}.$$

**Definition 2.** *For a given quaternion* $\vec{q} = (q_1, q_2, q_3, q_4)^T$*, the so-called* quaternion multiplication matrix $Q_m(\vec{q})$ *and* reversed quaternion multiplication matrix $Q_m^r(\vec{q})$ *are defined as:*

$$Q_m(\vec{q}) := \begin{pmatrix} q_1 & -q_2 & -q_3 & -q_4 \\ q_2 & q_1 & -q_4 & q_3 \\ q_3 & q_4 & q_1 & -q_2 \\ q_4 & -q_3 & q_2 & q_1 \end{pmatrix}$$

$$Q_m^r(\vec{q}) := \begin{pmatrix} q_1 & -q_2 & -q_3 & -q_4 \\ q_2 & q_1 & q_4 & -q_3 \\ q_3 & -q_4 & q_1 & q_2 \\ q_4 & q_3 & -q_2 & q_1 \end{pmatrix}$$

*For convenience, for a* $3-d$ *vector* $vecw \in \mathbb{R}^3$ *the quaternion multiplication matrix is defined such that the* $q_1$*-part is assumed* $0$*:*

$$Q_m(\vec{w}) := Q_m(\begin{pmatrix} 0 \\ \vec{w} \end{pmatrix}),\ \vec{w} \in \mathbb{R}^3$$

$$Q_m^r(\vec{w}) := Q_m^r(\begin{pmatrix} 0 \\ \vec{w} \end{pmatrix}),\ \vec{w} \in \mathbb{R}^3$$

*Finally, in similar spirit, the* cropped *quaternion multiplication matrix omitting the first column shall be defined as:*

$$Q_m^c(\vec{q}) := Q_m(\vec{q}) \cdot \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

*The expressions for* $Q_m^{r,c}(\vec{q}), Q_m^{r,c}(\vec{w})$ *and* $Q_m^c(\vec{w})$ *are defined accordingly.*

Using this definition, the quaternion product with a quaternion $\vec{p}$ can be replaced by its matrix representation:

$$\vec{q} \bullet \vec{p} = Q_m(\vec{q}) \cdot \vec{p},$$

where $\bullet$ denotes the quaternion product.

Although the quaternion product is not commutative, it is possible to reverse the multiplication order using the previously defined "reversed" multiplication matrix $Q_m^r$ is used [K$^+$99]:

$$\vec{q} \bullet \vec{p} = Q_m(\vec{q}) \cdot \vec{p} = Q_m^r(\vec{p}) \cdot \vec{q}$$

**Corollary 2** (Rotation matrix from quaternion)**.** *Given a unit quaternion $q_b^n$ representing a spatial rotation, the respective rotation matrix $R_b^n$ rotating a local vector from $\{b\}$ to coordinates in $\{n\}$ is given by:*

$$R_b^n(\vec{q}) = \begin{pmatrix} {q_1}^2 + {q_2}^2 - {q_3}^2 - {q_4}^2 & 2\,q_2\,q_3 - 2\,q_1\,q_4 & 2\,q_1\,q_3 + 2\,q_2\,q_4 \\ 2\,q_1\,q_4 + 2\,q_2\,q_3 & {q_1}^2 - {q_2}^2 + {q_3}^2 - {q_4}^2 & 2\,q_3\,q_4 - 2\,q_1\,q_2 \\ 2\,q_2\,q_4 - 2\,q_1\,q_3 & 2\,q_1\,q_2 + 2\,q_3\,q_4 & {q_1}^2 - {q_2}^2 - {q_3}^2 + {q_4}^2 \end{pmatrix}$$

*See for example [K$^+$99] for explanation.*

# 2. State of the Art and Related Work

This chapter's intention is to review the State of the Art and all necessary fundamentals of underwater technology, filtering and dynamic modelling before getting into required details in the subsequent chapter.
At first, in section 2.1, today's AUV technology will be briefly reviewed, including some details on the AUV developed in the SMIS project. Afterwards, the basics of general state estimation and filtering are presented in section 2.2, followed by section 2.3 on sensor noise and techniques for noise characterization. These two sections provide the basis to introduce the special characteristics of underwater navigation and localization as well as sensors required for the same, in section 2.4.
To be able to compute accuracy bounds for navigation, characteristics of the famous Posterior Cramér-Rao bounds in non-linear discrete-time filtering problems are given in section 2.5. Afterwards, the details on AUV dynamic models are discussed in section 2.6, before their use to improve navigation solutions is outlined in section 2.7. This chapter is concluded by an overview of the most relevant related work used in this thesis, in section 2.8.

## 2.1. AUV Technology

The list of AUVs used throughout the world in both science and industry is huge. This section is intended to at least give a brief overview of several example AUVs.
AUVs differ in shape size and designated application, e.g. mapping, exploration, surveillance and monitoring. In this work, only AUVs for civil purposes are considered although many military applications exist as well [HSV+03]. Usually, AUVs for deep-sea applications tend to be bigger and heavier than the ones for shallow seas. In the latter case systems exist which can be carried by a person whereas for deep-sea applications AUVs tend to weigh more than a ton due to the various deep-sea requirements, see section 2.1.1 for example. Unlike Remotely Operated Vehicles (ROV), AUVs are usually designed for longer distance operations resulting in mostly slender body shapes [CSZ11] with limited actuation dedicated to forward and turning movements [XLJ15]. In contrast, ROVs are operated via a tether for stationary operations with flexible actuation [ZC03].

AUV technology up to the year of 2001 is summarized in [Bli01] but many advances have been made since. Examples of commercially available AUVs are: Hugin [MVKS04] and REMUS [PGP+11] by the Kongsberg company of Norway, Bluefin 21 [McN14] by Bluefin Robotics, Explorer [FP00], SAUVIM [YCI+98] by Marine Autonomous Systems Engineering, Marlin MK3 by Lokheed Martin, Autosub6000 [McP09], Dorado [Kir07] by MBARI, R2D4 [UTA+07] by Mitsui Enineering and Shipbuilding, SeaOtter [Hor05]

by Atlas.
Many of these are used world-wide within the industry, some in military and in research. Many more systems exist though, e.g. research systems DEDAVE [DED] by the Fraunhofer IOSB and Infante [SP07] by the University of Lisbon. The famous Woods Hole Oceanographic Institution makes use of the REMUS AUVs.

Listing details, history and additional systems available could be treated in an entire chapter. However, as it is not important for this work, this is not considered any further.

### 2.1.1. The SMIS AUV

The SMIS AUV was built during the SMIS project as an experimental platform.
An image of the AUV being recovered during trials is given in figure 2.1.

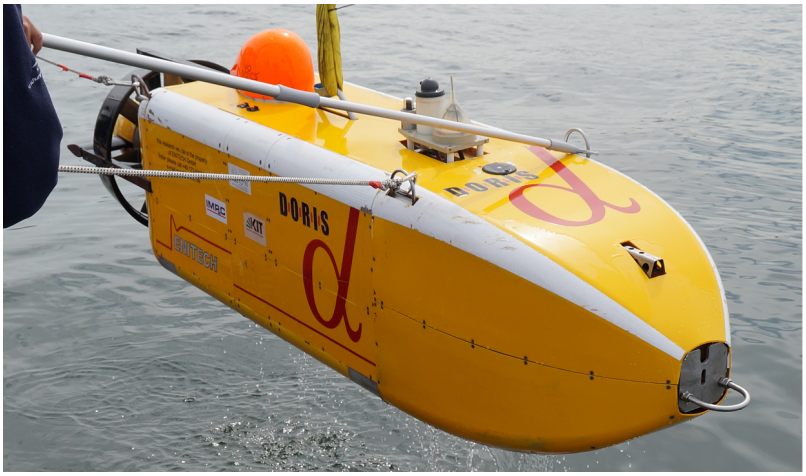


Figure 2.1.: Photograph of the SMIS AUV being recovered from the water during tests at Lake Constance in May 2016. The orange bulb in tha back covers the USBL modem for mechanical protection and hydrodynamics (Picture courtesy of IMPaC GmbH, Hamburg).

An overview of the technical specifications is given in table 2.1. Structurally, the AUV contains a stainless steel skeleton. The modular hull is made out of ABS [1]. It can easily be (un-)screwed on board of a ship for maintenance.

The thruster and rudder system is similar to the one is described in [Ole13] using a duct. Details are given in section 2.6.3.

[1] Acrylonitrile Butadiene Styrene

Table 2.1.: Technical specifications of the SMIS AUV

| Type | Dimension | Remark |
|---|---|---|
| Length | $3.47m$ | |
| Diameter | $0.72m \times 0.72m$ | of cuboid torso |
| Weight | ca. $1000kg$ | in air |
| Buoyancy | static $\sim 5 - 10kg$ | usual trimming/balancing |
| Depth rating | up to $6000m$ | |
| Main propulsion | $110V$, $3.6kW$ brushless DC | usually operated at $\sim 1kW$ |
| Max speed | $\sim 5kn$ | |
| Cruise speed | $\sim 3kn$ | i.e. ca. $1.5\frac{m}{s}$ |
| Battery | $20kWh$ LiPo | including reserve |
| Operation time | $\sim 15 - 20h$ | per charge |

For basic missions, the AUV is equipped with the AHRS model XSens MTi-700, a Doppler velocity log (DVL) Worhorse navigator WHN 600, and Evologics S2C 7/17 USBL modems for underwater communication and localization, see section 2.4.2. Additionally, the basic setting contains a CTD probe measuring conductivity, temperature and depth/pressure [Bak81]. Moreover, the AUV is equipped with a GPS receiver, WLAN and radio transceivers for localization and communication at the surface.
For positive buoyancy in (sea) water, the AUV is equipped with buoyancy syntactic foam [KP04]. Most of the AUV components possess densities above water density. Thus, the AUV would simply sink to the sea bottom without counter-measurements. Each $kg$ foam of used provided a positive net buoyancy of approximately $+0.33\,kg$ in water. Since this foam needs to counter a lot of weight from other components, the foam itself makes up a huge portion of the overall weight of the AUV in air.

The AUV uses so-called pressure-neutral technology explained in [Lüc10] and [Thi11]. This means, most of the electronics is *not* separated from water and pressure through special pressure tanks. Instead, it is cast into blocks of special silicone to prevent direct water contact and short circuits. The deep-sea pressure itself is not an immediate issue for the components up to some point - as long as they consist of solid or liquid materials only. Some capacitors must be replaced beforehand in case they contain gas filled components, though, making this technology not that easy to master and handle. Some of the AUV's interior is flooded by water during diving operations since there is no need to for a huge pressure tank.

Details of the sensors which are relevant for this work are described in chapter 3.

## 2.2. State Estimation and Filtering

In this work and in relevant literature, the technique of estimating some system state with known noisy system transition from noisy measurements (and known optional control inputs) is coined *state estimation*. Depending on the application, the system state can be anything of particular interest, usually with physical meaning such as position,

velocity or temperature just to name a few. Typically, the state is represented by a real-valued vector $\vec{x}_n$ for a (discrete) time step $n$. In engineering applications, estimating the system state is the obligatory foundation for performing control tasks [LM67]. In this work, *filtering* denotes the general process of performing recursive state estimation.
Traditionally, state estimation is performed probabilistically as in the fundamental work of Thrun et al [TBF05]. That means, both the system transition and observations are determined by their respective noise characteristics, see section 2.2.1.
Obviously, navigation and localization is a classic state estimation problem, but also system identification can be considered a state estimation problem with fixed states.
For completeness, the fundamentals of state estimation including assumptions made herein will be briefly reviewed in the following.

### 2.2.1. System Description and Assumptions

This work follows the notations, explanations an conventions used in [TBF05] and [TMN98]. The most important of these are summarized here.

*Completeness*: given the system state $\vec{x}_n$ for a time step $n$ is denoted *complete* if incorporating additional information like past states and measurements does not yield a better prediction of the following system state, i.e. it is the best predictor of the future [TBF05]. Basically, this allows to use popular strictly recursive filtering techniques on the state history described by a (first-order) Markov chain. However, care must be taken in practice to check whether this completeness assumptions justified or if it is at least a reasonable simplification.

*System transition and observation models*: In the most general form, using the completeness assumption, the system transition model is given as a conditional probability density function (pdf) $p(\vec{x}_n \mid \vec{u}_n, \vec{x}_{n-1})$ where $\vec{u}_n$ denotes the control input applied to the system at step $n-1$. The observation model is then given by the pdf $p(\vec{z}_n \mid \vec{x}_n)$ with observation $\vec{z}_n$.

Due to the prominence of the *Bayes rule* for conditional probabilities, filters operating with these assumptions and pdfs are generally called *Bayes filters*. See [TBF05] for a detailed explanation and derivation.
Usually, the pdfs are implicitly defined by discrete system and observation equations of the type

$$\vec{x}_n = f_n(\vec{x}_{n-1}, \vec{w}_n) \qquad \text{or} \tag{2.1}$$

$$\vec{x}_n = f_n(\vec{x}_{n-1}, \vec{u}_n, \vec{w}_n) \tag{2.2}$$

where $\vec{w}_n$ denotes the system transition noise. If the system equation is given as an ordinary differential equation (ODE), then equations 2.1, 2.2 must be derived through discretization from their continuous time counterparts, equation 2.4. Discretization of the systems considered in this work can be found in section 3.8.2.
Additionally, the observation equation is generally given as:

$$\vec{z}_n = h_n(\vec{x}_n, \vec{v}_n) \tag{2.3}$$

with observation noise vector $\vec{v}_n$. If the noise characteristics are known or approximated via Additive White Gaussian Noise (AWGN), see section 2.3 for details, these system transition models imply a particular density in the sense of the general system transition and observation, e.g. $p(x_n \mid u_n, x_{n-1})$ is then defined by:

$$x_n - f_n(x_{n-1}, u_n) = \vec{w}_n \sim \mathcal{N}(0, Q_n).$$

Many more variations of this basic setting exist in literature, but this is the most important one needed for this work.

### 2.2.2. Notes on Stochastic Differential Equations

In many applications, the standard system and observation equations for a continuous time system are written, e.g., as

$$\dot{x} = f(x(t), u(t), w(t)) \tag{2.4}$$
$$z(t) = h(x(t), v(t)). \tag{2.5}$$

Sometimes the time parameter $t$ is included explicitly in $f$ and often noise is assumed additive with process and measurement noise $w(t), v(t)$. The latter are usually assumed white and often modelled Gaussian. In engineering literature, this model is often referred to as *state-space representation*.

It must be mentioned here, that the mathematical operation $\frac{d}{dt}$ is in general *not* sound with respect to the noise within the equations. For correct treatment, this would require viewing the equations as a *stochastic differential equation* with the respective (huge) theory [Nis15]. In many cases, this is omitted by using a (close-to) equivalent discrete version of the equations where noise differentiation is not an issue any more.

If it is required to leave the system equations in the original continuous time ODE shape, then Kalman-Bucy filters should be considered [BJ87]. In this work however, it is assumed the system equations are given as or transformed to a discrete time representation for direct use in numerical implementations. The issues arising from the continuous-time formulation are not examined here in detail and the focus is on the (non-linear) discrete time equivalent of the respective systems as is done in practically all the literature used in this work, e.g. again [TBF05].

### 2.2.3. Short Review of Filtering algorithms

The *Bayesian Filtering* approaches are all based on the famous (recursive) *Bayes Filter Algorithm* [TBF05]:

Given $bel(x_{n-1}), u_n, z_n$ at time step $n$, compute:

$$\overline{bel}(x_n) = \int p(x_n \mid u_n, x_{n-1})\, bel(x_{n-1})\, dx_{n-1} \quad \text{[Prediction]} \tag{2.6}$$
$$bel(x_n) = \eta \cdot p(z_n \mid x_n)\, \overline{bel}(x_n) \quad \text{[Update]} \tag{2.7}$$

for each discrete time step $n$, with $\eta$ being a normalization constant and initial belief $bel(x_0)$, ideally $bel(x_0) = p(x_0)$.

The core trade-off between accuracy and computation time is usually determined by the way the belief-function $bel(x_n)$ and $\overline{bel}(x_n)$ are tracked. The choice of implementation of the Bayesian algorithm depends on the type of transition and measurements functions, noises, desired accuracy and additional assumptions. The most common of the these approaches are briefly reviewed in the following.

#### Kalman Filter and Extended Kalman Filter

Originally introduced in [K$^+$60], the Kalman Filter (KF) and its extension celebrate great popularity throughout all of robotics research and technology. This is mostly due to its simplicity and effectiveness despite requiring quite heavy assumptions on the observed system: both the system transition and observations, equations 2.1 and 2.5, are required to be linear in both state and noise. Moreover, the noise needs to be white and Gaussian. In that case, the Kalman Filter provides an optimal solution to the filtering problem w.r.t. the expected RMS error.

The Extended Kalman Filter (EKF) [KB61] used for non-linear systems works like the Kalman Filter through analytical linearization of the system transition and observation. The linearization is performed by Taylor expansion requiring the functions to be differentiable w.r.t. state. Thus, the EKF approximates the functions through a local linear form and thus the EKF is in general not optimal. Due to its simplicity and performance the EKF is still very popular in many applications, including underwater navigation examined herein. Sometimes even higher order Taylor expansions are used to approximate the functions locally.

#### Information Filter

Both the KF and the EKF can be implemented in an Information Filter (IF) type setting [Gre11], i.e. instead of using the covariance matrices $\Sigma$ their inverse $J = \Sigma^{-1}$, the information matrix (provided it exist), is used. This has several practical advantages. The IF analogues can be derived from the KF and EKF equations using the famous Woodburry Matrix Identiy, equation 1.10. With respect to the mostly theoretical considerations in this work, the IF and Extended IF (EIF) formulation are only of minor importance here.

#### Filters using Statistical Linearization

Instead of using analytical linearization as is done for the EKF, also so-called *Statistical linearization* can be performed to approximate the system transition and observations functions locally. Its most basic type is the Unscented Kalman Filter (UKF) assuming AWG noise sources, introduced in [JU97]. More recent and general approaches were

introduced in [SH13, SH14, SPH15] as the Smart Sampling Kalman Filter (S2KF) providing various advantages, e.g. the noise needs to be Gaussian but not additive.
These approaches do not require an explicit derivative of the system and measurement equations. Instead, the state distribution is assumed Gaussian as well as the (joint) distribution of state and measurement. The mean and covariance of the states are computed from (Gaussian distributed) samples which are forwarded through the system equations and evaluated afterwards.
The S2KF is similar to the Gaussian Particle Filter [KD03] where particles weighted by importance sampling are used to estimate both state means and covariances for a Gaussian distribution based on the particle distribution and weights.

#### Monte Carlo Filter (Particle Filter)

Though Monte Carlo Filtering can be achieved through several ways, the most popular used in robotics is the Particle Filter (PF) [AMGC02]. Since its Monte Carlo nature and implementation is quite similar to the approximate Cramér-Rao Bounds computed in this work, see section 2.5, the PF is quickly reviewed here.

The basic idea is to approximate the belief function by a sum of particles, i.e.

$$bel(x_n) \approx \frac{1}{M} \sum_{k=1}^{M} \delta(x_n - x_n^{[k]}) \tag{2.8}$$

with $M$ "sufficiently big" and particles $\mathfrak{X}_n = \{x_n^{[1]}, \dots, x_n^{[M]}\}$ where $\delta$ denotes the Dirac distribution. The Prediction step, equation 2.6, is then performed by simply forwarding the particles through the system transition equation:

$$\overline{bel}(x_n) = \frac{1}{M} \sum_{k=1}^{M} \delta(x_n - f(x_{n-1}^{[k]}, u_n, \epsilon_n^{[k]})), \tag{2.9}$$

where $\epsilon_n^{[k]}$ for $k = 1, \dots, M$ denotes individually sampled system noise. This yields an intermediate particle set $\overline{\mathfrak{X}}_n$:

$$\overline{\mathfrak{X}}_n = \{\overline{x}_n^{[1]}, \dots, \overline{x}_n^{[M]}\},\ \overline{x}_n^{[k]} = f(x_{n-1}^{[k]}, u_n, \epsilon_n^{[k]}).$$

Afterwards, the Update step is performed by resampling the particle set. In case *importance sampling* is used, $M$ particles are drawn from $\overline{\mathfrak{X}}_n$ based on their so-called *importance weight* $w_n^{[k]}$ by evaluating the appropriate density

$$w_n^{[k]} := p(z_n \mid \overline{x}_n^{[k]}), \quad k = 1, \dots, M. \tag{2.10}$$

In general, evaluating the density introduced to the observation model by an arbitrary observation noise influence can be tricky.

Many other resampling strategies exist though and this step is crucial for the stability and accuracy of the PF [TBF05]. Two examples are Low Variance Resampling where particles are drawn by using only one random number for efficiency. The other one is

Stratisfied Sampling: particles are grouped into subsets with a two-step sampling process proportional to group's summed weights first and individual weights secondly.

More information on PF can be found in the manifold literature, e.g. [DM98].

## 2.3. Sensor Noise Characterization

In section 2.2 it was implicitly assumed that sensor noise is modelled by some random variable with specific density and with known influence on the system. In order to examine real-world sensors, however, some concepts and tools for modelling sensor noise should be examined first which is achieved in this section. Doing this with sufficient accuracy will yield the possibility to deal with sensor noise within various filtering approaches.

Noise influences arise from various physical reasons, such as electrical, capacitive and thermal effects just to name a few. Additional technical effects, e.g. from quantization, also play a role here. Noise can take all kinds of shapes and relevance [Tuz02].

In chapter 3, it is explained how sensor noise can be modelled in practice by typically assuming a biased noise component together with an unbiased (e.g. white) noise source.

### 2.3.1. Power Spectral Density

The so-called power spectral density (PSD) is a quite common concept to characterize noise sources by a Fourier-type of analysis by, loosely speaking, estimating the noise signal's strength over different (ideally: all) frequencies.

See [Beu15], p. 428, for explanation of the name *Power* Spectral Density. The PSD's physical unit depends on the signal it is based on: the square of $x$'s original unit divided by $Hz$. Some simpler introduction on PSD can be found in several course notes, e.g. [psd].

For a given a real valued or complex signal $x(t)$, i.e. a stochastic process, let

$$\hat{x}_T(\omega) = \frac{1}{\sqrt{T}} \int_0^T x(t) \cdot e^{-j\omega t} dt. \tag{2.11}$$

Then the PSD $S_{xx}$ of $x(t)$ is defined as follows.

$$\begin{aligned} S_{xx}(\omega) &= \lim_{T\to\infty} \mathbb{E}\left[|\hat{x}_T(\omega)|^2\right] \\ &= \lim_{T\to\infty} \frac{1}{T} \int_0^T \int_0^T \mathbb{E}\left[x^*(t_1)x(t_2)\right] \cdot e^{j\omega(t_1-t_2)} dt_1 dt_2, \end{aligned} \tag{2.12}$$

provided the limit exists. $x^*$ denotes the complex conjugate of $x$. This can be considered a 'truncated' version of the Fourier Transform and taking the expected value is only necessary for signals where the classic Fourier Transform is not applicable. Otherwise,

the *energy* spectral density is used, defined the squared norm of the Fourier transform, i.e. $S'_{xx}(\omega) = |S(\omega)|^2$ where $S(\omega)$ denotes the Fourier transform of $x(t)$ if it exists.

The PSD shares an important relation with the autocorrelation function $r_{xx}$ of a stationary random process $x$:

$$r_{xx}(\tau) = \mathbb{E}\left[x(t) \cdot x^*(t-\tau)\right] \tag{2.13}$$

$$= \lim_{T_F \to \infty} \frac{1}{T_F} \int_{-T_F/2}^{T_F/2} x^*(t)x(t+\tau)dt \tag{2.14}$$

Then, the Wiener-Khinchin theorem [BD06] for stationary random processes $x$ states that:

$$S_{xx}(\omega) = \int_{-\infty}^{\infty} r_{xx}(\tau) \cdot e^{-j\omega\tau} d\tau \tag{2.15}$$

where $r_{xx}$ denotes the autocorrelation function meaning $r$ and $S$ are Fourier pairs. Note that if $S_{xx}(\omega) = s_{xx}$ constant, then $r_{xx}(\tau) = s_{xx}\delta(\tau)$, where $\delta(\cdot)$ denotes the Dirac delta function.

For *ergodic* [Beu15] stochastic processes, the auto-correlation function can be approximated by

$$r_{XX}(\tau) \approx \frac{1}{2T} \int_{-T}^{T} x(t)x^*(t+\tau)dt$$

for large $T$. This is essentially why the $1/T$ term appears in many cases in these considerations: the correlation function for ergodic processes can be approximated by a function's average over time. This follows from the definition of ergodic, i.e. the mean and covariance of the stochastic process can be estimated by taking the mean of some observed interval of this signal.

Analogously, the *cross correlation* $r_{xy}$ of two signals $x$,$y$ is defined as

$$r_{xy}(\tau) = \mathbb{E}\left[x^*(t) \cdot y(t-\tau)\right] \tag{2.16}$$

which, in practical terms, is often approximated by

$$\tilde{r}_{xy}(\tau) = \frac{1}{T-\tau} \int_{\tau}^{T} x^*(t) \cdot y(t-\tau)dt \tag{2.17}$$

This will be needed later in section 3.7.1.

### 2.3.2. White Noise

Although used widely throughout the Robotics literature, the concept of White Noise requires caution if it is supposed to be used properly in mathematical terms. In continuous time, White noise is a purely theoretical concept that does not exist in practice due requiring infinite signal energy. The concept can only be approximated in practice, e.g. by a constant PSD over finite bandwidth. Keeping this in mind and using the

required care, the theoretical concept can be transmitted to discrete time White Noise where the term "white" is less problematic. This sections is intended to explain this in more detail.

A sound formal definition for stochastic process White noise $W_t$ can be given as follows:

$$\mathbb{E}[W_t] = 0, \quad \forall t \geq 0 \tag{2.18}$$

$$cov(W_t, W_s) = R \cdot \delta(t - s). \tag{2.19}$$

Note that the covariance is defined through the Dirac function which is a formal concept and can only be approximated in real-world applications.
To receive discrete time noise values $w_k$ by formal integration of White Noise, the previous definitions yield [Wen11]:

$$w_k := \frac{1}{\Delta t} \int_{t_{k-1}}^{t_k} W(\tau) d\tau \tag{2.20}$$

$$\Rightarrow \mathbb{E}[w_k] = 0, \tag{2.21}$$

$$var(w_k) = \frac{R}{\Delta t} \tag{2.22}$$

and the $w_k$ are independently and identically distributed by White Gaussian Noise with $R$ denoting the PSD constant sometimes denoted $s_{xx}$.

### 2.3.3. Allan Variance

A simple way to characterize and estimate noise properties related to the PSD is practically achievable by the Allan Variance. This was originally designed to characterize the frequency stability in oscillators but can also be used to characterize the different noise types in inertial sensors, see [ESHN08]. Especially, Allan Variance can provide estimates of *biased* noise within these sensors. In some applications, biased noise terms can cause much greater difficulties than non-biased ones, see [AP08] as an example in a different research area.

Although, a characteristic Allan variance curve does not uniquely define the types of noises involved in producing this curve [Teh83], it can still be used as a valuable guideline when modelling the sensor's noise sources.
The basic idea is as follows: A consecutive row of, e.g. gyro, measurements at rest over several hours is recorded. These are clustered into adjacent uniform intervals by duration $T$. The mean value (integral divided by interval length) over these intervals is computed and the variance $\sigma^2(T)$ of these means is evaluated. This provides characteristic values for each $T$. Plotting these variances vs. cluster sizes $T$ in a loglog-plot provides the so-called Allan Variance plot. It is, thus, quite easy to generate and compute. Moreover, within the loglog-plot several characteristic noise and bias influences can be identified easily by checking the slopes and constants, though this identification

is still not unique. Examples of characteristic curves found in [ESHN08]. Using the signal's PSD $S_\Omega(f)$, the operation described above can theoretically be described via the equation:

$$\sigma^2(T) = 4\int_0^\infty S_\Omega(f)\frac{\sin^4(\pi fT)}{(\pi fT)^2}\,df, \qquad f = 2\pi\omega.$$

Since Allan Variance was also used and published for AHRS sensors [VBLS14] used in the evaluation part of this work, it is also employed here to specify some noise constants, see section 4.3.1.

## 2.4. Underwater Navigation and Localization

Underwater navigation is a challenging field, mostly due to the absence of GPS underwater, see [LB16] for a detailed discussion regarding the state of the art. This requires complementing basic inertial sensors by the use of rather sophisticated acoustic positioning and velocity sensors described in section 2.4.2.

Basic underwater navigation is performed using a combination of inertial measurements (acceleration and turn rates), pressure measurements (depth), velocity measurements (using the Doppler effect), acoustic positioning, see section 2.4.2, and a compass [PSSL14]. The measurements must be combined through some state estimation or filtering scheme. This will be explained in detail in chapter 3 and this section is just intended to provide an overview.

In case a map is known in advance, it can also be used to improve the navigation accuracy. This is especially favourable in special shallow water applications where a map may be known in advance, e.g. generated in previous missions or via a surface vessel. As is shown in [KC13], this can also be examined in a Bayesian setting for which Cramér-Rao Bounds, see section 2.5 can be computed.

The next step would be to use techniques like simultaneous localization and mapping (SLAM) techniques for improving the navigation accuracy. SLAM in the underwater domoin has become quite popular in the recent years, but it is mostly omitted in this work as it would break its scope. Both sonar and camera-based techniques have been employed. [NLR05] demonstrates early approach from the $2000's$ using a synthetic aperture sonar. A more recent work is given in [LNP+14] using some probability hypothesis density filter for handling multiple objects. As in [LNP+14], in [JLK+17], camera-based SLAM techniques for the UW-domain are presented. Please refer to the large number of references given in these papers for more details on underwater SLAM.

The remainder of this section will firstly present the coordinate frames used for underwater navigation, then give some introduction to underwater sensors and conclude by a sketch/outline of how underwater navigation is typically performed.

### 2.4.1. Coordinate Frames for Navigation

This work mostly follows the notation used in [Fos11], which uses the so-called SNAME convention ("Society of Naval Architects and Marine Engineers"), defining several navigation frames and names used:

*ECI*: Earth-centered inertial reference frame $\{i\}$ for earth navigation. Origin located at the center of earth with axis' not rotated by earth but being fixed in space. This frame is not truly physically inertial as it is revolving around the sun. However, in this context it is "sufficiently inertial" for the considered navigation tasks.

*ECEF*: Earth-centered Earth-fixed reference frame $\{e\}$ has the same origin as the ECI but the axes are fixed to the earth so they rotate with it. For velocities appearing in typical marine applications, the forces arising in combination with Earth rotation $\omega_e = 7.2921 \times 10^{-5} \frac{rad}{s}$ are negligible. Thus, $\{e\}$ can be considered inertial here as well. This is not true for typical airborne vehicles though.

*NED*: North-East-Down coordinate system $\{n\}$ with origin $o_n$ defined relative to the Earth's reference ellipsoid. Usually the origin moves with the craft. For marine navigation, a (local) Flat Earth Navigation system can be assumed. This means, the NED-origin is fixed to a local point and the tangent plane in this point defines a local Cartesian NED-frame which can assumed to be inertial.

*BODY*: Body-fixed reference frame $\{b\}$, also in NED-style, i.e. the $x$-axis points in direction of the vehicle's "nose", the $y$-axis points in direction of vehicle's starboard and the $z$-axis points downwards, perpendicular to its "belly". Often but not necessarily, the BODY-origin is located at the AUVs center of gravity. This frame is also sometimes referred to in lower case letters "body".

Please refer to [Fos11] and [Wen11] for further details and overviews of the frames. Figure 2.2 illustrates the BODY-frame and the naming for its respective linear and angular velocities.
Using these frames, velocities are for example given as

$$\vec{v}^{\,e}_{b/n} \;:\; \{b\} \text{ relative to } \{n\} \text{ expressed in } \{e\}$$

meaning the linear velocity of frame $\{b\}$ relative to $\{n\}$ expressed in coordinates of $\{e\}$ denoted by superscript.
In some cases, the notation of [Wen11] is used which uses a slightly different style. Velocity vectors are denoted as:

$$\vec{v}^{\,e}_{nb} \;:\; \{b\} \text{ relative to } \{n\}^b \text{ expressed in } \{e\} \tag{2.23}$$

which represents the linear velocity of frame $\{b\}$ relative to $\{n\}^b$ expressed in coordinates of $\{e\}$. The index is reversed and without slash w.r.t. previous Fossen notation. *However*, in [Wen11] the frame $\{n\}$ is usually not fixed to the earth but always fixed to the moving vessel. Here, the difference is explicitly indicated by the $b$ written in the superscript denoting an $\{n\}$ frame fixed to the BODY-origin. Refer to [Wen11] for more details. It should be noted that this difference in style and notation can cause some confusion when studying the literature.

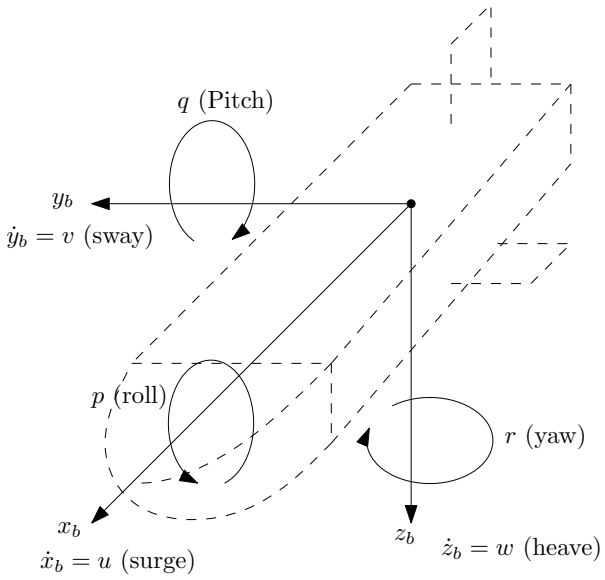


Figure 2.2.: BODY axis with velocities in the SNAME convention according to [Fos11]. Angles and angular velocities follow right-hand rule.

**Definition 3** (AUV global position, orientation and body velocity)**.** *Using Roll, Pitch, Yaw angles, the AUV's generalized global NED position and orientation $\vec{\eta}$, also referred to as* pose *in the following, reads:*

$$\vec{\eta} := (x, y, z, \phi, \theta, \psi)^T \tag{2.24}$$

*and the AUV body-fixed velocities $\nu$ are denoted:*

$$\vec{\nu} := (u, v, w, p, q, r)^T$$

*where $u, v, w$ are usually called surge, sway and heave and $p, q, r$ are called roll-rate, pitch-rate and yaw-rate.*

The herein defined location of the BODY-origin within the SMIS AUV is illustrated in figure 2.3.

If linear and angular magnitudes are to be mentioned separately, also the notation from [Fos11] is used:

$$\begin{aligned}
\vec{p}^{\,n}_{b/n} &= (x, y, z)^T \\
\vec{\Theta}^{n}_{b/n} &= (\phi, \theta, \psi)^T \\
\vec{\nu}^{\,b}_{b/n} &= (u, v, w)^T \\
\vec{\omega}^{b}_{b/n} &= (p, q, r)^T
\end{aligned}$$

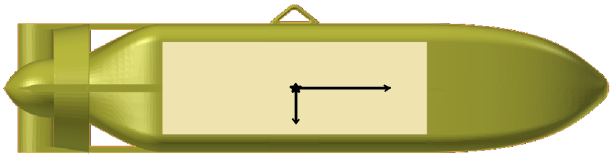

Figure 2.3.: BODY origin location (star-shaped center) within the SMIS AUV, see section 2.1.1: centralized within the AUV's longitudinal axis, below the handle (triangle shape on top). The origin's distance to the AUV's nose is ca. $1.79m$ and to its tail $1.68m$.

In order to transform between different sensor frames fixed to the AUV, some computation is necessary. For BODY-frames $\{b_1\}, \{b_2\}$ fixed to the AUV, let $R^{b_1}_{b_2}$ denote the fixed rotation matrix between these frames, given by constant angles $\vec{\Theta}_{b_1,b_2} = (\phi_{1,2}, \theta_{1,2}, \psi_{1,2})$. Then angular velocities using notation $\vec{\omega}^{b_i}_{b_i/n} = (p_i, q_i, r_i)^T$ are related through:

$$\vec{\omega}^{b_1}_{b_1/n} = T(\vec{\Theta}_{b_1,b_2}) \cdot \vec{\omega}^{b_2}_{b_2/n} \approx \begin{pmatrix} 1 & 0 & \theta_{1,2} \\ 0 & 1 & -\phi_{1,2} \\ 0 & \phi_{1,2} & 1 \end{pmatrix} \cdot \vec{\omega}^{b_2}_{b_2/n} \quad \text{(for small } \vec{\Theta}_{b_1,b_2}\text{)}$$

provided the matrix $T$ is non-singular. Since $\vec{\Theta}_{b_1,b_2}$ is fixed, the same type of equation holds for angular accelerations:

$$\dot{\vec{\omega}}^{b_1}_{b_1/n} = T(\vec{\Theta}_{b_1,b_2}) \cdot \dot{\vec{\omega}}^{b_2}_{b_2/n}$$

For $\vec{p}^{b_1}_{b_2/b_1}$ denoting the fixed coordinates of $\{b_2\}$ relative to $\{b_1\}$ given in frame $\{b_1\}$, the linear body-velocities can be translated as follows:

$$\begin{aligned} \vec{p}^{n}_{b_2/n} &= \vec{p}^{n}_{b_1/n} + R^{n}_{b_1} \cdot \vec{p}^{b_1}_{b_2/b_1} \\ \dot{\vec{p}}^{n}_{b_1/n} &= R^{n}_{b_1} \cdot \vec{\nu}^{b_1}_{b_1/n}, \quad \dot{\vec{p}}^{n}_{b_2/n} = R^{n}_{b_2} \cdot \vec{\nu}^{b_2}_{b_2/n} \\ \Rightarrow \dot{\vec{p}}^{n}_{b_2/n} &= \dot{\vec{p}}^{n}_{b_1/n} + R^{n}_{b_1} \cdot \left( \vec{\omega}^{b_1}_{b_1/n} \times \vec{p}^{b_1}_{b_2/b_1} \right) \\ \Rightarrow R^{n}_{b_2} \cdot \vec{\nu}^{b_2}_{b_2/n} &= R^{n}_{b_1} \cdot \left( \vec{\nu}^{b_1}_{b_1/n} + \vec{\omega}^{b_1}_{b_1/n} \times \vec{p}^{b_1}_{b_2/b_1} \right) \\ \Rightarrow R^{b_1}_{b_2} \cdot \vec{\nu}^{b_2}_{b_2/n} &= \vec{\nu}^{b_1}_{b_1/n} + \vec{\omega}^{b_1}_{b_1/n} \times \vec{p}^{b_1}_{b_2/b_1} \end{aligned} \tag{2.25}$$

Thus, the second derivative w.r.t. NED position is given by:

$$\ddot{\vec{p}}^{n}_{b_2/n} = \ddot{\vec{p}}^{n}_{b_1/n} + R^{n}_{b_1} \cdot \left( \vec{\omega}^{b_1}_{b_1/n} \times \vec{\omega}^{b_1}_{b_1/n} \times \vec{p}^{b_1}_{b_2/b_1} + \dot{\vec{\omega}}^{b_1}_{b_1/n} \times \vec{p}^{b_1}_{b_2/b_1} \right)$$

and the derivative w.r.t. body velocity reads as:

$$R^{b_1}_{b_2} \dot{\vec{\nu}}^{b_2}_{b_2/n} = \dot{\vec{\nu}}^{b_1}_{b_1/n} + \dot{\vec{\omega}}^{b_1}_{b_1/n} \times \vec{p}^{b_1}_{b_2/b_1} \tag{2.26}$$

**Definition 4** (Notation for estimated vs. true values)**.** *If applicable, the* estimated *coordinate frame will be denoted by a hat:*

$$\{\hat{n}\}: \textit{ estimated coordinate frame for the true frame } \{n\}$$

*The same hat notation will be used for other estimated quantities, e.g.:*

$$R_b^{\hat{n}}: \textit{estimated rotation matrix from } \{b\} \textit{ to } \{n\}$$

In the actual navigation implementation, see chapter 3, quaternions are used to represent the AUV's orientation.
However, since it is convenient to illustrate the orientation using Roll, Pitch and Yaw angles, $\phi, \theta, \psi$, they were computed from the quaternions as follows where applicable: the approximate resulting covariance $\Sigma_{RPY}$ of the Roll, Pitch, Yaw angles can be computed via a linearization $\Sigma_{RPY} = J \cdot \Sigma_q \cdot J^T$ where $J$ is the Jacobian (w.r.t. $\vec{q}$) of a function transforming a quaternion $\vec{q}$ into a RPY-vector.

### 2.4.2. Sensors for Underwater Navigation

Electromagnetic (EM) wave propagation in water is heavily impaired compared to its propagation in air or vacuum [Kar15]. The specific attenuation depends on many additional aspects like the amount and type of particles inside the (sea) water and it is strongly dependent on the EM frequency [ASSS04]. However, the node-to-node communication range is mostly limited to short-range applications [CWD+10]. Although submarine technology uses EM communication at Very Low Frequency (3-30 kHz) and Extremely Low frequency (3-300 Hz) [Moo67, UG09] penetrating sea water at several tens and even several hundreds of meter, respectively, these techniques are neither applicable nor useful for underwater robotic applications, e.g. due to the required antenna size.

Due to the EM attenuation of sea water, both communication and localization in underwater applications is mostly restricted to acoustic transmission. See [PKL07] for an overview of challenges in underwater networks compared to terrestrial (EM) networks. The most intuitive manifestation of EM attenuation underwater is indicated in figure 2.4

Moreover, the resulting absence of GPS underwater yields a whole set of challenges in underwater localization compared to terrestrial and aerial localization. See [TDSW11, Vic98] for an overview of challenges w.r.t. underwater acoustic communication and localization.
[Hol16] provides further notes and recent research on underwater electromagnetics. A description of non-acoustic sensors used for scientific water or underwater measurements to, e.g., extract chemical or biological properties of the water can be found in [FK16].
In the following, the operation of the core inertial navigation sensors will be presented.

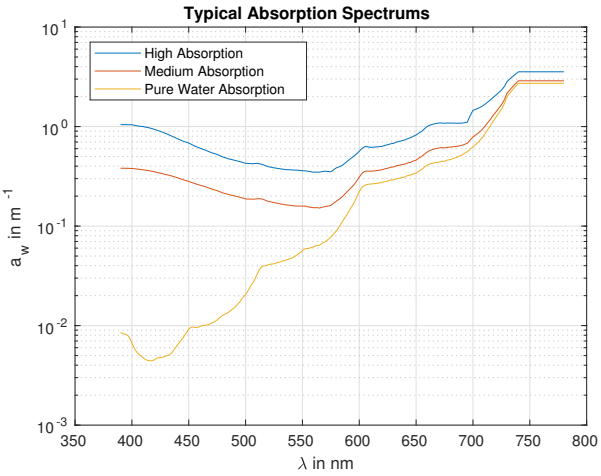


Figure 2.4.: EM attenuation of visible light in water by wavelength $\lambda$. Values based on [PS81] following an exponential attenuation $\propto exp(-a_w \cdot d)$ for distance $d$. The pure water spectrum explains why colours underwater are shifted towards blue (smaller wavelength).

### Inertial Measurements

Inertial measurements, i.e. linear accelerations (specific forces) and angular velocities are the most basic measurements used for navigation. From single and double integration, these measurements can be transformed into a position and orientation estimate, however due to inherent noise this estimate will drift heavily within a few seconds [Wen11] depending on the quality of the sensor. Nevertheless, also for underwater navigation these types of measurements are a vital component of the entire navigation scheme. Error models and details of these sensors are explained in section 3.3.

### Acoustic Velocity Measurements

A very common sensor used in underwater navigation is the so-called *Doppler Velocity Log* [WYS99]. Using the Doppler effect from acoustic signal reflections of the ground, the vehicle or device can estimate its velocity relative to the sea bottom as soon as the latter is in range. Quality depends on frequency, distance, device, configuration and angle towards the ground. Typical distances for AUV usage are within $10 - 200m$ [PSSL14]. If the water contains particles such as plankton or minerals, some of the acoustic signal will be reflected by these. In case the device can properly detect these reflections, it is common to use the term Acoustic Doppler Current Profiler (ADCP) instead to measure the approximate speed relative to the water body [FS89]. However, using this ADCP effect is much less reliable and is thus not considered in more detail in this work.

Error models and details of DVLs are presented in section 3.4.

### Acoustic Localization

The most important techniques for underwater acoustic localization from [PSSL14] are long baseline (LBL), short baseline (SBL) and ultra-short baseline (USBL) which are characterized by their baseline, i.e. the spatial distance between their receivers or transmitters [YW01]. This distance is typically in the order of 100-6000m for LBL systems, it is reduced to 3-50m for SBL systems, "shrinking" to the order of 0.1m in USBL systems. Figure 2.5 provides a schematic overview of the different techniques.

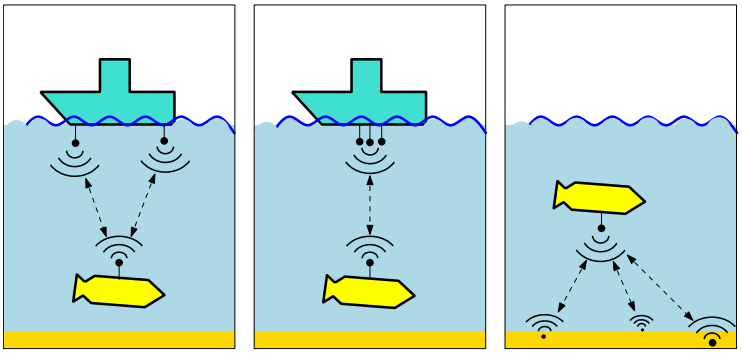

Figure 2.5.: Overview of SBL, USBL and LBL localization (left to right). Scheme according to [PSSL14]

All techniques have in common that they produce and receive acoustic signals via water. Evaluating the signal's travel time, delay and/or time differences of arrival yields information about the position of the target or receiver.

In LBL systems, previous to an AUV mission, an array of beacons is deployed to fixed localizations on the sea bed which can be used as a geo-reference after being localized sufficiently by a calibration localization procedure [YJBB07]. A underwater vehicle can use these beacons to triangulate its own position within the array. In SBL systems, the beacons are usually mounted on a ship's hull or the AUV/submerged vehicle itself (inverse SBL) in order to estimate the vehicle's position from relative time-of-arrival measurements [RPW06]. LBL is often used in deep-sea applications due to its accuracy despite the difficulty to calibrate the fixed transponders prior to the mission start. For recent advanced techniques w.r.t. LBL systems, refer to e.g. [PMdSR16] with suggestions to improve the time-of-flight measurements. Typically, the localization accuracy is reduced proportional to baseline.

The focus in this work lies on the USBL as this technology was used within the SMIS project due to their easy deployment and the additional communication abilities in the USBL modems used.

In this work, USBL modems are mostly considered w.r.t. their localization capabilities. The USBL's complementary communication capabilities, as in deliberately exchanging information messages, are just assumed as given and not considered in more detail here. For more information on that matter, please refer to [SB16] which also discusses the raw physical aspects of underwater acoustic signal propagation which in turn naturally affect the localization as well.

Physically, signal production and reception is performed through underwater microphones called *hydrophones*. Most hydrophones are based on the well-known piezoelectric effect [Gau02] in order to transform pressure change into an electric signal and vice-versa. Hydrophones with different depth ratings are commercially available worldwide, e.g. at the Teledyne company [tel]. Typically, the crutial part, the piezoelectric transducer, has a diameter of $2 - 3cm$.

In USBL modems, hydrophones are packed into rather small space, e.g. $(20cm)^3$. At least four non-planar hydrophones are required to determine the target's direction. The devices used in this work, Evologics [evo] S2C 7/17 USBL, use five hydrophones. Four of them are merely receivers and one of them is used both for transmitting and receiving signals. In USBL systems, the relative position or orientation is estimated by local phase differences from signal correlation of the received acoustic signal sent from a remote USBL device or beacon, see section 3.7.1. Additionally, the relative distance is estimated through the round-trip signal time in case of a call-response event. Figure 2.6 provides a photograph of the USBL modems. Due to its central role within AUV navigation, a


Figure 2.6.: Photograph of Evologics S2C 7/17 USBL modems by Evologics. Made during trials in the baltic sea. The left modem is the one mounted to the AUV, the right one was temporarily attached next to it for quick in-air testing (this should be done with care as it may harm the transmitters).

quick overview of its technical specifications is given in table 2.2.

Table 2.2.: Technical specifications of the Evologics S2C 7/17 USBL modems used, provided by the manufacturers manual [evo].

| **Type** | **Dimension** |
|---|---|
| Frequency band | $7 - 17kHz$, sweep-spread carrier (S2C) |
| Nominal SNR | $10dB$, required for normal operation |
| Acoustic connection | up to $6.9kbits/s$ in streaming mode,<br>$64$ Byte in single message |
| Power consumption | $2.5mW$ for stand-by<br>$\leq 1.3W$ for receiving<br>peak $10 - 80W$ for sending (configurable) |
| Size | $405mm$ total length<br>ø$175mm \times 145mm$ transducer head |
| Depth rating | $6000m$ (deep-sea Titanium housing) |
| Weight | $13.4kg$ in air, $8.9kg$ in water<br>(Titanium housing only) |
| Range up to | $9.5km$ (communication), $8.5km$ (localization)<br>under ideal conditions [NOW$^+$15],<br>but more typically up to $3km$ and above $30m$ |

The modems used in SMIS can also be acquired in two parts: the electronic housing within the metallic cylinder is then separated from the black USBL head containing the actual transducer. Both parts are then connected through a sub-sea cable. This makes it easier to integrate the device into AUVs. This type was also used in the SMIS AUV.

Internally, the modems use some outlier detection technique based on a received signal strength indicator (RSSI) and an estimated signal to noise ratio (SNR). If these measured values are not within a internal threshold, the modems will not provide positioning information.

There is a variety of physical challenges regarding the underwater acoustic channel. This includes refraction effects from varying sound velocity within different water layers, shadowing and multi-path effects [SFMC88], just to name a few.
An inhomogeneous sound velocity profile can have a big effect on the sound propagation. This can be simulated through a simple ray-tracing [CTSC10] based on Snell's law [Jen94]. That means, an acoustic ray satisfies the condition

$$cos(\beta_1) \cdot c_2 = cos(\beta_2) \cdot c_1$$

for any two given positions along the ray where $\beta_i$ is the ray's angle and $c_i$ the sound velocity at the respective depth. A sample sound velocity profile along the water column can be seen in figure 2.7 which has been computed by colleagues from the Leibnitz Institute for Baltic Research, Warnemünde (IOW) using [FM83] on the measured CTD data.

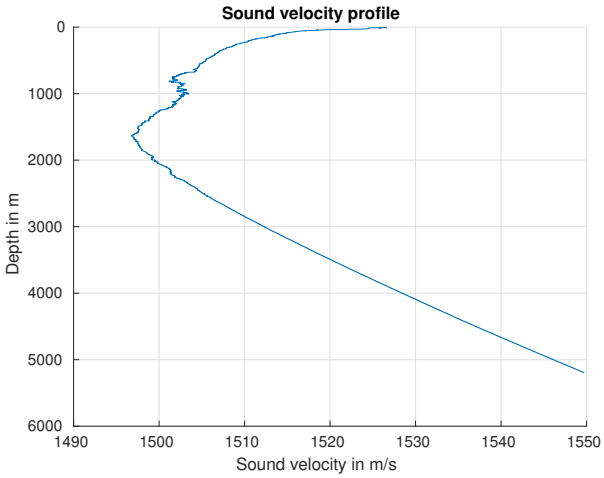


Figure 2.7.: A typical sound velocity profile in the Eastern Atlantic Ocean (courtesy of Leibnitz Institute for Baltic Research, Warnemünde). CTD data recorded in May 2014 on the research vessel Poseidon.

Figure 2.8 shows an extreme case of refraction on the sound propagation through an inhomogeneous sound velocity profile from figure 2.7. Depending on depth a small ray's initial horizontal angle $|\alpha| < 5^\circ$ will result in the ray bouncing between the edges of a depth-interval. This is due to the minimum at a certain depth, i.e. $\approx 1800\,m$ in the given example. This effect has been known since the mid 20th century as the SOFAR channel [Mil69]: Sound Fixing And Ranging channel. It is assumed that marine mammals may use this channel for long-range communication [Jan05] over $> 20\,km$.

However, in most communication and localization scenarios, where participants communicate mostly from surface to bottom or from bottom to mid-depth, the refraction effect is usually much smaller and sometimes even negligible.

This is intended as an example of one of the effects to be considered in ocean acoustics. More sophisticated sound propagation models exist, e.g. [oal], which are not considered here.

This section was intended as an introduction and overview of underwater acoustic localization technology. USBL's error models and its technical application in navigation are handled in detail in section 3.7.

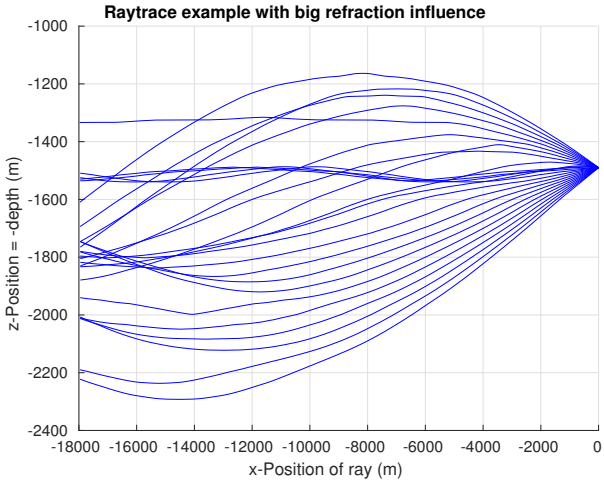


Figure 2.8.: Extreme refraction example simulated by ray-tracing based on Snell's law using the sound velocity profile from figure 2.7. Step size $\Delta_t = 0.01s$, simulation time $12s$, starting depth $1490m$ using $25$ rays in equidistant angles through narrow cone with opening angle $\approx 9.33°$. Axes are not at same scale. Simulation from [ONW+16] was used.

### 2.4.3. Navigation Outline

Classic AUV navigation using inertial and acoustic sensors is performed by estimating the AUV's navigation state consisting of [LLH+05]

- Position given as $NED$, see section 2.4.1, or as latitude, longitude and height
- Velocity of AUV relative to Earth, given in $\{n\}$ or $\{b\}$ coordinates
- AUV's orientation, given as quaternion or Roll, Pitch and Yaw
- Acceleration biases
- Turn rate (gyro) biases

It is common to use a Strapdown formulation [Wen11] for the system transition core by integrating gyroscope and acceleration measurements to receive orientation, velocity and position estimates. This means, the acceleration and turn rate measurements are integrated used to define the system transition equations. Despite not being accurate in terms of separate system and observation equations, this approach has major advantages: it allows for a simple implementation and it is applicable to any type of vehicle navigation. See section 3.8 for details.
In order to correct for accumulating position and velocity drift of Strapdown output due to the double integration, the Strapdown core is complemented by an additional state estimation scheme, usually an Extended Kalman Filter [PMdSR16] or Unscented Kalman Filter [ACC+16] which is used to incorporate velocity (DVL) and position (USBL, GPS) measurements into the state estimate. This scheme is depicted in figure 2.9.

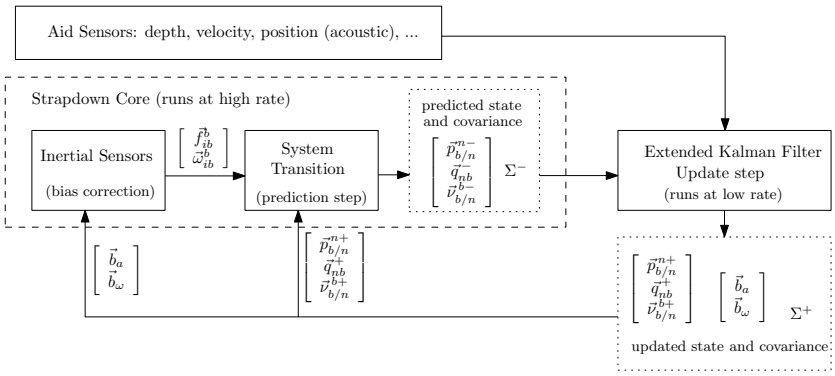


Figure 2.9.: Strapdown core integrates acceleration and angular velocity measurements at high rate while EKF corrects drift and bias over time with low rate. Figure according to the scheme from [PMdSR16]

It has to be noted in this set-up it not necessary to use an EKF as the central multi-sensor fusion core although it is quite convenient to use and provides satisfying results. Of course, PF can be used as well [Don12] but the increase in computation time is not

always worth the possible accuracy benefit except for more sophisticated settings using e.g. terrain-aided navigation [AH06].

The details of this procedure are one of the major parts in chapter 3.

## 2.5. Posterior Cramér-Rao Bounds for Filtering Problems

Comparing underwater navigation methods, section 2.4, often suffers from the lack of ground-truths. Instead, it is possible to get an idea of the achievable navigation accuracy through Posterior Cramér-Rao bounds (PCRB). Provided the mathematical modelling of the sensors is correct, the CRB can provide the "inevitable" level of uncertainty remaining in the state estimate - independent of the estimation algorithm.

The classic work in [TMN98] gives a general answer of how to derive PCRBs in non-linear discrete-time filtering for receiving an estimate of the accuracy which could be achieved for the respective state estimation. Following the notation[2] of [TMN98], the central a Cramér-Rao bound is an equation of the type seen in equation 1.9:

$$\mathrm{cov}_\theta(T) \geq J^{-1}$$

where the matrix-wise $A \geq B$ denotes that $A - B$ is positive semi-definite. An iterative lower bound in case of a discrete-time filtering approach can be computed as:

$$J_{n+1} = D_n^{22} - D_n^{21} \left(J_n + D_n^{11}\right)^{-1} D_n^{12} \tag{2.27}$$

where

$$D_n^{11} = \mathbb{E}\left[-\Delta_{x_n}^{x_n} \log p(x_{n+1} \mid x_n)\right] \tag{2.28}$$

$$D_n^{12} = \mathbb{E}\left[-\Delta_{x_n}^{x_{n+1}} \log p(x_{n+1} \mid x_n)\right] \tag{2.29}$$

$$D_n^{21} = \mathbb{E}\left[-\Delta_{x_{n+1}}^{x_n} \log p(x_{n+1} \mid x_n)\right] = \left[D_n^{12}\right]^T \tag{2.30}$$

$$D_n^{22} = \mathbb{E}\left[-\Delta_{x_{n+1}}^{x_{n+1}} \log p(x_{n+1} \mid x_n) - \Delta_{x_{n+1}}^{x_{n+1}} \log p(z_{n+1} \mid x_{n+1})\right] \tag{2.31}$$

For the popular AWGN system presented in section 2.2, omitting the control input $u_n$ here,

$$\begin{aligned} x_n &= f_n(x_{n-1}) + w_n \\ z_n &= h_n(x_n) + v_n \end{aligned}$$

with covariance matrices $Q_n$ and $R_n$ of the system noise $w_n$ and observation noise $v_n$, respectively. This yields:

[2] Omitting vector notation for now

$$
\begin{aligned}
J_n = Q_n^{-1} &+ \mathbb{E}\left[\left(\nabla_{x_n} h_n^T(x_n)\right) R_n^{-1} \left(\nabla_{x_n} h_n^T(x_n)\right)^T\right] \\
&- \left(\mathbb{E}\left[\nabla_{x_{n-1}} f_n^T(x_{n-1})\right] Q_n^{-1}\right)^T \\
&\cdot \left(J_{n-1} + \mathbb{E}\left[\left(\nabla_{x_{n-1}} f_n^T(x_{n-1})\right) Q_n^{-1} \left(\nabla_{x_{n-1}} f_n^T(x_{n-1})\right)^T\right]\right)^{-1} \\
&\cdot \left(\mathbb{E}\left[\nabla_{x_{n-1}} f_n^T(x_{n-1})\right] Q_n^{-1}\right). \qquad (2.32)
\end{aligned}
$$

In case that both the transition and observation equations can be linearized by a Taylor approximation with sufficient accuracy, at least within the region of interest, that is:

$$
\begin{aligned}
f_n(x_{n-1}) &\approx f_n(x_{n-1}^{(0)}) + f_n'(x_{n-1}^{(0)}) \cdot (x_{n-1} - x_{n-1}^{(0)}) \qquad (2.33) \\
h_n(x_n) &\approx h_n(x_n^{(0)}) + h_n'(x_n^{(0)}) \cdot (x_n - x_n^{(0)}) \qquad (2.34) \\
& \qquad (2.35)
\end{aligned}
$$

then it holds:

$$
\begin{aligned}
\mathbb{E}\left[\left(\nabla_{x_{n-1}} f_n^T(x_{n-1})\right)^T\right] &\approx f_n'(x_{n-1}^{(0)}) \\
\mathbb{E}\left[\left(\nabla_{x_n} h_n^T(x_n)\right)^T\right] &\approx h_n'(x_n^{(0)}).
\end{aligned}
$$

Here, $x_{n-1}^{(0)}$ denotes some suitable point for Taylor approximation, e.g. $x_{n-1}^{(0)} = \mu_{n-1}$ the current state mean with $F_n = f_n'(\mu_{n-1})$ and $H_n = h_n'(f(\mu_{n-1}))$ the predicted observation Jacobi.

Thus, equation 2.32 can be rewritten as:

$$
\begin{aligned}
J_n \approx \tilde{J}_n := Q_n^{-1} &+ H_n^T R_n^{-1} H_n \\
&- \left(F_n^T Q_n^{-1}\right)^T \left(J_{n-1} + F_n^T Q_n^{-1} F_n\right)^{-1} \left(F_n^T Q_n^{-1}\right) \qquad (2.36)
\end{aligned}
$$

The same equations hold if the original system equations are linearized in another manner, e.g. with the aim of receiving the same information matrices as the one in the nonlinear case, see [TMN98] (eq. (37),(38)).

Plugging equation 1.10, the Woodburry matrix identity, twice into the basic EKF equations, see section 2.2.3, yields:

$$
\begin{aligned}
\Sigma_n^{-1} &= \left(\overline{\Sigma}_n - \overline{\Sigma}_n H_n^T \left(H_n \overline{\Sigma}_n H_n^T + R_n\right)^{-1} H_n \overline{\Sigma}_n\right)^{-1} \\
&\overset{1.10}{=} \left(F_n \Sigma_{n-1} F_n^T + Q_n\right)^{-1} + \overline{\Sigma}_n^{-1} \overline{\Sigma}_n H_n^T \\
&\quad \cdot \left(\left(H_n \overline{\Sigma}_n H_n^T + R_n\right) - H_n \overline{\Sigma}_n \overline{\Sigma}_n^{-1} \overline{\Sigma}_n H_n^T\right)^{-1} \cdot H_n \overline{\Sigma}_n \overline{\Sigma}_n^{-1} \\
&= \left(F_n \Sigma_{n-1} F_n^T + Q_n\right)^{-1} + H_n^T R_n^{-1} H_n \\
&\overset{1.10}{=} Q_n^{-1} - Q_n^{-1} F_n \left(\Sigma_{n-1}^{-1} + F_n^T Q_n\right)^{-1} F_n^T Q_n^{-1} + H_n^T R_n^{-1} H_n
\end{aligned}
$$

Thus,

$$\tilde{J}_n = \Sigma_n^{-1}$$

in that case. This means the assumptions from equation 2.33 and 2.34, unsurprisingly, directly yield the optimality of the *Extended Kalman Filter* equations given by the general Cramér-Rao bounds. Further ideas on this are summarized in the following observation.

**Observation 1** (Optimality on State Estimation)**.** *For an AWGN system, achieving an optimal filtering result equal to the theoretical lower bound, given by Cramér-Rao, is equivalent to accurately computing the expected values in equation 2.32. This means, in every time step $n$, the following expressions must be computed or estimated accurately:*

$$E_f^{(\alpha)} := \mathbb{E}\left[\nabla_{x_{n-1}} f_n^T(x_{n-1})\right] \tag{2.37}$$

$$E_{ff}^{(\alpha)} := \mathbb{E}\left[\left(\nabla_{x_{n-1}} f_n^T(x_{n-1})\right) Q_n^{-1} \left(\nabla_{x_{n-1}} f_n^T(x_{n-1})\right)^T\right] \tag{2.38}$$

$$E_{hh}^{(\alpha)} := \mathbb{E}\left[\left(\nabla_{x_n} h_n^T(x_n)\right) R_n^{-1} \left(\nabla_{x_n} h_n^T(x_n)\right)^T\right] \tag{2.39}$$

Here, $\alpha$ indicates that this holds only for the popular but restrictive AWGN case. However, although the Cramer-Rao bound is only dependent on expected values in terms of mean and covariance, actually computing these values can be tedious and, in general, already requires knowledge of the entire distribution of $x_{n-1}$ meaning the entire distribution needs to be tracked rather than the expected values themselves. Thus, as is widely known, it is vital for the estimation algorithm to always keep an accurate track of the state probability, i.e. of the belief functions $bel(x_n)$ and $\overline{bel}(x_n)$ [TBF05]. How this is achieved exactly represents the core trade-off between performance and accuracy, e.g. EKF-simplification vs. a PF.

Moreover, this specific formulation is only applicable if the respective function $f_n$, $h_n$ are differentiable w.r.t. $x_{n-1}$, $x_n$.

### 2.5.1. Disadvantages of PCRB and Possible Alternatives

Computing PCRBs for a specific and rather sophisticated problem such as underwater navigation is almost never feasible analytically. Instead, they must be approximated using Monte Carlo simulations [ŠKT01]. In that case, all computed bounds are just *approximations* of the true PCRBs.
This is also what was done in this work. Whenever PCRB appears in this work, it is exclusively meant with regards to *approximate* PCRBs.

In some cases it can be more suitable to put a specific filtering scheme into the Monte Carlo loop in order to test its performance, see [MOS13]. In case of underwater navigation this means putting the EKF into the Monte Carlo loop since it is still the predominant filtering schemed applied in that domain.
In the simulations regarding the Model-Aided navigation accuracy, an EKF was used in the loop. Although this does not justify the name PCRB any more, it provides a practically more useful result than the pure PCRB. This is because it also includes how well the common navigation algorithm EKF can cope with MA navigation as an extension to an already existing navigation solution.

## 2.6. Dynamic Models for AUVs

As illustrated in [HHJ07], dynamic modelling of underwater vehicles is based on the classic work about solids moving through a liquid H. Lamb [Lam32] and G. Kirchhoff dating back to the $19th$ century. This led to the term *added mass* - a key concept of hydrodynamic modelling which will be discussed later. A few more historical notes can be found in the introduction of [HHJ07].

The state-of-the-art concept used for dynamic modelling of the AUV today and in this work are based on the well-known ordinary differential equations (ODEs) presented in [Fos11].

$$\dot{\vec{\eta}} = \mathbf{J}(\vec{\eta})\vec{\nu} \tag{2.40}$$

$$\mathbf{M}\dot{\vec{\nu}} + \mathbf{C}(\vec{\nu})\vec{\nu} + \mathbf{D}(\vec{\nu})\vec{\nu} + \vec{g}(\vec{\eta}) = \vec{\tau} \tag{2.41}$$

According to [Fos11], dynamics can be divided into two parts: the general geometry of moving objects, equation 2.40 - called *kinematics* - and the theory of forces acting on the body causing the motion - called kinetics, equation 2.41. For a detailed derivation of these equations, please refer to [Fos11].
The AUV pose $\vec{\eta}$ and body velocity $\vec{\nu}$ were introduced in section 2.4.1, see definition 3. The matrix

$$\mathbf{J}(\vec{\eta}) := \begin{pmatrix} R_b^n & 0_{3\times 3} \\ 0_{3\times 3} & T(\vec{\Theta}_{b/n}^n) \end{pmatrix} \in \mathbb{R}^{6\times 6}$$

is used for a general transformation from body- to NED-velocities. $T(\vec{\Theta}_{b/n}^n)$ was introduced in section 2.4.1. The shape is similar when quaternions are used instead of Euler

angles. This matrix and equation 2.40 is implicitly contained in the basic navigation algorithm presented in section 3.8 and will not receive detailed attention in this chapter.

An overview of the remaining expressions used in equation 2.41 is given in table 2.3

Table 2.3.: Overview of expressions used in the dynamic model, equation 2.41. Vectors in $\mathbb{R}^6$, matrices in $\mathbb{R}^{6\times 6}$

| Symbol | Name and explanation |
|---|---|
| $\mathbf{M}$ | *System inertia matrix*: mass, inertia and added mass. |
| $\mathbf{C}(\vec{\nu})$ | *Coriolis-centripetial matrix*. |
| $\mathbf{D}(\vec{\nu})$ | *Damping* and *Lift* from hydrodynamic drag etc. |
| $\vec{g}(\vec{\eta})$ | Gravitational and buoyancy forces and torque. |
| $\vec{\tau}$ | Actuator forces and torque. |

Usually, $\vec{\tau}$ also includes the forces applied by wind and waves. However, since only sub-sea models are considered here, $\vec{\tau}$ exclusively stands for the forces of control inputs, i.e. thrusters and rudders.

There are many ways to specify the model components. Most notably, the most challenging modelling parts can be identified as added mass, the damping model and the actuator forces. This will be explained in the following sections.

**Remark 1** (On terminology "dynamic model")**.** *In case the term "dynamic model" is used in the following, it is almost exclusively meant with regards to its second part, the kinetics, as this can be considered the core dynamic model.*

The dynamic model (kinetics) can be further divided into an active and a passive part. The former describes the modelling of the actuators represented by $\tau$ where the latter denotes the AUV's interaction with the surrounding liquid as a moving rigid body independent of actuation: $\mathbf{M}$, $\mathbf{C}(\vec{\nu})$, $\mathbf{D}(\vec{\nu})$ and $\vec{g}$.
Note that equation 2.41 is a six-dimensional sum of forces and torques. It has the character of a mass-damper-spring system with mass matrix $\mathbf{M}$, a damping matrix $\mathbf{D}(\vec{\nu})$ and $\vec{g}$ representing a spring through the torque by a separate gravity and buoyancy center.

### 2.6.1. Passive Part of Dynamic Model

**Definition 5** (Forces and Torque)**.** *Forces and torque with respect to the BODY-frame, see section 2.4.1, are often abbreviated by upper case letters:*

$$\vec{F} := (X, Y, Z, K, M, N)^T \in \mathbb{R}^6$$

*Here, $X, Y, Z$ denote forces and $K, M, N$ denote torque w.r.t. the body axes.*

Since each term of the dynamic model represents a force acting on the AUV as a rigid body, each component can be considered a separate force of unknown non-linear origin. It is common to approximate these forces by lower order terms, e.g. linear or

(cross-)quadratic. Using the SNAME convention, these dependencies are denoted by sub-scripts, e.g.:

$$Y_{example} = Y_{\dot{u}} \cdot \dot{u}$$

denotes a force within the BODY $y$-axis caused by surge acceleration $\dot{u}$. Theoretically, these parameters describing the linear or cross-quadratic influences might for example be derived from a Taylor series expansion of the "true" idealized mathematical model. However, the latter cannot be derived practically so approximations like this are necessary.

### Mass, inertia and added mass

The total mass matrix is composed of the mass/inertia matrix $\mathbf{M}_{RB}$ and the so-called *added mass matrix* $\mathbf{M}_A$

$$\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A$$

$$\mathbf{M}_{RB} = \begin{pmatrix} m \cdot I_{3\times 3} & -m \cdot \left[\vec{r}_g^{\,b}\times\right] \\ m \cdot \left[\vec{r}_g^{\,b}\times\right] & I_b \end{pmatrix} \tag{2.42}$$

$$I_b = I_g - m \cdot \left[\vec{r}_g^{\,b}\times\right]^2 \tag{2.43}$$

where $\vec{r}_g^{\,b}$ denotes the AUV's center of gravity in BODY-coordinates and $I_b$ is the inertia matrix w.r.t. the BODY-origin, $I_g$ the one w.r.t. center of gravity [Fos11].
$\mathbf{M}_{RB}$ is symmetric and positive definite [MW14], $\mathbf{M}_A$ is symmetric and positive semi-definite. Note that due to the skew-symmetry of the cross-product term $\left[\vec{r}_g^{\,b}\times\right]$, the upper right part of $\mathbf{M}_{RB}$ requires a negative sign for $\mathbf{M}_{RB}$ to be symmetric.

The AUV's mass and inertia is rather straight-forward to handle. Methods for deriving the inertia matrix can be found in basic literature [Gol11]. In theory, the parameters can be derived or at least estimated through help of the CAD[3] models of the AUV. This assumes though that all the sensors, electric cables, batteries etc. should at least be roughly known w.r.t. their mass and position.

**Remark 2** (On basic mass and inertia parameters)**.** *Due to changed payload sensor devices being replaced "live" on-board of the RV as well as different resulting trimming for positive buoyancy (by buoyancy syntactic foam, see section 2.1.1), there will almost always remain some uncertainty in the parameters: mass, inertia and even center of gravity/buoyancy to a varying degree.*
*Since the SMIS AUV is using pressure neutral technology, section 2.1.1, a significant amount of water is contained within the AUV's hull during operation. This water needs to be accounted for as well in mass and inertia terms which is problematic as the rigid body assumption does not hold in this regard.*

[3] Computer-Aided Design

The added mass matrix $\mathbf{M}_A$ comes from Kirchhoff's original equations about kinetic energy in the rigid body interacting with the surrounding fluid [CGN05].
For simplicity, it can be explained as the mass associated with (the energy of) moving water surrounding the AUV. This liquid is moving since it must make way for the AUV's nose and then close again behind the AUV's back. Thus, a certain kinetic energy is associated with this surrounding water - transferred by the AUV to the liquid. It should be intuitively obvious that finding a general description of the kinetic energy in such a system is quite tough - including (fluid) simulations of this setting.
In Kirchhoff's and Lamb's [Lam32] original formulation, the total kinetic energy is expressed through the quadratic terms $\frac{1}{2}\vec{\nu}^T\mathbf{M}_{RB}\vec{\nu}$ - kinetic energy of the rigid body - and $\frac{1}{2}\vec{\nu}^T\mathbf{M}_A\vec{\nu}$ - kinetic energy of the surrounding fluid [Fos11]. The key to estimating the entire energy is then to estimate a simplified version of the added mass matrix:

$$
\begin{aligned}
-\mathbf{M}_A &= \begin{pmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{pmatrix} \\
&\underset{\text{symm.}}{\overset{\text{AUV}}{=}} \begin{pmatrix} X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 & 0 & 0 & Y_{\dot{r}} \\ 0 & 0 & Z_{\dot{w}} & 0 & Z_{\dot{q}} & 0 \\ 0 & 0 & 0 & K_{\dot{p}} & 0 & 0 \\ 0 & 0 & M_{\dot{w}} & 0 & M_{\dot{q}} & 0 \\ 0 & N_{\dot{v}} & 0 & 0 & 0 & N_{\dot{r}} \end{pmatrix}
\end{aligned}
\tag{2.44}
$$

Here, the sign before $\mathbf{M}_A$ in equation 2.44 was just inserted for consistency with the notation in [Fos11]. It is not relevant for understanding this section.
In general, the matrices' components will *not* be constant but depend on many other influences - especially when waves are included. For simplicity, in the AUV modelling domain, it is however mostly assumed these parameters remain constant for diving missions beyond wave and wind influences.
The added mass values can be estimated analytically involving the AUV's shape. One of the earlier works describing this is [Iml61]. Similar equations can be found in [Ver09], [Pre01b] and [Pre01a]. Additionally, there is some commercially available software like WAMIT [LN06] providing the added mass quantities - even for rather complex structures such as ships where wave and wind influences cannot be neglected [Fos11].
The simplification made in equation 2.44 is valid if the AUV's hull is symmetric in the $xy$-plane and $xz$-plane [Fos11]. In case of the SMIS AUV, the former holds only with reasonable accuracy - due to the USBL modem and antennas on the AUV's top. It is possible to even just use the diagonal part of the matrix assuming a fully decoupled motion [MW14].

The Coriolis terms can also be derived from the original equations by Kirchhoff. For brevity, this is not explained in more detail. Instead, only the shape of $\mathbf{C}(\vec{\nu})$ is given

from [Fos12]:

$$\mathbf{C}(\vec{\nu}) = \mathbf{C}_{\mathbf{M}}(\vec{\nu}) = \begin{pmatrix} 0_{3\times 3} & -[(M_{11}\vec{\nu}_1 + M_{12}\vec{\nu}_2)\times] \\ -[(M_{11}\vec{\nu}_1 + M_{12}\vec{\nu}_2)\times] & -[(M_{21}\vec{\nu}_1 + M_{22}\vec{\nu}_2)\times] \end{pmatrix} \quad (2.45)$$

$$\text{where } \mathbf{M} = \mathbf{M}^T = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix}$$

Here, $\vec{\nu} = (\vec{\nu}_1^T, \vec{\nu}_2^T)^T$ describes the linear and rotational components of $\vec{\nu}$.

#### Notes on the Munk Moment

In general, for slender stream-lined bodies, it holds that $|X_{\dot{u}}| < |Y_{\dot{v}}|$. This means, unlike in the mass matrix $\mathbf{M}_{RB}$, the first three entries of the diagonal are not equal as they are in equation 2.42. This difference results in a moment of the type $(X_{\dot{u}} - Y_{\dot{v}})uv$ which is mathematically entering the equation through the Coriolis-terms. This moment is called the *Munk moment* which is known to have a destabilizing effect on the system dynamics [AC02]. In simulation, this effect can be quite devastating when damping coefficients are too small to attenuate the effect of this moment. This may yield quite chaotic trajectories in contrast to the actual smoother physical data.

On the one hand, when engineering an underwater vehicle, the Munk moment must always be considered in order to maintain enough stability in the vessel's movement. On the other hand, for model identification purposes it must be kept in mind that matching the added mass terms to appropriate damping coefficients can be a quite delicate matter.

#### Hydrodynamic Damping

There are numerous effects causing hydrodynamic damping such as skin friction, vortex shedding [Ach74] and lifting forces. These effects are neither easy to identify nor to model in detail. For further reading, please refer again to the fundamental work of Fossen [Fos11] and to the original work of Hoerner et al. [HB85] which provides a mixture of quite extensive theoretical and experimental research on lifting effects of bodies moving through air or liquid.

Explaining these effects in more detail is way beyond the scope of this work. Instead, only a brief description is given here for the major parts of damping used throughout the literature. As is known from undergraduate physics classes [Fal11], a solid body moving through a liquid (or gas) experiences some resistance against its movement, called *drag*. Depending on the type of resulting flow around the body, this resistance force is approximately proportional to the relative speed in case of a laminar flows or the relative squared speed in case of turbulent flows. Proportional constants depend on area projected in the moving direction, the fluids density and the Reynolds number [Fal11].

This justifies the use of non-linear drag $\mathbf{D}_n(\nu)$ (mostly assumed as quadratic) and linear drag $\mathbf{D}_l$ added by a lifting component $\mathbf{L}(\vec{\nu})$:

$$\mathbf{D}(\vec{\nu}) = \mathbf{D}_n(\vec{\nu}) + \mathbf{D}_l + \mathbf{L}(\vec{\nu}) \tag{2.46}$$

There are many ways to model the non-linear part. One option is to use the *full quadratic drag* is given as:

$$\mathbf{D}_n(\vec{\nu}) = \begin{pmatrix} |\vec{\nu}|^T \cdot \mathbf{D}_{n1} \\ |\vec{\nu}|^T \cdot \mathbf{D}_{n2} \\ |\vec{\nu}|^T \cdot \mathbf{D}_{n3} \\ |\vec{\nu}|^T \cdot \mathbf{D}_{n4} \\ |\vec{\nu}|^T \cdot \mathbf{D}_{n5} \\ |\vec{\nu}|^T \cdot \mathbf{D}_{n6} \end{pmatrix}$$

where $|\vec{\nu}|$ denotes the component-wise absolute value of $\vec{\nu}$. The matrices $\mathbf{D}_{ni}, i = 1, \ldots, 6$ have size $6 \times 6$. As such, using this full quadratic drag model requires up to $216$ parameters to be identified. [MW14] presents results on how to identify the full $216$ drag terms from (tank) experiments for a Remotely Operated Vehicle (ROV) moving at slow speed. However, in most AUV applications a lot of these (cross) drag parameters can be ignored [Fos11].

**Remark 3** (Damping parameter choice)**.** *For AUVs, it is common to simplify both the non-linear (quadratic) and the linear drag to a single matrix with the same shape as given in equation 2.44, see [Fos11] and [dSTMdS07] for example.*
*In this case:*

$$\mathbf{D}_n = \begin{pmatrix} X_{|u|u}\,|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{|v|v}\,|v| & 0 & 0 & 0 & Y_{|r|r}\,|r| \\ 0 & 0 & Z_{|w|w}\,|w| & 0 & Z_{|q|q}\,|q| & 0 \\ 0 & 0 & 0 & K_{|p|p}\,|p| & 0 & 0 \\ 0 & 0 & M_{|w|w}\,|w| & 0 & M_{|q|q}\,|q| & 0 \\ 0 & N_{|v|v}\,|v| & 0 & 0 & 0 & N_{|r|r}\,|r| \end{pmatrix}$$

*and accordingly for the linear drag:*

$$\mathbf{D}_l = \begin{pmatrix} X_u & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_v & 0 & 0 & 0 & Y_r \\ 0 & 0 & Z_w & 0 & Z_q & 0 \\ 0 & 0 & 0 & K_p & 0 & 0 \\ 0 & 0 & M_w & 0 & M_q & 0 \\ 0 & N_v & 0 & 0 & 0 & N_r \end{pmatrix}$$

*If not mentioned otherwise, this shape for linear and non-linear drag is used throughout this work.*

Following the notation from the beginning of this section, the parameters appearing in $\mathbf{D}_{ni}$ are named, e.g. $X_{|v|r}$ for the coupled influence of absolute sway velocity $|v|$ together with yaw-rate $r$ as a force along the $x$-axis. In [HHJ07] even more parameters are

included, e.g. $X_{vr}$ without the absolute value of $v$ or $r$. The selection of these further parameters is often driven by heuristic approaches. One important addition in this direction is given by the *body lift force and moment* [LM98] also appearing in [Pre01a] for example. This yields additional forces and moments of the type

$$\vec{l}_F^* = \begin{pmatrix} 0, & Y_{uvl} \cdot uv, & Z_{uwl} \cdot uw, & 0, & M_{uwl} \cdot uw, & N_{uvl} \cdot uv \end{pmatrix}^T$$

which are caused by the AUV moving through the water with a certain sideslip angle [Fos12]. I.e. a force and torque resulting from a coupled surge- and sway-motion relative to the water. This force is pushing and turning the slender AUV body within the direction of movement such that the AUV nose will be pointing into the actual direction of movement, the vector of AUV velocity relative to the water body. The equation is valid for small sidelsip angles. It can be basically explained as the same effect that is exploited in rudders for manoeuvring, see section 2.6.3.

For a slightly more general approach, we express the body lift and moments through a matrix, following the recommendation in [dSTMdS07]:

$$\mathbf{L}(\vec{\nu}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{uv} & 0 & 0 & 0 & Y_{ur} \\ 0 & 0 & Z_{uw} & 0 & Z_{uq} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & M_{uw} & 0 & M_{uq} & 0 \\ 0 & N_{uv} & 0 & 0 & 0 & N_{ur} \end{pmatrix} \cdot u$$

$$\vec{l}_F = \mathbf{L}(\vec{\nu}) \cdot \vec{\nu}.$$

This means a positive surge speed will stabilize the AUV in its motions in such a way that it will be forced to turn its nose into its direction of travelling. This is in agreement with physical observations of appropriate body shapes moving through water. Moreover, this can be considered a counter-weight to the aforementioned destabilizing Munk Moment within this section.
The lift effect is crucial for sensible modelling of the AUV's pitch motion: through the forward speed, the AUV is able to maintain a rather stable constant pitch angle despite the present gravity/buoyancy moment which is pushing the AUV to return to its "equilibrium" pitch angle. Usually, this angle is small but typically $> 0^\circ$, e.g. $5^\circ$, in case the AUV needs to dynamically balance a positive buoyancy to maintain a constant depth at constant surge speed. In simulation, the lifting effect will prevent the AUV from making an unrealistic oscillating pitch movement to balance between rudder forces/moments and the gravity/buoyancy moment.
Especially with regards to damping coefficients and in control applications, it is common to decouple the system into a longitudinal ($u$, $w$, $q$, $\theta$) and lateral subsystem ($v$, $p$, $r$, $\phi$, $\psi$) [VMMH$^+$13].

The choice of parameters with respect to drag is not unique in literature. For example, the additional parameters used in modelling a slender-shaped AUV in [HHJ07], e.g. $X_{vr}$ and the body-lift, are not contained in the model of the slower-moving cubic-shaped ROV in [MW14].
As such, the model will of course be dependent on the type of vehicle. Also, the overview

section of [Fos11] on AUV and ROV models, section 7.5.5 therein, does not explicitly mention the lift terms used in [HHJ07] which are derived from *potential theory* [Hoe65]. Instead, in the explanation of influences for hydrodynamic damping, section 6.4 of [Fos11], it is argued that the different effects (lift, friction etc.) are difficult to separate and thus, it is convenient to subsume them in the general shape of $\mathbf{D}$ given in equation 2.46.
As is noted in [HHJ07], the precise choice of the model will mostly be driven by its performance and accuracy in practice.

#### Gravity and buoyancy

The gravity and buoyancy influence can be described easily in case their origins $\vec{r}_g^{\,b}$ and $\vec{r}_b^{\,b}$ are known as well as the AUV's weight force magnitude $W = mg$ and buoyancy force magnitude $B = \rho V g$ with displaced volume $V$, water density $\rho$ and gravity constant $g$.

$$\vec{f}_g^{\,n} := (0, 0, W)^T, \quad \vec{f}_b^{\,n} := (0, 0, -B)^T$$
$$\vec{g}(\vec{\eta}) = -\begin{pmatrix} R_n^b(\vec{f}_g^{\,n} + \vec{f}_b^{\,n}) \\ \vec{r}_g^{\,b} \times R_n^b \vec{f}_g^{\,n} + \vec{r}_b^{\,b} \times R_n^b \vec{f}_b^{\,n} \end{pmatrix}$$

As is explained in [MW14], the full gravity/buoyancy parameters are not observable from experimental data. Instead, the gravity/buoyancy vector is expressed through

$$d_{BW} := B - W$$
$$\vec{\gamma} := \vec{r}_g^{\,b} \times R_n^b W \hat{z} - \vec{r}_b^{\,b} \times R_n^b B \hat{z}$$

where $\hat{z}$ denotes the unit vector pointing down in the global NED frame. Here, $d_{BW}$ denotes the (usually positive) *net buoyancy* of the vehicle in Newtons and $\vec{\gamma}$ denotes the moment applied to the AUV by the combined effect of gravity and buoyancy on the vehicle. As such, the gravity/buoyancy is reduced to four parameters which can be observed from experimental data through the relation:

$$\vec{g}(\vec{\eta}) = -\begin{pmatrix} -d_{BW} R_n^b \hat{z} \\ \vec{\gamma} \times R_n^b \hat{z} \end{pmatrix}$$

Please refer to [MW14] for further details.

#### Water current influence

How to deal with further influences such as wind, wave and currents can be found in [Fos12]. Since the AUV model is only considered for underwater applications, only the current influence is described here.

In AUV applications, the influence of water currents is usually heavily simplified from the physical complexity which it contains. Typically, for AUV applications the water current is modelled as an irrotational fluid as in [Fos12]:

$$\vec{\nu}_c = (u_c, v_c, w_c, 0, 0, 0)^T$$

where $\vec{\nu}_c^b = (u_c, v_c, w_c)$ denotes the linear fluid velocity in body frame. The current velocity in NED frame is then received through:

$$\vec{\nu}_c^n = R_n^b \vec{\nu}_c^b$$

Moreover, in some applications the current is considered to be stationary, i.e. constant during the considered time. Otherwise it can be simulated and/or modelled via a first-order Gauss-Markov process in NED-frame, see [Fos12], and estimated online[4].

In later chapters of this work, only a constant water current in NED-frame is estimated for the test data sets, see section 4.5.3.
In case a sufficiently accurate dynamic model exists, this model can in theory be used to estimate ocean currents from velocity and attitude measurements, see [FXCZ16] for a numerical simulation.
Note that usually, the presence of ocean currents requires more advanced methods for planning and control. A typical Lyapunov based approaches for dynamic positioning in the presence of ocean currents can be found in [AP07]. Plenty of relevant literature exists for AUV motion control. However, as this is not that relevant in this work, it is not considered in more detail. See section 4.5.5 for more details on AUV motion control.
When ocean currents are considered, the original equation(s) 2.41 are slightly adjusted using the vehicle velocity $\vec{\nu}_r = \vec{\nu} - \vec{\nu}_c$ relative to the (moving) water body. From [Fos11], [Fos12] and [HHJ07] we can then write:

$$\mathbf{M}_{RB}\dot{\vec{\nu}} + \mathbf{C}_{RB}(\vec{\nu})\vec{\nu} + \mathbf{M}_A\dot{\vec{\nu}}_r + \mathbf{C}_A(\vec{\nu}_r)\vec{\nu}_r + \mathbf{D}(\vec{\nu}_r)\vec{\nu}_r + \vec{g}(\vec{\eta}) = \vec{\tau}.$$

This means, under current influence the mass and inertia effects of the rigid body are separated from the ones referring to the added mass and all hydrodynamic effects are expressed in the water-referenced velocity $\vec{\nu}_r$ instead of $\vec{\nu}$.

### 2.6.2. Frame of Reference

From a physical perspective, it is most intuitive to use the center of gravity as the origin of the dynamic model, equation 2.41. However, from an operational point of view, it is more practical to use a fixed point of the body (e.g. body origin) since the center of gravity changes with the vehicle load. When switching to a different frame, the parameters appearing in 2.41 change as well. This has implicitly been done already in equation 2.42 and equation 2.43. Generalizing this, using the equations (3.25f) from [Fos11], this can also be done for the added-mass matrix $\mathbf{M}_A$. Suppose $\mathbf{M}_A$ is defined e.g. w.r.t. center

[4] this is by far not a simple task and will be explained later.

of gravity denoted by $\mathbf{M}_A^{RB}$. Then the added mass matrix in center of origin, $\mathbf{M}_A^{CO}$, can analogously be computed through:

$$\mathbf{M}_A^{CO} = \mathbf{H}^T(\vec{r}_g^b) \cdot \mathbf{M}_A^{RB} \cdot \mathbf{H}(\vec{r}_g^b) \quad \text{, where}$$
$$\mathbf{H}(\vec{r}_g^b) = \begin{pmatrix} I_{3\times3} & S^T(\vec{r}_g^b) \\ 0_{3\times3} & I_{3\times3} \end{pmatrix}$$

This allows to recompute the matrices into different frames for convenience.

### 2.6.3. Actuation: Thruster and Rudders

Depending on the desired taks and vehicle, many different settings of thrusters and rudders are possible. If precise dynamic positioning using a ROV is desired than it is favourable to use at least two thrusters along all body axis for full and easy 6DoF manoeuvring [THZG10]. For vehicles moving mostly straight ahead, e.g. for generating maps of the sea-floor, under-actuation is common [XLJ15].
The SMIS AUV is equipped with similar actuators as classic ships tend to have: a (ducted) propeller followed by rudders. Since both yaw motion and pitch motions for diving are required, the SMIS AUV is equipped with two coupled pitch/depth and two coupled yaw rudders. See figure 2.10 for an overview of the thruster-rudder configuration. Other actuation concepts are possible, e.g. replacing the rudders by more thrusters

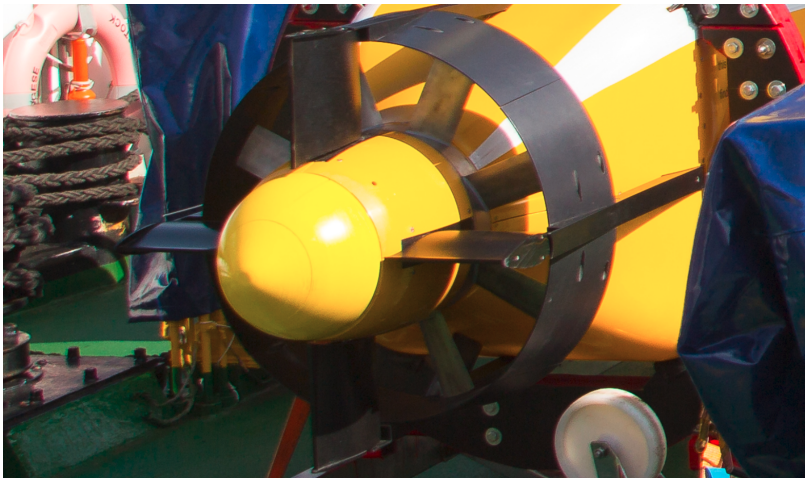

Figure 2.10.: Photograph of the thruster and rudder configuration at the SMIS AUV's rear: a silver/metallic ring motor propeller in a black duct followed by black rudders (Picture courtesy of Dr. David Kaiser from IOW).

or by using an adjustable thruster (or duct) called azimuth thruster [Fun09].

Naturally, the actuation topic is close to the question of AUV motion control. The mapping of control forces into the actuator commands or set points is coined allocation [FJP09].

Thruster modelling itself can be tedious [KC06] and can be considered a research field of its own. In this work, simplified models from literature are used which are supposed to provide sufficient accuracy for the most important influences on AUV dynamics.
Modelling is generally easier if the thrusters are not directly placed next to the rudders. However, the SMIS AUV set-up allows the AUV to perform slow turns even without forward speed yielding an increased manoeuvrability at the surface, e.g. for easier positioning when being lifted out of the sea.

### Thruster

The foundation of thruster modelling is usually a Bernoulli tube yielding equations of pressure and flow velocity under several assumptions like incompressibility [WY99]. Using linear momentum theory, ODEs can be derived using thrust, torque and RPM as variables, see [KC06] for various non-linear models with experiments. For simplicity just the equations from [HHJ07] is used. Force generated through the thruster is then given as:

$$\begin{aligned} F_T &= T_{|n|n}|n|n - T_{|n|u}(1-w)|n|u_r \\ &= T_{|n|n}|n|n - T_{|n|u1w}|n|u_r \end{aligned} \tag{2.47}$$

where $w$ denotes the wake fraction number and $n$ denotes the thruster's RPM. For simplicity, it is assumed constant and as such included in the second parameter $T_{|n|u1w}$. Both parameters are greater than zero. Note that a positive surge speed will result in a reduced force applied by the thruster. In equation 2.47, $w \in (0, 1)$ denotes the wake fraction number. Typical values are $0.1 - 0.2$ [SSF04]. In the following it is assumed that $w = 0.2$.

The lift generated by the rudder is dependent on the (axial) flow velocity of the surrounding water arriving at the rudder. Since the rudder is located behind the thruster, this velocity called $v_{wake}$ will in general not be equal to the relative surge speed $u_r$. Figure 2.11 illustrates the idealized velocities appearing in this context. Water will flow inside the thrusters front - usually $v_{in} = (1-w)u_r$, then it will receive some acceleration by the thruster due to a pressure gradient. This results in some axial flow speed $v_p$ within the thruster's center and will be increased to $v_{out}$ [Kim09] directly behind the thruster. Until it reaches the rudder, some it will lose some of it's former (theoretical) velocity and reach the rudder with velocity $v_{wake}$ [BLF00]. Note that this consideration does not take into account if there is a negative surge or something similar. This is neglected here as the AUV should always move at positive surge.
More sophisticated thruster models can be found in e.g. [PJS09].

The mentioned velocities are not simple to measure in practice, especially during AUV operation. As such, they should be estimated from known values: RPM and surge speed.

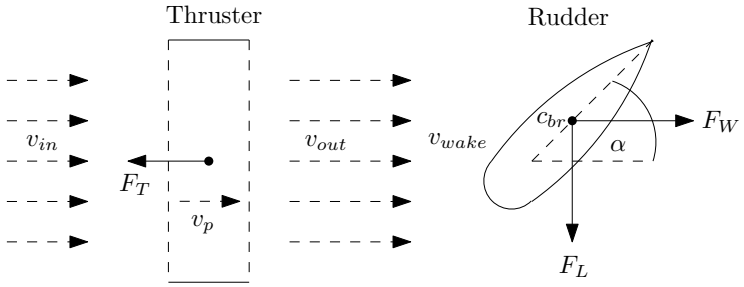


Figure 2.11.: Overview of some physical quantities for operation of thrusters and rudders, i.e. flow velocities and forces.

Using water density $\rho$, propeller area $A_p$ (assumed to be a disc), the velocities can be computed as follows.
Assuming $v_{in} = (1 - w) \cdot u_r$ for the propeller intake velocity, the following model equations are assumed [Kim09]:

$$v_{out} = \sqrt{\frac{2F_T}{\rho A_p} + v_{in}^2} \tag{2.48}$$

Axial flow velocity from [Kim09] and [WY99]:

$$v_p = \frac{1}{2} \cdot (v_{in} + v_{out}) \quad \text{(axial flow velocity)}$$

This provides estimates of $v_p, v_{in}, v_{out}$ and $v_{wake}$ from thrust and surge speed. As such they can be expressed through RPM and surge speed by equation 2.47.

The AUV's propeller rotation will result in torque $Q_T$ as a roll moment on the AUV. Through a simplified power equality assumption [WY99] this can also be estimated by:

$$Q_T \cdot \frac{n \cdot \pi}{30} = F_T \cdot v_p \cdot c_{Q_T}.$$

Although the equation given there uses the assumption of zero ambient flow velocity, it still makes sense from a simplified physical point of view as a force $F_T$ moving through water with speed $v_p$ is equal to the respective propeller's torque $Q_T$ moving at angular velocity $n \cdot \pi/30$. This estimate will be used to model propeller torque applied to the AUV. The effect of which can be observed in the form of a constant roll angle offset during AUV operation. The constant $c_{Q_T}$ is included for model fitting such that this rather strict assumption for the torque is slightly alleviated.
Alternatively, a similar expression as equation 2.47 can be used to model the thruster's torque, see [FB00]. However, despite the expected gain in accuracy this requires estimating/identifying two additional parameters as in equation 2.47.

To estimate the wake velocity, the formula from [SSF04] combined with equation 2.48 is used here:

$$v_{wake} = u_r + k_u \cdot \left( \sqrt{\frac{2F_T}{\rho A_p} + v_{in}^2} - v_{in} \right)$$

with $k_u \approx 0.5$ for rudders close to the propeller [SSF04]. In case the expression within the square root should become negative, then this term is set to $0$.

It must be noted that all the equations used in this section are simplifications of physically more complicated relations. This holds especially for estimating the propeller's torque and wake velocity [Car12].

### Rudders

The (idealized) lift and drag forces from the rudders can be modelled by the well-known equation:

$$F_L = c_W(\alpha) \cdot \frac{\rho}{2} \cdot v_{wake}^2 \cdot A_R(\alpha) \tag{2.49}$$

$v_{wake}$ can be replaced with the surge speed $u_r$ in case it is not placed directly behind the propeller. $A_R(\alpha)$ denotes the rudder's surface projected into flow direction. The drag and lift force will attack in the rudder's center of (hydrodynamic) pressure, see figure 2.11. For simplicity it is assumed this point is lying close to the rudder's axis of rotation (shaft) and can be ignored. Note that in practice this difference results in a small additional moment on the shaft which needs to be accounted for in low-level control of the rudder's angle [Ole13].

The drag and lift coefficients called $c_w(\alpha)$ values are explained, e.g. in the famous original work done by NASA who published $c_w$-tables for special rudder/wing profiles called NACA[5] air-foils [JWP33] and later [Som97]. They represent a proportionality factor to determine the lift and drag forces arising from an (pitched) rudder in water flow or a wing in air flow. The exact influence of $c_w$ on drag and lift is given in equation 2.49.

For a pitch/depth/height-rudder the resulting 3d force and torque in body-frame will then be typically given as

$$\vec{F}_L^{pitch} = \begin{pmatrix} -F_D, & 0, & -F_L \end{pmatrix}^T, \quad \vec{M}_L^{pitch} = \vec{l}_{rudder}^{\,b} \times \vec{F}_L^{pitch} \tag{2.50}$$

with rudder position $\vec{l}_{rudder}^{\,b}$ in body-frame. $\vec{M}_L$ is the resulting moment. For the yaw rudder

$$\vec{F}_L^{yaw} = \begin{pmatrix} -F_D, & -F_L, & 0 \end{pmatrix}^T, \quad \vec{M}_L^{yaw} = \vec{l}_{rudder}^{\,b} \times \vec{F}_L^{yaw} \tag{2.51}$$

with sign before $F_L$ depending on the rudder's angle convention.
By this convention and equations 2.50, 2.51, positive rudder angles will result in a diving motion and a turn towards the AUV's right, respectively.

[5] National Advisory Committee for Aeronautics

See section 4.5.2 for a derivation of the rudder coefficients from experimental data in SMIS.
In section 3.9 it is described how the AUV dynamic model was implemented in Simulink.

### 2.6.4. Notes on System Identification

The most general field dealing with the issue of generating a (e.g. dynamic) model of a (technical) system is called *System Identification*. A thorough discussion of this topic can be found in the classic work [Lju98].

Estimating model parameters for an assumed model structure from experimental data is a sub-topic of the science of System Identification. It also mainly deals with the issue of generating models (e.g. for control tasks) where no previous model structure is known or assumed at all.

In [Lju10], several perspectives and recommended future research in the field were discussed. Among several others, Ljung recommended to make more use of intersection with other seemingly different but structurally similar fields, e.g. Machine Learning. A more detailed recent analysis of this can be found in [Nel13].

Matlab provides a System Identification Toolbox, but it was not used on a notable level in this work.

As the complete theory is unnecessarily heavy-weighted for the context needed in this work, it is not considered in detail here.

#### As an Inverse Problem

Form a mathematical point of view, system identification can be considered as an Inverse Problem. An extensive research for its application to model parameter identification can be found in [Tar05].

However, the theory of Inverse Problems, especially with regards to so-called ill-posed problems goes beyond what is required in this AUV model context. As such, it is not employed in more detail and just left for the reader as a reference to tackle similar problems where the Inverse Problem theory might be mandatory.

## 2.7. Model-Aided Navigation

The concept of combining the worlds of dynamic modelling with inertial navigation is not uniquely named in literature. Usually it is coined Model-Aided navigation [HHG07] or (embedded) vehicle dynamics aiding [MOSV07] or just navigation aided by vehicle models [CGTH13]. As [HHG07] is closest to this work, it was chosen as the preferred terminology in the following.

The origin of Model-Aided navigation (MA navigation), i.e. complementing an inertial navigation system through a kinetic model, is often credited to [MK91] and [KBI99]. Although it dates back a few years already, it is not that common to use it in the robotics field. A rather sophisticated and new approach for aerial vehicles is given in [CSST14]. A rare example within the underwater domain for underwater gliders can be found in [WSY13]. It must be noted though that the classic probabilistic robotics approach would be the other way around: derive a dynamic model used for system transition and guide this system estimate through external measurements. However, since deriving this model can be quite tedious and the result may lack accuracy, inertial navigation systems often employ a very general system transition model [Wen11] as mentioned in section 2.4.3.

According to [CGTH13] there are two basic classes of MA navigation. On the one hand, the Koifman and Bar-Itzhack [MK91, KBI99] system structure using the vehicle model as an external virtual sensor. On the other hand, the Vasconcelos [VSOG10] system structure where the vehicle model is integrated into the EKF system equation. Since it is easier to extend an already existing inertial navigation system by the virtual sensor based on the dynamic model, only the first type was examined in this work. The outline of this approach is shown in figure 2.12 Note that [CGTH13] presents an approach to

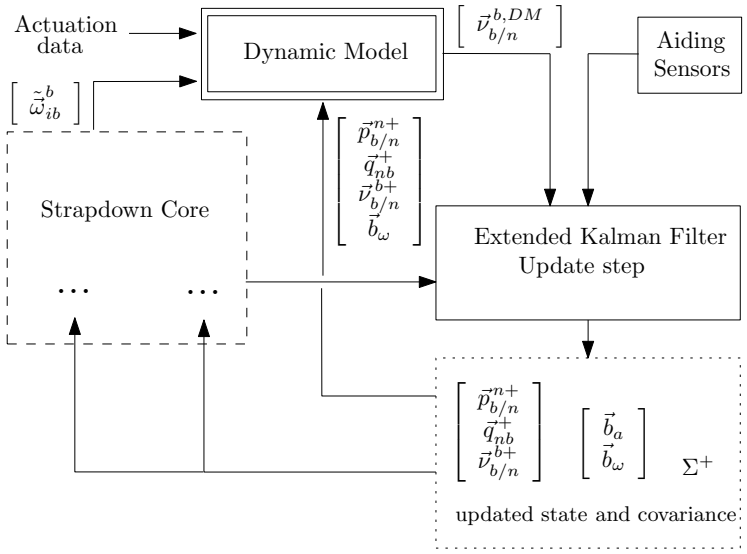


Figure 2.12.: Model-Aided navigation according to Koifman and Bar-Itzhack system structure [MK91]. Extension of figure 2.9. The vehicle model is integrated as an external aiding sensor. In general, the dynamic model output does not need to be limited to virtual velocity measurements.

unify both types of MA navigation.

The most straight-forward use of MA navigation according to figure 2.12 would be to

initiate the dynamic model through the current state estimate. Then model is simulated for a short time span, e.g. $0.25s$, using the known actuation set points and then use the final model output as the virtual measurement to be fed back into the filtering algorithm. This is how MA was implemented in this work.

There are a few theoretical and practical issues arising from MA navigation. First of all, as can be seen in figure 2.12, it is not clear how to model the virtual measurements' noise. This can be solved by modelling the noise level in the same or similar way as it was seen within the ground-truth during training or identification of the model - which is how it is achieved in this work. Moreover, the usual measurement assumptions of the measurement only depending on the current system state may not be correct. To account for these types of correlations, [HH11] proposed to introduce additional bias terms into the virtual measurement. This can be modelled as a Random Walk or a first order Gauss-Markov process, see section 3.2.

Of course, using a model which does not reflect the true behaviour of the vehicle motion or which is simply not accurate enough can make the entire navigation solution diverge. The same can happen if external disturbances such as the correct water current are not predicted or included in the model.

It is possible to use not only the linear velocities from the dynamic model as a virtual measurement as proposed in [Heg10]. However, explained in the same work, this is problematic as then the same source of information will provide two types of correlated virtual measurements making it difficult from a mathematical perspective how to model these correlations. Also, trusting too much into this source in the presence of noise and general modelling uncertainties can easily make the filter solution worse or even diverge.
Moreover, not all quantities given by the dynamic model output can be used easily in the filtering equations. Since for Strapdown formulations the turn rate is *not* an explicit part of the navigation system state, it is not straight-forward how to include additional virtual measurements of the turn rates into the system.

The choice of how to model all these effects in this work are based on simple rules which can easily be integrated into a real world system, consistent with [HH11] with several extensions. More details are given in the evaluation, section 4.7.3.

## 2.8. Overview of Related Work

The literature associated with the specific topics of this thesis can be found in the respective subsections. However, this section is supposed to clarify the main literature which was most influential for this work and to which this work relates to the most.

Based on inertial navigation described in [Wen11], navigation considered in this work is hugely based on the reviews in [PSSL14] and [LB16]. Ongoing challenges in AUV navigation were picked up from [KEW06] and [HMH07]: multi-vehicle navigation, improved near-bottom navigation, advances in state estimation research, improved navigation in the mid-water zone, increased level of autonomy including terrain-based techniques such as SLAM, were the focus was chosen on increased level of autonomy due to

MA navigation. The same technique is also useful for mid-water navigation [MWP+16] to replace DVL measurements.

The most prominent related work and source influencing the general treatment and understanding of the AUV's dynamic model is given by Fossen in [Fos11] and [Fos12] for treatment of ocean currents. Moreover, the work of Martin et al on ROV model identification [MW14] and [MW16] strongly influenced the choice and evaluation of the dynamic model. Despite not being directly applicable to a faster moving AUV with limited actuation, the insights from these papers are useful for both theoretical and practical considerations.

Considering the USBL modelling, this work is mostly related to the work of Morgado et al: [MOSV06] to survey basic schemes for USBL operation. The work in [MOS10] is closely related to the USBL model presented in section 3.7 which led to insights on how to best integrate the USBL equations into the EKF navigation scheme.

At its core, this work can be considered an extension and completion of both the work by Hegrenaes on Model-Aided navigation for AUVs as well as the [MOS13] which is examining PCRBs in the context of different USBL evaluation schemes in shallow water without DVL. In [MOSV14], similar MA navigation techniques were examined using the Vasconcelos [VSOG10] MA structure in a simulated experiment without DVL. The dynamic model identification scheme applied in this work is also based on the practical work in [HHJ07]. Hegrenaes' et al main contribution in his series [HHG07, HBH08, HH11] and his thesis [Heg10] was to introduce the concept of MA navigation in a practically usable manner for a real-world AUV. His foundation was used in this work for a feasibility analysis of this technique in deep-sea using various extensions.

The biggest extensions towards the work mentioned therein are:

- Focus on deep-sea applications
- The use of a full 6-DoF model unlike 3-DoF in [HH11]
- Insert a 3-DoF virtual velocity measurement instead of 2-DoF in [HBH08]
- Examine the benefit of MA navigation with focus on full sensor availability as opposed to sensor failure mostly evaluated in [HH11]
- Apply PCRB techniques similar to the work from [MOS13] on basic and MA navigation in deep-sea.

In [MHS+15], the dynamic model is integrated into the KF system equations instead of being treated as an external aiding sensor. Please refer to this work if needed as it was not considered in more detail here.

The general concept of PCRB is based on Tichavsky's [TMN98] and Bergman's work [Ber99], accompanied by the Monte Carlo simulation approach presented in [ŠKT01]; applied to underwater navigation as in [MOS13] and in [MOS09].

Further literature in related fields: Despite not being the same as navigation, the literature on tracking was also considered [LJ01] [LJ03]. The basic setting of estimating a robots position without the availability or with outliers in GPS is also relevant in other

fields, e.g. a UAV exploring indoors [DCD15], where it is popular to use techniques like stochastic cloning [RSS+13].

### 2.8.1. Related Work of SMIS Colleagues

Several fellow researchers were involved in the SMIS project.

At the point of submitting this work, S. Neumann was also working on his PhD thesis [Neu18] in the same domain with focus on multi-robot cooperation.
Putting that work in a nutshell: a cooperative localization approach for networks of mobile underwater agents is developed, where each agent benefits from overheard data traffic in the network. By measuring the angle of the incoming data packet via USBL and integrating the measurements with location informations in the package head, the algorithm estimates the global position of the listening agent.

Besides performing various other examinations on USBL and its error model, S. Neumann evaluated the position of the hydrophones within the USBL modems used - partly based on experimental data. The hydrophone positions were re-used in this work, see section 3.7.2.

For an overview of other collaborative publications made within the SMIS project, refer to section 1.1.1.

# 3. Sensor Error Models, Navigation Details and Implementation

This chapter is dedicated to details of sensor error models, Strapdown navigation and implementation outlines of these topics including the dynamic model implemented in Simulink [MAT17], MA navigation and the framework to compute the approximate PCRBs.

After briefly reviewing a few more details and notations on general filtering in section 3.1, the importance of modelling and being aware of biased noise terms in measurements is emphasized in section 3.2 presenting common bias models.
Then, the error models of the sensors needed for underwater navigation which were introduced in section 2.4 are presented and discussed with focus on how to include them into the navigation equations. The list is sorted in descending order of state derivative measured. Beginning with acceleration and turn rate measurements provided by the AHRS/IMU in section 3.3 (including magnetic compass), the error model of the main velocity sensor - the DVL - is given in section 3.4. Afterwards, the position sensors error models are presented. Beginning with the pressure sensor for depth measurements in section 3.5, a brief model of GPS measurements used for surfacing is given in section 3.6. Finally, the main part of the error models is dedicated to the important acoustic positioning via USBL devices in section 3.7 including theoretical discussions in the spirit of Cramér-Rao as well as the final chosen error model, its limitations and possible extensions.
In section 3.8 the mathematical details of Strapdown navigations are given including the necessary discretization of the continuous terms.
Finally, sections 3.9, 3.10 and 3.11 provide details on how the AUV dynamic model, the MA navigation scheme and the evaluation of approximate PCRBs were implemented, respectively.

## 3.1. Further Notes on Filtering

Before moving on to the sensor error models, this section is intended to provide a few more insights and details on filtering.

[TMN98] can be adapted for the general non-linear filtering problem formulated in section 2.2 if the control input $u_k$ is modelled explicitly. This is not strictly necessary for the considerations in the subsequent sections but it may yield some insight on the general filtering equations. Following the notation with explicit use of a control vector input

$U_n$

$$X_n = (x_0, \ldots, x_n), \quad Z_n = (z_0, \ldots, z_n), \quad U_n = (u_1, \ldots, u_n),$$

and treating $u_{n+1}$ like an additional system state having influence on the transition from $x_n$ to $x_{n+1}$, it can be derived:

$$\begin{aligned} p_n := p(X_n, Z_n, U_{n+1}) &\overset{1.1}{=} p(x_0, u_1) \cdot p(z_0 \mid x_0, u_1) \cdot \\ &\quad p(x_1, u_2 \mid x_0, u_1, z_0) \cdot p(z_1 \mid x_0, u_1, z_0, x_1, u_2) \cdot \\ &\quad (x_k, u_{k+1} \mid X_{k-1}, U_k, Z_{k-1}) \cdot p(z_k \mid X_k, U_{k+1}, Z_{k-1}) \cdot \\ &\quad \ldots . \\ &\quad p(x_n, u_{n+1} \mid X_{n-1}, U_n, Z_{n-1}) \cdot p(z_n \mid X_n, U_{n+1}, Z_{n-1}) \end{aligned}$$

Using the completeness and Markov assumption, most terms can be simplified:

$$\begin{aligned} p(x_k, u_{k+1} | \mid X_{k-1}, U_k, Z_{k-1}) &= p(x_k, u_{k+1} \mid x_{k-1}, u_k) \\ p(z_k \mid X_k, U_{k+1}, Z_{k-1}) &= p(z_k \mid x_k, u_{k+1}) \end{aligned}$$

for $k > 0$ and $k > 1$, respectively. Note that in case a specific system and observation equation is already given, see section 2.2.1, like the typical state-space model, the latter simplifications follow implicitly from the equations. This yields

$$\begin{aligned} p(X_n, Z_n, U_{n+1}) &= p(x_0) \cdot \prod_{k=1}^{n} p(x_k, u_{k+1} \mid x_{k-1}, u_k) \cdot \prod_{k=0}^{n} p(z_k \mid x_k, u_{k+1}) \\ &= p(X_{n-1}, Z_{n-1}, U_n) \cdot p(x_n, u_{n+1} \mid x_{n-1}, u_n) \cdot p(z_n \mid x_n, u_{n+1}) \end{aligned}$$

If needed, using this notation the control input $u_k$ can be considered a completely random and independent state vector that can be treated as a close-to-exact observation in every time step before it influences the following state.

As explained in e.g. [ŠKT01], estimation theory considered here can be categorized. Given observations at discrete time steps $1, \ldots, k$ we denote three different settings via:

- *Prediction*: estimate future state(s) at time step(s) $k+1, k+2, \ldots$;
- *Filtering*: estimate current state at time step $k$;
- *Smoothing*: estimate past states at time steps $1, \ldots, k-1$.

The emphasis in this work is on filtering with some applications of smoothing. Usually, when filtering is applied, the state estimate can be somewhat unsteady or discontinuous in case, e.g. periodic position estimate arrive only every 10th time step. Informally speaking, for post-processing this effect can be alleviated through smoothing which will make the state estimation history look more continuous.
For smoothing, the method by Rauch-Tung-Striebel (RTS) [RTS+65] was employed due to its simplicity in implementation. It works in two steps. First, a forward (Kalman) filter estimate is computed and afterwards a similar, easily computed estimation is performed backwards. See [RTS+65] for the original equations.

## 3.2. Sensor Bias

When dealing with sensors like IMUs and magnetic compasses, keeping track of the sensor bias is crucial. This requires being aware of two additional effects: the actual bias offset, compensated by a constant, and the bias drift, i.e. change of the bias over time. Both quantities must be known to be able to effectively estimate the bias terms.
In some cases the drift is negligible such that only the bias offset is important which may be calibrated before the actual filtering mission.
If a drift needs to be considered, the bias $b$ is often modelled as a first order Gauss-Markov process:

$$\dot{b} = -\frac{b}{\tau_b} + n_b$$

with time constant $\tau_b$ and white (Gaussian) noise $n_b$. Otherwise a simple Random Walk can be used:

$$\dot{b} = n_b.$$

Please note that just as in the system transition equations, section 2.2, using the time derivative $\dot{b}$ in combination with a Random Variable in form of the $n_b$ must be considered with mathematical care. Please refer to section 2.2.2 and 2.3.2 for explanation.

In terms of system theory this means the noise source $n_b$ is either sent through a first-order low-pass filter in the Gauss-Markov case

$$B(s) = N_b(s) \cdot \frac{1}{s + \frac{1}{\tau_b}},$$

or integrated over time in the random walk case

$$B(s) = N_b(s) \cdot \frac{1}{s},$$

where

$$B(s) := \mathcal{L}\{b\}(s), \quad N_b(s) := \mathcal{L}\{n_b\}(s)$$

denote the Laplace transforms. Gauss-Markov processes are popular in many electronic applications. However, due to the specific characteristics of the AHRS used in SMIS, see the Allan variance part in section 4.3.1, this work uses mostly Random Walk bias models.

The reasons for biased noise components are diverse, but one of the most common sources is temperature dependency, see section 2.3 for more details. In case of MEMS sensors, see section 3.3, there is always a gravity dependent bias commonly given as $g$ and $g^2$ dependent parameter. These effects are not modelled specifically in this work for simplicity and only the generic bias models mentioned above are used. Further information on bias in can be found in [Gel15, Wen11].

This section is concluded with emphasising the importance of dealing with bias terms. Unknown or unmodeled bias influences can heavily worsen the filter results up to the point of divergence [Fit71].

## 3.3. AHRS/IMU Error Model

An Inertial Measurement Unit (IMU) provides inertial measurements: angular turn rates from gyroscopes and linear accelerations or more accurately the so-called specific force including gravity [Wen11]. Many IMUs use Microelectromechanical systems (MEMS) [Jud01] for the physical realization of measurements. An AHRS system complements IMU measurements with magnetic measurements to generate a stabilized attitude information of the vehicle's orientation. The details of how to model this sensor is given in the following.

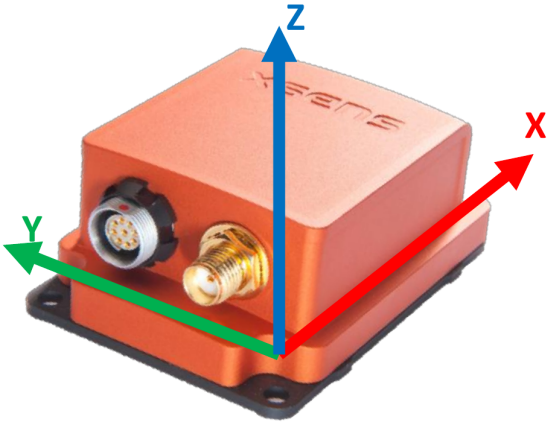


Figure 3.1.: IMU/AHRS unit, XSens MTi-100-series, used within the SMIS project including default local coordinate frame of the device. Measurements include linear accelerations, turn rates, magnetic measurements and GPS. The latter is performed via the optional golden antenna connector on the right, available at the surface only. (Image taken from [YSe12], courtesy of XSens Technologies B.V. )

For the measurement of angular turn rate and linear accelerations, taken by the gyroscope and the linear accelerometers, the following model is assumed:

**Definition 6** (AHRS Measurement Equations)**.** *The measured angular rate $\tilde{\vec{\omega}}_{ib}^{b}$ and measured specific force $\tilde{\vec{f}}_{ib}^{b}$ of the AHRS are assumed to be physically generated as [Wen11]:*

$$\tilde{\vec{\omega}}_{ib}^{b} = \vec{\omega}_{ib}^{b} + \vec{b}_{\omega} + \vec{n}_{\omega}$$
$$\tilde{\vec{f}}_{ib}^{b} = \vec{f}_{ib}^{b} + \vec{b}_{a} + \vec{n}_{a}$$

*where $\vec{\omega}_{ib}^{b}$ denotes the true turn rate, $\vec{b}_{\omega}$ denotes the sensor bias and $\vec{n}_{\omega}$ denotes the sensor noise. The noise is assumed WGN with usually simplified covariance matrix $\sigma_{\omega}^{2} I_{3\times 3}$ and analogously for the specific force: $\sigma_{a}^{2} I_{3\times 3}$.*

*Here, possible non-orthogonal configurations of the sensor axis are neglected.*

For high-quality sensors such as a Ring-Laser-Gyroscope the bias drift can be neglected for many applications ranging within e.g. less than an hour. The actual quasi-constant bias term needs to be calibrated through a start-up routine before the filtering mission. Otherwise, the bias drift can be modelled e.g. as a Random Walk or a first order Gaussian-Markov process as presented in section 3.2.

**Corollary 3** (Rotation matrix approximation for small estimation errors)**.** *When the difference between estimated orientation $\{\hat{n}\}$ and true orientation $\{n\}$ is small, quantified by angles $\vec{\Psi} = (\alpha, \beta, \gamma)^T$, a simplification based on Taylor expansion for* sin *and* cos *is common:*

$$\begin{aligned} R_b^{\hat{n}} &= R_{\hat{n}}^{\hat{n}} \cdot R_b^n \\ &\approx \left( I_{3\times 3} + \left[ \vec{\Psi} \times \right] \right) \cdot R_b^n \qquad (3.1) \\ &= \begin{pmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{pmatrix} \cdot R_b^n \qquad (3.2) \end{aligned}$$

Using the notation and explanation from [Wen11], measuring the turn rates via an IMU is physically governed by the equation:

$$\vec{\omega}_{nb}^b = \vec{\omega}_{ib}^b - R_n^b \left( \vec{\omega}_{ie}^n + \vec{\omega}_{en}^n \right)$$

In most marine or sub-sea applications, the Earth's rotation rate $\vec{\omega}_{ie}^n$ and the transport rate $\vec{\omega}_{en}^n$ can be omitted though which is assumed in the following. Note that this is not true for fast vehicles such as planes.

Using Euler angle notation, the orientation change is then received through:

$$\frac{d}{dt} \begin{pmatrix} \Phi \\ \theta \\ \Psi \end{pmatrix} = T(\Phi, \theta, \Psi) \cdot \vec{\omega}_{nb}^b$$

with $T(\Phi, \theta, \Psi)$ given in section 2.4.1. Due to singularities arising for $\theta = \pm 90^\circ$, it is favourable to use either the (simplified) equations by Bortz [Bor71]:

$$\begin{aligned} \dot{\vec{\sigma}} &\approx \vec{\omega}_{nb}^b + \frac{1}{2} \vec{\sigma} \times \vec{\omega}_{nb}^b + \frac{1}{12} \vec{\sigma} \times \left( \vec{\sigma} \times \vec{\omega}_{nb}^b \right) \qquad (3.3) \\ &\approx \vec{\omega}_{nb}^b + \frac{1}{2} \vec{\sigma} \times \vec{\omega}_{nb}^b \qquad (3.4) \end{aligned}$$

for the orientation vector $\vec{\sigma}$ with angle "encoded" as $|\vec{\sigma}| = \sigma$. This holds for small $\sigma$ and this formulation is useful to keep track of small angular increments over short-term intervals. Otherwise the respective quaternion formulation can be used:

$$\begin{aligned} q_b^n &= \begin{pmatrix} \cos(\frac{\sigma}{2}) \\ \sin(\frac{\sigma}{2}) \cdot \frac{\vec{\sigma}}{\sigma} \end{pmatrix} \in \mathbb{R}^4 \qquad (3.5) \\ \dot{q}_b^n &= \frac{1}{2} q_b^n \bullet \begin{pmatrix} 0 \\ \vec{\omega}_{nb}^b \end{pmatrix} \qquad (3.6) \end{aligned}$$

In the further work, the focus lies on this quaternion representation.
For completeness, the update of the Rotation matrix itself is described by:

$$\dot{R}_b^n = R_b^n \cdot \left[\vec{\omega}_{b/n}^{\,b}\times\right] \tag{3.7}$$

Analogue to the approximate rotation matrix error, equation 3.1, the influence of a faulty rotation matrix estimation can be expressed as:

$$\begin{aligned}\dot{R}_b^{\hat{n}} &= \frac{d}{dt}(R_n^{\hat{n}} \cdot R_b^n) \\ &= \dot{R}_n^{\hat{n}} \cdot R_b^n + R_n^{\hat{n}} \cdot R_b^n \left[\vec{\omega}_{b/n}^{\,b}\times\right]\end{aligned}$$

where $\dot{R}_n^{\hat{n}}$ represents the *rate of change regarding the orientation estimation* vs. the true rotation matrix. Since, $\dot{R}_n^{\hat{n}} \approx \frac{d}{dt}\left(I + \left[\vec{\Psi}\times\right]\right) = \frac{d}{dt}\left[\vec{\Psi}\times\right]$, this change is roughly equal to (or bounded by) the drift rate of the orientation tracking system in general. In [PSSL14] even in bad cases this is given as, e.g. $60°h^{-1}$ or $0.0001°h^{-1}$ for an excellent sensor. As $60°h^{-1} \hat{=} 2.91 \cdot 10^{-4}\text{rad}\,s^{-1}$, it follows for $\frac{d}{dt}\vec{\Psi} \approx 2.91 \cdot 10^{-4} \cdot (1,1,1)^T$:

$$\|\dot{R}_n^{\hat{n}}\|_2 \leq \|\dot{R}_n^{\hat{n}}\|_F \approx (6 \cdot (2.91 \cdot 10^{-4})^2)^{\frac{1}{2}} \frac{\text{rad}}{s} = 7.128 \cdot 10^{-4} \frac{\text{rad}}{s}$$

using the Frobenius norm [Dem97]. Thus, the term can be neglected here, and consequently:

$$\dot{R}_b^{\hat{n}} \approx R_n^{\hat{n}} \cdot R_b^n \left[\vec{\omega}_{b/n}^{\,b}\times\right] \tag{3.8}$$

In order to incorporate physical acceleration estimates based on inertial sensors into the equations used for NED-navigation and the Dynamic model , a slight adjustment must be made to the general equations from [Wen11] which is explained in the following.

From [Wen11], (3.151), using the vector notation used therein, we receive:

$$\begin{aligned}\dot{\vec{\nu}}_{eb}^{\,n} &= R_b^n \vec{f}_{ib}^{\,b} - (2\vec{\omega}_{ie}^{\,n} + \vec{\omega}_{en}^{\,n}) \times \vec{\nu}_{eb}^{\,n} + \vec{g}^{\,n} \\ \Rightarrow \dot{\vec{\nu}}_{eb}^{\,n} &\approx R_b^n \vec{f}_{ib}^{\,b} + \vec{g}^{\,n} \quad \text{(for slow navigation)}\end{aligned} \tag{3.9}$$

when, again, omitting the Earth's rotation rate $\vec{\omega}_{ie}^{\,n}$ and the transport rate $\vec{\omega}_{en}^{\,n}$. This can always be done in slow navigation scenarios like AUV navigation with speed in order of $2m/s$.

In case the flat earth navigation assumption is justified we can transform the notation from [Wen11] into [Fos11] via:

$$\vec{\nu}_{eb}^{\,n} \hat{=} \dot{\vec{p}}_{b/n}^{\,n} = R_b^n \cdot \vec{\nu}_{b/n}^{\,b} \quad \text{(for flat earth navigation).}$$

Thus, equation 3.9 can be transformed to:

$$\dot{\vec{\nu}}_{eb}^{\,n} \hat{=} \ddot{\vec{p}}_{b/n}^{\,n} \overset{3.7}{=} R_b^n \cdot \left(w_{b/n}^b \times \vec{\nu}_{b/n}^{\,b} + \dot{\vec{\nu}}_{b/n}^{\,b}\right) \overset{3.9}{=} R_b^n \vec{f}_{ib}^{\,b} + \vec{g}^{\,n}$$

which straightforwardly yields the following:

**Lemma 3** (Central Acceleration Equation)**.** *In slow, flat-earth navigation for a given specific force $\vec{f}_{ib}^{\,b}$, turn rate $\vec{\omega}_{b/n}^{\,b}$ and gravity $\vec{g}^{\,n}$, the SNAME body-velocity ODE is governed by the equation:*

$$\dot{\vec{\nu}}_{b/n}^{b} = \vec{f}_{ib}^{\,b} - \vec{\omega}_{b/n}^{\,b} \times \vec{\nu}_{b/n}^{b} + R_n^b \cdot \vec{g}^{\,n} \tag{3.10}$$

**Corollary 4.** *In case the body velocity is measured within a different frame $b_2$ than the specific force from $b_1$, equation 3.10 can be expanded by additionally using equation 2.25:*

$$\dot{\vec{\nu}}_{b_1/n}^{b_1} = \vec{f}_{ib_1}^{\,b_1} - \vec{\omega}_{b_1/n}^{\,b_1} \times \left( R_{b_2}^{b_1} \cdot \vec{\nu}_{b_2/n}^{b_2} - \vec{\omega}_{b_1/n}^{\,b_1} \times \vec{p}_{b_2/b_1}^{\,b_1} \right) + R_n^{b_1} \cdot \vec{g}^{\,n} \tag{3.11}$$

### 3.3.1. Magnetic measurements

Measurements of earth magnetic field can be incorporated into the Strapdown navigation equations as a classic Kalman style observation.

**Definition 7** (Magnetic compass measurement model)**.** *For a local earth's magnetic field vector $\vec{h}^{\,n}$ in NED-frame, the magnetic measurement model is assumed as:*

$$\vec{h}^{\,b} = R_b^{n,T} \vec{h}^{\,n} + \vec{v}_m$$

*i.e., the global magnetic field coordinates are measured within local body frame with additive noise.*

Note that usually, an additional bias source can be assumed as well. This is however neglected here. In [MOS13] it is assumed, the magnetometer is calibrated and compensated for bias, scale factors and nonorthogonality of the input axis prior to the mission, referring to [AS02].

In [MOS13] the magnetometer's AWGN is modelled through a standard deviation of $60\mu$G given by the manufacturer's manual although in the experiments, a much bigger experimentally more suitable value of $60$mG is used there - a value enlarged by $3$ (!) orders of magnitude. In the AHRS-manual by XSens this is given as $200\mu$G, see section 4.3.1. This underlines the practical difficulties of dealing with magnetic measurements - especially in the presence of electromagnetic disturbances e.g. from the AUV's motor combined with hard- and soft-iron effects [Ozy12].

Assuming only a small orientation estimation error, the measurement equation can be linearized by generating the derivatives for using the first-order Taylor expansion in an EKF:

$$H_q^m := \frac{\partial \vec{h}^{\,b}}{\partial q_b^n} = \begin{pmatrix} H_{q_1}^m \cdot \vec{h}^{\,n}, & \ldots, & H_{q_4}^m \cdot \vec{h}^{\,n} \end{pmatrix},$$
$$H_{q_i}^m = \frac{\partial R_n^b}{\partial q_i}$$

the detailed expressions can be found in the appendix, section B.1.

A sample vector for coordinates $49^\circ N,\ 8^\circ E$, a location close to Karlsruhe, Germany, is given as

$$\begin{aligned}\vec{h}^n &= (20611.33, 690.19, 43710.01)^T\ nT \text{ (nano-Tesla)} \\ &= (0.2061133, 0.0069019, 0.4371001)^T\ G \text{ (Gauß)}\end{aligned}$$

This value was provided by the $wrldmagm$ function from the Matlab [MAT17] Aerospace Toolbox based on [CMA+15] for the reference day $28th$ of September $2015$ which was one of the test days evaluated in chapter 4.

For the sensor data acquired within the SMIS-project, several further adjustments and pre-filtering were necessary due to several systematic errors. Details can be found in section 4.2.1.

## 3.4. DVL Error Model

The Doppler Velocity Log (DVL) is one of the *key sensors* for accurate underwater navigation. Despite being a quite sophisticated device in practice, its operation principle is quite simple. Several acoustic pings having a typically narrow spectrum are sent simultaneously towards the ground at typical distances within $10m - 200m$, depending on device and operation mode. Some of the signal is reflected by the ground and received by the device. Due to the movement of the AUV relative to the ground, the received pings' frequencies will be shifted due to the Doppler effect. By combining the shifts of all pings, a velocity of the device relative the ground can be estimated. This is illustrated schematically in figure 3.2

Given several simplifying assumptions, if the local sound velocity $c$ is approximately constant and known and if for $c \gg v$ an AWGN is assumed on measured shifted incoming frequency $f_r$ then the resulting measured velocity $\hat{v}$ within the instrument frame is also AWGN. Only the linear transformation of instrument frame velocities to world frame velocities implicitly introduces non-linearities since the transformation matrix depends on the AUV's orientation estimates. Also, this requires the transmitted frequency $f_t$ to be realizable with sufficient accuracy. Expressed in equations:

$$\begin{aligned} f_r &= f_t \cdot \left(\frac{1+\frac{v}{c}}{1-\frac{v}{c}}\right) \\ f_d &= f_r - f_t \\ \Rightarrow f_d &\approx f_t \cdot \frac{2v}{c} + \epsilon_{f_r} \\ \Rightarrow \hat{v}_{beam} &= \frac{c}{2f_t} \cdot \left(\hat{f}_d + \epsilon_{f_r}\right) \end{aligned} \tag{3.12}$$

Where $\hat{v}_{beam}$ denotes the measured velocity parallel to one of typically four DVL beams. Combining these four measurements in [WYS99] to a vector $\vec{v}_{beam}(t)$ yields:

$$\vec{v}^{\,b}_{DVL/e} \approx \vec{v}^{\,b}_{DVL/n} = T \cdot \vec{v}_{beam}(t)$$

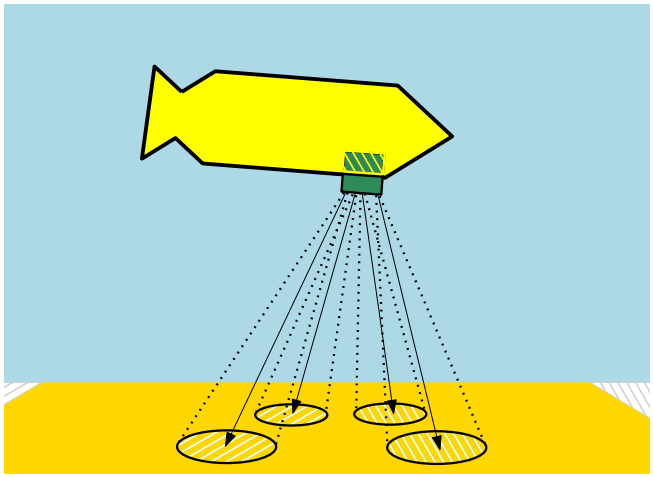

Figure 3.2.: Schematic Overview of DVL operation. Four acoustic pings are sent towards the ground along device-fixed angles (arrows) within the dotted cones representing the signals' spread. The signals reflections by the ground (ellipsoid areas) is partly received by the DVL device (dark green, attached to AUV) and processed for a velocity estimate.

Here, $T$ is a constant matrix. Exchanging $\{e\}$ with $\{n\}$ depends, again, on the justification of the flat-earth navigation assumption, see section 2.4.1. If the fourth row from $T$ in [WYS99] is ignored, then $T$ can be parameterized as follows:

$$T = \frac{1}{4} \begin{pmatrix} \cos(\Psi_1)\sin(\theta_1) & \cos(\Psi_2)\sin(\theta_2) & \cos(\Psi_3)\sin(\theta_3) & \cos(\Psi_4)\sin(\theta_4) \\ \sin(\Psi_1)\sin(\theta_1) & \sin(\Psi_2)\sin(\theta_2) & \sin(\Psi_3)\sin(\theta_3) & \sin(\Psi_4)\sin(\theta_4) \\ \cos(\theta_1) & \cos(\theta_2) & \cos(\theta_3) & \cos(\theta_4) \end{pmatrix}$$

In a typical $4$ beam configuration, where beams have a yaw offset of $90^\circ$ towards each other and a $30^\circ$ pitch angle with $0^\circ$ signifying a down-facing beam, this matrix simplifies to:

$$T \approx \begin{pmatrix} 0.0884 & -0.0884 & -0.0884 & 0.0884 \\ 0.0884 & 0.0884 & -0.0884 & -0.0884 \\ 0.2165 & 0.2165 & 0.2165 & 0.2165 \end{pmatrix}$$

For a more detailed modelling of the DVL measurements an additional bias component can be used for each beam, see [RPW06] for details.

Neglecting the bias, the error matrix in DVL coordinates assuming beam-wise indepen-

dent errors evaluates to

$$\begin{aligned} R_{DVL} &= \sigma_{DVL}^2 \cdot T \cdot T^T \\ &= \sigma_{DVL}^2 \cdot \begin{pmatrix} \frac{1}{32} & 0 & 0 \\ 0 & \frac{1}{32} & 0 \\ 0 & 0 & \frac{3}{16} \end{pmatrix}, \end{aligned}$$

meaning measurements in the instruments horizontal plane are less error prone than those in the vertical axis and the finale measurement model can be written as:

$$\tilde{\vec{v}}_{DVL/n}^{\,b} = \vec{v}_{DVL/n}^{\,b} + \mathcal{N}(0, R_{DVL}). \tag{3.13}$$

Here, $\sigma_{DVL}$ denotes the beam-wise standard deviation of the measured velocity. As of [WYS99], this value is typically below $1\%$ of speed travelled.

**Remark 4** (DVL Bias)**.** *It is suitable to treat the DVL measurements noise as a combination of white and coloured noise, i.e. to add a bias term into equation 3.13. The bias term itself can be treated as a Random Walk or first order Markov process, see section 3.2. For simplicity, this was neglected in this work but in commercial applications this is recommendable and more details can be found in [HH11] and again [RPW06].*

In case the DVL is not located sufficiently close to the BODY origin, i.e. it's relative position in BODY-coordinates $\vec{p}_{DVL/b}^{\,b}$ is not close to 0, where $\{DVL\}$ denotes the DVL-frame, then the coordinates can be transformed via:

$$\begin{aligned} &\vec{p}_{DVL/n}^{\,n} = R_b^n \cdot \vec{p}_{DVL/b}^{\,b} + \vec{p}_{b/n}^{\,n} \\ &\Rightarrow \dot{\vec{p}}_{b/n}^{\,n} = \dot{\vec{p}}_{DVL/n}^{\,n} - R_b^n \cdot \vec{w}_{b/n}^{\,b} \times \vec{p}_{DVL/b}^{\,b} \\ &\Rightarrow \vec{v}_{b/n}^{\,n} = R_b^n \cdot \left( \vec{v}_{DVL/n}^{\,b} - \vec{w}_{b/n}^{\,b} \times \vec{p}_{DVL/b}^{\,b} \right). \end{aligned} \tag{3.14}$$

In this case, the measurement model from equation 3.13 must be expanded to:

$$\vec{h}_{DVL}(\vec{v}_{b/n}^{\,b}) = \vec{v}_{DVL/n}^{\,b} + \mathcal{N}(0, R_{DVL}) \tag{3.15}$$

$$= \vec{v}_{b/n}^{\,b} + \vec{w}_{b/n}^{\,b} \times \vec{p}_{DVL/b}^{\,b} + \mathcal{N}(0, R_{DVL}) \tag{3.16}$$

$$\begin{aligned} &\approx \vec{v}_{b/n}^{\,b} + \left( \tilde{\vec{w}}_{b/n}^{\,b} - \vec{b}_\omega \right) \times \vec{p}_{DVL/b}^{\,b} \\ &\quad + \mathcal{N}(0, R_{DVL} + \left[ \vec{p}_{DVL/b}^{\,b} \times \right] \Sigma_{\vec{n}_\omega} \left[ \vec{p}_{DVL/b}^{\,b} \times \right]^T) \end{aligned} \tag{3.17}$$

where $\Sigma_{\vec{n}_\omega}$ denotes the turn-rate error covariance matrix from section 3.3.

Thus, an additional insecurity is introduced when the body origin is located far apart from the DVL origin. Practically speaking, you would like to have both IMU/AHRS and DVL close to each other. Preferably the AHRS should even be close to the AUV's center of gravity. That way the amount of additional local linear accelerations caused by rotation are kept small as well [Wen11].

If the DVL velocity is given in the NED frame, then the measurement equation can approximately be assumed as:

$$\vec{h}_{DVL}^{NED}(\vec{v}_{b/n}^{n}) = R_b^n \cdot \left(\vec{v}_{DVL/n}^{b} - \vec{w}_{b/n}^{b} \times \vec{p}_{DVL/b}^{b}\right) + \mathcal{N}\left(0, R_b^n \left(R_{DVL} + \left[\vec{p}_{DVL/b}^{b}\times\right] \Sigma_{\vec{n}_\omega} \left[\vec{p}_{DVL/b}^{b}\times\right]^T\right) R_n^b\right) \tag{3.18}$$

using the current estimate of the rotation matrix $R_b^n$. Physically, the DVL device is measuring velocities in its local DVL frame. However, the capabilities of commercial DVL devices can be good enough to provide NED-velocities from internal gyro and compass measurements. If data is given in such a format, it can be favourable to filter the velocity measurements in this transformed NED-format instead of re-transforming the measurements back to the local DVL frame prior to filtering. This is relevant in the evaluation, see section 4.2.1.

**Remark 5** (DVL operation limits)**.** *It must be noted that the model presented here is assuming that the DVL's is facing the ground rather directly. Angles within range of e.g.* $\pm 20^\circ$ *relative to the ground yield systematic bias in the data up to the point of making the measurements useless. This is mostly due to the fact that in these cases the ping response cannot be detected properly as the signal's echo on the ground may be deflected too drastically. Usually, the DVL manual will have warnings with respect to this scenario, giving recommendations for the range of pitch/roll angles.*

The effect of the previous remark can be seen in figure 3.3. This drift is especially prominent in the measured $z$-velocity.

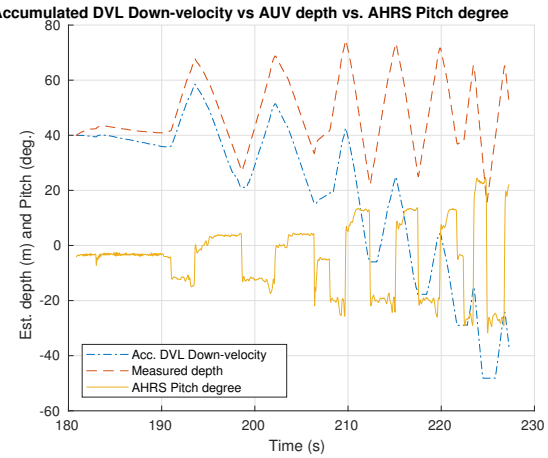


Figure 3.3.: Demonstration of bias/drift when accumulating the DVL Down-velocity. The velocity has been transformed to global coordinates by the DVL device itself. Drift appears stronger for bigger pitch angles. See section 4.2 for further description of the data set from $2015-09-28$.

In case the ground is sufficiently even, this error can be avoided by excluding measurements when the AUV's absolute Pitch or Roll is too big. Otherwise, in case the ground is very uneven and rugged, the faulty DVL data may cause severe problems. The ground's structural type may be known roughly prior to the mission (or online due to multi-beam and side-scan sonar images) and the DVL feed-back into the navigation scheme can be excluded. However, in these cases the drift error is growing drastically as the DVL is a key sensor in AUV navigation. Figure 3.3 also indicates why it may make sense in practice to include a bias term within the measurement in equation 3.13.

## 3.5. Pressure Sensor

By using pressure sensors the underwater navigation problem basically becomes a $2d$-problem regarding the AUV position since the global $z$-position can be measured via the pressure at a high sample rate without drift. Ideally, this will also require the AUV to keep track of the water column's CTD-profile to correctly compute depth from pressure [FM83].

However, the necessity for the AUV to keep track of a $3d$-orientation still remains.

In the following, it is assumed the water column profile is calibrated correctly, possibly online from CTD-measurements although this is not a simple task in practice. From [PSSL14] a standard deviation of $\sigma_{depth} = 0.1m$ is assumed using a simplified observation model in the form

$$h_{depth} = z^n + p^n_{d/b,z} + \mathcal{N}(0, \sigma^2_{depth}). \tag{3.19}$$

Here, $\vec{p}^{\,n}_{d/b} = R^n_b \cdot \vec{p}^{\,b}_{d/b}$ denotes the position of the depth-sensor in NED-frame and $\vec{p}^{\,b}_{d/b}$ its position in BODY-frame with $p^n_{d/b,z}$ representing its current $z$-component relative to the body origin, expressed in coordinates of $\{n\}$.

## 3.6. GPS error model

GPS measurements and their incorporation into the Strapdown algorithm can be treated quite extensively. See, [AC91] on existence of GPS solutions from Pseudo-Ranges and [ZQF16] in a loosely coupled GPS navigation scheme.
However, since it is only available on the surface, GPS does not receive detailed attention for AUV navigation, here. Instead, a simplified model is used, given by:

$$\vec{h}_{GPS} = \vec{p}^{\,n}_{b/n} + R^n_b \cdot \vec{p}^{\,b}_{GPS/b} + \mathcal{N}(0, \Sigma_{gp})$$

using a fixed positional covariance matrix $\Sigma_{gps}$ where $\vec{p}^{\,b}_{GPS/b}$ denotes the GPS antenna position in BODY-frame.
Both for simulation and evaluation, a diagonal covariance matrix $\Sigma_{GPS} = \sigma^2_{GPS} \cdot I_{3\times3}$ was used with $\sigma_{GPS} = 4m$, assuming a fairly stable and precise GPS reception.

See section 4.2.1 for a few more details on how several practical issues in dealing with GPS measurements were overcome to evaluate the real-world data.

## 3.7. USBL Error Model

In 2.4.2, the general functionality of a USBL device was explained, which allows measuring the relative direction and range of a vehicle towards another USBL device or beacon. Due to its outstanding importance in underwater navigation, the inevitable errors introduced by ranged USBL measurements is examined here in detail.

In the SMIS project, USBL modems model S2CR 7/17 by Evologics [evo] were used. Here, "S2C" stands for Sweep-Spread Carrier, a modulation technique based on linear chirps as explained in [KB02]. The technique was incorporated into Evologics' commercial USBL line. Accuracy and performance tests were performed by the manufacturer and reported in, e.g. [KKBY12].

In [NOW+15], results are reported for both communication and localization range in a deep-sea application ($5000$m). In that setting, decent communication was possible up to $8400$m slant range although the localization range was $1000$m below that. Still, this range is sufficient for many deep-sea applications with a support vessel or USV on the surface.

In [ONW+16], experiments were conducted and evaluated regarding the localization accuracy of an RV/USV localizing a USBL device fixed to the ground. Additionally, ray-tracing techniques were examined in order to improve the localization quality due to refraction effects from a non-constant sound velocity profile. The results of this paper influenced the choices made in the evaluation, chapter 4, in terms of horizontal distance of the USV to the AUV. Also, an additional simulation is dedicated to this publication in terms of estimating the achievable accuracy of an USV localizing an SBS, see section 4.6.

At first sight, the hydrophone setting can be compared to an inverse GPS measurement [ME06] using pseudoranges from each hydrophone (satellite) yielding a position estimate by numerical [Wen11] or analytical least-squares-solutions [AC91]. The absolute arrival times of acoustic signals could be evaluated for each hydrophone independently, yielding a least-squares remote position estimate. This is problematic in several ways as illustrated by the following example.

**Example 1** (Suboptimal pseudorange error model based on GPS)**.** *Given a typical remote position $\vec{r}$ to be tracked by a USBL device,*

$$\vec{r} = (500m, 300m, 200m)^T \quad \Rightarrow norm(\vec{r}) \approx 616.44m.$$

*The respective Jacobian $H(\vec{r})$ of the GPS-type pseudorange-measurement is given as:*

$$H(\vec{r}) = \frac{\partial \vec{h}_{Pseudorange}(\vec{r})}{\partial \vec{r}}$$

$$\Rightarrow V = (H^T H)^{-1} \overset{here}{\approx} 10^7 \cdot \begin{pmatrix} 0.9276 & -1.0213 & -0.7870 \\ -1.0213 & 2.0170 & -0.4722 \\ -0.7870 & -0.4722 & 2.6757 \end{pmatrix}$$

*where $V$ is the approximate (first-order) error propagation from pseudorange-measurements to position estimation. Note that due to the leading factor, in GPS-terms this yields a "horrible"* Dilution of Precision *[Lan99] of*

$$DOP(H, \vec{r}) = \sqrt{tr\left((H(\vec{r})^T H(\vec{r}))^{-1}\right)} \approx 7.5 \cdot 10^3$$

*Meaning even small measurement errors have a huge effect on the position estimate. A DOP value of $1$ would be considered an excellent value in GPS terms and a value above $20$ would be considered a poor value.*

*The values given in this example are based on a typical hydrophone set-up, namely the default set-up in this chapter as given by equation 3.21.*

*However, if the hydrophones were hypothetically placed much further apart, e.g. by a factor of $500$ which for equation 3.21 would yield a hydrophone array spaced $85m$ ($62m$) apart horizontally (vertically) reflecting an SBL (almost LBL) scenario, the DOP would would get much better for the same $\vec{r}$:*

$$DOP(H_{500}, \vec{r}) \approx 14.83$$

Example 1 denotes the numerical issues implied in USBL position estimates and although a pseudorange model could be used for theoretical considerations assuming an appropriately small measurement noise, it does not directly reflect the *true* physical capabilities of USBL localization. Physically, measuring absolute time-of-arrival for each hydrophone is problematic. Another disadvantage of this approach is highlighted in section 3.7.4. Instead, the time-*difference*-of-arrival can be used both for modelling and in practice.

For measuring the time-difference of an arriving signal at two hydrophones, the signal correlation can be used given in the classic work of [KC76] which is explained in the following section.

### 3.7.1. Time Difference of Arrival (TDOA)

Using the signal's correlation, the time difference of arrival (TDOA) can usually be estimated more accurately than measuring the absolute time of arrival [KC76], at least when several assumptions are justified for spatially close-by receivers (hydrophones). See [YHKA96, GTG+05] for surveys and applications in radio signals.
When a signal is sent from source $A$ to receivers $B_1$ and $B_2$ being spatially close to each other, the received noisy signals $x_1$, $x_2$ from original signal $s_1$ will have been affected by similar channel distortions. The received singals' cross correlation can then be evaluated easily and its maximum will yield a TDOA estimate. [KC76] provides a Cramér-Rao type bound on the minimum variance $\sigma_D^2$ of the measurable TDOA from correlation. This is used here, to estimate the order of magnitude which can be expected as a TDOA for the USBL setting.
[KB02] describes the basic technique which is used for achieving parallel localization and communication in Evologics USBL modems. In that work, linear frequency sweeps are employed using a frequency gradient of $\frac{4kHz}{ms}$, i.e. the frequency change over time

for a single chirp. Applying the same constant to a $7 - 17kHz$ signal yields a single sweep time of $T = 2.5ms$. The PSD of such a linear sweep signal is illustrated in figure 3.4. Following the notation from [KC76] with (cross) power spectrum $G_{x_1,x_2}$, it can

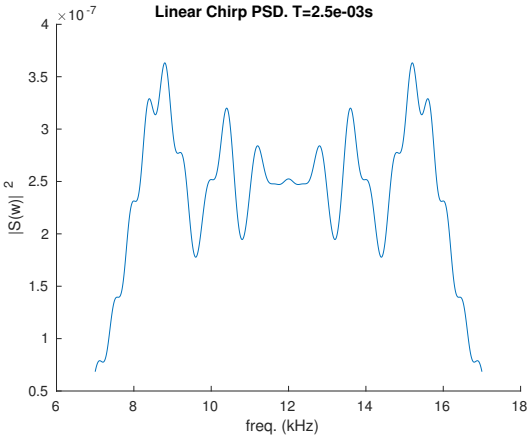


Figure 3.4.: Typical (unscaled) power spectral density of a linear chirp, given by a truncated Fourier Transform $S(\omega)$. The frequency band is limited by 7-17kHz, as is used in the S2C 7/17 USBL modems.

be assumed that:

$$\begin{aligned}\|\gamma_{12}\|^2 &:= \frac{\|G_{x_1,x_2}\|^2}{\|G_{x_1,x_1}\|\|G_{x_2,x_2}\|} \approx \frac{\|G_{s_1,s_1}\|^2}{\|G_{s_1,s_1} + G_{n_1,n_1}\|^2} \\ &\overset{\text{SNR}=10dB}{\approx} \frac{\|G_{s_1,s_1}\|^2}{\|G_{s_1,s_1} + \frac{1}{10}G_{s_1,s_1}\|^2} \approx \frac{1}{1.21}, \text{ for } 7kHz \leq f \leq 17kHz\end{aligned}$$

for a nominal signal-to-noise-ratio (SNR) of $10dB$ given in the USBL's manual, see table 2.2. For the latter simplification, the chirp spectrum is implicitly approximated by a rectangle. With formula (A4) from [KC76], it follows:

$$\begin{aligned}\sigma_D^2 &= \left[2T \int_0^\infty (2 \cdot \pi \cdot f)^2 \frac{\|\gamma_{12}\|^2}{1 - \|\gamma_{12}\|^2} df\right]^{-1} \\ &\approx \left[2T \int_{7 \cdot 10^3}^{17 \cdot 10^3} (2 \cdot \pi \cdot f)^2 \frac{1}{0.21} df\right]^{-1} \approx \left[2T \cdot 2.86 \cdot 10^{14} s^{-3}\right]^{-1} \\ &\Rightarrow \sigma_D \approx 8.37 \cdot 10^{-7} s \text{ for } T = 2.5ms\end{aligned}$$

This assumes, only one chirp of length $2.5ms$ is used for correlation. Since the acoustic signal transmitted by the sender consists of an audible series of chirps, e.g. 100 of them, $T$ can also be considered to have length $250ms$ [1]. In that case $\sigma_D \approx 8.37 \cdot 10^{-8} s$.

[1] For the S2C 7/17 USBL modems used in this work, $250ms$ is a reasonable estimate of a single message duration.

As mentioned before, this is just a rough theoretical estimate for the range of its lower bound. Due to many assumptions made, the true respective boundary for estimating the time difference from correlation should in general be greater than $\sigma_D \approx 8.37 \cdot 10^{-8} s$.

Moreover, a comparison of TDOA and different schemes for USBL localization using a self-made USBL system can be found in [MOS10].

### 3.7.2. Hydrophone Array Layout

As will be shown in the following sections, a USBL device requires at least $4$ non-planar hydrophones to estimate the direction of an incoming signal from TDOA measurements. For robustness, $5$ hydrophones are often used. More would be acceptable as well. However, that leads to spatial problems due to the typical modem's size in the order of $\sim (20cm)^3$ combined with the hydrophone having a typical diameter of $2 - 3cm$, as explained in section 2.4.2. On the one hand, the modem head runs out of room if too many hydrophones are used. On the other hand, hydrophones should be placed as far away as possible from each other. Thus, the additional information received from using more hydrophones could become practically useless since this would require some or all hydrophones to be placed too close to each other.
The hydrophone array layout considered primarily in this work, is presented in the following.

**Definition 8** (Hydrophone Array Matrix)**.** *Given a fixed USBL device's Cartesian reference frame $\{b\}_{USBL}$ which is usually located in the transducer head, e.g. within a hydrophone, the hydrophones' positions in that frame are denoted by $\vec{p}_{hydro,i}$ for $i = 1, \ldots, n_{hydro}$ and they are subsumed in the Hydrophone Array Matrix $P_{hydro}$:*

$$P_{hydro} = \begin{pmatrix} p^{(x)}_{hydro,1} & p^{(y)}_{hydro,1} & p^{(z)}_{hydro,1} \\ p^{(x)}_{hydro,2} & p^{(y)}_{hydro,2} & p^{(z)}_{hydro,2} \\ \cdots & & \\ p^{(x)}_{hydro,n_{hydro}} & p^{(y)}_{hydro,n_{hydro}} & p^{(z)}_{hydro,n_{hydro}} \end{pmatrix} \tag{3.20}$$

For the hydrophone set-up considered primarily in this work, the following matrix is used.

$$P_{\text{hydro}} = \begin{pmatrix} -d_x & 0 & -\frac{1}{5} d_z \\ 0 & d_y & -\frac{1}{5} d_z \\ 0 & -d_y & -\frac{1}{5} d_z \\ d_x & 0 & -\frac{1}{5} d_z \\ 0 & 0 & \frac{4}{5} d_z \end{pmatrix} \tag{3.21}$$

$$\text{where } d_x = d_y = 0.06m,\ d_z = 0.06m$$

The set-up of hydrophones based on equation 3.21 is sketched in figure 3.5. In principle, this hydrophone layout structure is identical to the pyramid structure described in [Ark12].

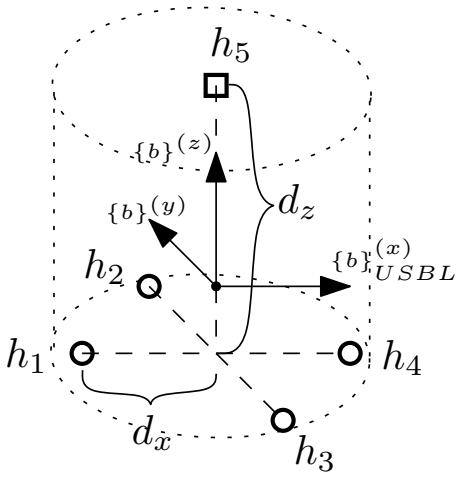


Figure 3.5.: Hydrophone Layout for the modem used as defined by equation 3.21 with origin frame $\{b\}_{\text{USBL}}$. Circles denote purely receiving hydrophones ($h_1, \ldots, h_4$) and the square denotes the transceiver hydrophone $h_5$. The dotted cylinder signifies the USBL modem's head. (Distances not to scale).

**Remark 6** (Hydrophone distances)**.** *The true hydrophone positions of the USBL modems used (S2C 7/17 USBL by Evologics) are not provided within the manufacturer's manual. Instead, they were firstly estimated from additional maintenance knowledge as well as the dimensions of the USBL head. Moreover, data-based tests were performed by Sergej Neumann in order to determine the positions with more accuracy. The details of this can be found in his thesis, see section 2.8.1.*

In order to use the TDOA measurements for direction estimates, further simplifications can be made. The most important one, the planar wave assumption, is presented in the following section.

### 3.7.3. Planar Wave approximation

The error model derived here follows the design and evaluation of an experimental USBL device described in [RMB+16] which is based on [AS87] and [YHKA96].

For a simplified, linearized estimation of the target position, the arriving acoustic wave is approximated by a planar wave (PW) as is often done in many similar applications for, e.g., electromagnetic wave propagation [Jac07]. This means, the acoustic wave caused by a small spherical source propagating through water is locally approximated by a plane instead of a spherical surface. This assumption is justified for "sufficiently" big sphere's radii. Figure 3.6 illustrates the concept of PW in case of USBL devices. The time it takes for the signal to propagate from one hydrophone located at $\vec{p}^{\,b}_{\text{hydro},i}$ to another one located at $\vec{p}^{\,b}_{\text{hydro},j}$ can then be assumed to be equal to the difference vector between the

hydrophones projected on the unknown target direction vector $\vec{d}^{\,b}$ - divided by signal propagation speed. Mathematically, this projection is achieved by the scalar product of both vectors, see equation 3.22.

PW approximation for USBL is quite common as it also used in [MOS13] and [RMB+16]. It will be shown in section 3.7.5, that the pure error resulting from the PW assumption is negligible for both theoretical and practical considerations. This is due to the hydrophones being spatially very close to each other making a big sphere's surface locally almost indistinguishable from a plane for radii greater than $10m$.

**Definition 9** (USBL Planar Wave (PW) equations)**.** *Following the notation in figure 3.6 where $\vec{r}^{\,b}$ denotes the target position and $\vec{d}^{\,b}$ the target's unit direction vector both in $\{b\}_{USBL}$, the PW equations for the TDOA measurements $\delta_{i,j}$ are given as:*

$$\delta_{i,j} = t_i - t_j = -\frac{1}{c}\vec{d}^{\,b,T} \cdot \left(\vec{p}^{\,b}_{hydro,i} - \vec{p}^{\,b}_{hydro,j}\right) \quad \text{for } i,j \in \{1,\dots,n_{hydro}\},\ i<j \tag{3.22}$$

*for hypothetical (i.e. not directly measured) absolute arrival times $t_i$ at hydrophone $i$ and local sound velocity $c$. Moreover, the $\delta_{i,j}$ are collected in a vector $\Delta_t$ as:*

$$\Delta_t = \left(\delta_{1,2}, \delta_{1,3}, \dots, \delta_{(n_{hydro}-1),n_{hydro}}\right)^T$$

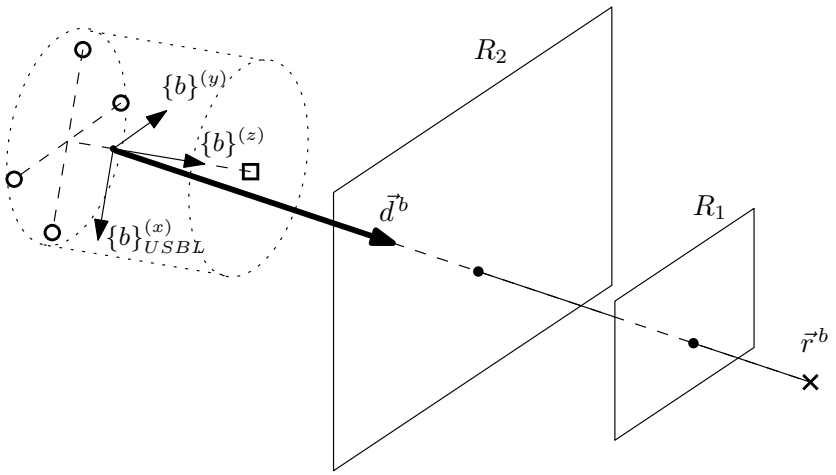


Figure 3.6.: Illustration of basic PW equations (from definition 9) as it is explained in [RMB+16]. Sound propagation is approximated as planes illustrated by rectangles $R_1, R_2$ along $\vec{d}^{\,b}$.

Further, let the *Difference Matrix* $D$ of dimension $5$ (for $5$ hydrophones) be defined as:

$$D = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix} \tag{3.23}$$

which can be easily generalized accordingly if alternate numbers of hydrophones are considered. This matrix $D$ represents the case where all possible hydrophone combinations $(h_i, h_j)$ for $i < j$ are used for estimating the target's position, i.e. all $\delta_{i,j}$ from equation 3.22 are measured and processed. Otherwise, if only a subset of the full hydrophone combinations are used, $D$ can be adapted easily by removing respective rows. Using $D$, the PW Approximation scheme can be denoted in short:

**Definition 10** (USBL Planar Wave Approximation Scheme)**.** *Using the previous definitions and equations, the Planar Wave Approximation scheme for relative USBL positioning can be subsumed as:*

$$S := D \cdot P_{hydro}, \tag{3.24}$$

$$S \cdot \vec{d} = -c \cdot \Delta_t \tag{3.25}$$

$$\Rightarrow \vec{d} = -c \cdot S^{+} \cdot \Delta_t. \tag{3.26}$$

*Here, $S^{+}$ denotes the (Moore-Penrose)* pseudoinverse *which is unique for any $S$. Usually it is computed as*

$$S^{+} = (S^T \cdot S)^{-1} \cdot S^T, \tag{3.27}$$

*provided the inverse $(S^T \cdot S)^{-1}$ exists which is true if $S$ has full rank $3$. This scheme yields a direction estimate $\tilde{\vec{d}}$ for a TDOA measurement $\tilde{\Delta}_t$.*

Note that if if $S$ does not have full rank, the underlying problem is ill-posed. This can only happen if the hydrophone set-up is singular, e.g. when all hydrophones are located in the same plane[2]. Without further notice this case is implicitly excluded here and $S$ is always assumed to have full rank.

Due to the PW approximation, there is a small systematic error w.r.t. the direction estimate $\vec{d}$ in equation 3.26. As can bee seen in section 3.7.5, this error is negligible. As such, for hypothetical perfect TDOA measurement $\Delta_t$, it holds with very high accuracy that:

$$\vec{d} = -c \cdot S^{+} \cdot \Delta_t \approx \vec{r}\frac{1}{\|\vec{r}\|} \quad \text{(Perfect measurement)} \tag{3.28}$$

[2]This would contradict basic physical intuition and is as such not of interest in this work.

Thus, if the range $r$ towards the target is measured by a round-trip time, the relative target's position can be estimated by:

$$\hat{\vec{r}} = \frac{1}{\|\tilde{\vec{d}}\|} \cdot \tilde{\vec{d}} \cdot \tilde{r} \tag{3.29}$$

where it is implicitly assumed that the resulting $\tilde{\vec{d}}$ from equation 3.26 does not have unit length for imperfect measurements, i.e. noisy TDOA. This yields the following corollary.

**Corollary 5** (USBL Direction Measurement Model)**.** *Assuming TDOA measurements with AWGN, i.e.*

$$\tilde{\Delta}_t = \Delta_t + \mathcal{N}(0, \Sigma_{tdoa}), \tag{3.30}$$

*then the USBL direction measurement can be modelled by the PW approximation as*

$$\vec{h}_d(\vec{r}) := \frac{\vec{r}}{\|\vec{r}\|} + \mathcal{N}(0, S^+ \left(c^2 \Sigma_{tdoa}\right) S^{+,T}) \tag{3.31}$$

*Usually, the TDOA is further simplified as i.i.d. TDOA with standard deviation constant $\sigma_{tdoa}$, i.e.*

$$\begin{aligned} \Sigma_{tdoa} &= \sigma_{tdoa}^2 \cdot I \\ \Rightarrow \vec{h}_d(\vec{r}) &= \frac{\vec{r}}{\|\vec{r}\|} + \mathcal{N}(0, c^2 \sigma_{tdoa}^2 (S^T S)^{-1}) \end{aligned}$$

*In this formulation, $\vec{r}$ denotes the target's position within the local USBL frame. Expressed with NED coordinates, this resolves to:*

$$\vec{r} = R_n^b \cdot \left(\vec{p}_{remote}^{\,n} - \vec{p}_{b/n}^{\,n}\right) - \vec{p}_{USBL/b}^{\,b} \tag{3.32}$$

*where $\vec{p}_{USBL/b}^{\,b}$ denotes the USBL "antenna" position in Body frame relative to the body origin.*

This corollary holds due to the linear processing of $\tilde{\Delta}_t$ in the PW approximation and applying covariance-rule $cov(A \cdot X, B \cdot Y) = A \cdot cov(X, Y) \cdot B^T$. It means that a ten-dimensional random Gaussian vector arising from the TDOA measurements is projected to a three-dimensional random Gaussian vector.

Note that equation 3.32 implies that the remote position of the target also adds another source of insecurity. If not mentioned otherwise, the target's position is assumed to be known with sufficient accuracy and only the vehicle's position $\vec{p}_{b/n}^{\,n}$ is of interest. This assumption is justified if the vehicle is communicating with the RV, the USV (position known from GPS) or the SBS (position known from calibration) and the additional insecurity can then be neglected. Otherwise this provides an additional AWGN term if the true $\|\vec{r}\|$ is ideally considered to be known. Moreover, it is even acceptable to intentionally normalize the direction measurement to 1 and then pretend the vector would still arise from the original source with AWGN. This will be briefly illustrated in section 4.3.3.

Complementary to the direction measurement from corollary 5 a range measurement model is derived based on [PSSL14]. The results from [FKJL11] report an acoustic slant range measurements to be modelled with AWGN having standard deviation $\sigma_r = 3m$ for one-way travel time (OWTT) measurements and $\sigma_r = 7m$ for round-trip range (RTR) nearly independent of the true range. In default USBL applications, only the RTR is important. Thus, it is used to derive a slant range measurement model.

**Corollary 6** (USBL Range Measurement Model)**.** *For a USBL device with $n_{hydro}$ hydrophones, the range-independent measurement error coefficient of*

$$\sigma_r = \frac{7}{\sqrt{n_{hydro}}} m \tag{3.33}$$

*yields a USBL range measurement AWGN-model of:*

$$\vec{h}_r(\vec{r}) = \|\vec{r}\| + \mathcal{N}(0, \sigma_r^2). \tag{3.34}$$

*Most notably, this error is treated as being independent of the direction measurement model from corollary 5.*

Note that this is a simplification omitting the actual distance between hydrophones which would yield a small additional term in the range measurement. Although in general, assuming the range error to be independent of the direction error should not be true for a real-world application, it is still a reasonable assumption as long as lower bounds in the sense of Cramér-Rao are of interest. In [LJ03] it is argued that decoupled filters (here: for range and direction) may outperform fully coupled "optimal" filters in practice. The $\sqrt{n_{\text{hydro}}}$ from equation 3.33 stems from a further simplification where each hydrophone measures the RTR independently from the others yielding a full slant range estimate through taking the sample mean of from hydrophone measurement which is, again, in the spirit of Cramér-Rao.

This error model only holds for the static USBL case. In case the participants are moving, more care is required. Usually, this leads to employing delayed state Kalman filters. The concept is described in [LARP98] which is needed when the measurement function $h_k$ is not only dependent on the current state $x_k$ but also of previous states $x_{k-1}, x_{k-2}, \ldots$. This can occur when considering either measurements requiring relevant processing time or, in this case, measurements with relevant latency or integration time. An early work addressing this issue can be found in [MGJ01] which uses a technique comparable to delayed state Kalman filter approaches. [Sta10] presents a delayed-state sigma point Kalman filter (UKF-type) for AUVs using USBL. A more recent approach to this problem is found in [XWD+16]. In practice, this is issue only important for large range (e.g. $> 1000m$) USBL measurements from AUVs. For theoretical PCRB considerations, this problem can be neglected or at least heavily simplified. As such, this problem is not considered in detail here - neither for practical nor for theoretical considerations, since the field data involving moving AUVs presented in chapter 4 contains only small-range measurements up to $500m$.

### 3.7.4. USBL Measurement Model Characteristics

**Remark 7** (Typical error)**.** *For a typical error considered here, the following error values were chosen:*

$$\sigma_{tdoa} = \frac{0.006}{c} s = 4 \cdot 10^{-6} s,$$
$$\sigma_r = \frac{7}{\sqrt{5}} m.$$

*This is smaller than the $\frac{0.01}{c}s$ chosen in [RMB+16]) and bigger than the theoretical one derived in section 3.7.1. For these parameters, the direction vector covariance matrix, corollary 5, for the full-TDOA processing using the hydrophone setting from equation 3.21 evaluates as:*

$$c^2\sigma_{tdoa}^2 (S^T S)^{-1} = c^2 \cdot \sigma_{tdoa}^2 \begin{pmatrix} \frac{250}{9} & 0 & 0 \\ 0 & \frac{250}{9} & 0 \\ 0 & 0 & \frac{625}{9} \end{pmatrix}$$
$$\approx 10^{-2} \cdot \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.25 \end{pmatrix}, \text{ for } \sigma_{toda} = 4 \cdot 10^{-6}$$

*Note that this specific diagonal shape is due to the full TDOA vector usage and the symmetric shape of the hydrophone array. Also note, the error is marginally larger in USBL's $z$-direction which is due to hydrophone $h_5$ being the only one not within the the main hydrophone plane of $h_1, \ldots, h_4$.*

*If the resulting error cloud around the true direction vector is transformed into a cone starting in the USBL's origin, this results in an apex angle of roughly $2.4^\circ$ to $7.6^\circ$ ($1\sigma$ to $3\sigma$ environment) along the device's $z$-axis.*

**Remark 8** (Compared error when using absolute time measurements)**.** *If the same PW scheme was used assuming the* absolute *signals' arrival times at the hydrophones could be measured modelled by AWGN* independently *for each hydrophone, this would result in the following error propagation matrix:*

$$\Sigma_{\text{t-abs}} = c^2 \cdot \sigma_t^2 \begin{pmatrix} \frac{1250}{9} & 0 & 0 \\ 0 & \frac{1250}{9} & 0 \\ 0 & 0 & \frac{3125}{9} \end{pmatrix}$$

*where $\sigma_t$ denotes the time measurements' standard deviation. Assuming this to have the same magnitude as $\sigma_{toda}$, i.e. $\sigma_t = 4 \cdot 10^{-6}$, yields*

$$\Sigma_{\text{t-abs}} = 10^{-2} \cdot \begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1.25 \end{pmatrix}.$$

*This means, the error propagation matrix would be $5$ the size the one from remark 7.*

*However, this does not even take into account that measuring the absolute signals' arrival times can be physically much more difficult than measuring the TDOA rendering this approach practically inferior.*

Note that the scheme from remark 8 still yields the same Gaussian error ellipsoid characteristics. It will yield the same numerical values as in remark 7 if smaller $\sigma_t^2$ are assumed. Thus, for theoretical considerations the schemes can be considered equivalent if appropriate constants are used. In [MOS13], the approach regarding 8 is chosen.

According to corollary 5, the full measurement error using both direction and slant range can be simulated for a target position $\vec{r}$ to be tracked by:

1. Sample a regular Gaussian Point Cloud $\mathcal{N}(\vec{r}, \|\vec{r}\|^2 \cdot S^+ \left(c^2 \Sigma_{tdoa}\right) S^{+,T})$,
2. Normalize and scale each resulting point by simulated range measurement $\tilde{r} = \|\vec{r}\| + \mathcal{N}(0, \sigma_r^2)$.

This results in typical error shapes in Cartesian space. For wide ranges, the shape resembles a "pancake" perpendicular to the range-vector which, according to [LJ03] is typical of phased array radars [DF83]. This is illustrated by simulations shown in figure 3.7 and 3.8. In mathematical terms, this pancake is an intersection of an ellipsoid (almost a sphere) with the crust of a much larger sphere intersecting the ellipsoid's center.

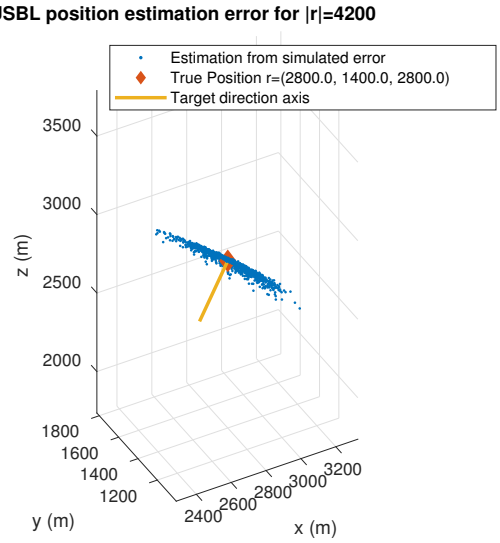


Figure 3.7.: Typical *pancake* error shape perpendicular to target's direction for wide range (4200m) USBL localization. USBL origin is same as coordinate system's origin $(0, 0, 0)$. Parameters: $\sigma_{tdoa} = 4 \cdot 10^{-6} s$, $\sigma_r = 7/\sqrt{5} m$ using $n_p = 1000$ points.

For closer ranges up to 100m depending on $\sigma_r$, the resulting error shape resembles a cigar along the range vector which is typical of continuous-wave radars [LJ03] where

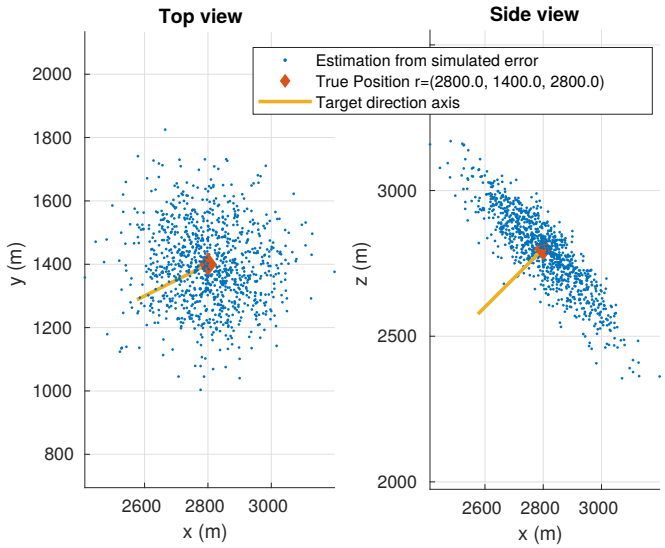


Figure 3.8.: Typical pancake error shape for wide range (4200m) USBL localization. Same data as in figure 3.7 from different views.

angle measurement is more accurate than range. A simulation of this effect for USBL is shown in figure 3.9.

**Remark 9** (Availability of direction and range measurements)**.** *Note that physically, direction estimates can be generated whenever the USBL device receives a valid signal from a target. As such, a one way message or signal is enough to generate a direction estimate. In contrast, the slant range estimate in a standard USBL device is only available from a round-trip signal (i.e. two-way message; with acknowledgement). Thus, to compute this estimate a full call and response cycle is needed causing a bottle-neck in the localization.*

*Otherwise a rather sophisticated synchronization scheme would be required to estimate slant ranges from one-way signals.*

### 3.7.5. Model Limitations

Care has to be taken in how the PW approximation is implemented. If only pairs of hydrophones are considered *separately* for angle estimation, the estimated elevation angle $\hat{\theta}$ is vulnerable to deviations in local sound velocity, depending on the true elevation $\theta$. That means, if a local sound velocity of $c_0$ is assumed although the true local sound velocity is $c_1$, the resulting estimated elevation angle $\hat{\theta}$ can strongly deviate from the true

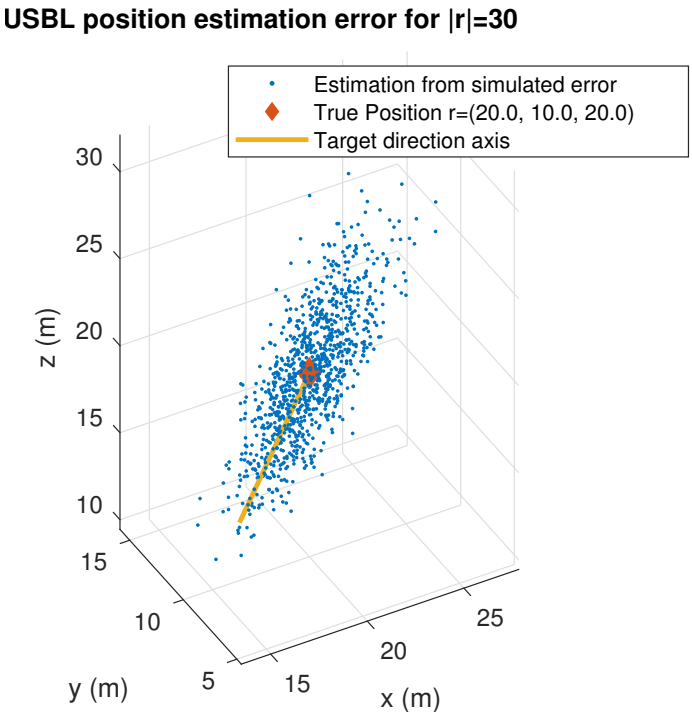


Figure 3.9.: Typical *Cigar* error shape perpendicular to target's direction for close range (30m) USBL localization. Parameters: $\sigma_{tdoa} = 4 \cdot 10^{-6} s$, $\sigma_r = 7/\sqrt{5} m$ using $n_p = 1000$ points.

value $\theta$. Figure 3.10 illustrates this error.

$$\hat{\theta}_{\text{sep}} = \frac{\pi}{2} - \arcsin\left(\frac{c_0}{c_1} \cdot \sin(\frac{\pi}{2} - \theta)\right) \tag{3.35}$$

Note that this formulation implicitly avoids the ambiguity arising included in the $2d$ planar-wave approximation depicted in figure 3.10 where the true direction of the received wave may also be mirrored along the $h_1$-$h_2$-axis marked $b$ for baseline.

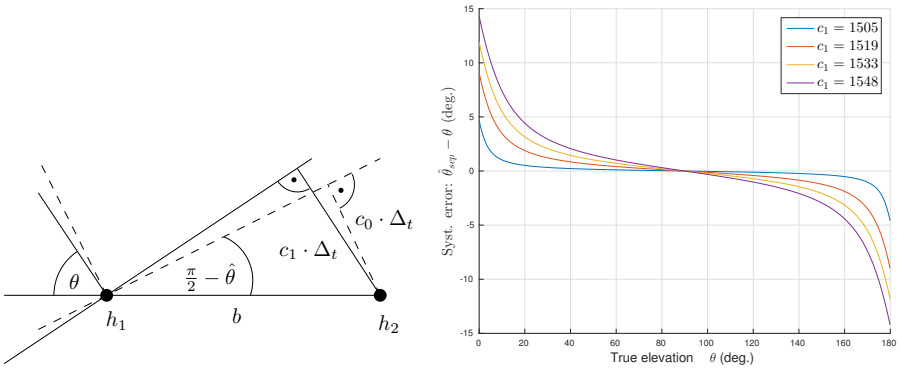


Figure 3.10.: Vulnerability to local sound velocity if only pairs of hydrophones $(h_1, h_2)$ are considered for a relative elevation angle estimation. Left image illustrates the elevation angle distortion, right image illustrates the resulting estimation error based on equation 3.35 if $c_0$ is wrongly assumed to be at $1500\,m/s$.

Using triplets of hydrophones with 2 respective TDOA can also yield a PW direction estimate. This requires to account for the ambiguities arising from a co-planar set-up of hydrophones. That means, any estimated direction from hydrophone triplet could arise from the same direction vector mirrored along the hydrophones' plane. The standard formulation using the matrix $S$ from the above section can not be used in this case.
The ambiguities can be cancelled if the single triplets estimates are combined to a total position estimate requiring a somewhat more tedious implementation as the standard PW scheme. In an additional theoretical investigation, 6 out of 10 possible triplets were used to generate direction estimates for targets of the type $\vec{r} = (\frac{5000}{tan(\theta)}, 0, 5000)^T$ as they occurred in [ONW+16]. The separate estimates were generated from a least-squares formulation which was evaluated within a close range of the true direction vector (up to $15^\circ$ apart from the true direction) using the true TDOA from $c_1 = 1548\frac{m}{s}$ with a falsely assumed velocity of $c_0 = 1500\frac{m}{s}$ within the estimation formula. For ease of implementation, the least-squares solution was generated from a brute-force sampling around the true direction $\vec{d}$ although this problem seems well suited for a standard Gauss-Newton Non-Linear Least-Squares procedure [Bjö96].
The results are shown in figure 3.11. The resulting direction estimate was generated by taking the mean of the separate triplet estimates. The overall angle error in figure 3.11

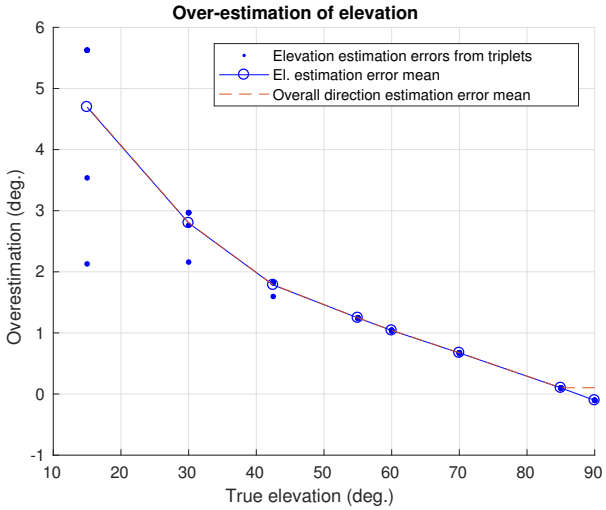


Figure 3.11.: For $\vec{r} = (\frac{5000}{\tan(\theta)}, 0, 5000)^T$ the hydrophone triplet scheme yields systematic error when assuming a local sound velocity of $c_0 = 1500\frac{m}{s}$ for a true local sound velocity of $c_1 = 1548\frac{m}{s}$. As can be seen, most of the angle error is affecting the resulting elevation estimate rather then azimuth estimate.

is computed as in equation 3.37. As it can be seen, this triplet scheme together with the falsely assumed local sound velocity may qualitatively - but not quantitatively - explain the systematic elevation over-estimation observed in [ONW+16].

The USBL's hydrophones are stored in a special oil instead of being exposed to seawater. Nevertheless, the oil's characteristic sound velocity may be subject to a similar change by pressure [MTB85] as it is the case for pure water [BSS+99] which, again, needs to be addressed in the implementation. The exact pressure dependencies go beyond the scope of this work and are thus not followed in more detail. The error arising from a flawed implementation can also be neglected in practice if the local sound velocity is measured or estimated from CTD data and fed to the USBL modem if significant changes occur (e.g. update $c_0$ within the modem if $|c_0 - c_1| \geq 3\frac{m}{s}$ while diving).

This error only occurs, if hydrophone pairs are considered *separately* for angle estimation (e.g. in pairs or triplets). However, *local sound velocity is not important for the direction estimate if all hydrophones are considered simultaneously*. That is due to the fact that the direction can be recovered alternatively to corollary 5 as:

$$\vec{d} = -c_0 \cdot S^+ \cdot (\frac{\Delta_\rho}{c_1}) \tag{3.36}$$

where $\Delta_\rho = \Delta_t \cdot c_1$ denotes the true RDOA

meaning the estimated incoming direction is just scaled by a constant factor $\frac{c_0}{c_1}$ leaving the direction estimate itself unchanged.

There is a small systematic distortion of the estimated direction vector $\vec{d}$ for the given hydrophone set-up. This error is due to the planar wave assumption yielding an accuracy loss depending on range. This error is measured by the angle-offset of the estimated direction vs. the true target's direction. This error is denoted as (systematic) *direction distortion* and quantified by the respective angle-offset. Given a perfect theoretical TDOA or RDOA measurement as denoted by $\Delta_t$ in equation 3.36, the resulting direction distortion $\alpha_{dist}$ can be examined as

$$\alpha_{dist} = \arccos\left(\frac{\vec{d}\cdot\vec{r}}{\|\vec{d}\|\|\vec{r}\|}\right) \tag{3.37}$$

This results in a systematic overall position distortion $r_{dist}$

$$r_{dist} = \alpha_{dist}\cdot\|\vec{r}\| \tag{3.38}$$

A small survey has shown that for the given hydrophone set-up, this systematic distortion is almost independent of azimuth and depends on the elevation angle $\theta$ of the incoming signal. Figure 3.12 shows that this systematic error is very small. Only for a

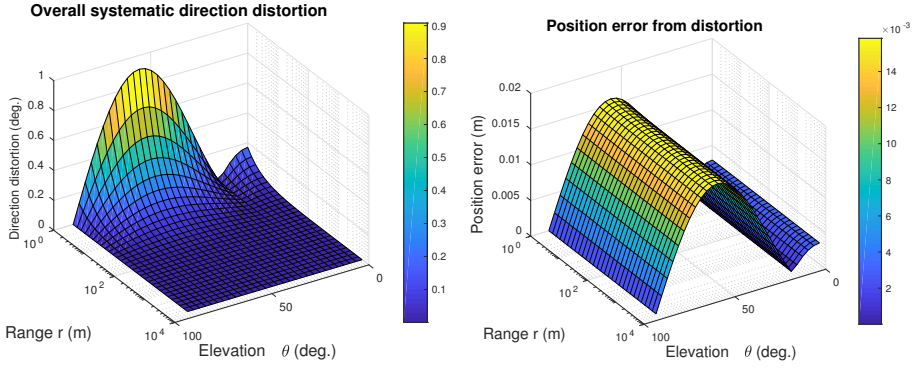


Figure 3.12.: Systematic PW angle distortion from equation 3.37 and the resulting positional error from equation 3.38. Note that the range is in logarithmic scale. Plot for fixed azimuth angle of $45°$ where systematic distortion has a maximum w.r.t azimuth.

hypothetically small close range of $1m$ towards the target $\alpha_{dist}$ gets as large as $\approx 1°$ and shrinks to $\approx 0.09°$ for $\|\vec{r}\| = 10$ getting even smaller as $\|\vec{r}\|$ increases. Further, estimating the resulting absolute position error in depending on $\|\vec{r}\|$ by equation 3.38 reveals that this error is mostly independent of range and remains smaller than $0.018m$ for all $\|\vec{r}\|$.
An accuracy of this magnitude is not to be expected in any ranged USBL application. *Thus, this systematic error is negligible* for the given hydrophone setup.

See the appendix, section B.2, for more details on the derivation for this systematic error.

### 3.7.6. Model Extensions

In [MOS13], an additional cross covariance between slant range and TDOA measurements is used. This is a justified assumption since the signal reception quality will influence both slant range and TDOA measurements. These cross covariances in [MOS13] only depend on $\sigma_{tdoa}$. Using conditional covariance, corollary 1, it can be shown easily that given the (combined) slant range measurement the TDOA covariance matrix still resolves to the typical shape from remark 8 for the given hydrophone layout. On the other hand, since $\frac{\sigma_r}{c} \gg \sigma_{tdoa}$, the TDOA measurements' influence on the slant range estimates is negligible. As such, this cross covariance is ignored in this work.

[Wen11] denotes two different means of introducing GPS-estimates into a navigation filter called loosely-coupled and tightly-coupled integration, see for example [ZQF16] for further readings on this topic. A loosely-coupled System fuses INS and GPS data based on GPS-position and velocity measurements whereas in a tightly-coupled system the raw Pseudorange and Deltarange-measurements are used instead of the resulting position estimate. This yields the advantage of stabilizing the position estimate by additional incomplete measurements in case only 3 or less satellites are visible which prohibits a full GPS-position output[3].
This concept of loosely- vs. tightly-coupled systems is expanded to USBL in simulation [MOSV06] and experiments [MOS13] where the known positions of reference transponders together with the TODA and slant range measurements are reformulated to pseudorange measurements which are then fed to the navigation filter like GPS-pseudorange measurements, see also example 1. Consequently, loosely-coupled integration is referred to the case where only *full* USBL-position measurements combining slant range and direction measurements like in equation 3.29 are fed to the navigation filter. It is not surprising that using only the full USBL position measurements will yield suboptimal results especially for EKF since the necessary multiplication operation violates the AWGN assumption. In the scheme presented in [MOSV06], the Pseudorange formulation does not yield additional information in itself but it successfully maintains the AWGN assumption on the measurements. However, the AWGN assumption also holds if direction and range are still treated as separate measurements. As such, using separate direction and slant range measurement models for the filter equations can be considered a tightly-coupled integration scheme if, additionally, the target's true position (reference point) is included as a constant, a periodic pseudo-measurement or as part of the state estimation vector.

Note that in [ZSC+16] a tightly-coupled *pseudorange* integration scheme is applied for an LBL instead of a USBL system. In a LBL scenario, where signal correlation yielding TDOA measurements can not be applied as in USBL applications, this pseudorange scheme is beneficial since the situation is comparable to classic GPS-positioning as the transponders are spaced widely apart from each other, see again example 1.
Keeping that in mind, the tightly-coupled alike technique used in this work is limited to separating the USBL measurements into range and direction measurements since in

[3] This assumes the clock-error needs to be corrected as well. Otherwise, if a precise clock is available, 2 or less visible satellites will lead to incomplete position output.

that cases the AWGN assumptions remain justified and can as such be used in a EKF setting.

Last but not least it should be mentioned that improving USBL technology is of course still a subject of today's research, see for example [PMdSR17] and [KKG$^+$17].

## 3.8. Strapdown Navigation Details

Given the error models provided in the previous sections, the classic Strapdown Navigation equations can be provided.
A typical (robotic) filtering setting, as given in section 2.2, requires a system transition model of the type $x_{n+1} = f(x_n, u_n, w_n)$ with state $x_n$, control input $u_n$ and system transition noise $w_n$ accompanied by an observation model $z_n = h(x_n)$. In that sense, it would be natural to use the AUV's dynamic model, introduced in section 2.6, as the system transition model and use the sensor error models presented within this chapter as observation model, fusing both through e.g. some Extended Kalman or Particle Filter.

However, in many navigation scenarios where inertial sensors are employed, it is common to derive a pseudo system transition model from the basic inertial sensors (gyroscope and accelerometer) by forwarding the measurements directly into fairly general system equations with noise parameters derived from the sensor's characteristics, see [Sav98a, Sav98b] for detailed review. Only the further measurements like magnetic compass, DVL, GPS etc. are then used as classic observations in a filtering setting as in [PMdSR16, LLH$^+$05, ACC$^+$16]. This approach provides many advantages such as general applicability, robustness, simplicity, execution speed and modularity, meaning the navigation core (sensors and code) can be acquired from a vendor as mostly a standalone navigation solution independent of specific AUV's dynamic characteristics. The scheme is illustrated in figure 3.13.
The same approach is employed in this work. Both the continuous time differential equations and their discrete version (implementation) for the system transition model are presented within the following subsections.

### 3.8.1. Continuous Time Equations

Since the central vector for estimating the dynamic model parameters is the vehicle velocity $\vec{\nu}^b_{b/n}$, given in body-coordinates, the system transition is formulated using this vector. This differs from other literature, e.g. [Wen11, Gel15], where the body velocity coordinates are estimated in the $\{n\}$-frame: $\vec{\nu}^n_{b/n}$. Consequently, the full time dependent state estimation vector $\vec{x} = \vec{x}(t)$ is given by position, velocity, orientation quaternion and optional IMU bias terms, see section 3.3:

$$\vec{x} = \left( \vec{p}^{\,n,T}_{b/n},\ \vec{\nu}^{\,b,T}_{b/n},\ \vec{q}^{\,n,T}_{b},\ \vec{b}^{\,T}_{a},\ \vec{b}^{\,T}_{\omega} \right)^T \in \mathbb{R}^{16} \tag{3.39}$$

Following the notations and equations 3.6, 3.10 and the ones given in section 2.4.1, the Strapdown equations can be formulated as follows (assuming, again, the simplifications are valid due to slow navigation):

$$\dot{\vec{p}}^n_{b/n} = R^n_b \cdot \vec{\nu}^b_{b/n} \tag{3.40}$$

$$\dot{\vec{\nu}}^b_{b/n} = \tilde{\vec{f}}^b_{ib} - \vec{b}_a - \left(\tilde{\vec{\omega}}^b_{b/n} - \vec{b}_\omega\right) \times \vec{\nu}^b_{b/n} + R^b_n \vec{g}^{\,n} - \vec{n}_a - \vec{\nu}^b_{b/n} \times \vec{n}_\omega \tag{3.41}$$

$$\dot{\vec{q}}^n_b = \frac{1}{2}\vec{q}^n_b \bullet \begin{pmatrix} 0 \\ \tilde{\vec{\omega}}^b_{nb} - \vec{b}_\omega \end{pmatrix} - \frac{1}{2}\vec{q}^n_b \bullet \begin{pmatrix} 0 \\ \vec{n}_\omega \end{pmatrix} \tag{3.42}$$

$$\dot{\vec{b}}_a = \vec{n}_{b_a} \tag{3.43}$$

$$\dot{\vec{b}}_\omega = \vec{n}_{b_\omega} \tag{3.44}$$

In this case, bias drifts are modelled as random walks although also first order Gauss-Markov processes could be used instead. The ODEs have been ordered such that the noise terms appear as (additive) state dependent noises in the last terms of the equations.

Equation 3.40 to 3.44 can be written in matrix form replacing the cross product by the skew symmetric matrix $S_3$ and the quaternion product by its (reverse) matrix multiplication representation as explained in section 1.4.2.

Then, using system matrix $A_{sd}(t)$, (pseudo) control input matrix $B_{sd}(t)$, vector $u_{sd}$, noise matrix $Q_{sd}(t)$ and noise vector $\vec{w}$

$$A_{sd_1}(t) = \begin{pmatrix} 0_{3\times3} & R^n_b(\vec{q}) & 0_{3\times4} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & -\left[(\tilde{\vec{\omega}}^b_{b/n} - \vec{b}_\omega)\times\right] & 0_{3\times4} & -I_{3\times3} & 0_{3\times3} \\ 0_{4\times3} & 0_{4\times3} & \frac{1}{2}Q^r_m(\tilde{\vec{\omega}}^b_{nb} - \vec{b}_\omega) & 0_{4\times3} & 0_{4\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times4} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times4} & 0_{3\times3} & 0_{3\times3} \end{pmatrix}$$

$$A_{sd_2}(t) = \begin{pmatrix} 0_{3\times3} & R^n_b(\vec{q}) & 0_{3\times4} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & -\left[(\tilde{\vec{\omega}}^b_{b/n} - \vec{b}_\omega)\times\right] & 0_{3\times4} & -I_{3\times3} & 0_{3\times3} \\ 0_{4\times3} & 0_{4\times3} & \frac{1}{2}Q^r_m(\tilde{\vec{\omega}}^b_{nb}) & 0_{4\times3} & -\frac{1}{2}Q^c_m(\vec{q}^n_b) \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times4} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times4} & 0_{3\times3} & 0_{3\times3} \end{pmatrix}$$

$$B_{sd}(t) = \begin{pmatrix} 0_{3\times3} & 0_{3\times3} \\ I_{3\times3} & R^{n,T}_b(\vec{q}) \\ 0_{4\times3} & 0_{4\times3} \\ 0_{6\times3} & 0_{6\times3} \end{pmatrix}, \quad u_{sd}(t) = \begin{pmatrix} \tilde{\vec{f}}^b_{ib} \\ \vec{g}^n \end{pmatrix}$$

$$Q_{sd}(t) = \begin{pmatrix} 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ -I_{3\times3} & -\left[\vec{\nu}^b_{b/n}\times\right] & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & -\frac{1}{2}Q^c_m(\vec{q}^n_b) & 0_{3\times3} & 0_{3\times3} \\ 0_{6\times3} & 0_{6\times3} & I_{6\times6} & \end{pmatrix}, \quad \vec{w} = \begin{pmatrix} \vec{n}_a \\ \vec{n}_\omega \\ \vec{n}_{b_a} \\ \vec{n}_{b_\omega} \end{pmatrix}$$

the Strapdown equations can be written in matrix form by a continuous-time state-space model:

$$\dot{\vec{x}}(t) = A_{sd_i}(t) \cdot \vec{x}(t) + B_{sd}(t) \cdot u_{sd}(t) + Q_{sd}(t) \cdot \vec{w} \tag{3.45}$$

where the matrices are time (and state) dependent. For simplicity, the noise vector is assumed to have an independent white Gaussian distribution, i.e.:

$$\vec{w} \sim \mathcal{N}(0_{12\times 1}, Q_{ahrs}), \text{ for}$$

$$Q_{ahrs} = \begin{pmatrix} \sigma_a^2 \cdot I_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & \sigma_\omega^2 \cdot I_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} & \sigma_{b_a}^2 \cdot I_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & \sigma_{b_\omega}^2 \cdot I_{3\times 3} \end{pmatrix}$$

Note that both versions $A_{sd_1}, A_{sd_2}$ and their combinations are possible. This is due to the fact that the system matrix itself contains state variables for solving the ODEs resulting in a non-unique matrix representation. This issue needs to be addressed when a (discrete) implementation is derived which is done in the following section. For observability of the gyro bias drift it is important not to let the last column contain all 0s only.

### 3.8.2. Discrete Time Equations

For numerical computing the continuous system, given by equation 3.40 to 3.44, must be transformed into a discrete-time counterpart. In general, this can be done via classical numerical methods for solving ODEs [DB13] like forth-order Runge-Kutta. However, given state-space formulation it is often more suitable to derive a form of the type

$$\begin{aligned} \vec{x}_{k+1} &= A_k \cdot \vec{x}_k + B_k \cdot u_k + \vec{w}_k \\ \vec{w}_k &\sim \mathcal{N}(0, Q_k) \end{aligned}$$

which can directly be used in (Extended Kalman) filtering. Note that both $A_k = A_k(\vec{x}_k)$ and $B_k = B_k(\vec{x}_k)$ are state dependent matrices.
In case equation 3.45 would feature time-independent matrices the discrete time counterpart can be obtained from [BH97] assuming zero-order hold on $u(t)$ by the matrix exponential over the sample time $t_s$:

$$\begin{aligned} A_k &= e^{At_s} \\ B_k &= \int_0^{t_s} e^{A\tau} d\tau \cdot B \\ Q_k &= \int_0^{t_s} e^{A\tau} Q e^{A^T \tau} d\tau \\ R_k &= \frac{R}{t_s} \end{aligned}$$

where $R$ denotes the AWGN covariance matrix for observations ins case they are received through integration over the sample interval.

The same scheme can be used for time-invariant matrices if the sample time is sufficiently small such that the matrices can be considered constant in these intervals. Nevertheless, even in the time-invariant case these expressions can become quite tedious to evaluate. Thus, even further simplifications can be applied in practice, e.g. approximating the matrix exponential through a first-order Taylor series yielding a simple Euler integration scheme.

The system given in the previous section can be discretized by making use of the matrices' special structure. The basic system transition matrix was discretized with preferably high accuracy whereas the noise influence was approximated through first order (Taylor) series.
The main problem for the discretization technique is the treatment of biases as explicit system states. Otherwise the discretization is not trivial but quite simple. As such, only the core ideas of the discretization routine including biases are given here whereas the details are given in the Appendix, section A.3.
The orientation update, equation 3.42, can be discretized independently by assuming the turn rate remains approximately constant $\vec{\omega}^b_{b/n} \approx \vec{\omega}^b_{b/n,k}$ during the sample interval time $t_s$, see [Sab06]:

$$\Delta\vec{\sigma}_k := t_s \cdot \vec{\omega}^b_{b/n,k}, \quad r_k := \begin{pmatrix} \cos(\|\Delta\vec{\sigma}_k\|/2) \\ \frac{\Delta\vec{\sigma}_k}{2} \cdot \operatorname{sinc}(\|\Delta\vec{\sigma}_k\|/2) \end{pmatrix} =: \begin{pmatrix} r_c \\ \frac{\Delta\vec{\sigma}_k}{2} \cdot r_s \end{pmatrix}$$
$$\vec{q}^n_{b,k+1} = Q^r_m(r_k) \cdot \vec{q}^n_{b,k}$$

Expanding the turn rate and simplifying further yields the approximation

$$\begin{aligned} \Delta\tilde{\vec{\sigma}}_k &:= t_s \cdot \left(\tilde{\vec{\omega}}^b_{b/n,k} - \vec{b}_\omega\right) \\ \Rightarrow \vec{q}^n_{b,k+1} &\approx Q^r_m(\begin{pmatrix} r_c \\ \frac{t_s}{2} \cdot \tilde{\vec{\omega}}^b_{b/n,k} \cdot r_s \end{pmatrix}) \cdot \vec{q}^n_{b,k} \\ &\quad - \frac{t_s r_s}{2} Q^c_m(\vec{q}^n_{b,k}) \cdot \vec{b}_{\omega,k} - \frac{t_s r_s}{2} Q^c_m(\vec{q}^n_{b,k}) \cdot \vec{n}_{\omega,k} \end{aligned} \tag{3.46}$$

where noise propagation is approximated through a first-order Taylor expansion. Other approximations are suitable as well, though. The terms $r_c$ and $r_s$ can also be omitted within the bias and noise part of the equation since they are close to 1 if the sample interval is sufficiently small.
For $x = \|\vec{\omega}^b_{b/n,k}\| \cdot t_1$ and $\vec{\omega}^b_{b/n,k} \approx \tilde{\vec{\omega}}^b_{b/n,k} - \vec{b}_{\omega,k}$ the matrix exponential can be given explicitly by [BS93]:

$$e^{t_1\left[\vec{\omega}^b_{b/n,k}\times\right]} = I + \operatorname{sinc}(x) \cdot t_1 \cdot \left[\vec{\omega}^b_{b/n,k}\times\right] + \frac{1}{2}\operatorname{sinc}^2\left(\frac{x}{2}\right) \cdot t_1^2 \cdot \left[\vec{\omega}^b_{b/n,k}\times\right]^2$$

Further, let a variant of this, using the measured turn rate in the linear term as opposed to the true/estimated one, be defined as:

$$e_*^{t_1\left[\vec{\omega}^b_{b/n,k}\times\right]} := I + \operatorname{sinc}(x) \cdot t_1 \cdot \left[\tilde{\vec{\omega}}^b_{b/n,k}\times\right] + \frac{1}{2}\operatorname{sinc}^2\left(\frac{x}{2}\right) \cdot t_1^2 \cdot \left[\vec{\omega}^b_{b/n,k}\times\right]^2$$

This yields the final structure which is used in the further experiments (further details can, again, be found in the Appendix, section A.3):

$$A_{sd,k} = \begin{pmatrix} I_{3\times3} & R_b^n \cdot e_i(t_s) & 0_{3\times4} & -R_b^n \cdot e_{ii}(t_s) & 0_{3\times3} \\ 0_{3\times3} & e_*^{t_s\left[\vec{\omega}_{b/n,k}^{\,b}\times\right]} & 0_{3\times4} & -e_i(t_s) & -t_s\mathrm{sinc}(x)\left[\vec{\nu}_{b/n,k}^{\,b}\times\right] \\ 0_{4\times3} & 0_{4\times3} & Q^* & 0_{4\times3} & -\frac{t_s r_s}{2}\cdot Q_m^c(\vec{q}_{b,k}^{\,n}) \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times4} & I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times4} & 0_{3\times3} & I_{3\times3} \end{pmatrix}$$

$$Q^* := Q_m^r\left(\begin{pmatrix} r_c \\ \frac{t_s\cdot r_s}{2}\cdot\tilde{\vec{\omega}}_{b/n,k}^{\,b} \end{pmatrix}\right), \quad \text{(only defined here for spacing)}$$

where

$$e_i(t_2) := \int_0^{t_2} e^{t_1\left[\vec{\omega}_{b/n,k}^{\,b}\times\right]} dt_1 \tag{3.47}$$

$$e_{ii}(t_s) := \int_0^{t_s} e_i(t_2) dt_2. \tag{3.48}$$

The latter expressions can be solved symbolically in Matlab, see again the code within the appendix. The formulation $e_*^{t_1\left[\vec{\omega}_{b/n,k}^{\,b}\times\right]}$ is used such that the part containing $b_\omega$ can be separated from $e^{t_1\left[\vec{\omega}_{b/n,k}^{\,b}\times\right]}$. As such, this (linear) part can be moved to the last column of the same row of $A_{sd,k}$.

It is assumed the discrete gyro and accelerometer noise samples are physically obtained through integration over the observation interval time $t_{obs}$. This is due to the fact that they are not truly a part of the system noise but rather an observation noise forwarded to the system equations. Usually $t_{obs}$ would be equal to $t_s$. However, this is not the case for the data examined in the next chapter. Thus, it is treated separately here:

$$\vec{n}_{a,k} = \frac{1}{t_{obs}} \int_{t_k}^{t_k+t_{obs}} \vec{n}_a(t) dt$$

$$\overset{2.20-2.22}{\Rightarrow} \vec{n}_{a,k} \sim \mathcal{N}(0, \frac{\sigma_a^2}{t_{obs}} I_{3\times3}) \tag{3.49}$$

The analogue applies for the discrete gyro noise.
Formally assuming an analogue a pseudo-integration time for the discretized bias (drift) terms, but using $t_s$ instead of $t_{obs}$, and collecting the result within matrix $Q_{ahrs,obs}$ allows a convenient formulation: the entire system noise can be discretized using a simple forward Euler integration scheme on the *original ODE* in the style of [Sab06].

$$\vec{w}_k \sim \mathcal{N}\left(0, t_s Q_{sd} \cdot Q_{ahrs,obs} \cdot (t_s Q_{sd})^T\right)$$

where $Q_{sd}$ denotes the matrix from the continuous version with the time dependent state variables being replaced by the respective discrete time states.

For the control input matrix $B_k$, the explicit formula, given above, was used by inserting the above given matrix exponential and using Matlab for solving the integral symbolically.

The resulting Matlab expressions for $e_i$ and $e_{ii}$ from equations 3.47 and 3.48 contain numeric difficulties due to the Matlab Symbolic Toolbox' slightly inferior handling of the $sinc$-function[4]. Depending on the evaluation, an apparent singularity may occur for turn rates close to 0, i.e. $\|\vec{\omega}\| \approx 0$. This singularity is removable in mathematical terms. In order to overcome it numerically, $\|\vec{\omega}\|$ was replaced by $10^{-12}$ in case it was smaller than that value.

#### Reduced version for constant bias terms

**Remark 10** (Bias Tracking)**.** *Note that for high-class sensors the bias drift can be assumed* 0 *over a large period of time. In that case, only an initial (non-trivial) calibration routine needs to be performed upon start-up. Afterwards, the bias can be assumed constant. As such, only the* $A_{sd_1}$*-type formulation would be important and the biases can be removed from the estimation vector* $\vec{x}$*. However, the formulation using explicit drifting biases is the most general one. From a practical point of view it is favourable to include them in the equations explicitly even when the drifts are mostly negligible .*

In case the bias terms are assumed constant over time, the system transition matrix can be reduced to the first $10 \times 10$ entries by integrating the gyro-bias parts into the terms containing the turn-rates and moving the acceleration bias into the (pseudo) control input, i.e. subtracting it from the measured specific force.

In this work only industrial grade IMUs were used, see section 4.3.1. As such, tracking the biases is essential. Moreover, it is necessary to receive an accurate navigation result since stabilized orientation output of the AHRS lacks an estimate of the biases - as without GPS, they are unobservable for the device [Wen11].

### 3.8.3. Including external measurements

The core Strapdown algorithm is integrating only specific force and gyro measurements from the AHRS within the system transition matrix $A_k$ and (pseudo) control inputs $B \cdot u_k$.

In order to correct the estimation drift over time, the measurements from aid sensors such as depth/pressure, magnetic compass, velocity, position etc. need to be included as well whenever available. This can be achieved by directly using the measurement equations given in the previous sections with appropriate noise constants.
At this point, many different filtering schemes can be employed, e.g. EKF, UKF or PF. Since all measurement equations are given as explicit differentiable equations with AWGN, the EKF is the simplest choice w.r.t. implementation. For this, the measurement models given in this chapter need to be derived w.r.t. the system state. This was done using the Matlab Symbolic Toolbox. After that, the general EKF equations can be used immediately.

[4] For R2017a, the Matlab Symbolic Toolbox replaces $sinc$ explicitly by an expression of the type $\frac{sin(x)}{x}$ which may yield unnecessarily "awkward" resulting expressions after derivation etc.

Figure 3.13 provides an overview of the EKF scheme. In practice an error state form [Wen11] from a linearized Kalman filter might be preferred to a direct estimate of the state vector. Online, the Strapdown core would run at a higher rate, e.g. $100Hz$, than

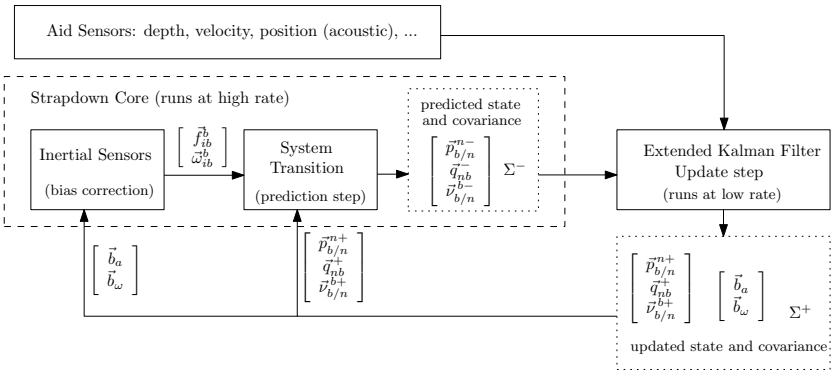


Figure 3.13.: Overview of AUV Navigation with Strapdown and EKF. Identical to figure 2.9.

the drift correction within the EKF, e.g. at $10Hz$. Also, it is possible to calculate predicted state covariance matrices denoted by $\Sigma^-$ in the figure only within the EKF step, saving computation time.
The formulation provided in the previous sections is, however, given as explicit state form instead of error state form. The formulation was chosen here, since it provides a more general and intuitive perspective and easier use of different filtering schemes like the PF. For a detailed discussion of linearized and extended Kalman filters the reader is referred to [May82].

## 3.9. 6-DoF AUV Dynamic Model in Simulink

The dynamic model has been implemented as described in the section 2.6 using Matlab Simulink [MAT17]. The result is briefly presented in this section.

For illustration, the model overview is separated into three main parts: the actuation model, seen in figure 3.14.
The hydrodynamic part can be seen in figure 3.15. The model also includes ocean current effects introduced via the block on the upper left part called “v_c_NED”.
The gravity/buoyancy and rotation matrix part is depicted in figure 3.16.
The appearing signals must be connected beyond the figures according to their names.

The naming and notation follows the conventions presented in section 2.6.
Further subsystems can be found in the Appendix, section C.1.

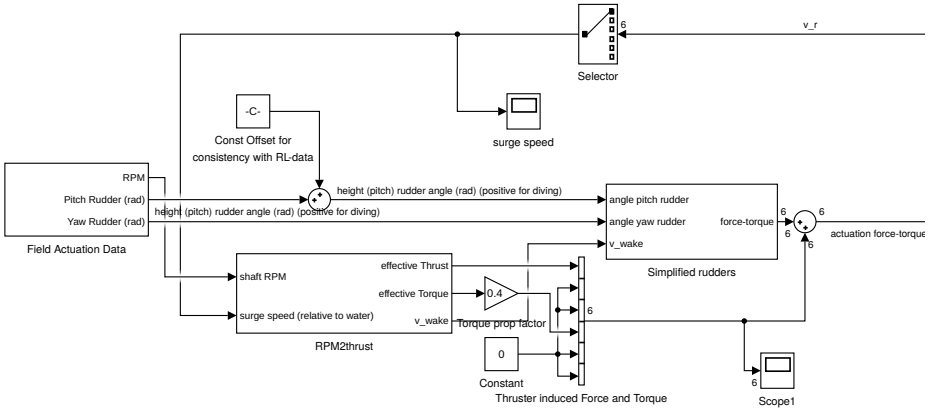


Figure 3.14.: Simulink model of the AUV thruster and rudder (overview) with field actuation data inserted for model identification.

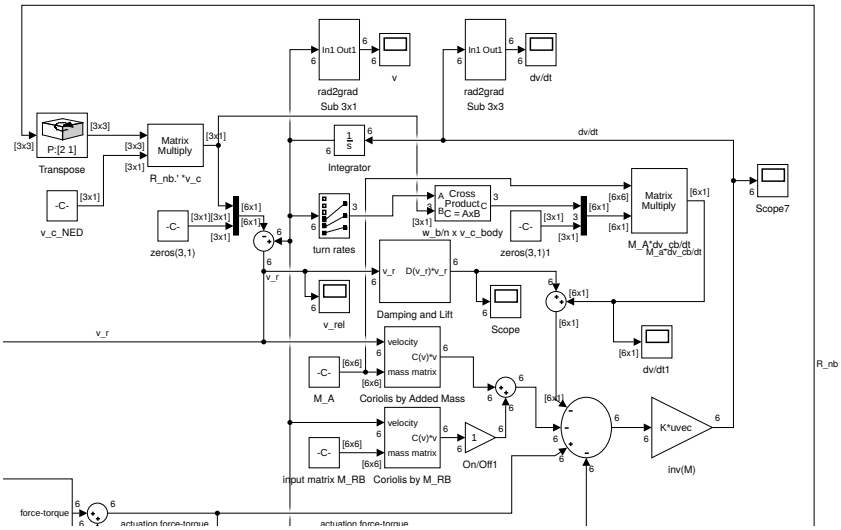


Figure 3.15.: Simulink model of the AUV hydrodynamic model (overview), including water current influences. The lowest incoming arrow of the big sum-element (bottom right) represents the influences of gravity and buoyancy $\vec{g}(\eta)$, see figure 3.16.

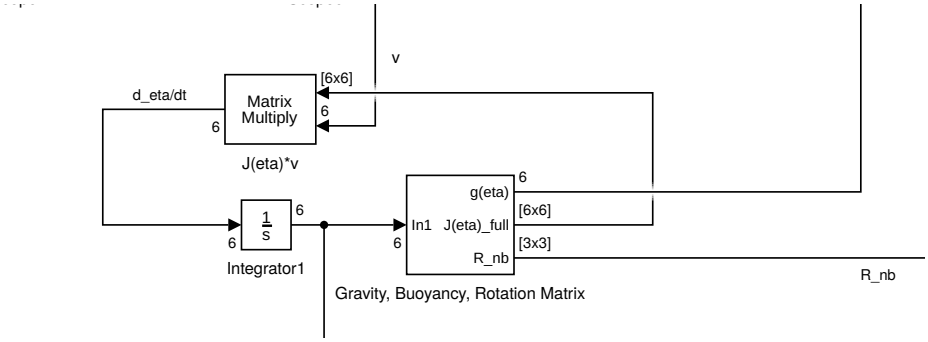


Figure 3.16.: Simulink model of gravity, buoyancy and rotation matrix (overview). The output of the integrator is the NED-position and orientation $\vec{\eta}$.

## 3.10. MA Navigation Implementation

As already mentioned in the explanation of MA navigation, section 2.7, only the scheme according to Koifman and Bar-Itzhack [MK91] [KBI99] [CGTH13] was implemented since this scheme can extend an already existing navigation system. Thus, it appears to be practically more relevant and usable than re-writing the estimation core from scratch as it is necessary for Vasconcelos [VSOG10] scheme.

The Simulink model presented in section 3.9 was used to generate the virtual measurements. This requires setting the (noisy) output of the AHRS and the EKF estimation as initial values for the Simulink model. Please refer to section 2.5.1 and 3.11 on why an EKF was used in the Monte Carlo loop.

During the main simulation considered, i.e. the lawn mower motion to scan the seabed in section 4.7.2, DVL sample rate is limited to at most $2Hz$. The work flow for integrating the MA scheme was chosen as follows:

1) Receive simulated DVL measurement at time step $k$
2) Use resulting estimated system state to initiate the Simulink model
3) Run Simulink for a simulated time of $0.3\,s$ representing 3 time steps[5]
4) Provide result as a virtual body velocity measurement at time step $k+3$
5) Proceed with step 1 at $k \leftarrow k+5$.

This scheme was chosen such that the initial velocity value for the dynamic model simulation always has a small estimation uncertainty which is the case after a successful DVL measurement.
Of course, this scheme can be used differently and in [HH11] it is shown the greatest

[5] Recall: one time step was chosen to have a duration of $0.1s$

benefit of this scheme could be during USBL or DVL failure. However, in this work the emphasis is on the benefit of MA navigation without significant sensor drop-outs.

In terms of sensor equations, the virtual dynamic model output can be integrated like the DVL equations with appropriate noise constants.
These constants were chosen such that they reflect the typical PCRB values seen in the standard navigation data, see section 4.7.2:

$$h_{MA}(\vec{\nu}_{b/n}^{b,DM}) = \vec{\nu}_{b/n}^{b} + \mathcal{N}(0, \Sigma_{MA})$$

Please note that in practice it would be favourable to include an additive bias term in the above equation for handling unknown correlations and systematic difference to the real AUV's dynamic [HH11]. For simplicity, they were omitted in this simulation.
This requires a switching scheme for $\Sigma_{MA}$ during turns as the estimation accuracy drops significantly during turns compared with the AUV following a straight line. As in [HH11] a switching scheme with hysteresis behaviour was chosen, where

$$\Sigma_{MA} = \sigma_{MA}^2 \cdot \begin{pmatrix} \frac{1}{32} & 0 & 0 \\ 0 & \frac{1}{32} & 0 \\ 0 & 0 & \frac{3}{16} \end{pmatrix}$$

$$\sigma_{MA} = 0.025 \quad \text{During straight maneuvers}$$

$$\sigma_{MA} = 0.8 \quad \text{During or close to turns}$$

Optionally, the MA scheme can be suppressed completely during turns.
Of course, much more sophisticated noise schemes can be considered but this one is an approach which could be used similar in practice - though most likely with different constants based on experimental data.
Please refer to section 2.7 why not more elements of the Simulink output were used. The same issues apply for using more than an effective $1\,Hz$ sample rate for integrating virtual measurements from the dynamic model.

Despite Simulink's generally fast runtime there is some significant overhead in starting up a simulation. This adds significant total computation time when starting multiple short time simulations.
Neither Simulink's accelerated nor its rapid acceleration nor the Matlab internal option for running Simulink models in parallel could alleviate the runtime issues. Thus, for speed-up only ten models were used at each respective discrete time step and the output was distributed equally such that each model instance provided the same amount of Monte Carlo realization with virtual measurements.

## 3.11. PCRB Implementation

In this section, the PCRB implementation is presented for computing approximate PCRBs in the underwater navigation setting.

There is a wide range of commercially available accelerometer and turn rate sensors which are usually categorized by their performance and noise characteristics in descending quality: navigation grade, tactical grade and industrial grade [Gel15, ZND+10]. There is even a category better than navigation grade - usually denoted strategic grade [HW10] - but sensors of this type are usually too specific, too big and too heavy or not applicable for AUV navigation. To compute possibly tight PCR bounds, navigation grade sensor constants are used in this chapter. All other noise constants such as the ones for USBL, DVL and depth/pressure remain the same as were presented in the previous sections, i.e. 3.7, 3.4 and 3.5 since they already reflect the noise constants of high quality sensors of their type.
For high-quality sensors, the ideal noise constants used for the PCRB computations here can be found in table 3.1.

Table 3.1.: Overview of ideal noise standard deviations modelled by Gaussian noise sources. Please note the physical units may differ from the ones used in standard literature. This is explained in section 4.3.4

| **Type** | **Range** [Wen11] | **Value used** | **Remark and Reference** |
|---|---|---|---|
| $\sigma_a$ | $\sim 17\,\mu g$ | $3\,\mu g$ | [ZND+10, Gel15, Hon05] |
| $\sigma_\omega$ | $0.01\,^\circ h^{-1}$ | $0.0035\,^\circ h^{-1}$ | Ring Laser Gyro [Hon09] |
| $\sigma_{b_a}$ | $< 100\,\mu g\, s^{-1}$ | $25\,\mu g\, s^{-1}$ | [Gel15] |
| $\sigma_{b_\omega}$ | $0.001 - 10\,^\circ h^{-1}\, s^{-1}$ | $0.0035\,^\circ h^{-1} s^{-1}$ | Ring Laser Gyro [Hon09] |
| $\sigma_m$ | ($\leq 60 \mu G$ in [MOS13]) | $30\,\mu G$ | Based on [Hon08]: $10\,\mu G$ |

The noise constants given in table 3.1 are based on the navigation grade commercially available Honeywell Ring Laser Gyro GG 1320AN [Hon09], the Honeywell QA 2000 (QA 3000) accelerometer series [Hon05] and the Honeywell HMC 1001 magnetometer series [Hon08]. Note that the estimate based on [Hon08] would result in $\sigma_m \approx 10\,\mu G$ which seems unusually optimistic. Following the experience from practical difficulties using the magnetometer in SMIS as well as in [MOS13], where the actual practical noise constant was set to $\sigma_m = 60\,mG$ (mills instead of microns!), the value used here is meant to be a compromise between the two.
Further error sources such as scale factor, temperature dependency and axis misalignment are neglected here. It must be noted that some of the values in table 3.1 seem somewhat optimistic but they reflect the ones used in literature. The noise constants used in [MOS13] for example are worse by several orders of magnitude. However, in that work low-cost sensors were used. In the setting of Cramér-Rao it should however be fine to use slightly over-optimistic noise constants.

The assumed sensor positions were slightly adjusted to the ones used in the SMIS AUV, see the Appendix section B.3.1 for the original ones and section B.3.2 for the ones used for the PCRB computations. The main difference is that the AHRS is positioned within the center of the Body frame and the DVL is positioned directly underneath it. All other sensor positions remain unchanged w.r.t. the original SMIS AUV.

The basic procedure to compute the PCRB is depicted in algorithm 1.
The code was implemented in Matlab. Jacobian matrices were generated via the Matlab

**input** : Set-points for reference trajectory, simulation constants, e.g. time steps $K$, number of MC trajectories $P$, initial values etc.
**output**: Approximate FIM $J_k$ for each time step

*Generate "continuous" reference trajectory via Simulink;*
*Derive specific forces $\vec{f}_k$ and turn rates $\vec{\omega}_k$ from reference trajectory;*
**for** $k \leftarrow 1$ **to** $K$ `/* Main time step loop */` **do**
  `/* Generate System noise, observation noise matrix and next reference system state */`
  $[Q_k, R_k, x_k] \leftarrow$ `SystemTransitionMain` $(x_{k-1}, \vec{f}_k, \vec{\omega}_k)$ ;
  $Q_k^{inv} \leftarrow Q_k^{-1}$ ; `/* Explicit inverse for speed-up */`
  $R_k^{inv} \leftarrow R_k^{-1}$;
  **for** $p \leftarrow 1$ **to** $P$ `/* Run this loop in parallel */` **do**
    $\left[x_k^{(p)}, F_k(p), H_k(p)\right] \leftarrow$ `SystemTransitionExtended` $(x_{k-1}^{(p)}, \vec{f}_k, \vec{\omega}_k)$ ;
    `/* Here,` $F_k(p) = \left(\nabla_{x_{k-1}} f_k^T(x_{k-1}^{(p)})\right)^T$ `state Jacobian and`
      $H_k(p) = \left(\nabla_{x_k} h_k^T(x_k^{(p)})\right)^T$ `observation Jacobian. */`
    $x_k^{(p)} \leftarrow x_k^{(p)} + \mathcal{N}(\vec{0}, Q_k)$ ; `/* individual noise sample */`
    $E_f(p) \leftarrow F_k(p)^T$;
    $E_{ff}(p) \leftarrow F_k(p)^T \cdot Q_k^{inv} \cdot F_k(p)$;
    $E_{hh}(p) \leftarrow H_k(p)^T \cdot R_k^{inv} \cdot H_k(p)$;
  **end**
  `/* Approximate expected values serially: */`
  $E_f \leftarrow \frac{1}{P} \cdot \sum_p E_f(p)$;
  $E_{ff} \leftarrow \frac{1}{P} \cdot \sum_p E_{ff}(p)$;
  $E_h \leftarrow \frac{1}{P} \cdot \sum_p E_h(p)$;
  `/* Iterative FIM according to Tichavsky: */`
  $D_k \leftarrow E_f \cdot Q_k^{inv}$;
  $J_k \leftarrow Q_k^{inv} + E_{hh} - D_k^T \cdot (J_{k-1} + E_{ff})^{-1} \cdot D_k$;
**end**

**Algorithm 1:** PCRB computation Pseudo-Code

symbolic toolbox, re-using the code for generating the basic navigation EKF, see section 3.8.2.
In the actual implementation, the history of Fisher matrices is not saved explicitly. Instead, only the current and previous FIM are kept in memory over time. For the history, only the square-root of the FIM's respective inverse's diagonal is saved. Otherwise the computing machine would easily run out of memory.

**Remark 11** (Inverse of $Q_k$)**.** *The system noise matrix given in section 3.8.2 does not have full rank which is due to the NED-part as well the quaternion-part. To overcome this, $Q_k$ is regularized here by adding some fake noise to its diagonal, i.e.*

$$Q'_k = Q_k + diag(\epsilon_f, \epsilon_f, \epsilon_f, 0, 0, 0, \epsilon_{qf}, \epsilon_{qf}, \epsilon_{qf}, \epsilon_{qf}, 0, \ldots, 0)$$

*with choice of*

$$\epsilon_f = 10^{-6}$$
$$\epsilon_{qf} = 0.01 * \sigma_\omega^2$$

*where $\sigma_\omega$ denotes the turn-rate measurement noise constant.*

The inner loop, beginning in line 7 of algorithm 1, was run in parallel using the Matlab *Parallel Computing Toolbox* via the parallel for-loop *parfor*. For e.g. $P = 1000$ and simulation time of $T = 10s$ the computation time could be reduced by a factor of $5$ using this parallelization. The most time consuming functions are given in lines 4 and 8 as they perform a huge amount of basic mathematical operations like multiplication coming from the results of symbolic derivatives and discretization. Turning these functions into Matlab-specific *MEX*-files [mex], a machine-specific compilation using a C-code analogue of the Matlab-code, yielded another speed-up of about $4$. In total, the originally required computation time of $200s$ could thus be reduced to about $14s$ in the mentioned case - very close to the actual simulated time of $10s$. Computation times refer to the employed hardware presented in section 4.1.

As mentioned in section 2.5.1, instead of computing a pure PCRB for the MA navigation evaluation in section 4.7.3 an EKF was put in the Monte Carlo loop. This was done because this captures more closely how the common underwater navigation work-horse - EKF - can cope with MA navigation as an extension to an already existing navigation solution. This should give a more realistic view of how MA navigation can improve the navigation accuracy in practice despite not being a true PCRB.
Otherwise, computing PCRBs for the MA navigation scheme examined in this work would just mean to include more DVL measurements over time with possibly different noise constants which does not at all reflect the difficulties contained in this scheme.

# 4. Evaluation

This chapter is organized as follows.
At first, the computers used for this evaluation are presented in section 4.1, followed by presenting the experimental evaluation data in section 4.2 and explaining the issues contained in those.
Then, the sensor constants needed for the models presented in chapter 3 are derived in section 4.3 including necessary adjustments made to process the given recorded data during the SMIS project. Afterwards, in section 4.4 the navigation algorithms and error models are tested on the experimental data to provide the necessary outputs used in section 4.5 for generating a reference dynamic model. This model and some of the navigation data is then re-used in the main PCRB simulations conducted in section 4.6 and section 4.7.

## 4.1. Hardware Configuration used for Evaluation and Simulation

Two computers were used for the evaluation, running Matlab Code.
Both were using Ubuntu 14.04LTS (64 Bit) running Code in Matlab R2017a. Where applicable, e.g. for MEX-files code generation, gcc 4.9 was used.

The specifications of the first computer:

- Intel Core i7-3770K CPU @ 3.50GHz×8,
- $4 \times 4$GB RAM (DIMM DDR3 1600 MHz by AMI)

This machine was used for most implementations, basic testing and evaluation of the navigation field data.

The second computer, called *SMISERS*, contained the following hardware:

- 2: Intel(R) Xeon(R) CPU E5-2630 v2 @ 2.60GHz (6 physical cores each)
- $4 \times 16$GB RAM (DIMM DDR3 1600 MHz by Kingston)

The latter PC was used mostly for the Monte Carlo Simulations. The default Parallel Toolbox setting was used, employing all 12 physical cores as workers despite the lower processors' clock rate.

## 4.2. Experimental Data Overview

The evaluation Data sets used in this work were recorded in at-sea trials in the Baltic sea at the end of September 2015. Experiments were conducted via the research vessel Elisabeth Mann Borgese (EMB) during its $113th$ mission (EMB113). It is a smaller to mid-sized ship (length ca. $56m$, width ca. $10.8m$) which is in service for the IOW.
Within the respective trials, the AUV was released into the sea for general sea-water testing and collecting data for refined controller design (depth and heading/course).
For all tests, the AUV was located via the USBL mounted within the RV's moon pool. The same device served for communicating status data and commands: desired heading, depth and/or desired rudder angles and thruster's RPM. For simplicity, the communication was set to a simple time division multiple access scheme with acknowledgements[1] for each message. Both AUV and RV sent their data and commands repeatedly at at $1Hz$. Above that, all IMU, CTD and pressure data was recorded at $10Hz$ and DVL data was recorded at $1Hz$ sample rate.

Three test day from the named sea trial, from September 27th to 29th were considered eligible for the experimental evaluation. The most used data was collected on September 28th. During that test day, the EMB was located ca. 60km east of the Baltic island of Bornholm at geographic location: $55°13'17.4"N 16°09'56.2"E$. Water depth in that area was at roughly 100m. An overview of the location within the Baltic sea can be seen in figure 4.1


Figure 4.1.: Location of main Baltic sea experiment from September 28th 2015 marked by the red pin at $55°13'17.4"N 16°09'56.2"E$. East of the Island of Bornholm, between Denmark (North) and Poland (South). (Image courtesy of Google Maps.)

The resulting navigation data is presented in section 4.4. Among other, less noteworthy

[1] acknowledgements are required for receiving slant ranges from the modems

maneuvers, the SMIS AUV performed several up-down spiralling motions as well as typical lawn mower patterns for testing.

Several smaller errors in the data sets (e.g. false conversion etc.) which could be corrected are not mentioned further here. However, there are a few complicating aspects in the data requiring attention explained in the following subsection.

Other experimental data which directly influenced choices and models in this thesis, mostly for the USBL analysis, is explained in [NOW+15] and [ONW+16].

### 4.2.1. Complicating Aspects in Data Sets

The software provided by Evologics is capable of automatically generating GPS positions from USBL measurements by the RV in case also a GPS and RV's AHRS sensor is available via the network. This data was extracted from the software after the experiments and synchronized with the data recorded on the AUV. Since the USBL measurements are transformed into a position estimate, the original separate direction and range measurement are not available any more. This is overcome by pretending the (normalized) direction extracted from the position estimate would still be a regular direction estimate with AWGN noise influence. In section 4.3.3, it is illustrated that this procedure is theoretically still nearly equivalent to the original scheme.
Moreover, the system broadcasting the RV's roll and pitch over the RV's network regularly did not work. Thus, only the ship's heading could be recorded. This introduces an additional error source into the data evaluated here which should be kept in mind when examining the results.

The magnetic measurements taken by the AHRS sensor do not have the expected magnitude. Even worse: there was a significant increase in the magnitude of the magnetic readings when the motor current is being switched on, see figure 4.2 - despite the AHRS being located in the AUV's nose to provide distance to wires carrying strong currents. Although the magnitude of the magnetic readings are biased they still provided a seemingly reliable source of information for stabilizing the yaw estimate. Thus, the compass readings were normalized before usage and only used to stabilize yaw without roll and pitch. This was included in accordingly in the Strapdown/filter implementation.

As mentioned by the end of section 3.4, apart from magnetic interference it would be ideal to place the IMU and DVL close to each other within the AUV's center. Otherwise, additional error sources (from apparent lineare acceleration) are introduced from pitching and yaw motions - which should approximately be pivoting around the AUV's center. However due to magnetic and spatial considerations, the AHRS was placed in the AUV's nose with the DVL placed $\approx 0.5m$ behind it facing downwards.
Above that, the DVL values were recorded after automatic transformation to the NED frame via the DVL's internal orientation sensors. Despite the convenience provided in some applications this yields some inaccuracies since the DVL's orientation is not entirely consistent with the orientation recorded from the AHRS and it is not entirely clear how the transformation was performed in detail. Nevertheless, the DVL's velocity output in NED frame was rather consistent with the AUV's USBL positions and was thus

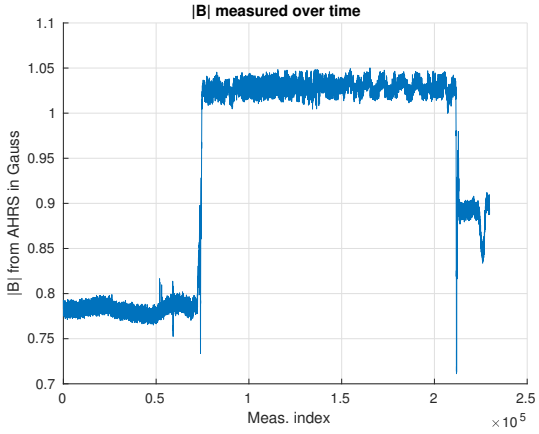


Figure 4.2.: Magnetic Readings from AHRS for $2015-09-28$. The significant increase/jump correlates with switching on the propeller's power. The expected local Earth magnetic field vector would have magnitude $\approx 0.5G$.

considered acceptable for further use. The DVL velocities in NED frame were transformed back the DVL coordinate system using its recorded orientation. Afterwards they were used in the filtering scheme from chapter 3. For cross-checking, the DVL data was also introduced into the filter equations as a measurement in NED-frame without providing significantly different results.

GPS data recorded by the AUV is partly quite noisy or biased. This is most likely due to the GPS antenna being partly submerged through waves which immediately results in heavily disturbed GPS reception quality. Since emphasis in this work is on the submerged navigation these GPS errors are not handled with special care. One option could be to low-pass filter the AUV's recorded GPS positions in advance.
To overcome some of the issues in a simple manner, GPS incorporation into the filter equations was suppressed immediately after intervals where GPS reception failed over some time, e.g. due to the antenna being underwater. That way, the stand-alone GPS receiver was given time to find a stable global position estimate before the measurements are allowed to enter the AUV navigation filter.

### 4.2.2. Pre-Filtering Data

Since the formulation given in section 3.8.2 uses the measured turn rates and specific force within the pseudo system transition matrix. This formulation is quite prone to heavy outliers - be it through communication errors or sensor failures themselves. As such it is important to use some kind of wild-point filtering in advance. For simplicity, obvious turn-rate and specific force outliers were re-set to $0$. Of course, this is not really sound but provides the quickest solution to maintain the Strapdown's stability.

For generating the numerical derivatives from experimental data, a Butterworth filter [SB98] was used. This is necessary for generating the angular accelerations for identification of the dynamic model since only turn rates are measured physically by the AHRS.

## 4.3. SMIS Sensors Characteristics and Adjustments for Recorded Data

In this section, the sensors used in SMIS are characterized in more detail by deriving numeric values for their noise parameters. Additionally, further necessary adjustments for using the sensor data are explained.

### 4.3.1. AHRS/IMU Noise: XSens MTi-700

Following the manufacturer's notes on the industrial grade AHRS by XSens which was used in SMIS, given in [VBLS14], the sensors gyro noise can be modelled as a superposition of AWGN (angle velocity random walk of $0.01^\circ s^{-1}$) together with a biased noise term having a drift parameter (bias stability) of $10^\circ h^{-1}$, see section 3.2. This can be evaluated in figure 4.3 which is in good correspondence with the plot given in [VBLS14]. However, the reported value of $10^\circ h^{-1}$ seems slightly over-optimistic w.r.t. the actual values within the plot[2]. The figure has been created in Matlab using the methods described in section 2.3.3.

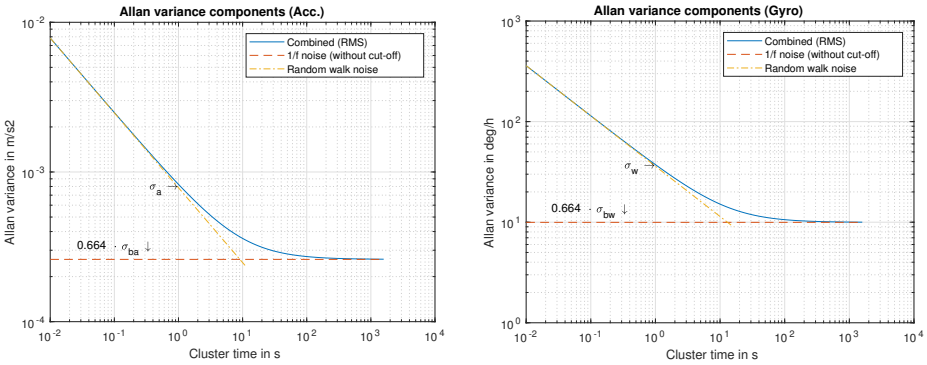


Figure 4.3.: Ideal theoretical Allan variance curves. Left: for Accelerometer using $1/f$-noise with parameter $40\mu g$ and random walk with parameter $80\mu g/\sqrt{Hz}$. Right: $1/f$-noise with parameter $15^\circ h^{-1}$ together with angle velocity random walk (i.e. constant PSD) with parameter $0.01^\circ/(s/\sqrt{Hz})$. Except for the gyro's $1/f$-noise, values are in agreement with [VBLS14].

[2]This reported value has been increased to $12^\circ h^{-1}$ in a more recent version of the paper

The two dominating noise types appearing in figure 4.3 can be explained as follows, see [ESHN08]: The (angle velocity) random walk noise can be modelled as a AWGN noise source and the $\frac{1}{f}$ noise [Hoo76] can be modelled as a drifting bias. The latter can be designed as a random walk itself where the respective constant determines the bias drift rate.
Alternatively, the bias term could be modelled as a first order Gauss-Markov process. However, in terms of Allan variance this would yield different curves which do not correspond as much with the plots from [VBLS14].

In the following, the AWGN together with a random walk bias is used for modelling the XSens Gyro and Accelerometers using the parameters given in figure 4.3.

### 4.3.2. DVL Noise: Worhorse Navigator WHN 600

The DVL used in the experiments features the same beam configuration as is described in section 3.4. The measurement error is given in [wor]. In this manual, the standard deviation of velocity is given as approximately $0.16 - 0.3\%$ of speed travelled, i.e.

$$\sigma_{DVL} \approx 5 \cdot 10^{-3} \frac{m}{s} \text{ at typical speed } \nu < 3 \frac{m}{s} \tag{4.1}$$

Unlike the AHRS noise, this value can be directly used for the discrete navigation equation since it already denotes typical deviations in the data recorded for discrete time steps.

For the additional insecurity introduced by the turn rate measurement, however, the noise discretization is applied as in equation 3.49.

### 4.3.3. Adjustment to USBL measurements

The true direction vector estimate being prone to AWGN yielding a ellipsoidal error shape w.r.t. direction could not be extracted from the data recorded since it is saved as a combined range and direction (i.e. position) estimate. A small simulation revealed that the error arising from retrospectively separating the combined measurement into separate direction and slant-range measurements is negligible. The insight is illustrated in figure 4.4. There, the original simulated noisy direction measurements were normalized ("flattened") to the unit sphere's surface and then evaluated in an EKF setting that expects to be fed the true direction with original AWGN. Despite the discrepancy, the changed data still provided virtually unchanged results from the original noisy direction processing: the difference in the direction estimate was below $0.08°$.
TDOA-noise in the figure was set to $\sigma_{tdoa} = 4 \cdot 10^{-6} s$ as in section 3.7.4. The initial direction estimate was set to $(0.78, 0.5, 0.33)^T$ instead of the ground-truth $(0.81, 0.49, 0.32)^T$. As can be seen for the second sample, the absolute angle error of the flattened estimate can even temporarily be slightly smaller than the non-flattened.
Note that this is a different issue than the one explained in section 3.7.6 which just illustrates why range and direction measurements should favourably be treated separately for correct treatment in an EKF filtering.

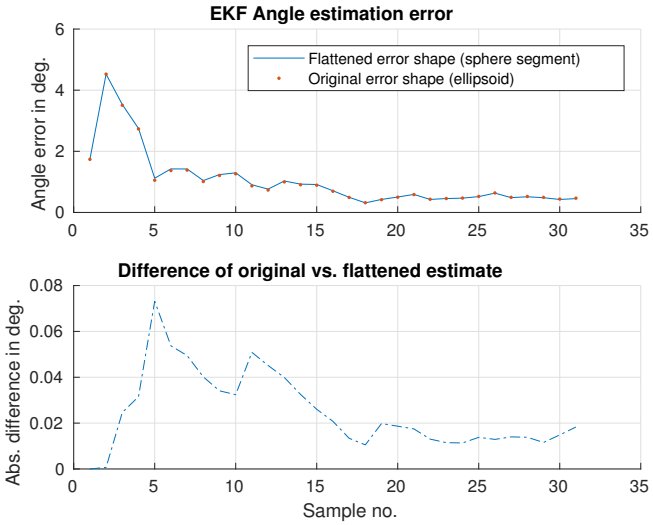


Figure 4.4.: EKF simulation for USBL: direction processed as ideal AWGN error shape (ellipsoid) vs. flattened (sphere surface segment) error shape. Both are treated as regular direction sources with AWGN. Target is fixed to $\vec{r} = (500, 300, 200)^T\ m$ relative to receiving device.

### 4.3.4. Summary of Noise Constants

The sensors used in the SMIS experiments were modelled by adjusting the theoretical values given in the respective data sheets [xse] or manuals in order to achieve a suitable value for computations. See table4.1 as an overview.

Table 4.1.: Overview of noise standard deviations modelled by Gaussian noise sources. "Data sheet" column refers to the values given from the manuals and "Value used" denotes the actual value used instead for computations.

| **Type** | **Data sheet** | | **Value used** | |
|---|---|---|---|---|
| | Typical | Maximum | Strapdown | Orientation stabilization |
| $\sigma_a$ | $80\mu g$ | $150\mu g$ | $150\mu g$ | $4.5mg$ |
| $\sigma_\omega$ | $0.01^\circ s^{-1}$ | $0.015^\circ s^{-1}$ | $0.015^\circ s^{-1}$ | $0.015^\circ s^{-1}$ |
| $\sigma_{b_a}$ | $40\mu g\, s^{-1}$ | – | $40\mu g\, s^{-1}$ | – |
| $\sigma_{b_\omega}$ | $10^\circ h^{-1} s^{-1}$ | – | $15^\circ h^{-1} s^{-1}$ | $15^\circ h^{-1} s^{-1}$ |
| $\sigma_m$ | $200\mu G$ | – | $2mG$ | $2mG$ |

The units in the table are given such that using Gaussian noise sources will yield physically consistent dimensions. Usually, the respective values are given as e.g. $80\mu g/\sqrt{Hz}$ for the noise density (represented by $\sigma_a$) and $40\mu g$ for the bias constants (represented by drift rate $\sigma_{b_a}$). Analogue standard deviations for Gaussian noise are then received

by taking the same numerical values and informally dropping the $/\sqrt{Hz}$ part from the density constant (square integral over all frequencies [ESHN08]) and multiplying a $1/s$ to the bias drift constant.

Note that using Gaussian noise samples is a simplifying assumption. It is also reasonable and common to use first-order Gaussian-Markov biases instead, see section 3.2.
The sensor positions assumed for the SMIS AUV can be found in the appendix, section B.3.1.

### 4.3.5. Sampling Time

The physical sampling frequency of the AHRS is at $100\,Hz$ but it can be configured up to $400\,Hz$. The output was recorded at $10Hz$. This means, the noise samples are integrated over a $10ms$ interval but the discretization interval has a length of $100ms$. Thus, the assumed discretization/integration times for gyroscope and accelerometer are not consistent with the true physical integration times in this case. This is the reason for the adjustment to the measurement matrix made in equation 3.49. The respective values $\Delta t_{obs} = 10ms$ vs. $\Delta t = 100ms$ can be inserted directly for the matrices representing gyro and acceleration noise.

### 4.3.6. Stabilizing Pitch and Roll

In the absence of velocity and position measurements, an AHRS can at least be used to track the vehicle's orientation from a Strapdown algorithm by typically fusing angular velocity measurements (gyro) with earth's magnetic readings and specific force measurements [Wen11]. When the system's apparent acceleration is mostly due to gravity, the specific force can be used to stabilize the orientation (roll and pitch) estimates from the gyroscopes whereas the compass measurements can be used to stabilize the heading. Similar to [Sab06] a threshold switching scheme is used.

$$\vec{h}_{grav} = f_{ib}^{b} \approx -R_{n}^{b} \cdot \vec{g} + \mathcal{N}(0, f_{grav} \cdot \Sigma_{a})$$

with $f_{grav} = 90$ in case $\|f_{ib}^{b} - \vec{g}\| < 0.2$ and $f_{grav} = \infty$ otherwise. The threshold value is according to [Sab06] and the factor $f_{grav}$ was chosen heuristically. For robustness, this scheme is also applied when position and velocity estimates are available.

Note that the XSens AHRS sensors already provide orientation estimates from an internal fusing routine. However, in order to also receive a guess of the estimation quality (covariance matrices) and the gyro/acceleration biases requiring external sensor input, the Strapdown formulation needs to be performed anyway.

Naturally, this scheme is very useful while the AUV is following straight lines at constant speed since in that case there is no additional acceleration and as such the specific force merely represents the gravity vector. However during turns and circles, the specific force always contains a Coriolis/centripetal part as can be seen in equation 3.41. Thus, the quality of the gravity stabilization is impeded during turns.

## 4.4. Navigation Results from Experimental Data

The EKF navigation filter was implemented as described in the previous sections and the previous chapter. This evaluation is mainly intended to verify the sensor models from chapter 3 on real-world data and in order to generate the required data for parametrizing an AUV dynamic model.
Both the error models and the dynamic model is then used for simulation and derivation of the approximate navigation PCRBs.

An overview of the result for the experimental data from $2015-09-28$ is given in figure 4.5. Within the smoothened navigation result, typical $3\sigma$-environments for North-

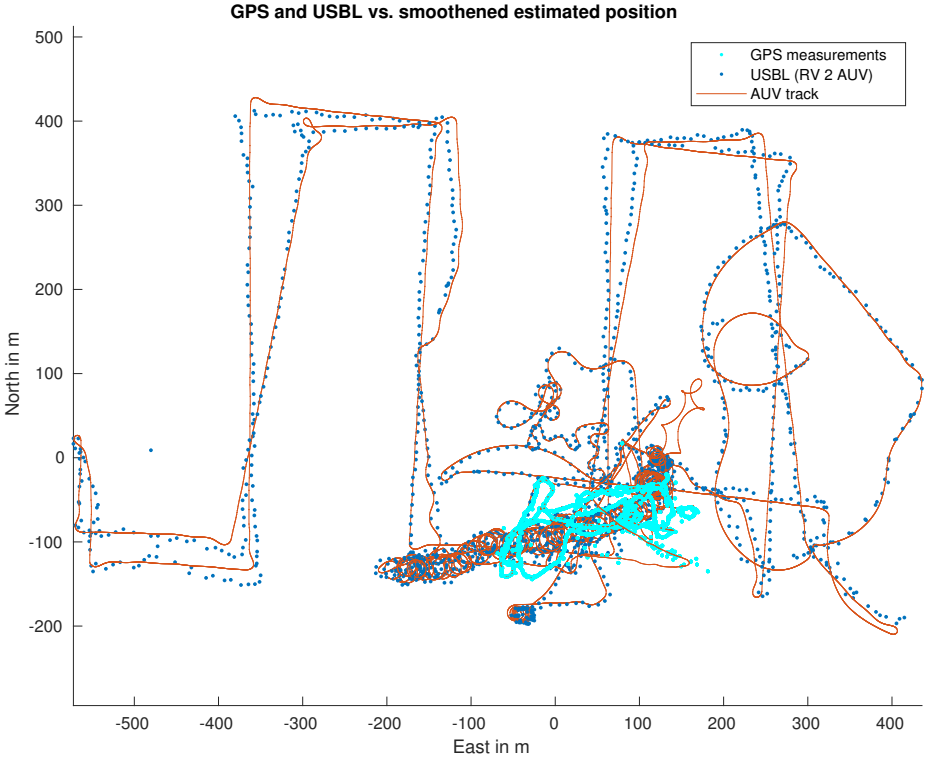


Figure 4.5.: EKF navigation applied to the experimental data recorded on $2015-09-28$ using the RV's localization of the AUV via USBL underwater and partly GPS above water. Topside view with local NED-frame. Data used here was recorded during ca. $3h$ and $50min$.

East position have a magnitude of $2.5-3m$ for times with regular USBL and DVL availability. The data used was recorded during a diving test for ca. $230min$.

Most GPS/USBL outliers have been filtered by basic outlier rejection using respective p-values from the chi-squared distribution as is standard for similar Kalman applications, see [Gel15] and [MLA01].
Some systematic offset/bias within the data is still visible w.r.t. the USBL positions of the RV tracking the AUV (blue dots). The exact source of this problem is not known but it is most likely due to the missing roll/pitch of the RV as described in section 4.2.1, as well as various inaccuracies in assumed sensor positions, alignments etc. All of these error sources can be overcome in commercial AUV operations - as opposed to prototypical developments performed in the SMIS project which this data is based on, see section 1.1.1.

The depth profile of the navigation result shown in figure 4.5 is given in figure 4.6.

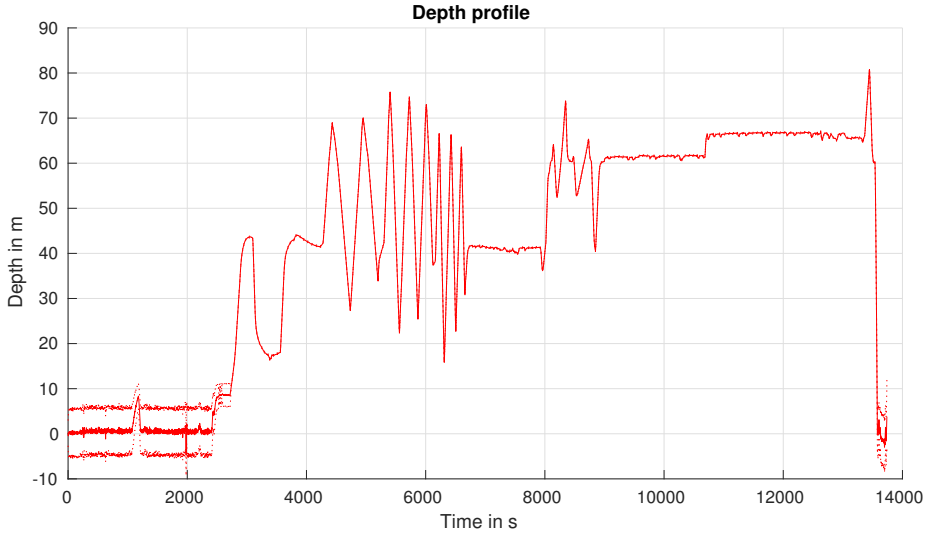


Figure 4.6.: Depth profile from the navigation result shown in figure 4.5. Includes $3\sigma$-environment (dashed line) which is only visible during surfacing - as then the depth measurements covariance is artificially increased to account for waves on the surface.

The vertical zig-zag movement seen between $4000s < t < 7000s$ corresponds with the spiralling-movement which can be seen beneath and left of the GPS measurements in figure 4.5. An extract of the estimated (smoothed) body-fixed velocities can be found in figure 4.7

Despite the flaws within the experimental data base, the filtering results can be considered overall consistent. Especially when comparing the pure USBL-measurements with the pure DVL track estimate, the data was considered sufficiently consistent for testing. The filtering scheme is valuable and effective for the given task. As such, this evaluation fulfilled its purpose for the further theoretical considerations.

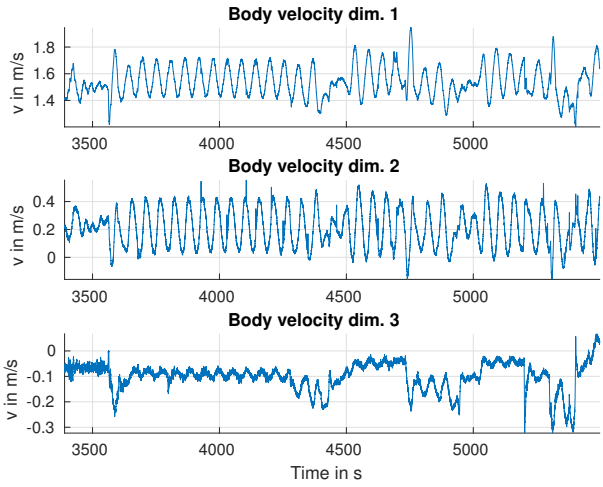


Figure 4.7.: Excerpt of estimated (smooth) body-velocities from the navigation result shown in figure 4.5. The estimated $3\sigma$-environment on this track varies mostly between $0.07-0.15\,\frac{m}{s}$ ($x, y$-velocities) and $0.04-0.21\,\frac{m}{s}$ ($z$-velocities) during DVL-availability. Time window corresponds with the data used for the reference model parameters in section 4.5.5.

## 4.5. Dynamic Model Identification

The results for identifying a dynamic model of the SMIS AUV are presented here.
This includes estimating thruster and rudder coefficients from real-world data as well as estimating the ocean currents present in the navigation results from section 4.4 on which the dynamic model is based. Estimating the ocean current is necessary for correct derivation [HHJ07].
This section is concluded by deriving a reference model used for simulation.

### 4.5.1. Thruster Coefficients

Assuming the general model assumptions w.r.t. thruster output velocity etc. are correct, then the only parameters which need to be identified for the thruster are $T_{|n|n}$ and $T_{|n|u1w}$ from section 2.6.3. As mentioned in [HHJ07], this can be done from open water propulsion tests.

An estimate of $T_{|n|n}$ was obtained from laboratory measurements performed by colleagues of Enitech in order to test the suitability and operation range of the thruster. The results can be seen in figure 4.8. The respective experiment was conducted under steady-state conditions, i.e. the thruster's position remained quasi-constant the entire time. It must be noted, however, that this does not mean that the advance speed can be considered exactly $0$ since the the thrust applied to the water body within a relatively

small ($\sim 5m \times 2m \times 1.5m$) basin did yield a notable movement of the water by circulating around the entire basin through means of a metal wall within the middle. As such, the relative advance speed $u_r$ is greater 0 and the true value for $T_{|n|n}$ should be somewhat larger.

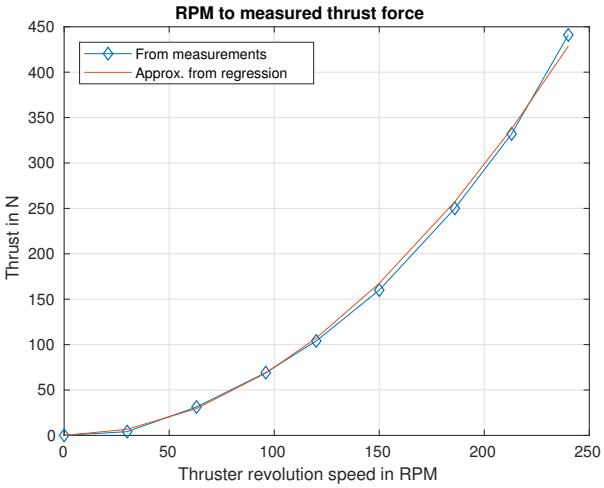


Figure 4.8.: Measured thrust for various revolution speeds at steady state. Used for estimation of parameter $T_{|n|n} \approx 7.4411 \cdot 10^{-3}$ from quadratic approximation in section 2.6.3. Measurements were taken within a small water tank by attaching the isolated thruster to a force-torque sensor attached to a fixed frame. (Values courtesy of G. Körner from Enitech.)

Using similar relations and orders of magnitudes as the ones given in [FB00] the second thruster coefficient was estimated to be $T_{|n|u1w} \approx 100 \cdot (1 - w) \cdot T_{|n|n} = 0.5953$. In fact, receiving a more accurate value of this parameter is not that easy. In the experiment given above, it would be necessary to measure the intake velocity of the surrounding water if open water propulsion tests as mentioned in [HHJ07] are not available.

### 4.5.2. Rudder Coefficients

In [Ole13] rudder's drag and lift coefficients were measured through experiments. Since the SMIS AUV is using the same rudder shape, the values were extracted and used here as: The third graph (projected surface, yellow squares) of figure 4.9 was, generated separately from the approximate rudder shape which is not mentioned in [Ole13]. Note that the lift coefficient is decreased for angles of magnitudes $\geq 22°$. This is due to some stalling effect kicking in at this angle which is typical for many rudder/wing profiles [And99], [JWP33]. In practice, the rudder set points of the SMIS AUV were limited to $\pm 25°$.

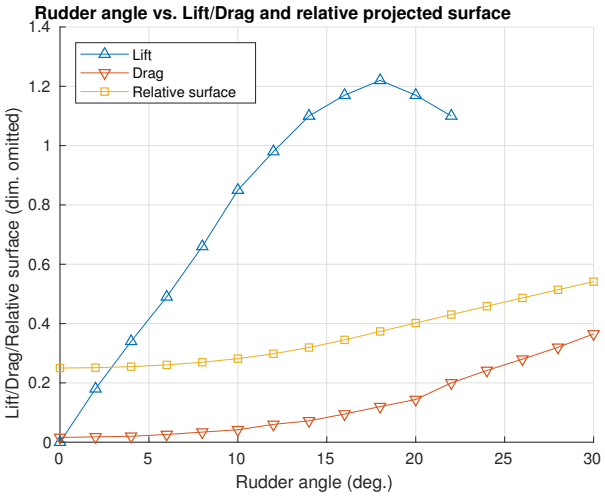


Figure 4.9.: Rudder drag and lift coefficients for the rudders used as given in [Ole13]. Relative projected surface computed by known geometry of the rudder (yellow squares). To receive the actual projected surface, multiply this value by the rudder's cross-area.

The entire rudder's surface is approximately:

$$A_R = 0.180m \cdot 0.1434m = 0.025812m^2$$

this needs to be multiplied by the relative value to receive the true projected surface.

When inserting the example tables for lift, drag and projected surface in equation 2.49 it can be seen that approximating the lift as a linear function of angle and the drag by a quadratic function of angle, as was proposed in [HHJ07], is justified. The estimated values from this procedure are (constants' dimensions omitted):

$$F_L \approx 16.85 \cdot \alpha \cdot v_{wake}^2 \quad \text{lift approximation}$$
$$F_D \approx 8.73 \cdot \alpha^2 \cdot v_{wake}^2 \quad \text{drag approximation}$$

for angles in range of $\pm 25°$. This is intended as an approximation as initial values for possibly finer or more suitable values.

### 4.5.3. Estimating Water Current

As mentioned in section 2.6.1, for using the filtered data in the estimation of the dynamic model the body velocity $\nu_r := \vec{\nu}_{b/w}^{\,b}$ relative to the water is required. This means, the water current must be either measured from a DVL or ADCP device, see section 2.4.2, through measuring the water relative velocity or it estimated from the data. The latter can happen either online or in post-processing.

In this work, the method proposed by [HHJ07] is employed. There, it is assumed the water current remains constant and irrotational within the global NED-frame. Moreover, only velocity within the NE-plane are considered using the relation

$$\vec{\nu}^b_{b/n} = (\vec{c}_1 \ \vec{c}_2) \cdot \begin{pmatrix} u^n_c \\ v^n_c \end{pmatrix} + \vec{\nu}^b_{b/w}, \tag{4.2}$$

where $\{w\}$ denotes the a water-fixed frame and $\vec{c}_1, \vec{c}_2$ denote the first two columns of $R^b_n$. $u^n_c$ and $v^n_c$ denote north and east velocity of the water current.

This means, body velocity is separated into velocity relative to the water body plus the (fixed/stationary) water velocity relative to the NED frame. Note that in the literature, the frame $\{w\}$ is also referred to as *FLOW* in case its x-axis is aligned with the relative flow/stream of the water [Fos11]

The left hand side of equation 4.2 is measured in many samples from the navigation algorithm - most notably from the DVL measurements. Thus, the $5$ unknown can be extracted from a linear LS scheme if multiple samples are used. For this to work, it is firstly required that the actuator set points are almost the same for all samples. Secondly, the AUV must swim in different directions. This can be achieved by letting the AUV swim a rectangle or a spiral pattern. In the first case, it is absolutely essential that the curvy part is excluded from the LS procedure since the rudder is changing its angle during the curves.

The results provided from this procedure are summarized in table 4.2 The results from

Table 4.2.: Identified currents within the data sets using scheme from [HHJ07].

| **Test day** | **Estimated water current velocity** | |
|---|---|---|
| | **North** | **East** |
| 2015-09-28 | $-0.0155\frac{m}{s}$ | $-0.1450\frac{m}{s}$ |
| 2015-09-29 | $-0.0906\frac{m}{s}$ | $-0.0830\frac{m}{s}$ |

the table are, however not to be considered completely reliable and accurate as the steady-state input assumptions from [HHJ07] did not strictly hold in the given data sets. However, this provides at least an estimate of the current strength within the test time.

The data set of $29 - 09$ was not used further in this thesis, but for completeness, the estimated current is provided anyway.

### 4.5.4. Classical Identification

Although the AUV dynamic model presented in section 2.6 is non-linear w.r.t. its variable $\vec{\nu}$, it is in fact linear in many of its parameters, namely: mass, inertia, added mass, damping and lift parameters and gravity/buoyancy represented by $d_{BW}, \vec{\gamma}$. As such, the measured or estimated body-fixed velocity $\vec{\nu}$ and body-fixed accelerations $\dot{\vec{\nu}}$ can be

inserted into equations 2.41 for fixed points in time. The resulting equation can be reorganized as

$$\mathcal{L}(\vec{\nu}, \dot{\vec{\nu}}) \cdot \vec{\beta} = \vec{\tau}^* \tag{4.3}$$

From this, a least-squares or weighted least-squares scheme can be applied, if the equation is applied to a multiple of measurement points. For more details on this scheme or the shape of $\mathcal{L}$ etc., see the Matlab code listed in the Appendix, section C.2.2.

This is the basic scheme which is applied in [MW14], [MW16] and with some adjustments in [HHJ07]. Among a few other changes, the latter also includes upper and lower limit constraints on the parameters for the least-squares estimation. Depending on which parameters are known in advance, $\vec{\beta}$ can be smaller or bigger. All parameters which are known/defined in advance can be moved to the right-hand side of equation 4.3, the $\vec{\tau}^*$-part. Otherwise, $\vec{\tau}^*$ only contains the (assumed known) thruster forces and moments. The rudder forces and moments, however can also be moved to the left hand side of the equation such that the rudder coefficients are contained in $\vec{\beta}$.

In either case, before trying to solve equation 4.3, the stacked system matrix' $\mathcal{L}(\vec{\nu}, \dot{\vec{\nu}})$ rank should be checked in order to make sure the least squares solution yields enough information.

It must be noted that even in the presence of currents, it is not possible to estimate *both* the parameters of $M_{RB}$ as well as $M_A$ at the same time. As such, one of them must be fixed in advance.

**Remark 12** (Success of classical identification)**.** *The success of this approach depends heavily on the following things:*

- *Correctness of the assumed thruster model/dynamics,*
- *Quality of the predetermined parameters,*
- *Noise levels within the measured/estimated velocity and acceleration data.*

Additionally, the approach presented in [MW14] is meant for a slow-moving fully actuated ROV (remotely operated vehicle) rather than a under-actuated AUV with faster movement. Naturally, the same approach is not applicable without adjustments and assumptions.

**Remark 13** (Ideal identification)**.** *Theoretically, when the measured velocities are noise-free and all model assumptions hold, i.e. the thruster model, position of center of gravity, mass/inertia are assumed correctly, then all hydrodynamic parameters can be easily deducted using a trajectory as simple as:*

- *Constant RPM, i.e.* 220*,*
- *Constant non-zero rudder angles (i.e.* $5 - 10°$*),*
- *E.g.* $\geq 100$ *sample points*

*Of course, more changes in all actuation set points will generally yield a more stable result. Additionally, practically speaking it is naturally more suitable to use two data sets: one for identification and the other one for validation [HH11].*

One of the sample tracks which has been used to try the classical estimation scheme is given in figure 4.10. The same data set was used to alternatively determine parameters

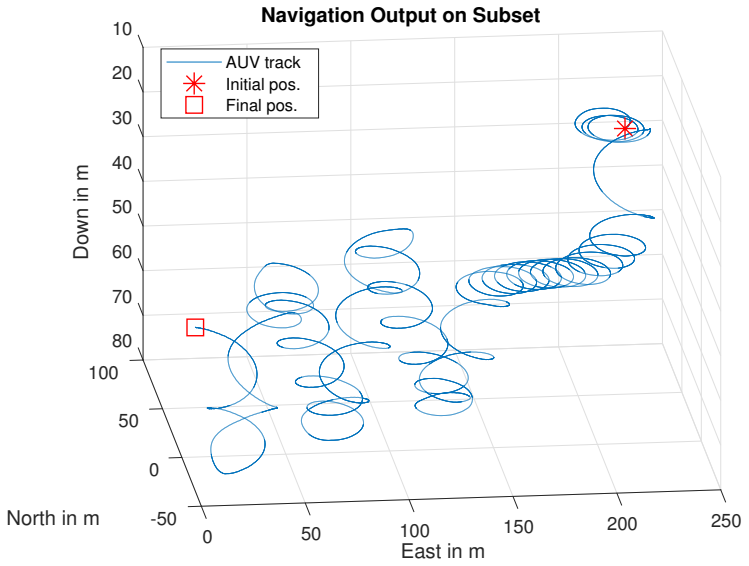


Figure 4.10.: NED positions of reference data set for reference model identification. This is part of the spiralling-movement seen in figure 4.5. The AUV had its yaw rudder fixed to ca. $-5^\circ$ (left turn) while the pitch rudder changed between $\pm 5^\circ$ and $\pm 10^\circ$. The drift towards South-West is due to water currents. Track duration is ca. $35min$.

for a reference model, see section 4.5.5. The corresponding velocities of the given track can be seen in the previous sections, figure 4.7.

Unfortunately, the classic identification failed on all available data sets which must be due to the reasons mentioned in remark 12. The resulting parameters were physically infeasible and were as such rejected. Of course, it is possible to use introduce upper and lower limits as in [HHJ07]. However, even the results from this approach were highly unsatisfactory.

Despite remaining curiosity on this subject, the experimental identification was not investigated further due to the following reasons: In a commercial or at least more advanced level of AUV usage than it was possible within the SMIS project, the unknown AUV dynamic details which are most likely to cause the failure (e.g. more knowledge of the thruster dynamics and added mass/inertia terms etc.) are not really an issue

any more. In fact, this problem can be considered solved in practical terms, see again [HHJ07] for example. As such, investing more time into this would not yield a great deal of scientific insight and thus this topic was dropped.

Instead, to generate a reference model, a different approach was chosen which is presented in section 4.5.5.

### 4.5.5. Reference Model Parameters

Based on the field data, parameters for a reference model were deducted.
For this, the Simulink model was fed with the same inputs as the AUV: thruster's RPM and rudder angles on a designated part of the trajectory presented in section 4.4. The reference trajectory can be seen in figure 4.10.

The Simulink parameters were successively adapted such that linear and angular velocity as well as the AUV's orientation resembled the navigation output data - while still maintaining physical consistency, e.g. due to symmetry, the hydrodynamic parameters of the $x$,$y$,$r$-sub-model for yaw-motion should not be too different from the $x$,$z$,$q$-sub-model for pitch-motion. Also, the model should remain fairly general to possibly avoid over-fitting. For this, the vehicle was assumed to have a fixed mass of $1000kg$ and a fixed inertia pre-estimated from this mass being rather homogeneously distributed over the entire cubic shape. Additionally, the added mass/inertia was initially estimated through formulas based on the vehicle hull shape as are given in [Bre82]. The respective Matlab codes for the latter step can be found in the Appendix, section C.2.1, for completeness. Additionally, the center of gravity is assumed to coincide with the center of body origin, i.e. $r_g^b = (0, 0, 0)^T$.
All four latter assumptions are simplifications but they suffice to deduct a representative reference model. Unlike the true mass/inertia, the added mass parameters were refined as well during the identification process.
In a commercial non-prototypical setting, mass, inertia and gravity/buoyancy should be known quite accurately from CAD data and respective dry-tests.

Figure 4.11 illustrates the comparison between model and experimental data w.r.t. angular velocities. In figure 4.12, the model match w.r.t linear velocities is illustrated. The orientation match is given in figure 4.13.

As the model position deviation is accumulated over time via integrating acceleration and velocity errors, there is a significant drift w.r.t. the position output of the model. This can be observed in figure 4.14 which is not surprising for the long track time of $35min$.

The full parameters can be found in the Appendix, section C.2.

#### Heading and Depth Controller

To make the AUV dynamic model follow straight lines and desired depths, simple controllers were added to the model. The basic structure can be seen in figure 4.15.

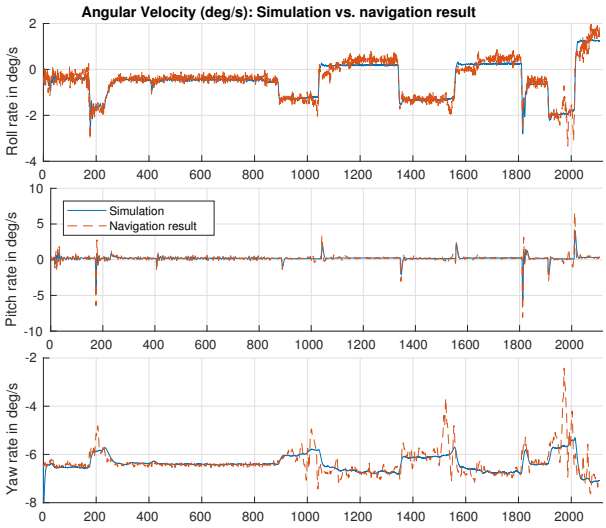


Figure 4.11.: Angular Velocity of the Reference model vs. the navigation result. The miss-match in terms of RMSE is $0.2326^{\circ}s^{-1}, 0.2770^{\circ}s^{-1}$ and $0.3688^{\circ}s^{-1}$ for the roll, pitch and yaw rate, respectively.

For heading control, a $PID$ [Sko03] controller was used and depth control is achieved by a $PI$ controller, here. Please note that for a real AUV it may be more suitable to use a cascaded depth controller for both depth and Pitch, as was done in e.g. [RKD+15]. The real SMIS AUV also used such a cascaded structure but the version used here is by far enough to suit the simulation requirements. In both cases, the controller element output was limited to $\pm 25^{\circ}$ through the Matlab built-in saturation option.

The Matlab auto-tune function was used to generate suitable controller gains. The resulting parameters, including the Matlab typical $N$-value for making the $D$-element feasible were as follows:

$$\begin{aligned} &\text{Heading} \quad \text{controller parameters :} \\ &\qquad P = 2.0403,\ I = 0.0120959,\ D = 1.55826,\ N = 0.680647 \\ &\text{Depth} \quad \text{controller parameters :} \\ &\qquad P = 0.04874,\ I = 0.0008030845. \end{aligned}$$

These controller gains are targeted at a fixed surge speed of ca. $1.43\frac{m}{s}$. It must be noted that designing general accurate controllers for underactuated AUVs is by far not a simple task - especially for the real-world AUV [Ver09] [SSF04]. However, in this simplified simulation it is suitable to use the Matlab built-in functions. Moreover, it is also common for real-world AUV controllers to design them for a constant surge speed [Fos11].

Additionally, both controllers use one of Matlab's default *anti-windup-method* [SP12]

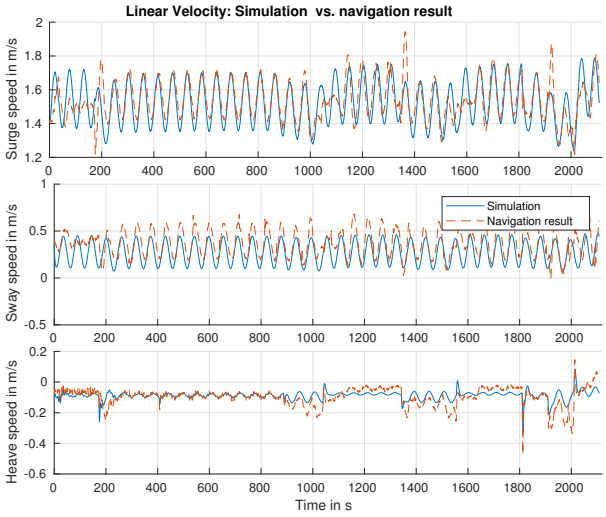


Figure 4.12.: Linear velocities of the reference model vs. the navigation result. The significant periodical change in in x- and y-velocity is due to the water current influencing the AUV's dynamic behaviour. As such, the linear velocity match strongly depends on the proper yaw-turn-rates. Nevertheless, the match w.r.t. RMSE is $0.0882\frac{m}{s}, 0.1349\frac{m}{s}$ and $0.0472\frac{m}{s}$ for surge, sway and heave, respectively.

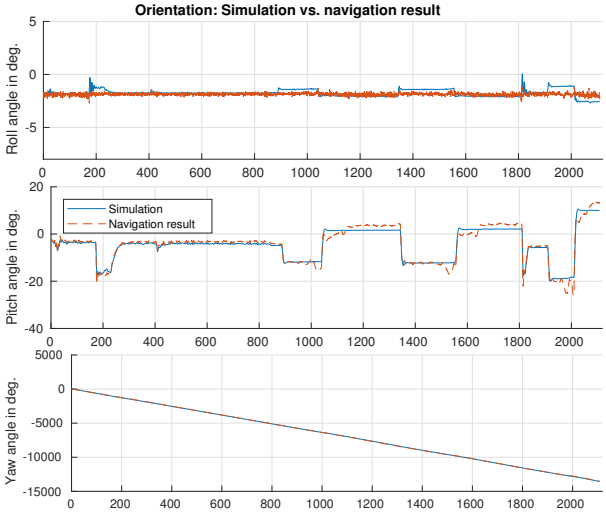


Figure 4.13.: Orientation of the simulated model vs. the navigation result. The match w.r.t. the RMSE is $0.3465°$, $1.6515°$ and $21.9996°$ for Roll, Pitch and Yaw, respectively. The latter also includes a drift over time which is not visible in the plot.

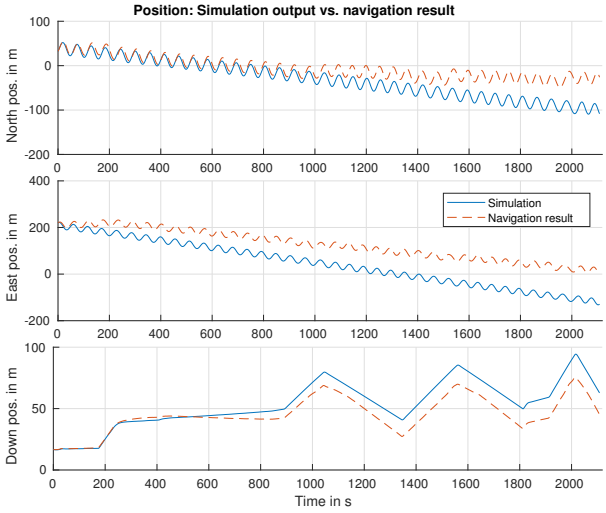


Figure 4.14.: Resulting drift of the simulated NED position during the $35min$ of simulated time. In the end, the simulated position deviates from the navigation result by $82.3m$ $130.6m$ and $17.6m$ in North, East and Down coordinates, respectively.

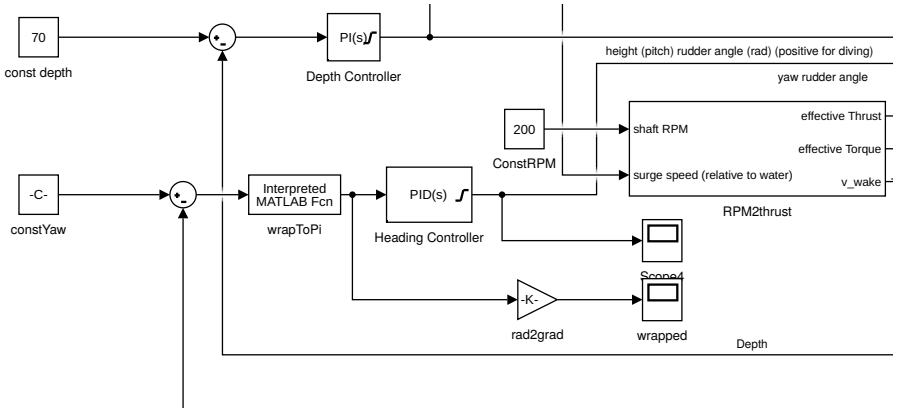


Figure 4.15.: Simple heading and depth controller structure used for the model in Simulink. The lower left arrow pointing up towards the leftmost sum-symbol carries the actual model's value of Yaw. See section 3.9 for more details on the remaining model in Simulink.

"Clamping" which is very important especially for the depth-controller in order to prevent the $I$-part of the controller from accumulating too much control difference over time.
For the heading controller to work properly, the Matlab $wrapToPi$-function was used on the control difference. Otherwise, the AUV might perform unexpectedly large turns to change direction.
The controllers were used to generate the reference trajectory in section 4.7.2.

## 4.6. Accuracy of SBS Calibration by Ship/USV

In order to receive an idea of how well the RV/USV can calibrate the SBS position on the ground in deep sea, a simulation was carried out.

In this simple scenario, using a fixed position for the SBS and an almost deterministic path of the tracking RV/USV, a complete PRCB computation as described in section 2.5 and 3.11 is somewhat overweighted. Thus, the simulations carried out for this section were replaced by a simple EKF localization simulation.
This section is also intended as a pre-test for the scenario of a moving AUV being tracked by the RV/USV.

The calibration of the SBS position from the RV (or USV) after mooring it to the ground was simulated using the USBL model constants from section 3.7.4.
Additionally, the RV's position was assumed to be simply known by GPS with a standard deviation of $\sigma_{GPS} = 4m$, see section 3.6. The RV's orientation was assumed to be directly measurable with AWGN having standard deviation of $\sigma_{RPY} = 1^\circ$. This is

a simplification but it should be technically feasible using sufficiently accurate sensors, e.g. Gyrocompass, DGPS, Fiber-optic Gyro (FoG) etc. since unlike for AUVs, weight and size of these devices is not an issue for an RVs.
The RV prescribed a (noisy) circle around the SBS mooring position

$$r_{SBS} = (10, 5, 4992)^T$$

with radius of three cable lengths ($= 555.6m$) and surge speed of five knots ($2.5\frac{m}{s}$) for 20 minutes, finishing almost the full ($85\%$) circle. The set-up can be seen on the left of figure 4.16. The simulated RV periodically located the SBS through acoustic pings for

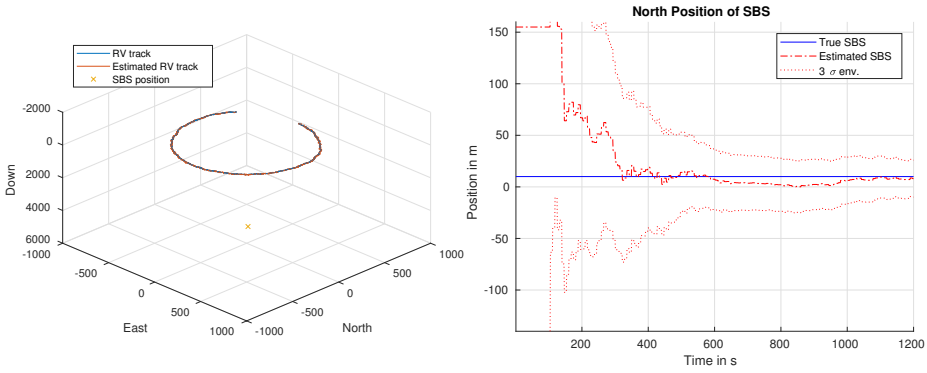


Figure 4.16.: Set-up of a simulated RV tracking the SBS in deep-sea ($5000m$) depth from a calibration circle of radius $\sim 555m$ (left image). Illustration of convergence rate for the RV's EKF scheme w.r.t. the SBS's North position (right image).

all ca. $7s$ keeping in mind the signal propagation round-trip time. Noise was added to the USBL measurements according to section 3.7.4. This set-up similar to the one which was used in [ONW+16].

There is a trade-off in the RV's circle radius. Bigger radii theoretically provide higher lateral resolution but take longer to perform ship-wise, are more prone to refractions by sound velocity variations and may exceed favourable slant ranges for communication and localization, see [ONW+16] and [NOW+15].

For the estimation, an EKF scheme was used which wrongly assumed the SBS to be initially located at $\tilde{r}_{SBS} = (155; -180; 5050)^T$ with an initial covariance matrix of $\Sigma = \text{diag}\left(100^2, 100^2, 20^2\right)$.
The simulation of the RV's movement contains a deterministic part, i.e. forwarding in the circle, and an additive Gaussian noise with standard deviation of $0.5 \cdot u \cdot \Delta t$ in the RV's xy-plane and $u \cdot \Delta t$ in its z-plane for a simplistic wave simulation. Similar noise was added to the RV's orientation. Roll and Pitch are drawn independently at random from $N_{angle} \sim \mathcal{N}(0, (5^\circ)^2)$. The RV's heading is aligned with the circle path with the same (independent) $N_{angle}$ added. The deterministic part is not included in the prediction made by the EKF. Instead, for the prediction step it is assumed that the RV is doing a random walk motion with AWGN steps having covariance $100 \cdot Q$ where $Q$ is the

true system noise. Thus, the positioning and orientation (un-)certainty of the RV comes from the GPS and orientation measurements. Consequently, both uncertainties remain close to the initially assumed GPS (and orientation) standard deviation of $\sigma_{GPS} = 4m$ ($\sigma_{RPY} = 1°$) in each dimension during all simulation time.
This discrepancy of true system transition and assumed system transition was chosen because during a real experiment it might not be practical to generate an accurate model of the RV's circling movement. Assuming a random walk instead with step sizes within a multiple of the surge speed (multiplied by step time) would be much easier to handle. Of course, better transition models can be assumed as well. In that case the estimation procedure can be accelerated - provided the motion assumptions hold.

In this simulation, the estimated SBS position using an EKF scheme converged rather quickly. The convergence behaviour can be seen on the right of figure 4.16 and on both sides of figure 4.17. In the end of this simulation, the EKF assumed a lateral position

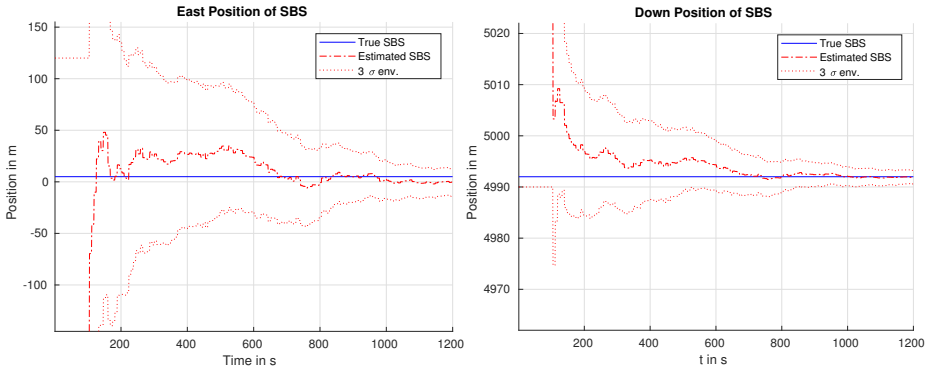


Figure 4.17.: Continuation of figure 4.16. Illustration of convergence rate for the RV's EKF scheme w.r.t. the SBS's East and Down position.

accuracy of approximately $\pm 17.9m$ w.r.t. the $3\sigma$-environment. In contrast, depth can be resolved faster end better under these ideal circumstances due to the USBL's pancake error shape, see section 3.7.4. The RV's modem is mostly facing down resulting in a fine resolution of the $z$-axis due to the measured slant ranges. The assumed final accuracy of the SBS's down position was at $\approx \pm 1.33m$ w.r.t. the $3\sigma$-environment.

Despite being sub-optimal due to non-linearities in the USBL measurement model, the EKF performed well in this example.
Simulation time was negligible compared with real-time and took just a few seconds in all cases presented in this section.

In the following subsections, the results with respect to different USBL availability are presented: direction estimates only vs. slant range estimates only.

### 4.6.1. USBL Direction Only

For this simulation, the exact same parameters and noise samples from section 4.6 were used with the only difference: the simulated slant range measurements were suppressed and prevented from being "fed" to the EKF. The results are presented in figure 4.18. At

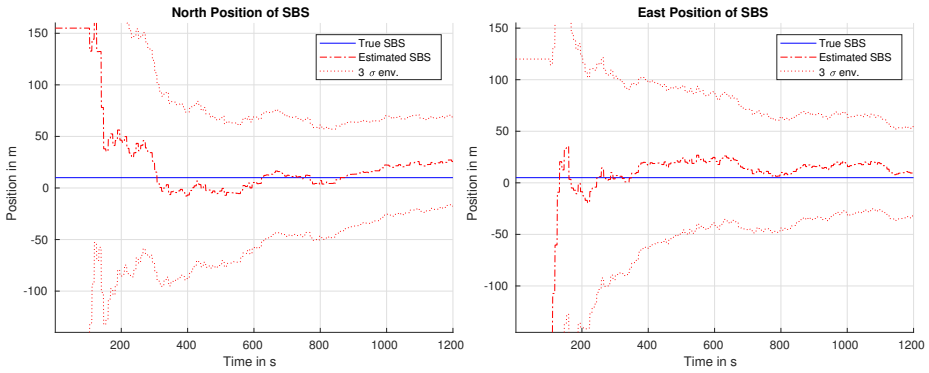


Figure 4.18.: Illustration of convergence rate for the RV's EKF scheme w.r.t. the SBS's North and East position *using only direction measurements*. Down position cannot be determined in this case.

the end of the simulation, the EKF assumed a lateral position accuracy of $\pm 43.2m$ w.r.t. the $3\sigma$-environment. The vertical (down-axis) accuracy, however, remained practically unchanged at the initial covariance of $(20m)^2$. This is not surprising as this configuration is comparable to determining the intersection of multiple straight lines which are almost parallel to each other. Numerically, this mapping has a bad condition number [Deu08] and is thus extremely prone to noise.
However, as this localization scheme does not require the participants to wait for the complete round-trip time, the messaging frequency from the SBS could be increased drastically: e.g. one message for each second. In that case, the North and East position of the SBS is determined very quickly but the Down position remains as unclear as ever. The results are shown in figure 4.19 Lateral certainty finishes at $\approx \pm 16.5m$ in the simulation.
A more detailed analysis of navigation accuracy using mostly direction measurements from non-acknowledged one-way messages will be part of the work in progress[3] [Neu18], see section 2.8.1, and will not be considered in more detail here.

### 4.6.2. USBL Slant Range Only

Using the same parameters and noise samples from section 4.6 but suppressing direction measurements from being fed to the EKF yielded the results presented in figure 4.20. The EKF scheme resulted in assuming approximately $20.81m$ lateral and $1.38m$ vertical

[3] work in progress as of autumn 2017

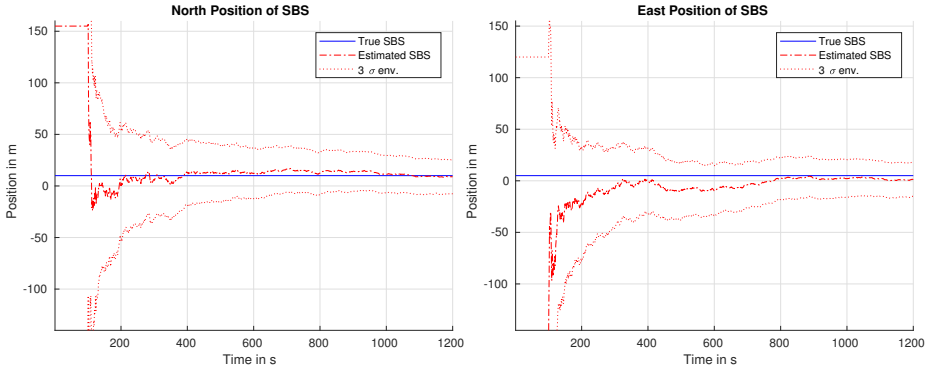


Figure 4.19.: Illustration of convergence rate for the RV's EKF scheme w.r.t. the SBS's North and East position using *only direction measurements at increased rate: one for each second*. Down position can still not be determined in this case.

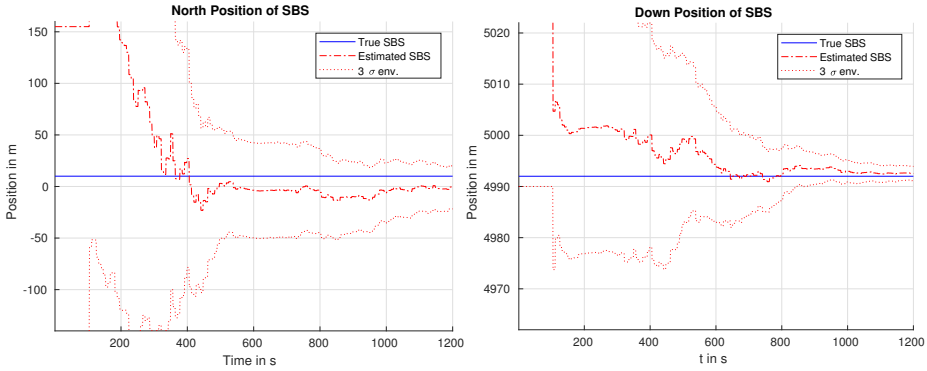


Figure 4.20.: Illustration of convergence rate for the RV's EKF scheme w.r.t. the SBS's North and Down position *using only slant range measurements*.

positioning accuracy w.r.t the $3\sigma$-environments.
It can thus be easily concluded that although the direction measurement is supporting the lateral localization, the majority of the SBS's position information is deducted from the *slant range measurements rather than the direction measurements* in this RV-SBS calibration set-up. This is clear due to the explanations given in section 4.6.1.

## 4.7. Approximate PCRBs for Deep-Sea Navigation

In this section, the main results from simulation regarding the navigation accuracy are presented.

At first, in section 4.7.1 the number of necessary Monte Carlo realizations is considered, before performing the main simulations.
In section 4.7.3, the results for the theoretical navigation accuracy of a reference deep-sea AUV mission is presented, consisting of a spiralling movement to dive to the sea-bottom followed by a typical lawn mower pattern to scan it. Afterwards, in section 4.7.3 the same mission is simulated using MA navigation for the lawn mower part upon having a dynamic model available after reaching the see bottom for an identification routine requiring DVL reception.
The results from Non-MA and MA navigation are compared in section 4.7.4.
Finally, some secondary results are presented in section 4.7.5.

### 4.7.1. Pre-Simulation: Number of MC realizations needed

In order to estimate how many realizations need to be contained in the Monte Carlo simulation, a pre-test was performed.

Using a short spiralling movement of $10$ seconds, the number of realizations $P$ was tested for $P \in \{10^6, 10^5, 10^4, 7000, 3500, 2250, 10^3, 500, 100\}$, using an initial state distribution of $10^4 \cdot \Sigma_{init}$.
It was assumed that using a number of $P = 10^6$ realizations should yield sufficiently accurate results. The accuracy loss when using less particles is illustrated in figure 4.21.

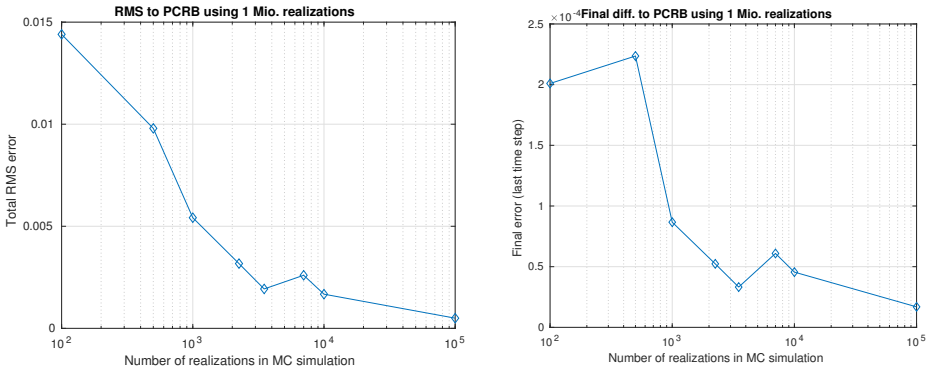


Figure 4.21.: Semi-logarithmic illustration of relative accuracy using increasing numbers of realizations within the MC simulation on the reference data set ($10s$ simulated time). Left: Total RMS compared with $P = 10^6$ case. Right: Final error compared with $P = 10^6$ case.

The approximate time needed to compute the $10s$ of simulated time (on SMISERS, see section 4.1) were as follows: $\{1h, 7min, 50s, 37s, 23s, 18.8s, 14s, 11s, 8s\}$ w.r.t. a decreasing number of realizations $P$.
Figure 4.21 indicates that using $P = 3500$ is a reasonable trade-off between accuracy and reduced computing time. As such, in the following, $P = 3500$ was used if not mentioned otherwise. Nevertheless, using only $P = 1000$ should also yield reasonable results close to the assumed true PCRB. Note that in [MOS13], only $P = 800$ was used.

### 4.7.2. Non-MA Navigation Accuracy

A reference trajectory was generated via Simulink using the reference model parameters from section 4.5.5. This reference trajectory consists of

- AUV starts at position $(0, 0, 20)^T$ with initial speed $\vec{v}_b = (1.5, 0, 0)^T$ and all its orientation angles set to 0,
- The AUV uses a downwards spiral to get close to the bottom (idealized as a plane of depth $5000m$ (ca. $3h$ simulated time),
- Upon getting close enough to the sea-bottom, the AUV performs a calibration motion (up/down-spirals) to generate data for model identification (ca. $15min$),
- Afterwards, the AUV is performing a Lawnmower-motion over the sea-bed to simulate a mapping mission (ca. $5.2h$ simulated time).

The motion to finally float to the surface is omitted in this simulation.

At the beginning of the first maneuver (diving down), we assume the initial AUV navigation state is known almost perfectly, i.e.

$$\Sigma_{init} = \begin{pmatrix} \sigma^2_{0,NED} \cdot I_{3\times3} & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{0,\vec{\nu}} \cdot I_{3\times3} & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{0,\vec{q}} \cdot I_{4\times4} & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{0,\vec{b}_a} \cdot I_{3\times3} & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{0,\vec{b}_\omega} \cdot I_{3\times3} \end{pmatrix}$$

with abbreviating $0$ in place of zero-matrices of appropriate size, using idealized constants:

$$\sigma_{0,NED} = 0.01;\ \sigma_{0,\vec{\nu}} = 0.001;\ \sigma_{0,\vec{q}} = 0.0013 \cdot \frac{\pi}{180}$$
$$\sigma_{0,\vec{b}_a} = 0.0018;\ \sigma_{0,\vec{b}_\omega} = 9.15 \cdot 10^{-4} \cdot \frac{\pi}{180}.$$

The initial MC realizations are distributed accordingly, assuming a normal distribution around $\Sigma_{init}$.
After the first maneuver, each follow-up maneuver starts with using the final states/realizations from the previous simulated maneuver.

### Diving Down via Spiral

To generate the spiralling movement, the Pitch rudder was set to constant $25°$, yaw rudder to $-5.2°$. This results in a AUV's Pitch degree (diving angle) of $\theta = -30.97°$ and a constant surge speed of ca. $u = 1.14\frac{m}{s}$.
The reference trajectory used for diving down in figure 4.22 is very similar to the one used in [MOS13].

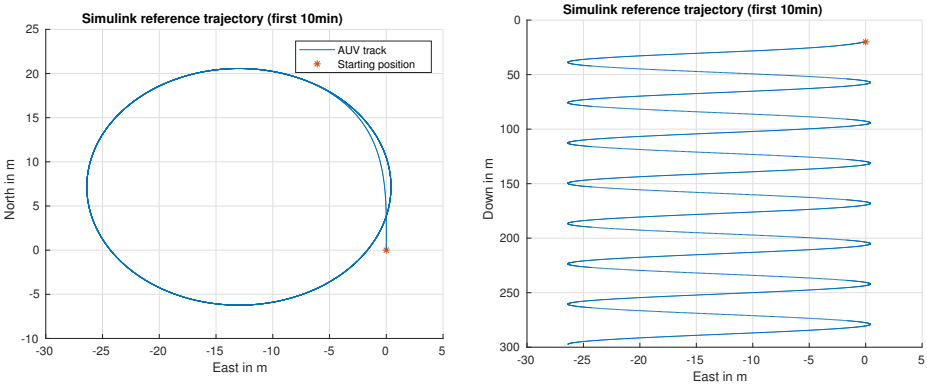


Figure 4.22.: First section (10 of ca. 177min) of the diving motion using a spiral. Pitch rudder is set constantly to $25°$, yaw rudder to $-5.2°$. The full trajectory finishes at a depth of $4917m$. Left: Top view. Right: Side View.

During the maneuver, the RV/USV served as a reference point clockwise circling the AUV's sprial's center at a cable length (183) with surge speed of $2.5\frac{m}{s}$. This movement was chosen in order to avoid big discrepancies in North vs. East tracking accuracy. This can happen if, e.g., the RV remains in fixed position at about $20m$ east of the spiral's center. In that case, the North accuracy would be better than the East accuracy which can be explained easily through the results from section 4.6.

The net diving speed of ca. $0.45\frac{m}{s}$ is bigger than the experimental one of the SMIS AUV. During a sea-test in the Atlantic Ocean in May 2015 (the same cruise that yielded the data for [ONW+16]), the AUV had a net diving speed of ca. $0.21\frac{m}{s}$.
However, in that case, the Pitch rudder was not used at quasi-full deflection and, to avoid drowning, the AUV's buoyancy was significantly bigger than the ones used in the experiments evaluated in section 4.5.5 and others. As such, a net diving speed of $0.45\frac{m}{s}$ would not be beyond reason for the original SMIS AUV.

Using SMISERS, see section 4.1, with $P = 3500$ took about $5 : 12h$ of computation time. The resulting PCRBs w.r.t. NED-coordinates are shown in figure 4.23.

Note that in figure 4.23, the PCRB for the Down-coordinate is not visible as it is too small $0.03m$. This is due to the depth-measurements taken from the pressure sensor.

Finally, the approximate Bias PCRB can be found in figure 4.25.

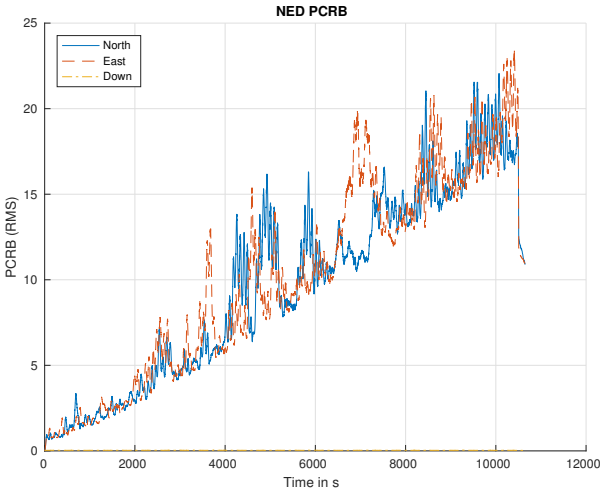


Figure 4.23.: PCRB result for the diving motion in the full $176min$ of simulated experiment w.r.t. NED coordinates (in $m$). The periodic "anti-cyclic" ripples are due to the position of the trajectory relative to the reference point (RV/USV) cycling clockwise ($R = 183m$) around the counter-clockwise AUV spiral, see figure 4.22. The final drop corresponds with the availability of DVL-measurements close to the sea-floor ($5000m$ depth).

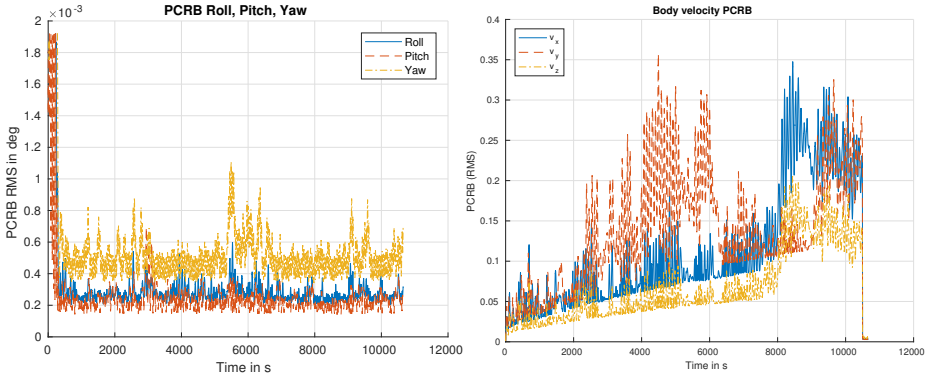


Figure 4.24.: Approximate PCRB for Roll/Pitch/Yaw (left, in $deg$) and Body velocities (right, in $\frac{m}{s}$) w.r.t. the Diving motion. Orientation accuracy in terms of degree is the result of somewhat idealistic sensor constants.

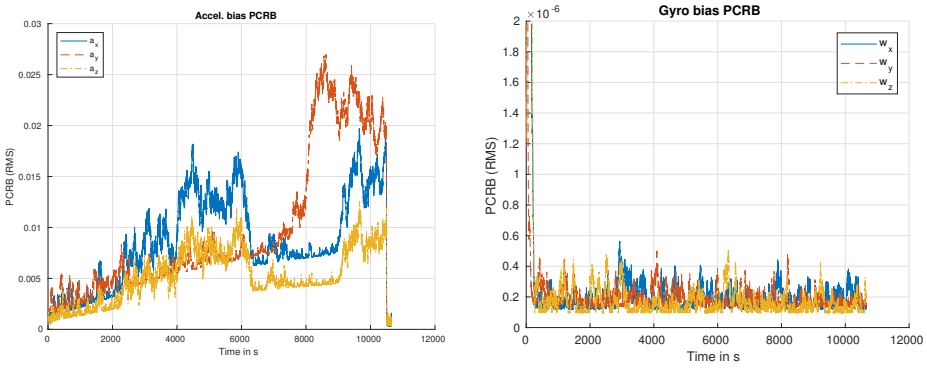


Figure 4.25.: PCRB results w.r.t. biases for Diving motion from section 4.7.2. Left: Acceleration bias ($m\,s^{-2}$). Right: Gyro Bias ($rad\,s^{-1}$). Again, the final drop (w.r.t. acceleration bias) is due to the availability of DVL measurements. Note that estimating the Acceleration bias without DVL measurements is tough and inaccurate yielding an accumulated inaccuracy. The Gyro biases estimation is, however, rather stable through use of magnetometers and Roll/Pitch stabilization from gravity.

A sample bias realization can be found in the Appendix, section D.1. Moreover, in that section, the same simulations found here were repeated without Compass and Roll/Pitch stabilization in order to underline the importance of orientation aiding techniques.

#### Identification Track

In order to identify the dynamic model parameters (online), the same actuation data was used as the one in section 4.5.4 and 4.5.5 as the movements given there are already enough to identify the model even through noisy data - providing the model assumptions are correct. Figure 4.26 shows the resulting track including the original water current influence.

For each Monte-Carlo realization, an individual model was identified and used for estimation afterwards.

#### Lawn Mower Mission over the Sea-floor

To simulate the main maneuver, e.g. a regular mapping scenario of the sea-floor, a lawn mower motion was simulated using the controllers from section 4.5.5. Assuming a sensor footprint of ca. $100m$ width, the AUV performed regular turns to follow lines spaced $100m$ apart from each other. The result can be seen in figure 4.27.
The resulting surveyed area has a size of ca. $1500m \times 1600m = 2.4\,km^2$ and the AUV

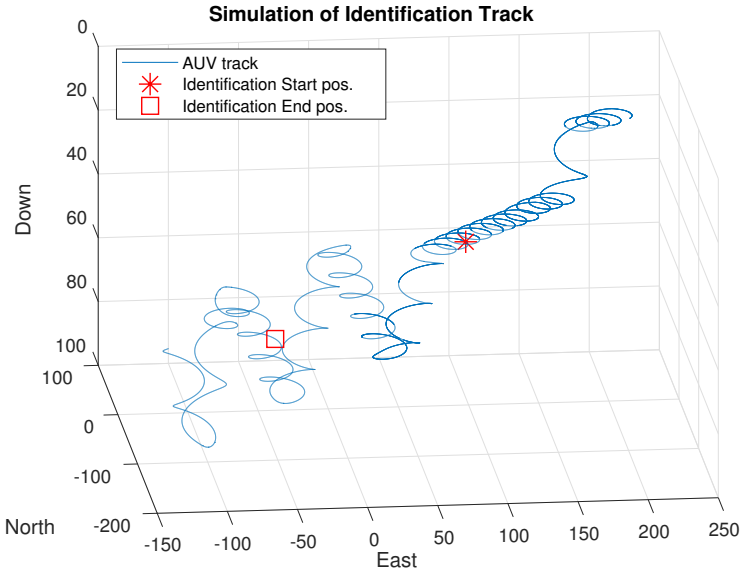


Figure 4.26.: Full simulated track re-using the actuation data from the experiment shown in figure 4.10. The sub-section which is used for identification is marked from star to square and reflects ca. $15min$ of real time (Depth is simplified to a scale $\leq 100m$). The same current influence as in figure 4.10 is used for the simulation to underline the track characteristics although the current is deactivated for the PCRB simulation.

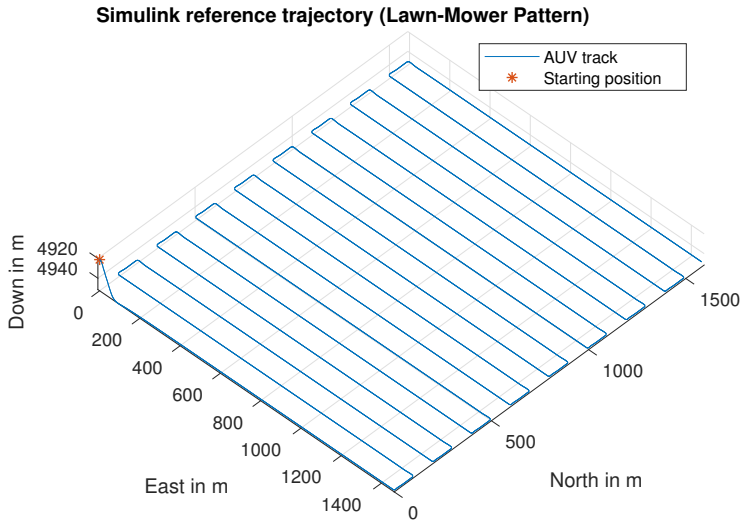


Figure 4.27.: Reference trajectory for a typical lawn mower motion pattern to map the sea-floor. Simulated time is $5.17h$ at a cruise speed of ca. $1.43\frac{m}{s}$. Simulation starts at a depth of $4920m$ reaching and keeping an almost constant depth of $4950m$ during the main maneuver.

follows a total track length of ca. $26.5\,km$ in that time.
The supporting USV prescribes a simple linear motion from $(0, 0)$ to $(1600, 1500)$, i.e. it moves along the diagonal of the surveyed square/rectangle. The speed is chosen such that it moves with constant speed arriving at the end point just in time when the AUV arrives there, too. More complex maneuvers are possible of course, but this one is useful to illustrate a few more details on the reference point positioning, as can be seen in the results.

The approximate PCRB result w.r.t. NED position can be seen in figure 4.28.
Through integrating the DVL measurements into the estimation process, the NED accuracy very slowly reaches a new asymptotic accuracy in the - impressive - order of theoretically only a few meters all the way at $5000m$ depth.

The approximate PCRB result w.r.t. body velocities and Roll/Pitch/Yaw is given in figure 4.29
The bias PCRB results are given in figure 4.30.
All in all, the navigation accuracy even in deep-sea is quite impressive - provided the idealistic assumptions hold, i.e. the sensor constants are accurate, and DVL measurements are available.

Computing the results for the Lawn Mower motion on SMISERS took about $8.5h$.

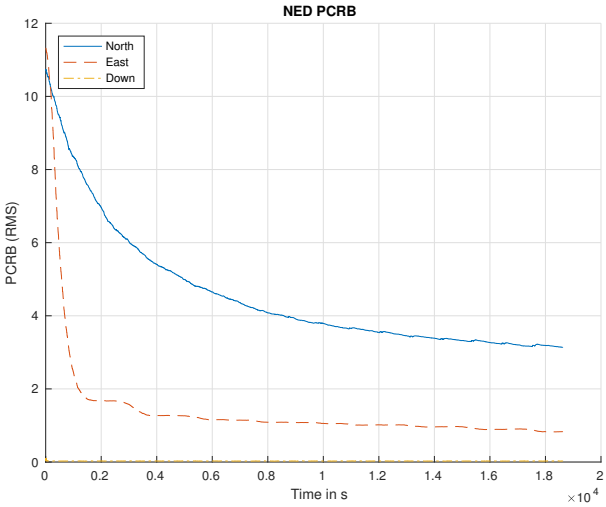


Figure 4.28.: PCRB result for Lawn Mower motion in the full $5.17h$ of simulated experiment w.r.t. NED coordinates (in $m$). The discrepancy between North and East position is due to the fact that the referencing RV/USV is almost always at the same North position of the AUV while "seeing" the AUV periodically either east or west of itself. A better North accuracy could be achieved by different RV reference positions.

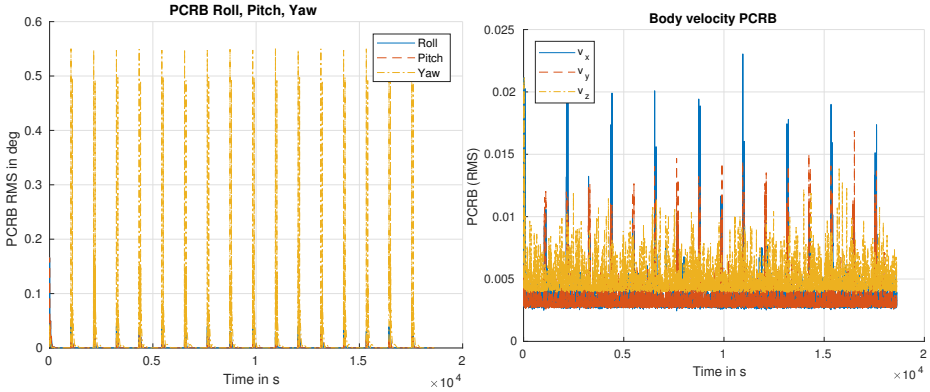


Figure 4.29.: Approximate PCRB for Roll/Pitch/Yaw (left, in $deg$) and Body velocities (right, in $\frac{m}{s}$) w.r.t. the lawn mower motion. All peaks reflect the AUV's turns in which e.g. the yaw estimation results are impeded. Some of the initial values for the body velocity PCRB were removed to zoom on the typical values during the main mission.

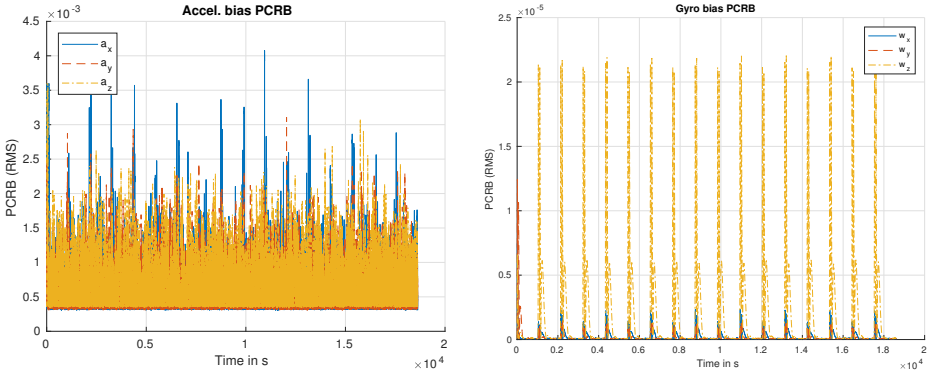


Figure 4.30.: PCRB results w.r.t. biases for the Lawn Mower motion. Left: Acceleration bias ($m\,s^{-2}$). Right: Gyro Bias ($rad\,s^{-1}$). Compared with figure 4.25, the Acceleration bias does not accumulate over time but follows an asymptotic which is due to the availability of DVL. Just as for the orientation PCRB, the Yaw part of the Gyro bias is periodically peaking at the AUV's turns. Some of the initial values for the acceleration bias PCRB were removed to zoom on the typical values during the main mission.

### 4.7.3. MA Navigation Accuracy

The results of using the basic MA navigation for the lawn mower motion are presented here.
The computed approximate NED PCRBs are given in figure 4.31.

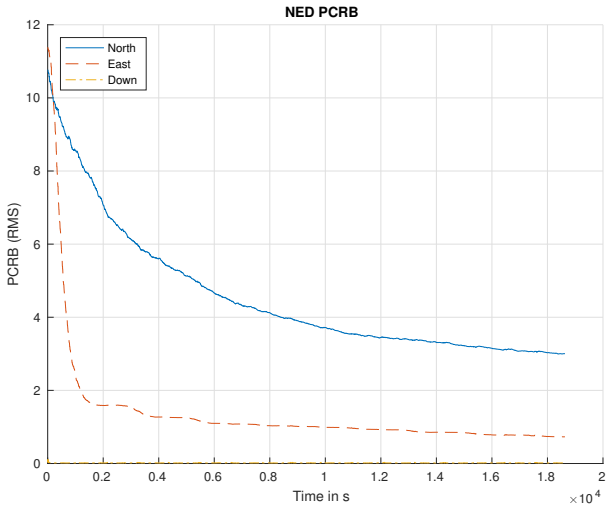


Figure 4.31.: Approximate PCRBs w.r.t. NED for the lawn mower mission using MA navigation. To the eye, no true difference is visible compared to the results from Non-MA scheme given in figure 4.28.

The results w.r.t. body velocity and acceleration bias are shown in figure 4.32.

The respective results for the gyro bias and the Roll, Pitch and Yaw angles are omitted here as they unsurprisingly look like the ones for the Non-MA scheme since they are practically unaffected by linear velocity measurements.

Through initiating the random number generator by the same seed as in section 4.7.2 the simulation including MA was given the same respective noise vectors as the ones used in the Non-MA simulation where applicable. This way, it was ensured the results become comparable.

Since the differences between the Non-MA and MA accuracies are barely visible through the pure plots, the results are compared directly in the following section.

### 4.7.4. Comparison of Non-MA vs. MA Navigation Accuracy

In this section, the results for the lawn mower reference track using Non-MA navigation in section 4.7.2 vs MA navigation in section 4.7.3 are compared to clarify both the benefit and limitations of MA navigation.

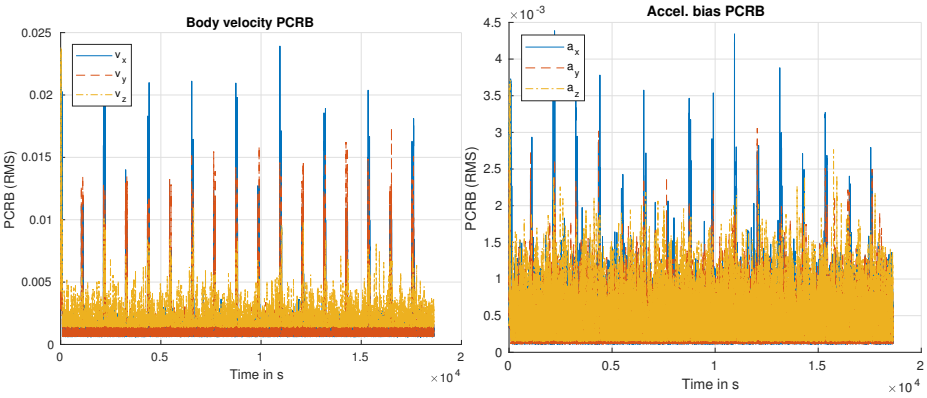


Figure 4.32.: Approximate PCRBs w.r.t. body velocity and acceleration bias for the lawn mower mission using MA navigation. Compared to the results for the Non-MA scheme, it can be seen that the values are shifted towards a slightly lower constant. On the other hand, they are also more noisy and peaking higher. Some initial values were skipped to highlight the typical main mission values only.

Since only the overall navigation accuracy improvement is of interest, only the North and East bounds were compared explicitly.

In an additional test, the MA-scheme was deactivated until the navigation system had settled to a better estimate on its own over time. For this, the MA-scheme remained deactivated over the first $5000$ seconds ($1.38h$). This scheme is called *MA-Settle-Time* or *MA-ST* for short in the following.
Figure 4.33 presents the approximate PCRBs for the three different schemes: Non-MA, MA and MA-ST.

The absolute and relative difference between the MA and the Non-MA scheme as well as the MA-ST and the Non-MA scheme w.r.t. the North estimate is shown in figure 4.34. The same absolute and relative difference w.r.t. the East estimate is given in figure 4.35.

As can be seen, the MA scheme can even decrease the estimation quality. The results indicate that it should only be used as an additional virtual sensor if the actual basic navigation estimate is already sufficiently accurate.
In the given scenario, the improvement in terms of absolute RMS is typically between $0.1m$ and $0.2m$ which reflects a relative improvement of ca $5\%$ (North) and $7 - 12\%$ (East) for the given lawn mower scenario.
Please note that for the time at which the MA scheme is activated, the North PCRB is still much bigger than the East PCRB. Possibly it is still too big to surely provide a benefit in that case. This may be the reason why the navigation accuracy is partly worse than the Non-MA scheme before reaching time $10000s$. This is the point at which the basic Non-MA North PCRB drops below $4m$ RMS in this specific scenario.

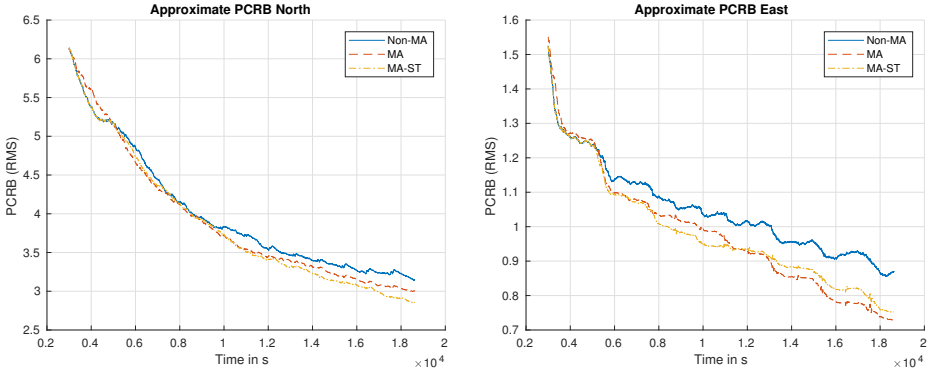


Figure 4.33.: Approximate PCRBs for North (left) and East (right) coordinates using the three considered MA-schemes. For clearer view of the region of interest, i.e. the later simulation time, the first $3000s$ of simulated time are not contained in the plots.

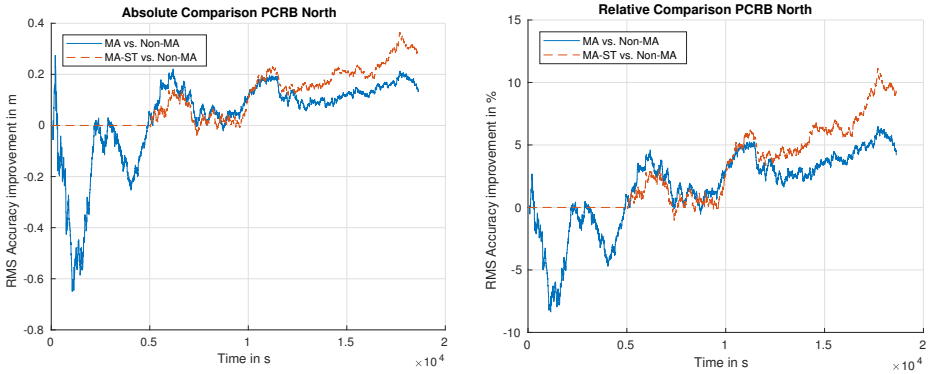


Figure 4.34.: Absolute (m) and relative (per cent) difference of the approximate North PCRBs, comparing MA and MA-ST to Non-MA on the lawn mower track. The decrease in accuracy during the first $5000s$ indicates that MA navigation can also decrease the accuracy in case the current Non-MA estimate is not sufficiently accurate.

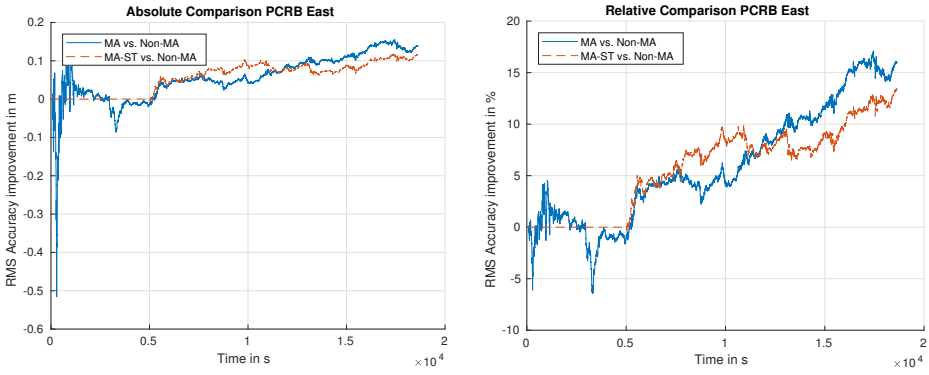


Figure 4.35.: Same results as shown in figure 4.34 - w.r.t. East coordinates.

### 4.7.5. Secondary Results

For completeness, it was tried to derive similar results for the sensors used in SMIS, represented by the sensor constants given in section 4.3.4.

It turned out that in simulation, it was not possible to maintain a stable estimate during all the time when using an EKF in the Monte Carlo loop. After ca. $7000\,s$ at a depth of $3100\,m$ the results began to diverge after the error bounds grew too big.
Thus, the results presented here are only given close to the point of divergence. Figure 4.36 yields the NED, figure 4.37 the body velocity and Roll, Pitch, Yaw and figure 4.38 the bias PCRBs, respectively.

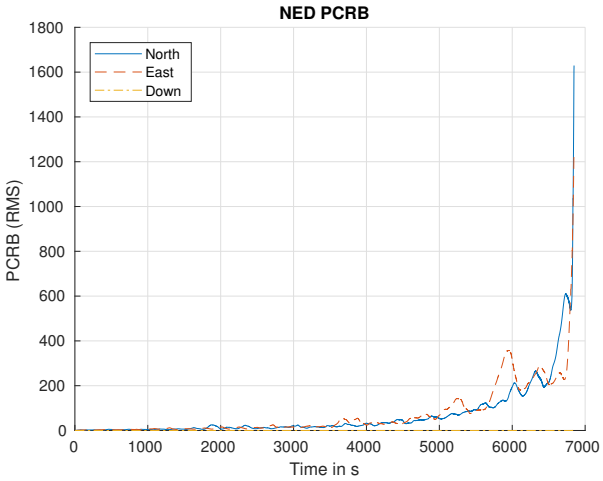


Figure 4.36.: Approximate PCRB results for NED using SMIS sensor constants on Diving-down motion. Plot only shows the first $1.94h$ of the mission after which the filter results diverged. This NED plot indicates the divergence clearly since the RMS accuracy reaches unacceptable values beyond $1000\,m$.

Since the results for Diving Down already reveal that the navigation solution will almost certainly diverge before the AUV will reach the Sea-Bottom, evaluating the navigation accuracy for the lawn mower motion was omitted.
This indicates that a Deep-Sea mission with the basic sensor setting of SMIS would not have been possible if only the AUV and the RV/USV took part in the mission. It could only work if the SBS was also available and the AUV managed to localize itself through the SBS.

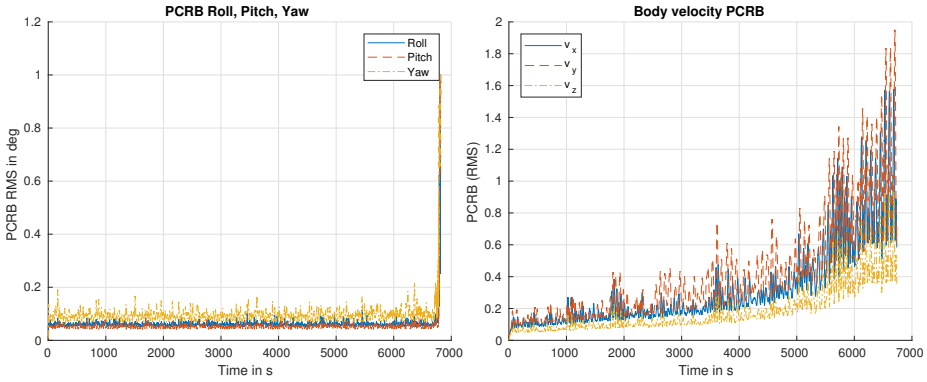


Figure 4.37.: Left: PCRB for orientation angles. Right: PCRB for velocities. Both are significantly worse than the values seen in section 4.7.2. Plot only shows the first $1.94h$ of the mission after which the filter results diverged.

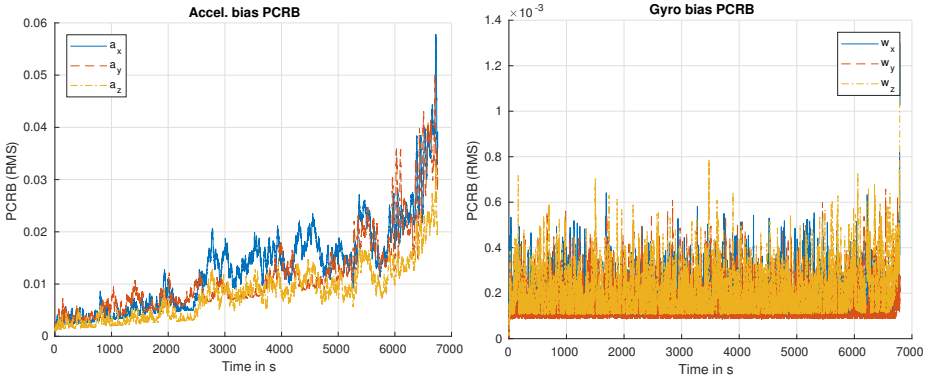


Figure 4.38.: Approximate bias PCRBs. Left: Acceleration bias PCRB. Right: Gyro Bias PCRB. Secondary results using SMIS sensor. Plot only shows the first $1.94h$ of the mission after which the filter results diverged. Before the bounds are diverging, the acceleration bias accuracy is similar to the one seen in section 4.7.2 since it is mostly estimated indirectly by the USBL having the same sensor constants in both sensor settings.

# 5. Conclusions, Discussion and Outlook

This chapter consists of Conclusions drawn from the work, in section 5.1, followed by discussing the results in section 5.2 and finally an Outlook and possible ideas for further research in section 5.3.

## 5.1. Conclusions

The conclusions that can be drawn from this work are as follows. Conclusions are sorted by their respective order of appearance in the thesis.

**Acoustic Positioning via USBL**
Despite some slightly inconsistent treatment in the literature, the best way to include the USBL measurements into an EKF based navigation solution is to separate the direction from the range measurement in order to be able to treat the noise affecting this measurement vector as additive, white and Gaussian as is required for the EKF.

The pre-test for RV to SBS calibration procedure was based on [ONW+16]. It lasted for ca. $20\,min$ and revealed an approximate final lateral accuracy of $6\,m$ w.r.t. RMS error in case the full USBL reception is given. Otherwise, if only the slant-ranges can be processed (due to bad reception quality on the noisy sea surface), the positioning RMS error is slightly bigger, in the order of $7\,m$ under the simulation assumptions. Using only direction measurements, this is increased to ca $14\,m$ for horizontal accuracy although the depth is close to unobservable in this case. Thus, it can be concluded that for this setting, the slant-range measurement is more important than the direction measurement.

**AUV Dynamic Model**
The structure of the AUV's dynamic model considered in this work (with cross-coupling in sway/yaw-rate and heave/pitch-rate) is in principle applicable to the real AUV motion. A simple maneuver such as spiralling up and down for a few minutes will already yield enough data to determine the model parameters with high accuracy. Of course, more variation in the motion will yield more robust results. This is also possible when sea current is present. In that case, the current must be estimated before applying the identification scheme.

**Deep-Sea Navigation**
The AHRS sensor used in SMIS is not accurate enough for full deep-sea navigation when the AUV is referencing itself only towards the USV. The navigation solution is very likely to diverge in that case. As such, a high-quality sensor ensemble is required.

Using high-quality sensors, deep-sea navigation under idealized conditions for an AUV referencing itself via USBL to a USV is possible in the tested scenario consisting of spiralling movement from surface to sea bottom followed by a typical lawn mower pattern. The expected accuracy in practice will be slightly worse than the ones generated here in simulation, i.e. above the observed order of $3 - 15\,m$ w.r.t. the expected RMS error.

**Model-Aided Navigation**
If the navigation system does not have a sufficient accuracy on its own, using the MA scheme can make the estimation result worse. Otherwise if the former is sufficient (ca. $< 4\,m$ in simulation), the MA scheme can further improve the navigation accuracy for another $3 - 12\%$, i.e. ca. $0.1 - 0.2m$ expressed in absolute values.

These results are supposed to serve as a guideline for the expected achievable accuracy in practice - despite the undeniable fact that practice is always more complicated and messy[1]. As such, the results in a practical deep-sea evaluation will most certainly be at least slightly worse.

## 5.2. Discussion

The discussion presented in this section provide a base for the outlook and future research given in the subsequent section.

**Acoustic Positioning via USBL and Deep-Sea Navigation**
Provided that the accuracy evaluated in this work is realistic for a real-world exploration mission in deep-sea the results are rather impressive. This is due to the error characteristics of a single USBL measurement: over a range of $5000\,m$ the absolute positioning error is in the order of $100\,m$ w.r.t. the RMS error.
Depending on the type of mapping sensor used, this accuracy should still be sufficient to generate a map of the sea-floor - at least in post-processing.
In practice, it should be favourable to use another point of reference for the AUV, e.g. another supporting AUV serving as an acoustic relay towards the surface while being present in the mid-water zone. Otherwise another reference station at the sea bottom can be used, e.g. the SBS. Of course, this largely increases the required autonomy and collaboration skills of the robotic team.
Unfortunately, it is not known if the given USBL setting would allow for a reliable communication and localization between AUV and SBS. This is due to the fact that the USBL device installed on the AUV is facing upwards. This could impede the signal spreading and reception when communication with another participant which is at bigger depth (i.e. underneath the AUV) than the AUV itself.
Alternatively, another USBL device facing downwards could be used to ensure AUV-to-SBS communication. However, in that case multipath and echoing effects become more relevant - similar to upwards communication close to the sea surface. Moreover, this type of localization is more prone to the topology of the sea bed since the terrain could impede the USBL by shadowing due to obstacles.
Moreover, this case is not to be considered a "pure" deep-sea mission any more because

[1] This was also confirmed for the SMIS AUV.

the deep-sea character (w.r.t. long-range) is dropped and from a navigation point of view the mission gets closer to a shallow sea application. Of course, the deep-sea scenario still contains multiple technical and control-related issues but this one very critical aspect of long-range is then alleviated.

A practical issue that was not addressed in detail in this work is is the significant signal runtime. This means, a more sophisticated slant-range measurement model is required, see section 3.7.5, and the resulting expected accuracy will be definitely worse then the ones seen in this work.

Another practical limitation of USBL for long-range applications must be kept in mind: Especially receiving messages at the noisy sea surface is problematic [NOW+15]. However, receiving range measurements is not an issue even in that case [ONW+16]. Also, the participant at the sea bottom receives messages very reliably which is the crucial part for the navigation.

**Dynamic Model and Model-Aided Navigation**
The derived results for MA navigation in this work are consistent with the results from [HH11]. In that work, the estimated improvement of the MA scheme during full sensor availability was in similar order of $0.1m$ for a real-world AUV application.
In [HH11] the focus of evaluating the MA navigation scheme was on providing robustness to the system in case of sensor failure. The navigation system is then stabilized when USBL and DVL are temporarily not available. It is assumed that this aspect could still be the greatest benefit of MA navigation.
This holds especially for mid-water localization where DVL is not available [MWP+16]. In that case however, it is important to estimate the ocean currents which may be difficult. In many deep-sea applications these currents can be neglected.

The extension to 6-DoF models as opposed to the work of Hegrenaes is useful if the AUV is not strictly working in a fixed plain parallel to the ground. This can even be the case if the AUV turns left or right - as this will often cause the AUV to also re-adjust its depth resulting in coupled motions which only 6-DoF models can cover. Of course, this type of model is harder to parametrize in practice.

Since the greatest benefit of MA navigation for the lawn mower motion appears to be the support during straight line motions, it is also possible to use a heavily simplified version of the AUV dynamic model providing virtual measurements: e.g. only the dependency of the AUV's surge speed on the propeller's RPM when rudder deflections are small.
Otherwise, as already mentioned in section 2.6, it must always be kept in mind that choosing a suitable model structure for the AUV motion always depends on the required task and type of AUV. There is no straight-forward method to do this in general although this work provides a guideline for typical slender body AUVs.
Moreover, a gap between the model and reality will always remain due to the many hydrodynamic simplifications made for the model. Ocean currents also remain a problem in practice due to the difficulty of estimation [FXCZ16].

The MA navigation scheme presented here can also be adjusted and more sophisticated methods can be used, e.g. the so-called H8-filtering in [NW13]. Although the applicability of these extensions in practice is not certain.

## 5.3. Outlook and Future Work

Based on the conclusions and discussion given in the previous sections, the outlook and possible future work in the area is presented here. Aaspects which were considered the most important and promising with relevance to this work's topics are presented.

**Small Additional Survey**
The expected navigation accuracy for the AUV localizing itself to the SBS could also be evaluated, i.e. for the lawn mower motion scenario - provided the localization is possible w.r.t. the USBL configuration.

**Model-Aided Navigation Benefit Threshold**
It would be interesting to determine the exact threshold at which the MA scheme becomes beneficial for the navigation solution - at least in theory. In the scenario considered here, it is expected to be in the order of $4\,m$ w.r.t. the RMS error of the horizontal position. A more accurate survey on this matter is, however, still required to be certain.

**Meta-Parameter Identification**
From a scientific point of view, it would be interesting to know if the meta-parameters in the dynamic model (i.e. mass, inertia and even thruster coefficients) could still be estimated from experimental data - given several simplifying assumptions of course. This might be possible if some correlation method would be considered: employ some general optimization routine such as Particle Swarm Optimization (PSO) on the meta-parameters which uses the identification of the classic parameters (hydrodynamic damping, lift etc.) to determine some model fit based on the remaining model-to-data-error. A procedure like this might be prone to over-fitting, but could still be interesting to investigate.
One of the benefits of such a procedure would be that the the navigation software itself could estimate all of the model parameters online without knowledge of the operator. The resulting model might be used for Model-Aided navigation or even for the purpose of parametrizing a motion controller based on that model.
However, besides some applicability in practice such an approach might not be necessary. This is because of the fact that AUV operation is very sophisticated and specific. Many tests and surveys must be conducted to "get to know" the type of AUV and even the individual AUV before it can even be considered for true every-day operation. Several parameters can also just be derived from CAD data and available commercial software to estimate the hydrodynamic added mass of the AUV. During these tests it is certain that the above mentioned meta-parameters will have to be determined anyway making the entire PSO approach less relevant.
Nevertheless, applying more sophisticated identification routines can definitely be considered valuable for the research field. One keyword in that direction is Multiple Model Adaptive Estimation. An example for space missions can be found in [MCL08].

**Machine Learning Approaches**
A very interesting and promising research branch which can be useful for both model identification and Model-Aided navigation might be found in Machine Learning methods [HTF11]. Here, the term Machine Learning is used although other names can be

inserted as well: Statistical Learning or Artificial Intelligence. What is meant in this context, is just the general pattern of "learning" (i.e. generalizing or even extrapolating) the output of a mathematical function based on possibly sparse pairs of known input-output samples. In this context, only the regression capabilities of Machine Learning are considered as opposed to the related classification capabilities.
In the recent work of the DFKI [WHK17] an estimation of hydrodynamic drag parameters was attempted using Machine Learning. Also, in [XZYC13] similar approaches were tested although in both cases only sub-models were identified and big efforts were required to generate the data. This is not surprising as Machine Learning requires a large number of training data.
Although these approaches are quite interesting, the real benefit of Machine Learning in this context might be to drop the model structure more ore less completely and only generate input-output mappings based on observed data. The idea behind this is that in typical AUV motion only very few motion types exist in the first place: mostly straight lines and turns. Focussing on observing these patterns online might yield some kind of motion "memory" in Machine Learning representation which is highly specialized on just these few motions. Once a specific certainty in this memory is achieved, it could be used for Model-Aided navigation. Of course, this is a more sophisticated task and it is almost certain that such a model will not be able to generalize to different motion types based on the training set. It will also not be directly suitable for automatic controller design based on this model. However, especially for these specific motions the Machine Learning approach might be useful to overcome the general gap between simulation/-modelling and the real motion behaviour as it might be able to also indirectly represent influences and effects that are not contained in the Fossen type dynamic models [Fos11]. Using this technique for Model-Aided navigation may be promising and also usable in practice. However, research in this direction would require another full PhD thesis.

**Others**
Finally, it must be said that there are also many completely different interesting and promising approaches to improve navigation accuracy, like SLAM and AUV cooperation, which were not examined in detail in this work.

# Appendix

# A. Additional Notation, Notes and Identities

## A.1. Miscellaneous Notation and Identities

For completeness, this section provides a quick (incomplete) review of selected Stochastic fundamentals introducing the notation used in the thesis.

### A.1.1. Measure Theory and Probability Spaces

A quick review of Probability Theory. A set of subsets $\mathcal{A} \subset \mathcal{P}(\Omega)$ with features

- $\emptyset \in \mathcal{A}$
- $A \in \mathcal{A} \Rightarrow A^c \in \mathcal{A}$ (complement)
- $A_n \in \mathcal{A} \Rightarrow \cup_{n=1}^{\infty} A_n \in \mathcal{A}$

is called a *sigma algebra* on $\Omega$. A triple $(\Omega, \mathcal{A}, P)$ with $\mathcal{A}$ being a sigma algebra over $\Omega$ and $P$ being a probability measure on $\mathcal{A}$, that is:

- $P(A) \geq 0 \quad \forall A \in \mathcal{A}$,
- $P(\Omega) = 1$,
- for pairwise disjoint $(A_i)_{i \geq 1}$ it holds $P(\sum_{i=1}^{\infty} A_i) = \sum_i P(A_i)$ ($\sigma$-additivity),

is called a *(continuous) Probability Space*.
For $(\Omega, \mathcal{A}, P)$ as above and $\Omega' \neq \emptyset$ with $\sigma$-algebra $\mathcal{A}'$ over $\Omega'$, a mapping $X : \Omega \to \Omega'$ is called $(\mathcal{A}, \mathcal{A}')$-measurable if:

$$X^{-1}(A') \in \mathcal{A}, \forall A' in \mathcal{A}'$$

In that case, $X$ is called a ($\Omega'$-valued) *Random Variable*.
A "natural" probability space is then defined by

$$P^X(A') := P(X^{-1}(A')), \quad A' \in \mathcal{A}'$$

#### Statistic Estimator

Let $\mathcal{X}$ be a sample space (in general, $\mathcal{X} = \mathbb{R}^n$) and let $x \in \mathcal{X}$ be realization of Random Variable $X : \Omega \to \mathcal{X}$. Let $\Theta$ be a *parameter space* for a class $\mathcal{P}$ of probability-measures (assumption of a probability distribution). Then $(\mathcal{X}, (P_\theta)_{\theta \in \Theta})$ is called a statistical model.

**Definition 11** (Estimator)**.** *An estimator for $\theta$ is a mapping*

$$T : \mathcal{X} \to \tilde{\Theta} \supset \Theta$$

*with a "suitable" superset $\tilde{\Theta}$ of $\Theta$.*

Expected value $\mathbb{E}_\theta$ w.r.t. parameter $\theta$. Variance $V_\theta$ w.r.t. parameter $\theta$.

### A.1.2. Densities

A function $f : \mathbb{R} \to \mathbb{R}$ is called a *density* if $f(x) \geq 0, \forall x \in \mathbb{R}$ and if $f$ is Riemann-integrable with

$$\int_{\mathbb{R}} f(x)\, dx = 1.$$

This defines a Probability measure $P$ on $\mathcal{B}$ by

$$P(B) := \int_B f(x)\, dx, \quad B \in \mathcal{B}$$

using the necessary tools and conventions from the Lebesgue-integral yielding a distribution for $X$.
Martingale density $f_X$ from a joint density $f_{X,Y}$:

$$f_X(x) = \int_{\mathbb{R}} f_{X,Y}(x, y)\, dy$$

Conditional density $f_{Y|X}(x, y) := \frac{f_{X,Y}(x,y)}{f_X(x)}$ for $f_X(x) > 0$.
Two continuous Random Variables $X, Y$ are stochastically independent iff $f_{X,Y}(x, y) = f_X(x) \cdot f_Y(y)$.
Adding two Random Variables $X, Y$, i.e. $Z := X + Y$ means convolving their respective densities: $f_{X+Y}(t) = (f_X * f_Y)(t) = \int_{\mathbb{R}} f_X(t - y) \cdot f_Y(y) dy$.

See [Geo13] for more details.

## A.2. Filtering

Following the scheme from section 3.1, it can be derived:

$$
\begin{aligned}
&p(X_n, Z_n, U_n) \overset{1.1}{=} \\
&\quad p(x_0) \cdot p(z_0 \mid x_0) \cdot \\
&\quad p(u_1 \mid x_0, z_0) \cdot p(x_1 \mid x_0, z_0, u_1) \cdot p(z_1 \mid x_0, z_0, u_1, x_1) \cdot \\
&\quad p(u_2 \mid x_0, z_0, u_1, x_1, z_1) \cdot p(x_2 \mid x_0, z_0, u_1, x_1, z_1, u_2) \cdot p(z_2 \mid x_0, z_0, u_1, x_1, z_1, u_2, x_2) \cdot \\
&\quad \ldots. \\
&\quad p(u_k \mid X_{k-1}, Z_{k-1}, U_{k-1}) \cdot (x_k \mid X_{k-1}, Z_{k-1}, U_k) \cdot p(z_k \mid X_k, Z_{k-1}, U_k) \cdot \\
&\quad \ldots. \\
&\quad p(u_n \mid X_{n-1}, Z_{n-1}, U_{n-1}) \cdot p(x_n \mid X_{n-1}, Z_{n-1}, U_n) \cdot p(z_n \mid X_n, Z_{n-1}, U_n)
\end{aligned}
$$

Using the simplifications as in section 3.1 by Markov and completeness:

$$
\begin{aligned}
p(x_k \mid X_{k-1}, Z_{k-1}, U_k) &= p(x_k \mid x_{k-1}, u_k) \\
p(z_k \mid X_k, Z_{k-1}, U_k) &= p(z_k \mid x_k)
\end{aligned}
$$

It can further be assumed, again by completeness, that control input $u_k$ is chosen as a reaction to only the current (complete) state $x_{k-1}$. As such,

$$
p(u_k \mid U_{k-1}, X_{k-1}, Z_{k-1}) = p(u_k \mid x_{k-1})
$$

The result can be simplified as:

$$
\begin{aligned}
&p(X_n, Z_n, U_n) = \\
&\quad p(x_0) \cdot \prod_{k=1}^{n} p(u_k \mid x_{k-1}) \cdot \prod_{k=0}^{n} p(z_k \mid x_k) \cdot \prod_{k=1}^{n} p(x_k \mid x_{k-1}, u_k) \\
&\quad = p(X_n, Z_n, U_n) \cdot p(u_n \mid x_{n-1}) \cdot p(z_n \mid x_n) \cdot p(x_n \mid x_{n-1}, u_n)
\end{aligned}
$$

The terms $p(u_k \mid x_{k-1})$ subsume a control strategy based on a precomputed or online computed plan (like a trajectory) which can itself be hidden within the state vector $x_k$, e.g. by the current time-step $k$.

## A.3. Discretization of the Strapown equations

The discretization of the Stapdown ODE, see section 3.8.2, was performed via the following Matlab script:

```
%% part II: discretization of entire system
% This uses special structure of the continuous system S_sd which yields
% some regularity in S_sd^2, S_sd^3, ...

% Add another symbolic variable for abbreviation
```

```
syms norm_w real
w_tmp = sym('w', [3,1], 'real')
assumeAlso(norm(w_tmp) == norm_w );
% w_tmp is abbreviation for "w_nb_b_meas - b_w" as an approximation of the
% true turn rate

% Split r_k's into separate parts
r_k_scalez = [cos(norm_w *tau/2); sinc( norm_w *tau/(2*pi)) * ones(3,1)];
r_k_w  = r_k_scalez.*[1; w_nb_b_meas *tau/2 ]; % measured
Sk4_d = quaternion_multiplication_matrix_reversed( r_k_w ); % r_k

disp(['Generating discretized system transition manually ',
'by using special structure of A_sd. This step takes some ',
'time to compute']);

S = -skew_symmetric(w_tmp);
S_w_meas = -skew_symmetric(w_nb_b_meas);
syms t1 t2 positive
% use represantion from INS-book (Wendel) instead to avoid numerical
% inaccuracies leading to slight imaginary part
x = norm_w*t1; %norm(norm_w^2*t1);
etS =          simplify(eye(3,3) + sinc(x/pi)*t1*S +  ...
               1/2 * (sinc(x/(2*pi)))^2 *t1^2 *S^2);
etS_w_meas =   simplify(eye(3,3) + sinc(x/pi)*t1*S_w_meas +  ...
               1/2 * (sinc(x/(2*pi)))^2 *t1^2 *S^2);

etSi1 = int( etS, t1, [0, t2] );
etSi2 = int( etSi1, t2, [0, tau] );

% Update: first order gyro-bias influence on body-velocity added
A_sd_d = simplify(...
[ zeros(3,3), R_nb*subs(etSi1, t2, tau), ...zeros(3,4), -R_nb*etSi2, zeros(3,3);...
  zeros(3,3), subs(etS_w_meas, t1, tau), zeros(3,4), -subs(etSi1, t2, tau),...
    -tau* sinc(norm_w*tau/pi) *skew_symmetric(v_nb_b);
  zeros(4,3+3), Sk4_d , zeros(4,3), -tau/2 * ...
               quat_multiplication_matrix(:,2:end)*diag(r_k_scalez(2:end)); ...
  zeros(3,3+3+4+3+3);...
  zeros(3,3+3+4+3+3);
]) ...
        + blkdiag( eye(3,3), zeros(7,7),  eye(6,6) );

disp(['Checking correctness of discretization through sample values. ',
'Should yield small values with absolute sum close to zero (or machine acc.)']);


syms dt positive

disp('Discretizing control input matrix.');
% Control input matrix
%B_sd_d = dt*B_sd % first order
%B_sd_d = simplify(dt*B_sd + 1/2*dt^2 *A_sd*B_sd) % second order
%B_sd_d = simplify( int( simplify(A_sd_d*B_sd) , tau, [0 dt] ) )
tic
B_sd_d = simplify( int( A_sd_d, tau, [0 dt] ) *B_sd )
toc

disp(['Accurate Discretizing of noise matrix is currently deactivated ',
'since it takes very long and it yields singularities!'])

% for discrete values, the respective observation
% "integration" time t_obs needs to be incorporated
Q_ahrs_obs = Q_ahrs;

Q_ahrs_obs(1:6,1:6) = Q_ahrs(1:6,1:6)/dt_obs;

% optional since it should be almost 1 everywhere
Q_sd_d_optional_noise_influence = sym(ones(size(Q_sd)));
Q_sd_d_optional_noise_influence(4:6,4:6) =  ...
        Q_sd_d_optional_noise_influence(4:6,4:6) * sym(sinc(norm_w*dt/pi));
Q_sd_d_optional_noise_influence(7:10,4:6) = ...
        Q_sd_d_optional_noise_influence(7:10,4:6) .* (sym(ones(4,1)) ...
        * subs(r_k_scalez(2:end).', tau, dt) );
```

```
Q_sd_d = Q_sd.*Q_sd_d_optional_noise_influence;

disp('Using simplified version with quasi-linear approximation from QB-EKF paper.');
Q_sd_total_d = (dt*Q_sd_d*Q_ahrs_obs*(dt*Q_sd_d.'));
% random walk part (11:16) treated differently:
Q_sd_total_d = simplify( blkdiag( Q_sd_total_d(1:10,1:10), ...
        tau*Q_ahrs_obs(7:end,7:end) ) )

% Replace tau by dt for final matrix
A_sd_d = simplify(subs( A_sd_d, tau, dt ))
```

# B. Various Notes on Underwater Sensor

## B.1. Additional Equations for various sensor models

Additional equations for the linearization of magnetic readings from section 3.3.1:

$$H^m_{q_1} = \begin{pmatrix} 2\,q_1 & 2\,q_4 & -2\,q_3 \\ -2\,q_4 & 2\,q_1 & 2\,q_2 \\ 2\,q_3 & -2\,q_2 & 2\,q_1 \end{pmatrix}$$

$$H^m_{q_2} = \begin{pmatrix} 2\,q_2 & 2\,q_3 & 2\,q_4 \\ 2\,q_3 & -2\,q_2 & 2\,q_1 \\ 2\,q_4 & -2\,q_1 & -2\,q_2 \end{pmatrix}$$

$$H^m_{q_3} = \begin{pmatrix} -2\,q_3 & 2\,q_2 & -2\,q_1 \\ 2\,q_2 & 2\,q_3 & 2\,q_4 \\ 2\,q_1 & 2\,q_4 & -2\,q_3 \end{pmatrix}$$

$$H^m_{q_4} = \begin{pmatrix} -2\,q_4 & 2\,q_1 & 2\,q_2 \\ -2\,q_1 & -2\,q_4 & 2\,q_3 \\ 2\,q_2 & 2\,q_3 & 2\,q_4 \end{pmatrix}$$

## B.2. Additional Equations for USBL error model

Using definition 10 for the given hydrophone setting $P_{hydro}$ from equation 3.21, the respective pseudoinverse $S^+$ evaluates as:

$$S^+ = \begin{pmatrix} -\frac{20}{17} & -\frac{20}{17} & -\frac{40}{17} & -\frac{20}{17} & 0 & -\frac{20}{17} & 0 & -\frac{20}{17} & 0 & \frac{20}{17} \\ -\frac{20}{17} & \frac{20}{17} & 0 & 0 & \frac{40}{17} & \frac{20}{17} & \frac{20}{17} & -\frac{20}{17} & -\frac{20}{17} & 0 \\ 0 & 0 & 0 & -2 & 0 & 0 & -2 & 0 & -2 & -2 \end{pmatrix}$$

The algebraic expression for estimating the systematic angle distortion by Planar Wave assumption, see section 3.7.3 and 3.7.5 can be produced by the following Matlab code using the Symbolic Math Toolbox:

```
syms el r az positive

% only consider quarter azimuth (due to symmetry) and eleveation 0 to 90Âř
assume( 0<=az & az<=pi/2 )
assume( 0<=el & el<=pi/2 )

% polar coordinate representation
r_dir = r*[ sin(pi/2-el)*cos( az ) ; sin(pi/2-el)*sin(az); cos(pi/2-el) ];
```

```
% general rdoa
rdoa_sym = USBLPseudorange_rdoa( r_dir, P_hydro );

% transformed direction (PW projection)
dir_tr = simplify(-Sp*rdoa_sym)

tr_r = simplify( norm(dir_

% overall angle distortion by scalar product
angle_distorion = simplify(acos( dir_tr.' * r_dir / (tr_r * r)));

% evaluate angle miss-match by distance and evaluation
angle_range = linspace(0,90,30)*pi/180;
r_range = logspace(log10(1),log10(6000),30);
[el_mesh, r_mesh] = meshgrid(angle_range, r_range);

amm_eval = double(subs( angle_distorion ,{el, az, r}, ...
                    {el_mesh, pi/4, r_mesh} ));
```

## B.3. SMIS AUV Sensor Positions in Body frame

### B.3.1. Sensor Positions within SMIS AUV

The (approximate) positions of the sensors within the SMIS AUV used for basic navigation, i.e. AHRS, GPS, depth/pressure, USBL and DVL, were as follows:

$$\begin{aligned}
\vec{p}^{\,b}_{GPS/b} &= (0.377, 0.03, -0.488)^T \\
\vec{p}^{\,b}_{d/b} &= (0.013, 0, 0.276)^T \\
\vec{p}^{\,b}_{USBL/b} &= (-0.469, 0, -0.363)^T \\
\vec{p}^{\,b}_{DVL/b} &= (0.817, 0, 0.358)^T \\
\vec{p}^{\,b}_{AHRS/b} &= (1.429, 0, 0)^T
\end{aligned}$$

Basic navigation was performed within the AHRS-frame.

### B.3.2. Sensor Positions for PCRB computation

The Sensor positions for the PCRB computations were slightly adjusted to the ones given in the original AUV in order to have a slightly more general positioning for referencing

with other works.

$$\begin{aligned}
\vec{p}_{GPS/b}^{\,b,PCRB} &= (0.377, 0.03, -0.488)^T \\
\vec{p}_{d/b}^{\,b,PCRB} &= (0.013, 0, 0.276)^T \\
\vec{p}_{USBL/b}^{\,b,PCRB} &= (-0.469, 0, -0.363)^T \\
\vec{p}_{DVL/b}^{\,b,PCRB} &= (0, 0, 0.358)^T \\
\vec{p}_{AHRS/b}^{\,b,PCRB} &= (0, 0, 0)^T
\end{aligned}$$

The main differences are: the AHRS is assumed to be within the center of the body frame and the DVL is positioned directly underneath it. All other (absolute) sensor positions remain unchanged to the original SMIS AUV.

# C. On the AUV Dynamic Model

## C.1. Further details of the AUV dynamic model in Simulink

In this section, further details of the Simulink model given in section 3.9 are presented. The thruster model in Simulink is presented in figure C.1. Figure C.2 illustrates details

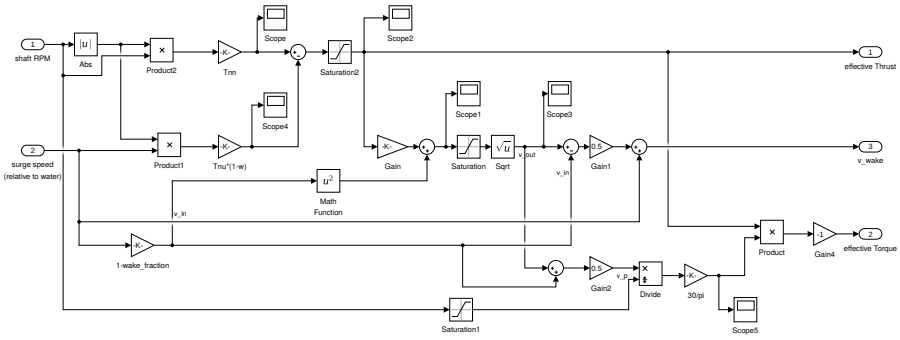


Figure C.1.: Simulink model of the AUV thruster (detailed view).

of the rudder model in Simulink. In figure C.3, details of Simulink model for the hydro-

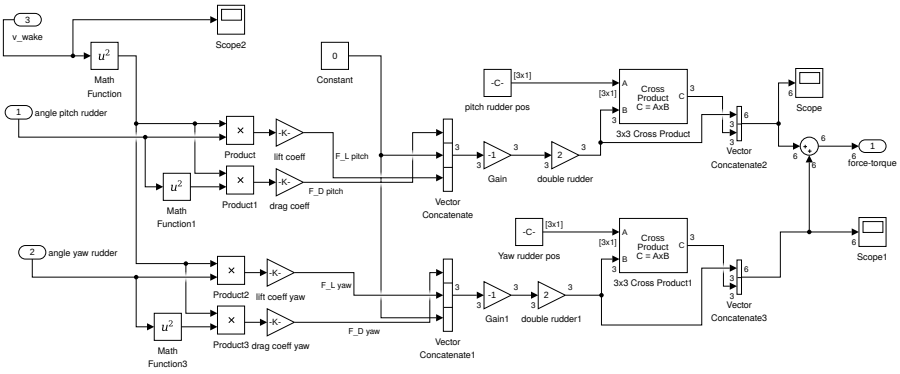


Figure C.2.: Simulink model of the rudder forces (detailed view).

dynamic forces through damping and lift are presented.

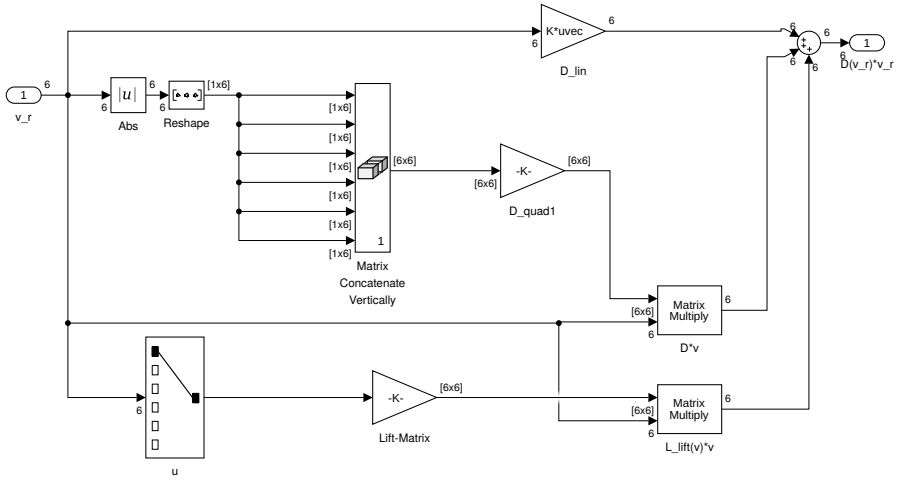


Figure C.3.: Simulink model of hydrodynamic Damping and Lift (detailed view) using linear and quasi-quadratic damping.

## C.2. Reference Model Parameters

See section 4.5.5.

$$M_{RB} = \begin{pmatrix} 1000.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1000.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1000.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 181.12 & 0 & 0 \\ 0 & 0 & 0 & 0 & 663.69 & 0 \\ 0 & 0 & 0 & 0 & 0 & 663.69 \end{pmatrix}$$

$$M_A = \begin{pmatrix} 797 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1021.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1021.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 48.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 512.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 512.2 \end{pmatrix}$$

$$D_{nl} = \begin{pmatrix} 32.47 & 0 & 0 & 0 & 0 & 0 \\ 0 & 970.15 & 0 & 0 & 0 & -223.0 \\ 0 & 0 & 1003.38 & 0 & 156.1 & 0 \\ 0 & 0 & 0 & 1.68 & 0 & 0 \\ 0 & 0 & -146.0 & 0 & 550.0 & 0 \\ 0 & 146.0 & 0 & 0 & 0 & 550.0 \end{pmatrix}$$

$$D_{l} = \begin{pmatrix} 60.55 & 0 & 0 & 0 & 0 & 0 \\ 0 & 349.43 & 0 & 0 & 0 & -200.0 \\ 0 & 0 & 349.43 & 0 & 150.0 & 0 \\ 0 & 0 & 0 & 70.0 & 0 & 0 \\ 0 & 0 & -50.0 & 0 & 500.0 & 0 \\ 0 & 50.0 & 0 & 0 & 0 & 500.0 \end{pmatrix}$$

$$L_{lift} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 332.52 & 0 & 0 & 0 & -23.4 \\ 0 & 0 & 332.52 & 0 & 39.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 199.51 & 0 & 300.0 & 0 \\ 0 & -119.71 & 0 & 0 & 0 & 230.0 \end{pmatrix}$$

$$d_{BW} = 98.1$$

$$\vec{\gamma} = \begin{pmatrix} 5.0 \\ 0 \\ 169.95 \end{pmatrix}$$

lift coefficient yaw: 42.18

lift coefficient pitch: 29.11

drag coefficient: 8.73

$$c_{Q_T} = 0.4$$

Further rudder parameters, given relative to the body origin:

$$\text{yaw rudders: } (-1.4210, \pm 0.2440, 0)^T$$
$$\text{pitch rudders: } (-1.4210, 0, \mp 0.2440)^T$$

And thruster parameters were set as follows:

$$T_{|n|n} = 7.813155 \cdot 10^{-3}, \quad T_{|n|u1w} = 0.5357592, \quad w = 0.2,$$

see section 4.5.1.

### C.2.1. Added mass

The added mass estimate used initially in section 4.5.5 was calculated using the following script based on [Bre82].

```
function [ ] = simplified_added_mass( a, b )
%SIMPLIFIED_ADDED_MASS estimate SMIS AUV added mass
%   based on ellipsoid and cubic shape estimate
%   input: a>b -- ellipsoid's "radii"
%
%   according to added mass paper BRE052

assert( a>b );

a_el = 1.05*a; b_el = 1.05*b;
e = axes2ecc( a_el, b_el );

% log: natural logrithm
alpha0 = (1-e^2)/e^3 *(log((1+e)/(1-e)) -2*e);
beta0 = (1-e^2)/e^3 *(e/(1-e^2) - 1/2*log((1+e)/(1-e)) );

k1 = alpha0/(2-alpha0);
k2 = beta0/(2-beta0);

rho = 1030; % water density

% translational added mass
base_added_mass_transl = 4/3*rho*pi*a_el*b_el^2;
m_a_x = base_added_mass_transl*k1
m_a_y = base_added_mass_transl*k2


% rotational added mass/inertia
base_added_mass_rot_pitch = 4/15 * rho ...
        *pi*a_el*b_el^2 *(a_el^2 + b_el^2)
kp_pitch = (a_el^2 - b_el^2)^2 *(beta0-alpha0) / ...
        ( (a_el^2+b_el^2)* (2*(a_el^2-b_el^2) -(a_el^2+b_el^2)*(beta0-alpha0)))

% pitch and yaw moment:
m_a_p = base_added_mass_rot_pitch*kp_pitch

% by the equations, added roll mass would be zero for ellipsoid
%base_added_mass_rot_roll = 4/15*rho*pi*a*b^2*(b^2 + b^2)
%kp_roll = (b^2- a^2)^2 *(alpha0-beta0) / ...
%               ( (a^2+b^2)* (2*(b^2-a^2) -(a^2+b^2)*(alpha0-beta0)))

M_A_ellipsoid = diag( [ m_a_x, m_a_y, m_a_y, 0, m_a_p, m_a_p] )

% added values based on cuboid
m_a_x_cuboid = 0.42*rho*(2*b)^2 *2*a; % BRE052
m_a_y_cuboid = 1.51*pi*rho*b^2 *2*a; % Readings6-bepps

m_a_roll_cuboid = 0.234*pi*rho* b^4 *2*a; % Readings6-bepps
m_a_pitch_cuboid = 0.234*pi*rho* a^4 *2*b;

M_A_cuboid = diag( [ m_a_x_cuboid, m_a_y_cuboid, m_a_y_cuboid, ...
        m_a_roll_cuboid, m_a_pitch_cuboid, m_a_pitch_cuboid] )

end
```

The previous code expects the AUV's semi-length $a = 3.47/2$ and semi-width $b = 0.72/2$ as input parameters. Unfortunately, the suggested estimates based on ellipsoidal and cubic shapes differ quite drastically (almost by an order of magnitude w.r.t. inertia). As the SMIS AUV is neither ellipsoidal nor really cubical, a hybrid between the two was used as an initial guess:

$$M_{A,initial} = \begin{pmatrix} 77.8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1021.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1021.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 48.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 512.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 512.2 \end{pmatrix}$$

which is almost the same as the one listed in section C.2 except for the $x$-parameter.

### C.2.2. Classical Identification

The code for generating the matrices and vectors for the classical identification scheme from section 4.5.4 is as follows:

```
syms mass positive
syms r_gb_x r_gb_y r_gb_z real
% inertia matrix assumed diagonal ?
syms I_xx I_yy I_zz positive
syms I_xy I_xz I_yz real
syms m11 m22 m26 m33 m35 m44 m55 m66 real
syms v1 v2 v3 v4 v5 v6 real
syms a1 a2 a3 a4 a5 a6 real
%syms w_b_x w_b_y w_b_z real
% current velocity in body coordinates:
syms v_cx v_cy v_cz real

syms q1 q2 q3 q4 real
q_nb = [q1; q2; q3; q4];

R_nb = quat2rotm_sym([q1;q2;q3;q4]);

v_c_ned = [v_cx; v_cy; v_cz; 0; 0; 0]; % in NED frame
% => v_c_body = R_bn*v_c_ned = R_nb.'v_c_ned

%w_b = [w_b_x w_b_y w_b_z].'; % AUV turn rate --> already contained in v
w_b = [v4; v5; v6];
% current body accel. (according to Fossen) for irrotational const current
a_c = [-skew_symmetric(w_b)*R_nb.'*v_c_ned(1:3); 0; 0; 0];

% inertia matrix w.r.t. center of gravity
% for transformation to center of gravity, see Fossen equation (3.35)
inertia_matrix_cog = ...
    [ I_xx, -I_xy, -I_xz;
     -I_xy,  I_yy, -I_yz;
     -I_xz, -I_yz,  I_zz];
% center of gravity
r_gb = [r_gb_x; r_gb_y; r_gb_z]; % coordinates in actual body-frame
M_RB = [ mass*eye(3,3), -mass*skew_symmetric(r_gb);
     mass*skew_symmetric(r_gb), inertia_matrix_cog - mass*skew_symmetric(r_gb)^2 ];

disp('Checking mass matrix symmetry');
simplify(M_RB - M_RB.')
if sum(sum(ans)) ~= 0
```

```
    warn('failed.');
else
    disp('OK.');
end

% v1:v6 body-velocity w.r.t. original body-frame
v = [v1; v2; v3; v4; v5; v6];
a = [ a1 a2 a3 a4 a5 a6 ].'; % body acceleration
% reduced added mass matrix defined in original body-origin:
M_A = ...
    [ m11 0 0 0 0 0 ;
    0 m22 0 0 0 m26 ;
    0 0 m33 0 m35 0 ;
    0 0 0 m44 0 0;
    0 0 m35 0 m55 0;
    0 m26 0 0 0 m66 ];

% only valid since irrotational current:
v_r = v - blkdiag(R_nb.',zeros(3,3))*v_c_ned;
a_r = a - a_c;

cvv = simplify(coriolis2(M_RB,v,0)*v + coriolis2(M_A,v_r,0)*v_r)

m = [ mass, I_xx, I_yy, I_zz, I_xy, I_xz, I_yz, m11, m22,...
                m26, m33, m35, m44, m55, m66].';

% use derivative to receive A*m shape
Ct = jacobian(cvv,m)

Mt = jacobian( M_RB*a+M_A*a_r, m )


use_simplified_drag = 1;
use_ITTC = 0;

if use_simplified_drag == 1

    % quadratic drag
    syms d11 d22 d26 d33 d35 d44 d55 d53 d66 d62  real
    d_nl = [d11 d22 d26 d33 d35 d44 d55 d53 d66 d62 ].';

    D_nl = ...
        repmat( abs(v_r.'), 6, 1 ) .*...
        [ d11 0 0 0 0 0 ;
        0 d22 0 0 0 d26 ;
        0 0 d33 0 d35 0 ;
        0 0 0 d44 0 0;
        0 0 d53 0 d55 0;
        0 d62 0 0 0 d66 ];

    % ITTC resistance law
    if use_ITTC == 1
        % see Fossen Handbook, p. 130

        syms d22r d26v d62r d66v real
        d_nl = [d_nl; [d22r d26v d62r d66v].'];

        D_nl = D_nl + ...
            [ zeros(1,6);
            0, d22r*abs(v_r(6)), 0,0,0, d26v*abs(v_r(2));
            zeros(3,6);
            0, d62r*abs(v_r(6)), 0,0,0, d66v*abs(v_r(2));
            ];
    end

    % linear drag
    syms dl11 dl22 dl26 dl33 dl35 dl44 dl55 dl53 dl66 dl62 real
    d_l = [dl11 dl22 dl26 dl33 dl35 dl44 dl55 dl53 dl66 dl62].';
    D_l = ...
        [ dl11 0 0 0 0 0 ;
        0 dl22 0 0 0 dl26 ;
        0 0 dl33 0 dl35 0 ;
```

```
    0 0 0 dl44 0 0;
    0 0 dl53 0 dl55 0;
    0 dl62 0 0 0 dl66 ]

% Update: use lift matrix according to LAUV-paper
syms l22 l26 l33 l35 l53 l55 l62 l66 real
L_lift = ...
    [ 0 0 0 0 0 0 ;
    0 l22 0 0 0 l26 ;
    0 0 l33 0 l35 0 ;
    0 0 0 0 0 0;
    0 0 l53 0 l55 0;
    0 l62 0 0 0 l66 ]*v_r(1)
b_l = [l22 l26 l33 l35 l53 l55 l62 l66].';
% note: Hegrenaes uses additional cross-coupling to u by v and r

d_total = [d_nl; d_l;b_l];

dvv = (D_nl+D_l + L_lift)*v_r;

Dt = simplify(jacobian( dvv, d_total ))

syms pitch_rudder_pos_x pitch_rudder_pos_y pitch_rudder_pos_z real
syms yaw_rudder_pos_x yaw_rudder_pos_y yaw_rudder_pos_z real

%     pitch_rudder1_pos = get_default_param_value('elevator1_origin');
%     yaw_rudder1_pos = get_default_param_value('yaw_rudder_origin');
pitch_rudder1_pos = [pitch_rudder_pos_x; pitch_rudder_pos_y; ...
                     pitch_rudder_pos_z];
pitch_rudder2_pos = pitch_rudder1_pos;
pitch_rudder2_pos(2) = -pitch_rudder2_pos(2);
yaw_rudder1_pos = [yaw_rudder_pos_x; yaw_rudder_pos_y; ...
                   yaw_rudder_pos_z];
yaw_rudder2_pos = yaw_rudder1_pos;
yaw_rudder2_pos(3) = - yaw_rudder2_pos(3);

syms alpha_yaw alpha_pitch real
% update: use separate lift coefficients for yaw and pitch rudder.
% Only drag coefficient will be shared
syms lift_coeff_yaw lift_coeff_pitch drag_coeff positive

%% Intermezzo: Right-hand side:

% delta B-W (weight vs buoyancy)
syms dBW gravity_const positive
% gravitiy/buoyancy-force/torque vector (replacing r_bb and r_gb)
syms gamma1 gamma2 gamma3 real
gamma = [gamma1; gamma2; gamma3 ];
%syms r_bb_x r_bb_y r_bb_z real
%r_bb = [r_bb_x ; r_bb_y ; r_bb_z ];

syms Tnn Tnu1mW RPM A_p water_density wake_fraction_number positive
%syms Tnn Tnu1mW RPM A_p positive
%water_density = 1050;
%wake_fraction_number = 0.2;

[f_load, v_wake] = compute_simplified_DMLoadVector( Tnn, Tnu1mW, RPM, v_r(1),...
            water_density, A_p, wake_fraction_number ) %water_density ca 1050

%% Back to left-hand side (need expression for v_wake)

gravityBuoyancy_FT = simplified_gravity_buoyancy_body( dBW, gamma , R_nb.' )

gBg = [dBW; gamma];
Gt = jacobian(gravityBuoyancy_FT, gBg );


rudder_FT = simplify(getRudderForceTorque( v_wake, alpha_pitch,...
    alpha_yaw, ...
    pitch_rudder1_pos, pitch_rudder2_pos,...
    yaw_rudder1_pos, yaw_rudder2_pos, ...
    drag_coeff, lift_coeff_yaw, lift_coeff_pitch ))
```

```
        rld = [lift_coeff_yaw; lift_coeff_pitch; drag_coeff];

        Rt = jacobian( rudder_FT, rld )

        disp(['System matrix and vector of unknown parameters.',...
              '[Mt+Ct, Dt, -Rt] [m; d_total]']);
        % -Rt since it comes from right-hand side
        L = [Mt+Ct, Dt, Gt, -Rt]  % input v, dv/dt=a
        dm_param = [m; d_total; gBg; rld];
        disp('Size of L:'); size(L)

        disp('Checking correctness of rewritten equations. All 0 means success');
        simplify( L*dm_param - (M_A*a_r + M_RB*a + coriolis2(M_RB,v,0)*v + ...
                    coriolis2(M_A,v_r,0)*v_r + (D_l + D_nl +L_lift)*v_r + ...
                    gravityBuoyancy_FT - rudder_FT) )
        if sum(abs(ans)) ~= 0
            warning('Error, 0 was expected!');
        else
            disp('OK.');
        end

        %% Observability
        rank( L )
        % depends on the variation in L from time step to time step
        %disp('Checking observability');
        %N_states = dim( dm_param );
        %rank( repmat(L, N_states, 1) )



        %% Derive linearized error matrix for generalized least-squares
        z_residual = f_load - L*dm_param; %simplify(f_load - L*dm_param);

        error_influence_vector = [v; a; q_nb; v_c_ned(1:3)];

        %J_z = simplify(jacobian( z_residual, error_influence_vector ))
        J_z = zeros(size(z_residual,1),size(error_influence_vector,1)) % tmp: omit
    end
```

# D. Further Results on PCRB

## D.1. Additional PCRB results

A sample for the acceleration and gyro biases simulated by a random walk is given in figure D.1. To demonstrate the importance of an accurate orientation estimation, the

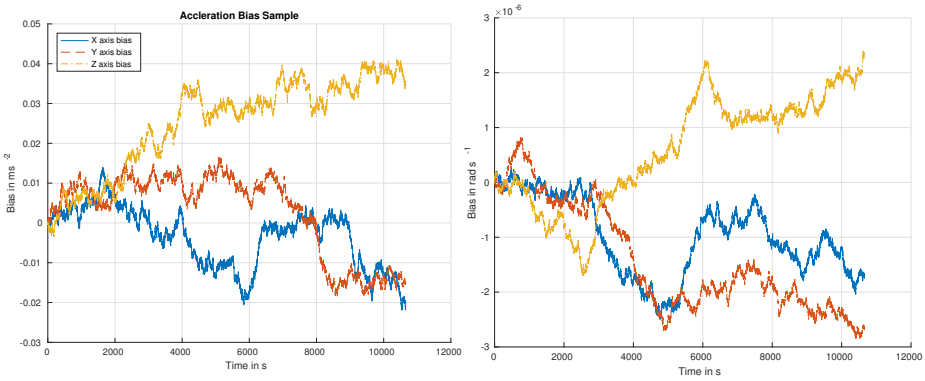


Figure D.1.: Left: Acceleration sample bias. Right: Gyro sample bias as was used in section 4.7.2. Both were generated from Random Walks as explained in section 3.2 and 4.3.1 using the discrete time equivalent given in section 3.8.2 with the Sensor constants from table 3.1.

same simulated experiment as was conducted in section 4.7.2 was performed again *without Compass and Roll/Pitch stabilization*. The resulting approximate PCRBs are shown in figure D.2, D.3 and D.4.
It must be noted that in this computation, the actual bias magnitudes were reduced by a factor of ca. 3 although the filtering algorithm assumed the full bias magnitude. It can be clearly seen, that without orientation stabilization the accuracy results are much worse than the ones given in section 4.7.2 despite reducing the bias influence. Otherwise, the results would look even worse.

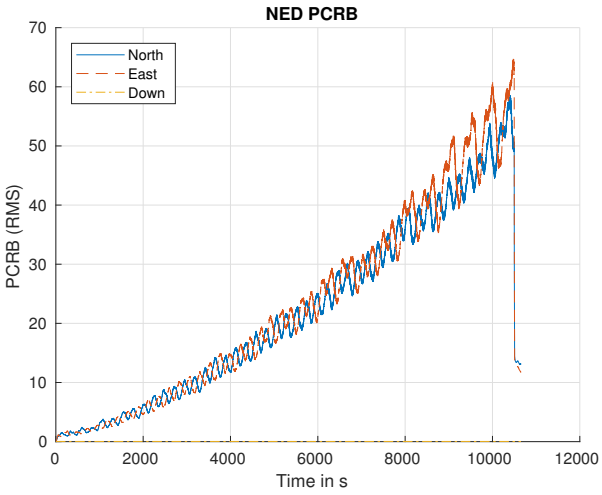


Figure D.2.: Approximate PCRB results w.r.t. NED coordinates without the use of Compass and Roll/Pitch stabilization.

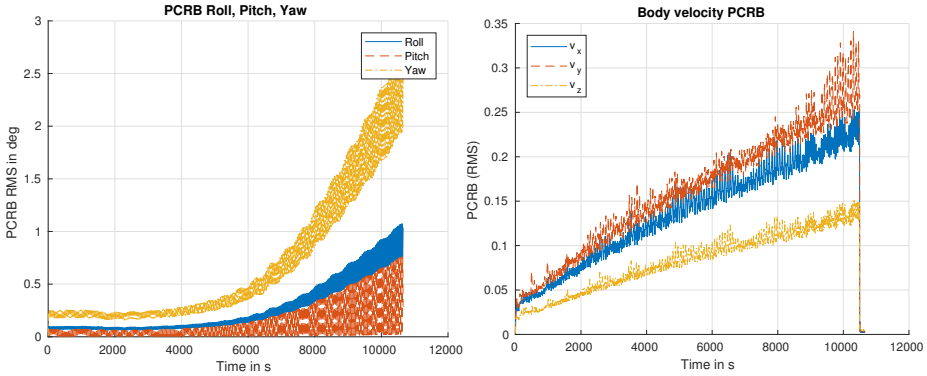


Figure D.3.: Left: PCRB for Orientation angles. Right: PCRB for velocities. The increasing orientation estimation inaccuracy beginning at $t = 6000s$ is crucial for the overall accuracy.

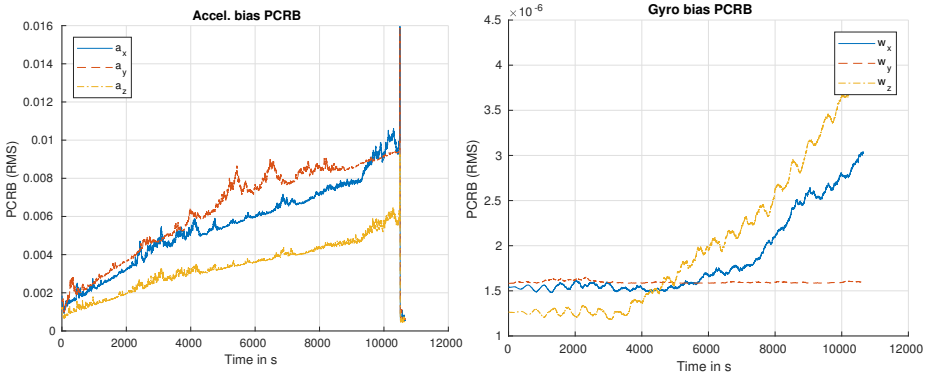


Figure D.4.: Approximate Bias PCRBs. Left: Acceleration bias PCRB. Right: Gyro Bias PCRB which degenerates over time and is much worse without the use of Compass measurements, see figure 4.25.

# List of Figures

# List of Tables

# Bibliography


[AC91] Jonathan S Abel and James W Chaffee. Existence and uniqueness of GPS solutions. *IEEE Transactions on Aerospace and Electronic Systems*, 27(6):952–956, 1991.

[AC02] Jamie M Anderson and Narender K Chhabra. Maneuvering and stability performance of a robotic tuna. *Integrative and comparative biology*, 42(1):118–126, 2002.

[ACC+16] B Allotta, A Caiti, R Costanzi, F Fanelli, D Fenucci, E Meli, and A Ridolfi. A new AUV navigation system exploiting unscented kalman filter. *Ocean Engineering*, 113:121–132, 2016.

[Ach74] Elmar Achenbach. Vortex shedding from spheres. *Journal of Fluid Mechanics*, 62(2):209–221, 1974.

[AH06] Kjetil Bergh Anonsen and Oddvar Hallingstad. Terrain aided underwater navigation using point mass and particle filters. In *Position, location, and navigation symposium, 2006 IEEE/ION*, pages 1027–1035. IEEE, 2006.

[AMGC02] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.

[And99] John David Anderson. *Aircraft performance and design*. McGraw-Hill Science/Engineering/Math, 1999.

[AP07] A Pedro Aguiar and António M Pascoal. Dynamic positioning and waypoint tracking of underactuated AUVs in the presence of ocean currents. *International Journal of Control*, 80(7):1092–1108, 2007.

[AP08] Panos Aliferis and John Preskill. Fault-tolerant quantum computation against biased noise. *Physical Review A*, 78(5):052331, 2008.

[Ark12] Mikhail Arkhipov. Designing a USBL system based on a square pyramid array with a complete set of three-element arrays. In *2012 Oceans*, pages 1–9. IEEE, 2012.

[AS87] JS Abel and J Smitht. The spherical interpolation method for closed-form passing source localization using range difference measurements. In *in Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*. Citeseer, 1987.

[AS02] Roberto Alonso and Malcohn D Shuster. Complete linear attitude-independent magnetometer calibration. *Journal of the Astronautical Sciences*, 50(4):477–490, 2002.

[ASSS04] Ahmed I Al-Shamma'a, Andrew Shaw, and Saher Saman. Propagation of electromagnetic waves at MHz frequencies through seawater. *IEEE Transactions on Antennas and Propagation*, 52(11):2843–2849, 2004.

[AW95] John David Anderson and J Wendt. *Computational fluid dynamics*, volume 206. Springer, 1995.

[Bak81] DJ Baker. Ocean instruments and experiment design, 1981.

[BD06] Peter J Brockwell and Richard A Davis. *Introduction to time series and forecasting*. Springer Science & Business Media, 2006.

[Ber99] Niclas Bergman. Recursive bayesian estimation. *Department of Electrical Engineering, Linköping University, Linköping Studies in Science and Technology. Doctoral dissertation*, 579:11, 1999.

[Beu15] Ottmar Beucher. *Signale und Systeme: Theorie, Simulation, Anwendung: Eine beispielorientierte Einführung mit MATLAB*. Springer-Verlag, 2015.

[BGRH14] F. Boeck, M. Golz, S. Ritz, and G. Holbach. SMIS - subsea monitoring via intelligent swarms, design challenges of an autonomous seabed station. In *ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering Volume 8B: Ocean Engineering*, San Francisco, California, USA, 2014.

[BGRH16] Florin Boeck, Matthias Golz, Sebastian Ritz, and Gerd Holbach. Automated landing and mooring in deep-sea environment using suction buckets for unmanned vehicles. In *ASME 2016 35th International Conference on Ocean, Offshore and Arctic Engineering*, pages V007T06A027–V007T06A027. American Society of Mechanical Engineers, 2016.

[BH97] Robert Grover Brown and Patrick YC Hwang. Introduction to random signals and applied kalman filtering: with MATLAB exercises and solutions. *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions, by Brown, Robert Grover.; Hwang, Patrick YC New York: Wiley, c1997.*, 1997.

[BJ87] Richard S Bucy and Peter D Joseph. *Filtering for stochastic processes with applications to guidance*, volume 326. American Mathematical Soc., 1987.

[Bjö96] Åke Björck. *Numerical methods for least squares problems*. SIAM, 1996.

[BLF00] Mogens Blanke, Karl-Petter Lindegaard, and Thor I Fossen. Dynamic model for thrust generation of marine propellers. *IFAC Proceedings Volumes*, 33(21):353–358, 2000.

[Bli01] D Richard Blidberg. The development of autonomous underwater vehicles (AUV); a brief summary. In *Ieee Icra*, volume 4, 2001.

[bmw] Homepage of german federal ministry of economic affairs and energy. `http://www.bmwi.de`. Accessed: 2017-07-14.

[Bor71] John E Bortz. A new mathematical formulation for strapdown inertial navigation. *IEEE transactions on aerospace and electronic systems*, (1):61–66, 1971.

[Bre82] CE Brennen. A review of added mass and fluid inertial forces. Technical report, BRENNEN (CE) SIERRA MADRE CA, 1982.

[BS93] Dennis S Bernstein and Wasin So. Some explicit formulas for the matrix exponential. *IEEE Transactions on Automatic Control*, 38(8):1228–1232, 1993.

[BSS+99] VA Belogol'Skii, SS Sekoyan, LM Samorukova, SR Stefanov, and VI Levtsov. Pressure dependence of the sound velocity in distilled water. *Measurement Techniques*, 42(4):406–413, 1999.

[BWXW99] Chunli Bai, Chen Wang, X Sunney Xie, and Peter G Wolynes. Single molecule physics and chemistry. *Proceedings of the National Academy of Sciences*, 96(20):11075–11076, 1999.

[Car12] John Carlton. *Marine propellers and propulsion*. Butterworth-Heinemann, 2012.

[CGN05] Paola Causin, Jean-Frédéric Gerbeau, and Fabio Nobile. Added-mass effect in the design of partitioned algorithms for fluid–structure problems. *Computer methods in applied mechanics and engineering*, 194(42):4506–4527, 2005.

[CGTH13] Philipp Crocoll, Lorenz Görcke, Gert F Trommer, and Florian Holzapfel. Unified model technique for inertial navigation aided by vehicle dynamics model. *Navigation*, 60(3):179–193, 2013.

[CMA+15] Arnaud Chulliat, Susan Macmillan, Patrick Alken, Ciaran Beggan, Manoj Nair, Brian Hamilton, Adam Woods, Victoria Ridley, Stefan Maus, and Alan Thomson. The US/UK world magnetic model for 2015-2020. 2015.

[CSST14] Philipp Crocoll, Justus Seibold, Georg Scholz, and Gert F Trommer. Model-aided navigation for a quadrotor helicopter: A novel navigation system and first experimental results. *Navigation*, 61(4):253–271, 2014.

[CSZ11] Jian Cao, Yumin Su, and Jinxin Zhao. Design of an adaptive controller for dive-plane control of a torpedo-shaped AUV. *Journal of Marine Science and Application*, 10(3):333–339, 2011.

[CTSC10] Giuseppe Casalino, Alessio Turetta, Enrico Simetti, and Andrea Caiti. RT 2: A real-time ray-tracing method for acoustic distance evaluations among cooperating AUVs. In *OCEANS 2010 IEEE-Sydney*, pages 1–8. IEEE, 2010.

[CWD+10] Xianhui Che, Ian Wells, Gordon Dickers, Paul Kear, and Xiaochun Gong. Re-evaluation of RF electromagnetic communication in underwater sensor networks. *IEEE Communications Magazine*, 48(12):143–151, 2010.

[DB13] Peter Deuflhard and Folkmar Bornemann. *Gewöhnliche Differentialgleichungen*. Walter de Gruyter, 2013.

[DCD15] Davide Dardari, Pau Closas, and Petar M Djurić. Indoor tracking: Theory, methods, and technologies. *IEEE Transactions on Vehicular Technology*, 64(4):1263–1278, 2015.

[DED] Deep diving autonomous underwater vehicle for exploration: "DEDAVE" (product sheet). `https://www.iosb.fraunhofer.de/servlet/is/5382/DeDAvE.pdf?command=downloadContent&filename=DeDAvE.pdf`. Accessed: 2018-02-21.

[Dem97] James W Demmel. *Applied numerical linear algebra*. SIAM, 1997.

[Deu08] Peter Deuflhard. Numerische mathematik: Eine algorithmisch orientierte einfÃijhrung. *De Gruyter Lehrbuch*, 2008.

[DF83] Fred Daum and Robert Fitzgerald. Decoupled kalman filters for phased array radar tracking. *IEEE Transactions on Automatic Control*, 28(3):269–283, 1983.

[DM98] Pierre Del Moral. Measure-valued processes and interacting particle systems. application to nonlinear filtering problems. *Annals of Applied Probability*, pages 438–495, 1998.

[Don12] Glenn T Donovan. Position error correction for an autonomous underwater vehicle inertial navigation system (INS) using a particle filter. *IEEE Journal of Oceanic Engineering*, 37(3):431–445, 2012.

[DRS02] Steven D'Hondt, Scott Rutherford, and Arthur J Spivack. Metabolic activity of subsurface life in deep-sea sediments. *Science*, 295(5562):2067–2070, 2002.

[dSTMdS07] Jorge Estrela da Silva, Bruno Terra, Ricardo Martins, and Joao Borges de Sousa. Modeling and simulation of the lauv autonomous underwater vehicle. In *13th IEEE IFAC International Conference on Methods and Models in Automation and Robotics*, 2007.

[ESHN08] Naser El-Sheimy, Haiying Hou, and Xiaoji Niu. Analysis and modeling of inertial sensors using allan variance. *IEEE Transactions on instrumentation and measurement*, 57(1):140–149, 2008.

[evo] EvoLogics GmbH, company homepage. `https://evologics.de/`. Accessed: 2016-12-17.

[Fal11] Gregory Falkovich. *Fluid mechanics: A short course for physicists*. Cambridge University Press, 2011.

[FB00] Thor I Fossen and Mogens Blanke. Nonlinear output feedback control of underwater vehicle propellers using feedback form estimated axial flow velocity. *IEEE Journal of oceanic Engineering*, 25(2):241–255, 2000.

[Fit71] R Fitzgerald. Divergence of the kalman filter. *IEEE Transactions on Automatic Control*, 16(6):736–747, 1971.

[FJP09] Thor I Fossen, Tor Arne Johansen, and Tristan Perez. *A survey of control allocation methods for underwater vehicles*. INTECH Open Access Publisher, 2009.

[FK16] David Fries and William Kirkwood. *Non-Acoustic Sensors*, pages 423–440. Springer International Publishing, Cham, 2016.

[FKJL11] Maurice F Fallon, Michael Kaess, Hordur Johannsson, and John J Leonard. Efficient AUV navigation fusing acoustic ranging and side-scan sonar. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2398–2405. IEEE, 2011.

[FM83] Nick P Fofonoff and Robert C Millard. Algorithms for computation of fundamental properties of seawater. 1983.

[Fos11] Thor I Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[Fos12] Thor I Fossen. How to incorporate wind, waves and ocean currents in the marine craft equations of motion. *IFAC Proceedings Volumes*, 45(27):126–131, 2012.

[FP00] James Ferguson and Allan Pope. Explorer-a modular AUV for commercial site survey. In *Underwater Technology, 2000. UT 00. Proceedings of the 2000 International Symposium on*, pages 129–132. IEEE, 2000.

[FS89] Charles N Flagg and Sharon L Smith. On the use of the acoustic doppler current profiler to measure zooplankton abundance. *Deep Sea Research Part A. Oceanographic Research Papers*, 36(3):455–474, 1989.

[Fun09] Isao Funeno. Hydrodynamic optimal design of ducted azimuth thrusters. In *First International Symposium on Marine Propulsors smp*, volume 9, 2009.

[FXCZ16] Shuangshuang Fan, Wen Xu, Zheng Chen, and Fumin Zhang. Nonlinear observer design for current estimation based on underwater vehicle dynamic model. In *OCEANS 2016-Shanghai*, pages 1–5. IEEE, 2016.

[Gau02] Dip-Ing ETH Gustav Gautschi. Piezoelectric sensors. In *Piezoelectric Sensorics*, pages 73–91. Springer, 2002.

[GBRH16] Matthias Golz, Florin Boeck, Sebastian Ritz, and Gerd Holbach. A ballast system for automated deep-sea ascents. In *ASME 2016 35th International Conference on Ocean, Offshore and Arctic Engineering*, pages V007T06A028–V007T06A028. American Society of Mechanical Engineers, 2016.

[Gel15] Johannes Gelze. *Construction of a MEMS-based integrated navigation system for the use in a pressure-tolerant, deapsea-capable AUV*. dissertation, TU Berlin, 2015.

[Geo13] Hans-Otto Georgii. *Stochastics: introduction to probability and statistics*. Walter de Gruyter, 2013.

[Gol11] Herbert Goldstein. *Classical mechanics*. Pearson Education India, 2011.

[Gre11] Mohinder S Grewal. Kalman filtering. In *International Encyclopedia of Statistical Science*, pages 705–708. Springer, 2011.

[GTG+05] Sinan Gezici, Zhi Tian, Georgios B Giannakis, Hisashi Kobayashi, Andreas F Molisch, H Vincent Poor, and Zafer Sahinoglu. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *IEEE signal processing magazine*, 22(4):70–84, 2005.

[HB85] Sighard F Hoerner and Henry V Borst. *Fluid-dynamic lift: practical information on aerodynamic and hydrodynamic lift*. LA Hoerner, 1985.

[HBH08] Oyvind Hegrenas, Einar Berglund, and Oddvar Hallingstad. Model-aided inertial navigation for underwater vehicles. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1069–1076. IEEE, 2008.

[Heg10] Øyvind Hegrenæs. Autonomous navigation for underwater vehicles. 2010.

[Hen13] Norbert Henze. *Stochastik für Einsteiger: eine Einführung in die faszinierende Welt des Zufalls*. Springer-Verlag, 2013.

[HH11] Øyvind Hegrenaes and Oddvar Hallingstad. Model-aided INS with sea current estimation for robust underwater navigation. *IEEE Journal of Oceanic Engineering*, 36(2):316–337, 2011.

[HHG07] Øyvind Hegrenæs, Oddvar Hallingstad, and Kenneth Gade. Towards model-aided navigation of underwater vehicles. *Modeling, Identification and Control*, 28(4):113–123, 2007.

[HHJ07] Oyvind Hegrenaes, Oddvar Hallingstad, and Bjorn Jalving. Comparison of mathematical models for the HUGIN 4500 AUV based on experimental data. In *Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies, 2007. Symposium on*, pages 558–567. IEEE, 2007.

[HMH07] Per Espen Hagen, Oivind Midtgaard, and Oistein Hasvold. Making AUVs truly autonomous. In *OCEANS 2007*, pages 1–4. IEEE, 2007.

[Hoe65] Sighard F Hoerner. *Fluid-dynamic drag: practical information on aerodynamic drag and hydrodynamic resistance*. Hoerner Fluid Dynamics, 1965.

[Hol16] John J. Holmes. *Ocean Electromagnetics*, pages 177–196. Springer International Publishing, Cham, 2016.

[Hon05] Honeywell. Honeywell qa2000 accelerometer (data sheet). Technical report, Honeywell Aerospace, 2005.

[Hon08] Honeywell. Honeywell hmc1001 magnetometer (data sheet). Technical report, Honeywell Aerospace, 2008.

[Hon09] Honeywell. Honeywell gg1320an digital laser gyro (data sheet). Technical report, Honeywell Aerospace, 2009.

[Hoo76] FN Hooge. 1/f noise. *Physica B+ C*, 83(1):14–23, 1976.

[Hor05] Willi Hornfeld. Atlas auvs for offshore applications: Status and perspectives. In *ASME 2005 24th International Conference on Offshore Mechanics and Arctic Engineering*, pages 627–636. American Society of Mechanical Engineers, 2005.

[HSV+03] Per Espen Hagen, Nils Størkersen, Karstein Vestgård, Per Kartvedt, and Geir Sten. Operational military use of the HUGIN AUV in norway. *Proc. UDT Europe 2003*, pages 123–130, 2003.

[HTF11] Trevor J.. Hastie, Robert John Tibshirani, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2011.

[HW10] Songlai Han and Jinling Wang. A novel initial alignment scheme for low-cost INS aided by GPS for land vehicle applications. *The Journal of Navigation*, 63(4):663–680, 2010.

[Iml61] Frederick H Imlay. The complete expressions for added mass of a rigid body moving in an ideal fluid. Technical report, DAVID TAYLOR MODEL BASIN WASHINGTON DC, 1961.

[IOW] IOW: bio-physical interactions, workgroup of J.J. Waniek. `https://www.io-warnemuende.de/che-ag-bio-physical_interactions.html`. Accessed: 2018-02-21.

[Jac07] John David Jackson. *Classical electrodynamics*. John Wiley & Sons, 2007.

[Jan05] VINCENT M Janik. Underwater acoustic communication networks in marine mammals. *Animal communication networks*, pages 390–415, 2005.

[Jen94] Finn Bruun Jensen. *Computational ocean acoustics*. Springer Science & Business Media, 1994.

[JLK+17] Jongdae Jung, Yeongjun Lee, Donghoon Kim, Donghwa Lee, Hyun Myung, and Hyun-Taek Choi. AUV SLAM using forward/downward looking cameras and artificial landmarks. In *Underwater Technology (UT), 2017 IEEE*, pages 1–3. IEEE, 2017.

[JU97] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *AeroSense'97*, pages 182–193. International Society for Optics and Photonics, 1997.

[Jud01] Jack W Judy. Microelectromechanical systems (MEMS): fabrication, design and applications. *Smart materials and Structures*, 10(6):1115, 2001.

[JWP33] Eastman N Jacobs, Kenneth E Ward, and Robert M Pinkerton. The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel. 1933.

[K+60] Rudolph Emil Kalman et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[K+99] Jack B Kuipers et al. *Quaternions and rotation sequences*, volume 66. Princeton university press Princeton, 1999.

[Kar15] Evangelia A Karagianni. Electromagnetic waves under sea: Bow-tie antennas design for Wi-Fi underwater communications. *Progress In Electromagnetics Research M*, 41:189–198, 2015.

[KB61] Rudolph E Kalman and Richard S Bucy. New results in linear filtering and prediction theory. *Journal of basic engineering*, 83(1):95–108, 1961.

[KB02] Konstantin G Kebkal and Rudolf Bannasch. Sweep-spread carrier for underwater communication over acoustic channels with strong multipath propagation. *The Journal of the Acoustical Society of America*, 112(5):2043–2052, 2002.

[KBI99] Mark Koifman and IY Bar-Itzhack. Inertial navigation system aided by aircraft dynamics. *IEEE transactions on control systems technology*, 7(4):487–493, 1999.

[KC76] Charles Knapp and Glifford Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4):320–327, 1976.

[KC06] Jinhyun Kim and Wan Kyun Chung. Accurate and practical thruster modeling for underwater vehicles. *Ocean Engineering*, 33(5):566–586, 2006.

[KC13] Bharath Kalyan and Mandar Chitre. A feasibility analysis on using bathymetry for navigation of autonomous underwater vehicles. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 229–231. ACM, 2013.

[KD03] Jayesh H Kotecha and Petar M Djuric. Gaussian sum particle filtering. *IEEE Transactions on signal processing*, 51(10):2602–2612, 2003.

[KEW06] James C Kinsey, Ryan M Eustice, and Louis L Whitcomb. A survey of underwater vehicle navigation: Recent advances and new challenges. In *IFAC Conference of Manoeuvering and Control of Marine Craft*, volume 88, pages 1–12, 2006.

[Kim09] Jinhyun Kim. Thruster modeling and controller design for unmanned underwater vehicles (UUVs). In *Underwater Vehicles*. InTech, 2009.

[Kir07] WJ Kirkwood. Development of the DORADO mapping vehicle for multibeam, subbottom, and sidescan science missions. *Journal of Field Robotics*, 24(6):487–495, 2007.

[KKBY12] Konstantin G Kebkal, Oleksiy G Kebkal, Rudolf Bannasch, and Sergey G Yakovlev. Performance of a combined USBL positioning and communication system using S2C technology. In *2012 Oceans-Yeosu*, pages 1–7. IEEE, 2012.

[KKG+17] K Kebkal, O Kebkal, E Glushko, V Kebkal, L Sebastião, A Pascoal, J Gomes, J Ribeiro, H SIlva, M Ribeiro, et al. Underwater acoustic modems with integrated atomic clocks for one-way travel-time underwater vehicle positioning. In *Proceefings of the Underwater Acoustics Conference and Exhibition (UACE)*, 2017.

[KP04] Ho Sung Kim and Pakorn Plubrai. Manufacturing and failure mechanisms of syntactic foam under compressionâŸĘ. *Composites Part A: Applied Science and Manufacturing*, 35(9):1009–1015, 2004.

[KRD+15] M Kurowski, E Rentzow, D Dewitz, T Jeinsch, B Lampe, S Ritz, R Kutz, F Boeck, S Neumann, and D Oertel. Operational aspects of an ocean-going USV acting as communication node. In *Proceedings of the 14th Conference on Computer and IT Applications in the Maritime Industries (COMPIT15)*, pages 486–498, 2015.

[Lam32] Horace Lamb. *Hydrodynamics*. Cambridge university press, 1932.

[Lan99] Richard B Langley. Dilution of precision. *GPS world*, 10(5):52–59, 1999.

[LARP98] Thomas Dall Larsen, Nils A Andersen, Ole Ravn, and Niels Kjølstad Poulsen. Incorporation of time delayed measurements in a discrete-time kalman filter. In *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, volume 4, pages 3972–3977. IEEE, 1998.

[LB16] John J Leonard and Alexander Bahr. Autonomous underwater vehicle navigation. In *Springer Handbook of Ocean Engineering*, pages 341–358. Springer, 2016.

[LJ01] X Rong Li and Vesselin P Jilkov. Survey of maneuvering target tracking: Iii. measurement models. In *International Symposium on Optical Science and Technology*, pages 423–446. International Society for Optics and Photonics, 2001.

[LJ03] X Rong Li and Vesselin P Jilkov. Survey of maneuvering target tracking. part i. dynamic models. *IEEE Transactions on aerospace and electronic systems*, 39(4):1333–1364, 2003.

[Lju98] Lennart Ljung. System identification. In *Signal analysis and prediction*, pages 163–173. Springer, 1998.

[Lju10] Lennart Ljung. Perspectives on system identification. *Annual Reviews in Control*, 34(1):1–12, 2010.

[LLH+05] Chong-Moo Lee, Pan-Mook Lee, Seok-Won Hong, Sea-Moon Kim, Woojae Seong, et al. Underwater navigation system based on inertial sensor and doppler velocity log using indirect feedback kalman filter. *International Journal of Offshore and Polar Engineering*, 15(02), 2005.

[LM67] Ernest Bruce Lee and Lawrence Markus. Foundations of optimal control theory. Technical report, MINNESOTA UNIV MINNEAPOLIS CENTER FOR CONTROL SCIENCES, 1967.

[LM98] Dominique Legendre and Jacques Magnaudet. The lift force on a spherical bubble in a viscous linear shear flow. *Journal of Fluid Mechanics*, 368:81–126, 1998.

[LN06] Chang-Ho Lee and J Nicholas Newman. WAMIT user manual. *WAMIT, Inc*, 2006.

[LNP+14] Chee Sing Lee, Sharad Nagappa, Narcis Palomeras, Daniel E Clark, and Joaquim Salvi. SLAM with SC-PHD filters: An underwater vehicle application. *IEEE Robotics & Automation Magazine*, 21(2):38–45, 2014.

[Lüc10] Martin Lück. Aufbau druckneutraler, autonomer unterwasserfahrzeuge für die tiefsee. 2010.

[MAT17] MATLAB. *version 9.2.0.556344 (R2017a)*. The MathWorks Inc., Natick, Massachusetts, 2017.

[May82] Peter S Maybeck. *Stochastic models, estimation, and control*, volume 3. Academic press, 1982.

[MCL08] Jeremy M Marschke, John L Crassidis, and Quang M Lam. Multiple model adaptive estimation for inertial navigation during mars entry. *AIAA Paper*, 7352, 2008.

[McN14] Marcia McNutt. The hunt for MH370. *Science*, 344(6187):947–947, 2014.

[McP09] Stephen McPhail. Autosub6000: A deep diving long range AUV. *Journal of Bionic Engineering*, 6(1):55–62, 2009.

[ME06] Pratap Misra and Per Enge. *Global Positioning System: Signals, Measurements and Performance Second Edition*. Lincoln, MA: Ganga-Jamuna Press, 2006.

[mex] Introduction to MEX files. `https://www.mathworks.com/help/matlab/matlab_external/introducing-mex-files.html`. Accessed: 2017-11-06.

[MGJ01] Magne Mandt, Kenneth Gade, and Bjørn Jalving. Integrating DGPS-USBL position measurements with inertial navigation in the HUGIN 3000 AUV. In *Proceedings of the 8th Saint Petersburg International Conference on Integrated Navigation Systems, Saint Petersburg, Russia*, pages 28–30, 2001.

[MHS+15] Alain Martinez, Luis Hernandez, Hichem Sahli, Yunier Valeriano-Medina, Maykel Orozco-Monteagudo, and Delvis Garcia-Garcia. Model-aided navigation with sea current estimation for an autonomous underwater vehicle. *International Journal of Advanced Robotic Systems*, 12(7):103, 2015.

[Mil69] D Michael Milder. Ray and wave invariants for SOFAR channel propagation. *The Journal of the Acoustical Society of America*, 46(5B):1259–1263, 1969.

[MK91] SJ Merhav and M Koifman. Autonomously aided strapdown attitude reference system. *Journal of Guidance, Control, and Dynamics*, 14(6):1164–1172, 1991.

[MLA01] S Grewal Mohinder, R Weill Lawrence, and P Andrews Angus. Global positioning systems, inertial navigation, and integration. *NY: John Wiley & Sons, Inc Publication*, 2001.

[Moo67] Richard K Moore. Radio communication in the sea. *IEEE spectrum*, 4(11):42–51, 1967.

[MOS09] Marco Morgado, P Oliveira, and Carlos Silvestre. Posterior cramér-rao bounds analysis for INS/USBL navigation systems. *IFAC Proceedings Volumes*, 42(18):20–25, 2009.

[MOS10] Marco Morgado, Paulo Oliveira, and Carlos Silvestre. Experimental evaluation of a usbl underwater positioning system. In *ELMAR, 2010 proceedings*, pages 485–488. IEEE, 2010.

[MOS13] Marco Morgado, Paulo Oliveira, and Carlos Silvestre. Tightly coupled ultrashort baseline and inertial navigation system for underwater vehicles: An experimental validation. *Journal of Field Robotics*, 30(1):142–170, 2013.

[MOSV06] Marco Morgado, Paulo Oliveira, Carlos Silvestre, and José Fernandes Vasconcelos. USBL/INS tightly-coupled integration technique for underwater vehicles. In *2006 9th International Conference on Information Fusion*, pages 1–8. IEEE, 2006.

[MOSV07] M Morgado, P Oliveira, C Silvestre, and JF Vasconcelos. Vehicle dynamics aiding technique for USBL/INS underwater navigation system. *IFAC Proceedings Volumes*, 40(17):111–116, 2007.

[MOSV14] Marco Morgado, Paulo Oliveira, Carlos Silvestre, and José Fernandes Vasconcelos. Embedded vehicle dynamics aiding for USBL/INS underwater navigation system. *IEEE Transactions on Control Systems Technology*, 22(1):322–330, 2014.

[MPD+14] David Meyer, Ralf D Prien, Olaf Dellwig, Joanna J Waniek, and Detlef E Schulz-Bull. Electrode measurements of the oxidation reduction potential in the gotland deep using a moored profiling instrumentation. *Estuarine, Coastal and Shelf Science*, 141:26–36, 2014.

[MTB85] MJP Muringer, NJ Trappeniers, and SN Biswas. The effect of pressure on the sound velocity and density of toluene and n-heptane up to 2600 bar. *Physics and Chemistry of Liquids an International Journal*, 14(4):273–296, 1985.

[MVKS04] Roar Marthiniussen, K Vestgard, Rolf Arne Klepaker, and N Storkersen. HUGIN-AUV concept and operational experiences to date. In *OCEANS'04. MTTS/IEEE TECHNO-OCEAN'04*, volume 2, pages 846–850. IEEE, 2004.

[MW14] Stephen C Martin and Louis L Whitcomb. Experimental identification of six-degree-of-freedom coupled dynamic plant models for underwater robot vehicles. *IEEE Journal of Oceanic Engineering*, 39(4):662–671, 2014.

[MW16] Stephen C Martin and Louis L Whitcomb. Fully actuated model-based control with six-degree-of-freedom coupled dynamical plant models for underwater vehicles: Theory and experimental evaluation. *The International Journal of Robotics Research*, 35(10):1164–1184, 2016.

[MWP+16] Lashika Medagoda, Stefan B Williams, Oscar Pizarro, James C Kinsey, and Michael V Jakuba. Mid-water current aided localization for autonomous underwater vehicles. *Autonomous Robots*, 40(7):1207–1227, 2016.

[Nel13] Oliver Nelles. *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*. Springer Science & Business Media, 2013.

[Neu18] Sergej Neumann. *(PhD thesis in progress, working title): Cooperative Localization in Mobile Underwater Acoustic Sensor Networks*. 2018.

[Nis15] Makiko Nisio. Stochastic differential equations. In *Stochastic Control Theory*, pages 1–30. Springer, 2015.

[NLR05] Paul M Newman, John J Leonard, and Richard J Rikoski. Towards constant-time SLAM on an autonomous underwater vehicle using synthetic aperture sonar. *Robotics Research*, pages 409–420, 2005.

[NOW+15] SERGEJ NEUMANN, DAVID OERTEL, HEINZ WÖRN, MARTIN KUROWSKI, DETLEF DEWITZ, JOANNA J WANIEK, DAVID KAISER, and ROBERT MARS. Towards deep-sea monitoring with smis-experimental trials of deep-sea acoustic localization. In *submitted to 18th International Conference on Climbing and Walking Robots*. World Scientific, 2015.

[NW13] Li Ning and Ding Wei. Model-aided strapdown inertial navigation integrated method for AUV based on H8 filtering. In *Computational and Information Sciences (ICCIS), 2013 Fifth International Conference on*, pages 1088–1092. IEEE, 2013.

[oal] Ocean acoustics library. `http://oalib.hlsresearch.com/`. Accessed: 2016-05-05.

[Ole13] Eugen Olenew. Entwicklung von manövrier-und steuereinrichtungen für ein druckneutrales unterwasserfahrzeug. 2013.

[ONW+16] D Oertel, S Neumann, H Worn, M Golz, and JJ Waniek. Reducing elevation angle errors of long-range deep-sea acoustic localization by ray tracing and depth measurements. In *Advanced Intelligent Mechatronics (AIM), 2016 IEEE International Conference on*, pages 178–183. IEEE, 2016.

[Ozy12] Talat Ozyagcilar. Calibrating an ecompass in the presence of hard and soft-iron interference. *Freescale Semiconductor Ltd*, 2012.

[PGP+11] Michael Purcell, Dave Gallo, Greg Packard, Mark Dennett, Marcel Rothenbeck, Andy Sherrell, and Sylvain Pascaud. Use of REMUS 6000 AUVs in the search for the air france flight 447. In *OCEANS 2011*, pages 1–7. IEEE, 2011.

[PJS09] Luca Pivano, Tor Arne Johansen, and Øyvind N Smogeli. A four-quadrant thrust estimation scheme for marine propellers: Theory and experiments. *IEEE Transactions on Control Systems Technology*, 17(1):215–226, 2009.

[PKL07] Jim Partan, Jim Kurose, and Brian Neil Levine. A survey of practical issues in underwater networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(4):23–33, 2007.

[PMdSR16] Breno C Pinheiro, Ubirajara F Moreno, Joao TB de Sousa, and Orlando C Rodríguez. Kernel-function-based models for acoustic localization of underwater vehicles. *IEEE Journal of Oceanic Engineering*, 2016.

[PMdSR17] Breno C Pinheiro, Ubirajara F Moreno, Joao TB de Sousa, and Orlando C Rodríguez. Kernel-function-based models for acoustic localization of underwater vehicles. *IEEE Journal of Oceanic Engineering*, 42(3):603–618, 2017.

[PPFS12] Mario Prats, Javier Pérez, J Javier Fernández, and Pedro J Sanz. An open source tool for simulation and supervision of underwater intervention missions. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2577–2582. IEEE, 2012.

[Pre01a] Timothy Prestero. Development of a six-degree of freedom simulation model for the REMUS autonomous underwater vehicle. In *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, volume 1, pages 450–455. IEEE, 2001.

[Pre01b] Timothy Timothy Jason Prestero. *Verification of a six-degree of freedom simulation model for the REMUS autonomous underwater vehicle*. PhD thesis, Massachusetts institute of technology, 2001.

[PS81] Louis Prieur and Shubha Sathyendranath. An optical classification of coastal and oceanic waters based on the specific spectral absorption curves of phytoplankton pigments, dissolved organic matter, and other particulate materials. *Limnology and Oceanography*, 26(4):671–689, 1981.

[psd] The power spectral density and the autocorrelation (course notes). `https://personal.egr.uri.edu/chelidz/courses/mce567/handouts/psdtheory.pdf`. Accessed: 2016-12-31.

[PSSL14] Liam Paull, Sajad Saeedi, Mae Seto, and Howard Li. AUV navigation and localization: A review. *IEEE Journal of Oceanic Engineering*, 39(1):131–149, 2014.

[QCG+09] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. ROS: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

[RDK+15] Erik Rentzow, Detlef Dewitz, Martin Kurowski, Bernhard P Lampe, Sebastian Ritz, Robert Kutz, Matthias Golz, and Florin Boeck. Design and automation of an ocean-going autonomously acting USV. In *OCEANS 2015-Genova*, pages 1–6. IEEE, 2015.

[RHBG14] S. Ritz, G. Holbach, F. Boeck, and M. Golz. Challenges and specialties in design of an unmanned surface vehicle for hydrographic tasks. *Jahrbuch der Schiffbautechnischen Gesellschaft*, Volume 108:in Process, 2014.

[RKD+15] Erik Rentzow, Martin Kurowski, Detlef Dewitz, Bernhard P Lampe, and Joanna J Waniek. Temperature gradient following method for box-shaped AUV. *IFAC-PapersOnLine*, 48(16):37–42, 2015.

[RMB+16] Joel Reis, Marco Morgado, Pedro Batista, Paulo Oliveira, and Carlos Silvestre. Design and experimental validation of a USBL underwater acoustic positioning system. *Sensors*, 16(9):1491, 2016.

[Roo77] J Rooney. A survey of representations of spatial rotation about a fixed point. *Environment and Planning B: Planning and Design*, 4(2):185–210, 1977.

[RPW06] Paul Rigby, Oscar Pizarro, and Stefan B Williams. Towards geo-referenced AUV navigation through fusion of USBL and DVL measurements. In *OCEANS 2006*, pages 1–6. IEEE, 2006.

[RSS+13] Michailas Romanovas, Tobias Schwarze, Manuel Schwaab, Martin Traechtler, and Yiannos Manoli. Stochastic cloning kalman filter for visual odometry and inertial/magnetic data fusion. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 1434–1441. IEEE, 2013.

[RTB+15] Sebastian Ritz, Carl Thiede, Florin Boeck, Matthias Golz, and Gerd Holbach. Challenges in design and automation of autonomous waterborne vehicles. *at-Automatisierungstechnik*, 63(5):389–399, 2015.

[RTS+65] Herbert E Rauch, F Tung, Charlotte T Striebel, et al. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965.

[Sab06] Angelo M Sabatini. Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing. *IEEE Transactions on Biomedical Engineering*, 53(7):1346–1356, 2006.

[Sav98a] Paul G Savage. Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms. *Journal of guidance, control, and dynamics*, 21(1):19–28, 1998.

[Sav98b] Paul G Savage. Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms. *Journal of Guidance, Control, and Dynamics*, 21(2):208–221, 1998.

[SB98] Ivan W Selesnick and C Sidney Burrus. Generalized digital butterworth filter design. *IEEE Transactions on signal processing*, 46(6):1688–1694, 1998.

[SB16] Milica Stojanovic and Pierre-Philippe J. Beaujean. *Acoustic Communication*, pages 359–386. Springer International Publishing, Cham, 2016.

[SFMC88] Mrinal K Sen, LN Frazer, S Mallick, and NR Chapman. Analysis of multipath sound propagation in the ocean near 49 n, 128 w. *The Journal of the Acoustical Society of America*, 83(2):588–597, 1988.

[SH13] Jannik Steinbring and Uwe D Hanebeck. S2KF: The smart sampling kalman filter. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 2089–2096. IEEE, 2013.

[SH14] Jannik Steinbring and Uwe D Hanebeck. LRKF revisited: The smart sampling kalman filter (S2KF). *Journal of Advances in Information Fusion*, 9(2):106–123, 2014.

[Sko03] Sigurd Skogestad. Simple analytic rules for model reduction and PID controller tuning. *Journal of process control*, 13(4):291–309, 2003.

[ŠKT01] Miroslav Šimandl, Jakub Královec, and Petr Tichavskỳ. Filtering, predictive, and smoothing cramér–rao bounds for discrete-time nonlinear dynamic systems. *Automatica*, 37(11):1703–1716, 2001.

[smi] Subsea monitoring via intelligent swarms (SMIS) project homepage. `http://www.smis-monitoring.de`. Accessed: 2016-03-10.

[Som97] Dan M Somers. Design and experimental results for the S809 airfoil. Technical report, National Renewable Energy Lab., Golden, CO (United States), 1997.

[SP07] C Silvestre and A Pascoal. Depth control of the INFANTE AUV using gain-scheduled reduced order output feedback. *Control Engineering Practice*, 15(7):883–895, 2007.

[SP12] Hwi-Beom Shin and Jong-Gyu Park. Anti-windup PID controller with integral state predictor for variable-speed motor drives. *IEEE Transactions on Industrial Electronics*, 59(3):1509–1516, 2012.

[SPH15] Jannik Steinbring, Martin Pander, and Uwe D. Hanebeck. The smart sampling kalman filter with symmetric samples. *CoRR*, abs/1506.03254, 2015.

[SSF04] Roger Skjetne, Øyvind N Smogeli, and Thor I Fossen. A nonlinear ship manoeuvering model: Identification and adaptive control with experiments for a model ship. *Modeling, Identification and control*, 25(1):3, 2004.

[Sta10] M Jordan Stanway. Delayed-state sigma point kalman filters for underwater navigation. In *Autonomous Underwater Vehicles (AUV), 2010 IEEE/OES*, pages 1–9. IEEE, 2010.

[Sto03] Walter Stockwell. Angle random walk. *Application Note. Crossbow Technologies Inc*, pages 1–4, 2003.

[Tar05] Albert Tarantola. *Inverse problem theory and methods for model parameter estimation*. SIAM, 2005.

[TBF05] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[TDSW11] Hwee-Pink Tan, Roee Diamant, Winston KG Seah, and Marc Waldmeyer. A survey of techniques and challenges in underwater localization. *Ocean Engineering*, 38(14):1663–1676, 2011.

[Teh83] MM Tehrani. Ring laser gyro data analysis with cluster sampling technique. In *1983 Technical Symposium East*, pages 207–220. International Society for Optics and Photonics, 1983.

[tel] Teledyne, company homepage, hydrophone products. `http://www.teledyne-reson.com/product-category/hydrophones/`. Accessed: 2017-07-21.

[Thi11] Carl Thiede. Druckneutrale sensoren und aktoren für den dauereinsatz in tiefseefahrzeugen. 2011.

[THZG10] Nima Harsamizadeh Tehrani, Mahdi Heidari, Yadollah Zakeri, and Jafar Ghaisari. Development, depth control and stability analysis of an underwater remotely operated vehicle (ROV). In *Control and Automation (ICCA), 2010 8th IEEE International Conference on*, pages 814–819. IEEE, 2010.

[TMN98] Petr Tichavsky, Carlos H Muravchik, and Arye Nehorai. Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on signal processing*, 46(5):1386–1396, 1998.

[Tuz02] Vyacheslav Tuzlukov. *Signal processing noise*, volume 8. CRC Press, 2002.

[UG09] Carlos Uribe and Walter Grote. Radio communication model for underwater WSN. In *2009 3rd International Conference on New Technologies, Mobility and Security*, pages 1–5. IEEE, 2009.

[UTA+07] Tamaki Ura, Kensaku Tamaki, Akira Asada, Kei Okamura, Kenji Nagahashi, Takashi Sakamaki, Toshitaka Gamo, Kyoko Okino, Takashi Obara, Kenji Nakane, et al. Dives of AUV r2D4 to rift valley of central indian mid-ocean ridge system. In *OCEANS 2007-Europe*, pages 1–6. IEEE, 2007.

[VBLS14] A Vydhyanathan, G Bellusci, H Luinge, and P Slycke. The next generation xsens motion trackers for industrial applications, 2014.

[Ver09] JHAM Vervoort. *Modeling and control of an unmanned underwater vehicle*. PhD thesis, PhD thesis, University of Canterbury-University of Technology Eindhoven, 2009.

[Vic98] K. Vickery. Acoustic positioning systems. a practical overview of current systems. In *Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles*, pages 5–17, Cambridge, MA, USA, 1998.

[VMMH+13] Yunier Valeriano-Medina, A Martinez, L Hernández, H Sahli, Y Rodríguez, and Julio R Cañizares. Dynamic model for an autonomous underwater vehicle based on experimental data. *Mathematical and Computer Modelling of Dynamical Systems*, 19(2):175–200, 2013.

[VSOG10] JF Vasconcelos, C Silvestre, P Oliveira, and B Guerreiro. Embedded UAV model and LASER aiding techniques for inertial navigation systems. *Control Engineering Practice*, 18(3):262–278, 2010.

[Wen11] J. Wendel. *Integrierte Navigationssysteme: Sensordatenfusion, GPS und inertiale Navigation*. Oldenbourg, 2011.

[WHK17] Bilal Wehbe, Marc Hildebrandt, and Frank Kirchner. Experimental evaluation of various machine learning regression methods for model identification of autonomous underwater vehicles. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 4885–4890. IEEE, 2017.

[Woo50] Max A Woodbury. Inverting modified matrices. *Memorandum report*, 42(106):336, 1950.

[wor] Teledyne RD instruments workhorse navigator DVL. `https://www.usna.edu/AUVT/_files/documents/dvl.pdf`. Accessed: 2017-01-31.

[WSY13] Pengcheng Wang, Pratul K Singh, and Jingang Yi. Dynamic model-aided localization of underwater autonomous gliders. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5565–5570. IEEE, 2013.

[WY99] Louis L Whitcomb and Dana R Yoerger. Development, comparison, and preliminary experimental validation of nonlinear dynamic thruster models. *IEEE journal of oceanic engineering*, 24(4):481–494, 1999.

[WYS99] Louis Whitcomb, Dana Yoerger, and Hanumant Singh. Advances in doppler-based navigation of underwater robotic vehicles. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 1, pages 399–406. IEEE, 1999.

[XLJ15] Xianbo Xiang, Lionel Lapierre, and Bruno Jouvencel. Smooth transition of AUV motion control: From fully-actuated to under-actuated configuration. *Robotics and Autonomous Systems*, 67:14–22, 2015.

[xse] Mti 100-series data sheet. `https://www.xsens.com/download/pdf/documentation/mti-100/mti-100-series.pdf`. Accessed: 2017-06-26.

[XWD+16] Guangdi Xiao, Bo Wang, Zhihong Deng, Mengyin Fu, and Yun Ling. An acoustic communication time delays compensation approach for master-slave AUV cooperative navigation. *IEEE Sensors Journal*, 17(2):504–513, 2016.

[XZYC13] Feng Xu, Zao-Jian Zou, Jian-Chuan Yin, and Jian Cao. Identification modeling of underwater vehicles' nonlinear dynamics based on support vector machines. *Ocean Engineering*, 67:68–76, 2013.

[YCI+98] J Yuh, SK Choi, C Ikehara, GH Kim, G McMurty, M Ghasemi-Nejhad, N Sarkar, and K Sugihara. Design of a semi-autonomous underwater vehicle for intervention missions (SAUVIM). In *Underwater Technology, 1998. Proceedings of the 1998 International Symposium on*, pages 63–68. IEEE, 1998.

[YHKA96] Jari Yli-Hietanen, Kari Kalliojarvi, and Jaakko Astola. Low-complexity angle of arrival estimation of wideband signals using small arrays. In *Statistical Signal and Array Processing, 1996. Proceedings., 8th IEEE Signal Processing Workshop on (Cat. No. 96TB10004*, pages 109–112. IEEE, 1996.

[YJBB07] Dana R Yoerger, Michael Jakuba, Albert M Bradley, and Brian Bingham. Techniques for deep sea near bottom survey using an autonomous underwater vehicle. *The International Journal of Robotics Research*, 26(1):41–54, 2007.

[YSe12] YSens. XSens MTi-100-series (user manual). Technical report, XSens Technologies B.V., 2012.

[YSY+17] AM Yazdani, K Sammut, OA Yakimenko, A Lammas, Y Tang, and S Mahmoud Zadeh. IDVD-based trajectory generator for autonomous underwater docking operations. *Robotics and Autonomous Systems*, 92:12–29, 2017.

[YW01] Junku Yuh and Michael West. Underwater robotics. *Advanced Robotics*, 15(5):609–639, 2001.

[ZC03] Silvia M Zanoli and Giuseppe Conte. Remotely operated vehicle depth control. *Control engineering practice*, 11(4):453–459, 2003.

[Zha06] Fuzhen Zhang. *The Schur complement and its applications*, volume 4. Springer Science & Business Media, 2006.

[ZND+10] Pascal Zwahlen, Anne-Marie Nguyen, Yufeng Dong, Felix Rudolf, Marc Pastre, and Hanspeter Schmid. Navigation grade MEMS accelerometer. In *Micro Electro Mechanical Systems (MEMS), 2010 IEEE 23rd International Conference on*, pages 631–634. IEEE, 2010.

[ZQF16] Lin Zhao, Haiyang Qiu, and Yanming Feng. Analysis of a robust kalman filter in loosely coupled GPS/INS navigation system. *Measurement*, 80:138–147, 2016.

[ZSC+16] Tao Zhang, Hongfei Shi, Liping Chen, Yao Li, and Jinwu Tong. AUV positioning method based on tightly coupled SINS/LBL for underwater acoustic multipath propagation. *Sensors*, 16(3):357, 2016.