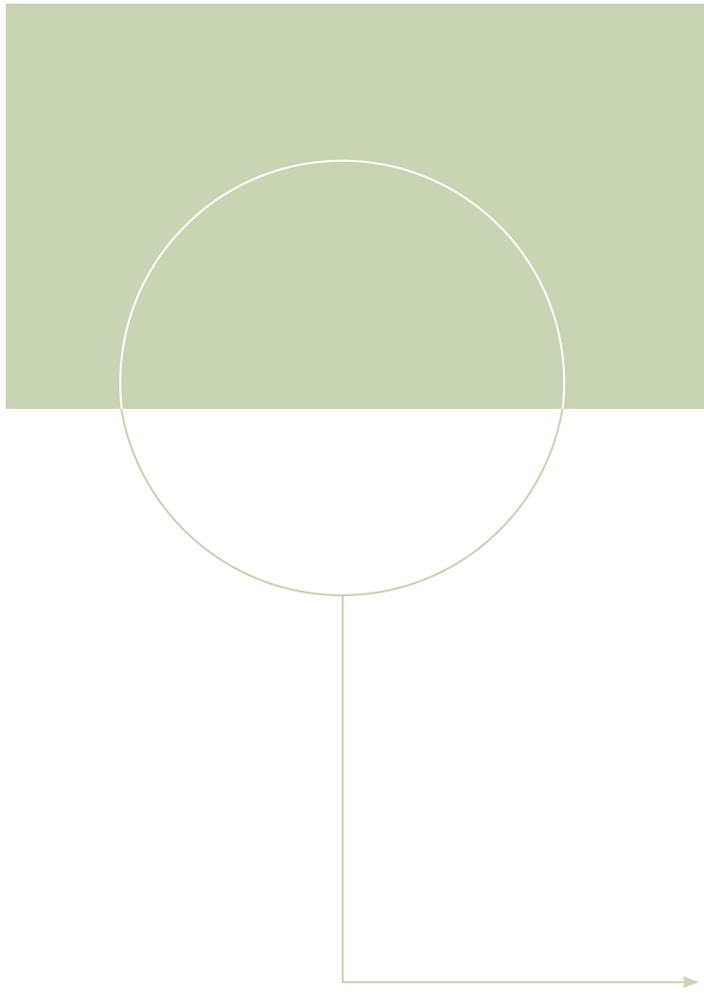


Master's thesis

2022

Bjørnar Reitan Hogstad

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

Bjørnar Reitan Hogstad

Side-Scan Sonar Imaging and Error-State Kalman Filter Aiding Unmanned Underwater Vehicle (UUV) to Autonomy

Through an error-state Kalman filter and an image construction pipeline, this thesis aims to develop the foundation for an easily deployable UUV.

July 2022

NTNU
Norwegian University of Science and Technology

NTNU
Norwegian University of Science and Technology

NTNU

Norwegian University of
Science and Technology

Side-Scan Sonar Imaging and Error-State Kalman Filter Aiding Unmanned Underwater Vehicle (UUV) to Autonomy

Through an error-state Kalman filter and an image construction pipeline, this thesis aims to develop the foundation for an easily deployable UUV.

Bjørnar Reitan Hogstad

Cybernetics and Robotics
Submission date: July 2022
Supervisor: Damiano Varagnolo
Co-supervisor: Simon Andreas Hagen Hoff

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Preface

This master's thesis is done as a part of the Masters of Science program in Cybernetics and Robotics at the Norwegian University of Science and Technology (NTNU). The work has been done under the supervision of Damiano Varagnolo and Simon Andreas Hagen Hoff. The code developed is available at GitHub[1].

First, I would like to express my gratitude to Simon; thanks for your patience, your thirst for coffee, and the countless hours spent both on and off track. It's been a blast! Further, I would like to thank Damiano for his enthusiasm and support, and Erlend A. Basso for his magic fingers. The help has been greatly appreciated. Finally, big thanks to friends and family for their support.

> *Working with hardware is as fun as it is painful, and I have had shedloads of fun.*[2]

[1] https://github.com/bjornrho/Navigation-brov2

[2] See Appendix A for alternative preface remarks.

Abstract

Surprisingly unexplored, and ripe with possibilities; the ocean is a key cog in Earth's planetary clockwork. The field of robotics is gradually extending to underwater environments along with technological advancements, allowing the ocean's potential to be harnessed.

This project aims to develop the foundation for an easily deployable Unmanned Underwater Vehicle (UUV) configuration for the purpose of simplifying and encouraging field testing, ultimately enabling research within underwater multi agent operations. To navigate underwater environments is one of the main challenges when transitioning into autonomous underwater operations.

In this project, a navigational scheme consisting of an error-state Kalman filter (ESKF) is implemented and tested. Experimental results show drift in position and heading, as is expected when using only interoceptive sensors, which leads to an unbounded growth in the uncertainty of the estimates. Performing detection in SSS images can help limit the estimate error. However, basic SSS images suffer from intensity variations and geometric distortions, complicating the use of computer vision methods. To address this, an image construction pipeline for SSS performing echo intensity normalization, slant range correction, blind zone removal, and geometric correction, is implemented. The pipeline is found to construct acoustic images which more accurately represent the sea floor, accounting for variations and distortions. This means that detection for the purpose of limiting navigational error can be performed on the resulting acoustic images.

Further research should focus on the development and implementation of a detector to use on processed acoustic images. Doing so, the complete system should be re-tuned and re-evaluated to accommodate the detector's performance.

Sammendrag

Havet er en essensiell brikke i jordens komplekse puslespill, bemerkelsesverdig uoppdaget og fullt av muligheter. Hånd i hånd med teknologisk utvikling er robotikk-feltet i ferd med å bokstavelig talt ta seg vann over hodet, slik at potensialet under vannoverflaten kan utnyttes. Formålet med denne oppgaven er å danne grunnlaget for en ubemannet undervannsfarkost som enkelt kan benyttes i forskningsøyemed. Det vil bidra til å forenkle og muliggjøre felteksperiment for å bane vei for multiagent undervannsforskning. Navigasjon under vann er én av hovedutfordringene knyttet til autonome undervannsoperasjoner.

I forbindelse med dette prosjektet har en navigasjonsløsning bestående av et error-state Kalman filter (ESKF) blitt implementert og evaluert. Eksperimentelle forsøk viser forflytting i tilstandsestimatene av farkostens posisjon og bevegelsesretning, noe som er forventet når kun interoseptive sensorer benyttes. Dette medfører en ubegrenset vekst i usikkerhet knyttet til estimatene. Veksten kan begrenses ved å utføre deteksjon i akustiske bilder fra en sidesøkende sonar, men bruk av deteksjonsmetoder er på ingen måte uproblematisk grunnet intensitetsvariasjoner og geometriske forvrengninger. I denne oppgaven implementeres en sekvens av prosesseringssteg bestående av intensitetsnormalisering, avstandskorreksjon, eliminering av blindsone, og geometrisk korreksjon. De ferdigprosesserte akustiske bildene framstår som mer presise representasjoner av havbunnen. Derfor er bildene bedre egnet for deteksjon i navigasjonsøyemed.

Videre arbeid bør omfatte implementasjon av en detektor for prosesserte akustiske bilder. Deretter bør hele systemet justeres og reevalueres i kontekst av detektorens ytelse.

Contents

Acronyms

AUV Autonomous Underwater Vehicle. 33, 75

CV Coefficient of Variation. 79, 80

DCP Dark Channel Prior. 69

DOF Degrees Of Freedom. 5, 6, 9, 10, 18, 21, 33

DVL Doppler Velocity Log. vii, 2, 19, 29–37, 40, 42, 45, 46, 52, 71, 78, 84, 86, 92, 94, 103

ECEF Earth-Centered Earth-Fixed. 6, 7

ECI Earth-centered inertial. 6, 7

EKF extended Kalman filter. 23, 24

EMI electromagnetic interference. 18, 37, 42

EnKF ensemble Kalman filter. 23, 24

ENU East, North, Up. 7, 36

ESKF error-state Kalman filter. iv–vii, 2, 3, 17, 20, 24–26, 28, 29, 31, 32, 34, 36–38, 41–53, 75, 78, 87, 91–94, 103

FBR First Bottom Return. 70, 71, 84, 85, 92

GNSS Global Navigation Satellite Systems. 1, 2

GPS Global Positioning System. 1, 2, 7, 18, 43, 75, 103

I2C Inter-Integrated Circuit. 33

IEKF iterated extended Kalman filter. 23, 24

IMU Inertial Measurement Unit. vii, 2, 18, 25–27, 29–31, 33–36, 42, 44, 45, 52, 92, 103

KNN K-Nearest Neighbor. 76, 77, 88, 89, 93

MIRA MIxed exponential Regression Analysis. 69, 70

NED North, East, Down. 7, 36, 37

NEES normalized estimation error squared. 39

NIS normalized innovation squared. 39, 48, 49, 51

PDF Probability Density Function. 16, 20, 21, 23, 24

RADAR RAdio Detection And Ranging. 18

RK Runge-Kutta. 15

ROV Remotely Operated Vehicle. 33, 41, 42, 44, 45, 65, 79, 86, 88, 92, 94

RPI Raspberry PI. vii, 33, 41, 52, 53, 78, 91

SLAM Simultaneous Localization And Mapping. 2, 75

SONAR SOund Navigation And Ranging. 61

SSS Side-Scan Sonar. iv, vii, 2, 3, 18, 41, 60–67, 71, 94, 103

TOF Time Of Flight. 60, 62

TVG Time Variable Gain. 69

UKF unscented Kalman filter. 23, 24

UUV Unmanned Underwater Vehicle. iv, 1, 2, 33, 72, 75

Part I

The secret to getting ahead is getting started.

Chapter 1

Introduction

1.1 Motivation

The oceans cover more than two-thirds of the Earth's surface, consequently playing an essential role when assessing solutions to tough challenges such as climate change and increasing population density. As all scientific, economic, political, and social decisions at some point depend on knowledge of the Earth, the importance of technology enabling underwater exploration and operations can not be understated.

Along with technological advancements, the field of robotics has extended to underwater environments through development of the Unmanned Underwater Vehicle (UUV). Set to perform surveying tasks such as sea floor mapping [1], ship hull inspections [2], environmental monitoring [3], and search operations [4], the UUVs have a wide field of application. Further, UUVs help reduce time, economic costs, and risk related to underwater tasks, making them prominent tools.

To reliably perform these tasks, UUVs are fundamentally dependent on robust and accurate navigation. Navigating underwater environments is challenging due to the attenuation of radio signals, severely restricting the availability of Global Positioning System (GPS) or other Global Navigation Satellite Systems (GNSS). Commonly, UUV navigation systems are based on dead reckoning and the use of inertial navigation systems, integrating accelerations and velocities to obtain position and orientation estimates [5]. Better accuracy can be obtained by fusing measurements from other sensors using state estimation solutions such as particle filters, high-gain observers, or one of many flavors of the Kalman filter.

However, all these solutions suffer from unbounded growth in estimate uncertainty due to the integration of noisy and biased measurements. The growth can be limited through the time and power consuming act of

resurfacing for GNSS updates. Doing so may be unfeasible for operations in confined spaces such as under ice [6]. Acoustic range methods can be employed to alleviate resurfacing, where the deployment of infrastructure such as long baseline and ultra-short baseline provides accurate position estimates of the UUV [7]. The main disadvantages of such methods are the time and cost related to deployment, the need for support vessels, and the limited range. Thus, acoustic range methods are well suited for operations in restricted areas like aquaculture and fish farming [8].

The inherent drift may be addressed through the use of Simultaneous Localization And Mapping (SLAM) solutions. SLAM aims to construct a model of the environment, while simultaneously locating the vehicle within it. The map allows the limiting of estimate uncertainty through relating location to previously observed parts of the environment [9]. Moreover, the ability to recognize and distinguish landmarks in the environment is a pivotal prerequisite.

Underwater images are known to be degraded due to factors such as light attenuation, lens distortion, and suspended particles, which is why acoustic sensors such as sonars are often preferred. Sonar technology has been around for over a century, and a wide variety of sonars are available; forward facing, synthetic aperture, and Side-Scan Sonar (SSS) to mention a few. The SSS is capable of efficiently ensonifying large areas of the sea floor [10]. Fusing SSS data for navigational purposes is done in [11, 12] with reduction in performance due to infrequent observation of targets. The construction of acoustic images which accurately represents the sea floor is imperative to enable proper detection. To this extent, processing of SSS data is required to address inherent intensity variations and geometric distortion.

In this thesis, an error-state Kalman filter (ESKF) based on Solà's formulation [13] and an image construction pipeline performing echo intensity normalization, blind zone removal, slant range correction, and geometric correction on SSS data has been implemented, tested, and evaluated. In addition to the ESKF and the pipeline, this thesis has made contributions to the foundation of an easily deployable Unmanned Underwater Vehicle (UUV) configuration through developing, modifying, and unifying[1]

- sensor drivers (IMU, DVL, pressure sensor, SSS, GPS),
- a hysteretic controller, a PD controller, a teleoperation solution,
- a trajectory planner, a trajectory publisher,
- a text user interface,

and automating start-up procedures using Tmuxinator.

[1] See Appendix B for a distinction of the developed, modified, and unified software.

1.2 Outline

This thesis covers the two partially disconnected subjects of state estimation and processing of sonar data for the purpose of connecting them in future work. Due to the fundamental differences between these subjects, it was beneficial to structure the thesis into the four following parts:

- **Part I:** Introducing and contextualizing the work, describing contributions, and outlining the thesis (this part).
- **Part II:** Presenting state estimation related theory, the hardware configuration, the error-state Kalman filter (ESKF) implementation, results from testing, and evaluation of the ESKF performance.
- **Part III:** Presenting relevant acoustic theory, the image construction pipeline for Side-Scan Sonar (SSS) data, results from testing, and evaluation of the pipeline performance.
- **Part IV:** Presenting suggested improvements and the conclusion of this work.

Although the thesis does not conform to the classic IMRAD-structure, an effort has been made to keep this structure within the separate parts. Consequently, separate parts can be read individually if the reader wishes to do so.

Part II

Where Are We Now?

Chapter 2

Primer on State Representation

Our ability to navigate surroundings greatly affects how we live life on planet earth and perform the tasks we are set to do. Much like humans, this fact also concerns robotics which inherently deals with things that move in the world. Within mobile robotics, navigation is to a great extent about estimating the state of a vehicle (e.g., robots, satellites, aircraft). The state is a set of quantities that seek to describe a vehicle's motion over time, and the act of estimating it is referred to as *state estimation*.

Reminiscent of how humans use various senses to obtain information about their surroundings, robots depend on various sensors with limited precision to do the same. Such sensors are often divided into two categories, namely *proprioceptive* and *exteroceptive* sensors. These terms were first introduced in 1906 by C. Sherrington [14] to describe the human nervous system, but have since become rather common also in engineering. In general proprioceptive sensors measure the state of the vehicle itself, such as accelerometer and gyrometer sensors do. Exteroceptive sensors on the other hand measure the state of the environment, such as time-of-flight transmitter/receiver sensors do. Thus, various sensor measurements can be combined into a state estimate, where the uncertainties involved should be tracked to provide information about the certainty of the estimate. As stated by T. Barfoot [15]:

> *In a way, state estimation is about doing the best we can with the sensors we have.*

If the state should fully describe a vehicle's motion over time, then the question of how the state is constructed is a fair one. A vehicle is typically free to both translate and rotate in its environment, consequently giving it 6 Degrees Of Freedom (DOF). Thus, the mathematical representation of translation and rotation should also have 6 DOF. This geometric configu-

ration combining position and orientation is referred to as the *pose* of the vehicle.

This chapter will present mathematical representations of the translation and rotation, accentuating cross-relations through brief the use of Lie theory, in addition to presenting a selection of integration techniques.

2.1 Reference Frames

Position, orientation, and motion are all relative to the point of view of which they are observed. As a result the concept of *reference frames* should be properly defined prior to delving into the representation of translation and rotation.

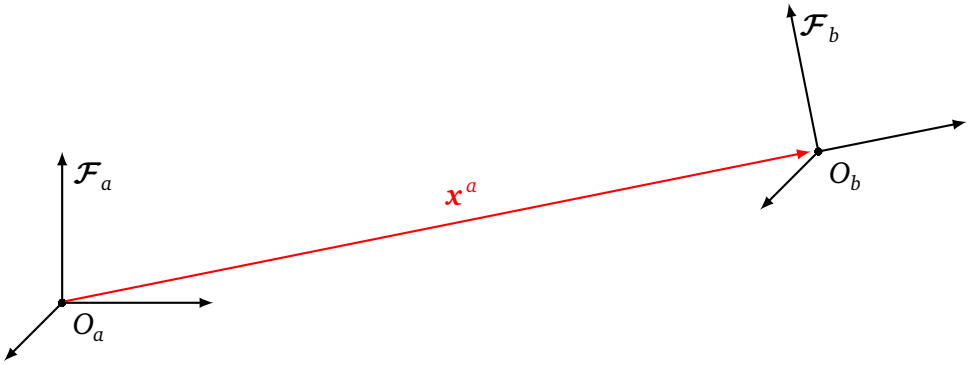


Figure 2.1: The coordinate vector $\boldsymbol{x}^a$ describes the origin of reference frame $\mathcal{F}_b$ with respect to the reference frame $\mathcal{F}_b$.

The position of a point in space can be described by a coordinate vector $\boldsymbol{x}^a \in \mathbb{R}^3$ consisting of three components, which represents the displacement in Euclidean space with respect to the frame of reference $\mathcal{F}_a$, as shown in Figure 2.1. Consequently, coordinate vectors are considered sufficient to represent the 3 DOF related to translation. Rotational motion is described by expressing the orientation of one frame, $\mathcal{F}_b$, with respect to another frame, $\mathcal{F}_a$. Both reference frames $\mathcal{F}_a$ and $\mathcal{F}_b$ are Cartesian coordinate frames, each of which defines a set of orthogonal axes which intersect at the origin O_a and O_b respectively [16].

Defining both Earth-centered and geographic reference frames is useful when analyzing the motion of a vehicle. As mentioned by T. Fossen [17], two examples of earth centered reference frames are the *ECI frame* and the *ECEF frame*:

- The Earth-centered inertial (ECI) frame $\mathcal{F}_{ECI}$ is an inertial frame for navigation where Newton's laws of motion apply. The origin of $\mathcal{F}_{ECI}$

is located at the center of the Earth and axes fixed with respect to the stars.

- The Earth-Centered Earth-Fixed (ECEF) frame $\mathcal{F}_{ECEF}$ also has its origin located at the center of the Earth, but the axes rotate relative to the ECI frame. This angular rate of rotation is $\boldsymbol{\omega}_{\epsilon} = 7.2921 \cdot 10^{-5}$ rad/s, which can be neglected for slow-moving vehicles. By doing so $\mathcal{F}_{ECEF}$ may be considered inertial.

On the other hand, the *NED frame* and the *ENU frame* are examples of geographic reference frames:

- The North, East, Down (NED) reference frame $\mathcal{F}_{NED}$ is the coordinate system with origin defined relative to Earth's reference ellipsoid, x axis pointing towards true *North*, y axis towards true *East*, and z axis downwards normal to Earth's surface.
- The East, North, Up (ENU) reference frame $\mathcal{F}_{ENU}$ has the same origin as $\mathcal{F}_{NED}$, although for ENU the x axis is pointing towards true *East*, the y axis towards true *North*, and z axis upwards normal to Earth's surface.

In general, $\mathcal{F}_{ECI}$ is primarily used in space applications, $\mathcal{F}_{ECEF}$ is used in navigation on Earth and systems such as the GPS, $\mathcal{F}_{ENU}$ is often used in aerial navigation, while $\mathcal{F}_{NED}$ is typically used in marine applications. The various reference frames are visualized in Figure 2.2.

In addition to Earth-centered and geographic reference frames, body-fixed frames are also of great use. The body-fixed reference frame is the moving coordinate system that is fixed to the vehicle. The position and orientation are often described using world frames such as $\mathcal{F}_{ECEF}$, while the linear and angular velocities and accelerations are described in the body frame.

2.2 Rotations

Unlike translation which can be sufficiently represented by coordinate vectors, the degrees of freedom related to rotation must be handled with care. Rotations do not live in a vector space[1], as they form a non-commutative group called the *Special Orthogonal group* denoted $SO(3)$. As such, rotations must be parameterized to provide mathematical representations of a vehicle's orientation. This section will inspect orientation representations such as *rotation vectors*, *Euler angles*, *rotation matrices*, and *unit quaternions*.

[1] In the mathematical sense of a set whose elements may be added together and scaled.

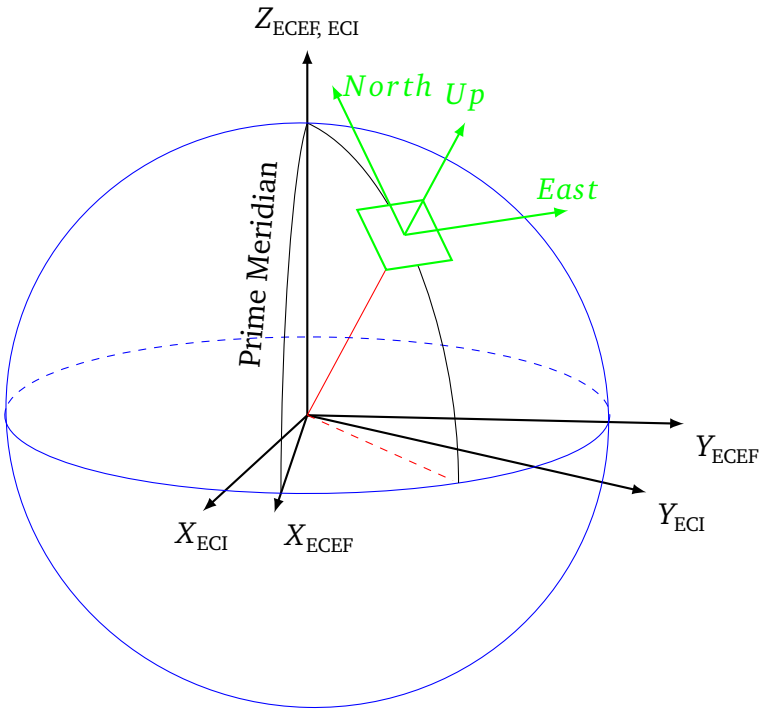


Figure 2.2: Visualization of the reference frames $\mathcal{F}_{ECI}$, $\mathcal{F}_{ECEF}$ and $\mathcal{F}_{ENU}$.

Rotation Vectors

Rotation vectors, also referred to as the angle-axis representation, describe rotations in three-dimensional Euclidean space using vectors co-directional with the rotation axis. The rotation vector can be written as

$$\boldsymbol{\phi} \triangleq \phi \boldsymbol{u} \tag{2.1}$$

where the length of the vector ϕ describes the magnitude of the rotation in radians, and the unit vector $\boldsymbol{u}$ is co-directional with the axis of rotation following the right-hand grip convention [13]. Note that a specific rotation can be represented by infinitely many rotation vectors, since a rotation vector with length $\phi + 2\pi K$, for any integer K, describes the same rotation as a rotation vector with length ϕ. Thus, the representation of rotation vectors can not be global.

Euler Angles

Euler angles constitute one of the most popular representations of rotation, a representation which dates back to 1751 and the works by L. Euler [18, 19]. They can be defined by composition of elemental rotations, where a

specific orientation can be represented through rotations around the axes of a coordinate system. Depending on the coordinate system, the elemental rotations can be considered either *intrinsic* or *extrinsic*:

- **Intrinsic rotations** occur around the axes of a moving coordinate system.
- **Extrinsic rotations** occur around the axes of a fixed coordinate system.

In addition to these conventions, there are numerous possible sequences defining different variants of Euler angles. For navigational purposes the set of Euler angles to parameterize the yaw, pitch, and roll of a vehicle is a popular choice, as presented by G. Bryan [20]:

- ψ denotes rotation around the *z-axis* in radians equivalent of *yaw*.
- θ denotes rotation around the *y-axis* in radians equivalent of *pitch*.
- ϕ denotes rotation around the *x-axis* in radians equivalent of *roll*.

A visualization of this parameterization using the intrinsic convention is shown in Figure 2.3.

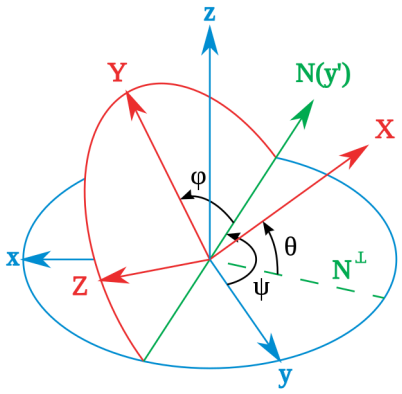


Figure 2.3: Euler angles parameterizing yaw ψ, pitch θ, and roll ϕ using a z-y-x sequence and intrinsic rotations. [21] (Licensed under CC BY 3.0)

Euler angles are a representation with exactly three parameters and they are associated with singularities, where certain sequences may fail to sufficiently represent a given rotation. This phenomena is better known as *gimbal lock*. To avoid gimbal lock certain restrictions must be enforced. However, independent of the representation used there are always only 3 underlying Degrees Of Freedom (DOF) associated with rotation. As proven by J. Stuelpnagel [22], no 3-dimensional parameterization can be both global and non-singular. This fact motivates the inspection of representations of higher dimensions than rotation vectors and Euler angles, such as *rotation matrices* and *unit quaternions*.

Rotation Matrices

A rotation matrix is a matrix $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ producing rotations to vectors through the matrix product, while maintaining the orthogonality condition and the relative orientation condition resulting in the following constraints

$$\boldsymbol{R}^\top\boldsymbol{R} = \boldsymbol{I} = \boldsymbol{R}\boldsymbol{R}^\top \qquad \det(\boldsymbol{R}) = 1 \tag{2.2}$$

These constraints limit the number of Degrees Of Freedom (DOF) from 9 to 3, thus matching up with the underlying number of DOF associated with rotation. As such, rotation matrices belong to $SO(3)$ [13].

Rotations about one basis vector of a coordinate system, also referred to as principle rotations, are represented by a specific set of rotation matrices. $\boldsymbol{R}_x(\bullet)$, $\boldsymbol{R}_y(\bullet)$, and $\boldsymbol{R}_z(\bullet)$ rotate vectors anticlockwise by a specified angle about the x-, y-, and z-axis using the right-hand rule with the following definitions:

$$\boldsymbol{R}_x(\phi) \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{c}(\phi) & -\mathrm{s}(\phi) \\ 0 & \mathrm{s}(\phi) & \mathrm{c}(\phi) \end{bmatrix} \tag{2.3}$$

$$\boldsymbol{R}_y(\theta) \triangleq \begin{bmatrix} \mathrm{c}(\theta) & 0 & \mathrm{s}(\theta) \\ 0 & 1 & 0 \\ -\mathrm{s}(\theta) & 0 & \mathrm{c}(\theta) \end{bmatrix} \tag{2.4}$$

$$\boldsymbol{R}_z(\psi) \triangleq \begin{bmatrix} \mathrm{c}(\psi) & -\mathrm{s}(\psi) & 0 \\ \mathrm{s}(\psi) & \mathrm{c}(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

Here, the abbreviations $\mathrm{s}(\bullet) \triangleq \sin(\bullet)$ and $\mathrm{c}(\bullet) \triangleq \cos(\bullet)$ have been used. Any rotation can be expressed as a product of these using matrix multiplication. For instance, the same rotation parameterized through Euler angles and visualized in Figure 2.3 can be expressed as follows using the principle rotations:

$$\begin{aligned} \boldsymbol{R}_{product} &= \boldsymbol{R}_z(\psi)\boldsymbol{R}_y(\theta)\boldsymbol{R}_x(\phi) \\ &= \begin{bmatrix} \mathrm{c}(\psi)\mathrm{c}(\theta) & \mathrm{c}(\psi)\mathrm{s}(\theta)\mathrm{s}(\phi) - \mathrm{s}(\psi)\mathrm{c}(\phi) & \mathrm{c}(\psi)\mathrm{s}(\theta)\mathrm{c}(\phi) + \mathrm{s}(\psi)\mathrm{s}(\phi) \\ \mathrm{s}(\psi)\mathrm{c}(\theta) & \mathrm{s}(\psi)\mathrm{s}(\theta)\mathrm{s}(\phi) + \mathrm{c}(\psi)\mathrm{c}(\phi) & \mathrm{s}(\psi)\mathrm{s}(\theta)\mathrm{c}(\phi) - \mathrm{c}(\psi)\mathrm{s}(\phi) \\ -\mathrm{s}(\theta) & \mathrm{c}(\theta)\mathrm{s}(\phi) & \mathrm{c}(\theta)\mathrm{c}(\phi) \end{bmatrix} \end{aligned} \tag{2.6}$$

The rotation matrix equivalent to a specific rotation vector $\boldsymbol{\phi}$ can be obtained through the *Rodrigues rotation formula*

$$\boldsymbol{R}(\boldsymbol{\phi}) \triangleq \boldsymbol{I}\cos\phi + [\boldsymbol{u}]_\times \sin\phi + \boldsymbol{u}\boldsymbol{u}^\top(1-\cos\phi) \tag{2.7}$$

Here, $[\bullet]_\times$ is the *skew operator* producing the cross-product matrix,

$$[\boldsymbol{a}]_\times \triangleq \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \tag{2.8}$$

also known as the *Lie algebra* of $SO(3)$. Lie theory is touched upon in Section 2.3.

Unit Quaternions

A unit quaternion $\boldsymbol{q}$ is defined as a complex number with one real part q_w and three imaginary parts $\boldsymbol{q_v} = [q_x, q_y, q_z]$, see Equation (2.9), where the unit norm constraint $\boldsymbol{q}^\top \boldsymbol{q} = 1$ is satisfied.

$$\boldsymbol{q} = \begin{bmatrix} q_w \\ \boldsymbol{q}_v \end{bmatrix} \tag{2.9}$$

Quaternions come with certain rules for addition and multiplication, which are used as part of their definition. The sum of two quaternions is defined as in Equation (2.10).

$$\boldsymbol{p} \pm \boldsymbol{q} = \begin{bmatrix} p_w \\ \boldsymbol{p}_v \end{bmatrix} \pm \begin{bmatrix} q_w \\ \boldsymbol{q}_v \end{bmatrix} = \begin{bmatrix} p_q \pm q_w \\ \boldsymbol{p}_v \pm \boldsymbol{q}_v \end{bmatrix} \tag{2.10}$$

The multiplication of two quaternions is defined as in Equation (2.11).

$$\boldsymbol{p} \otimes \boldsymbol{q} = \begin{bmatrix} p_w q_w - \boldsymbol{p}_v^\top \boldsymbol{q}_v \\ q_w \boldsymbol{p}_v + p_w \boldsymbol{q}_v + \boldsymbol{p}_v \times \boldsymbol{q}_v \end{bmatrix} \tag{2.11}$$

Quaternions satisfy the associative property, but the presence of a cross product in the definition of multiplication means that the quaternion product is non-commutative. Lastly, the inverse of a unit quaternion equals the conjugate according to $\boldsymbol{q}^{-1} = \boldsymbol{q}^*$, where the conjugate is defined as $\boldsymbol{q} = [q_w, -\boldsymbol{q}_v^\top]^\top$ [23].

Handling cross-relations, the rotation matrix equivalent to a specific quaternion denoted by $\boldsymbol{R} \triangleq \boldsymbol{R}\{\boldsymbol{q}\}$ can be obtained through the *quaternion to rotation matrix conversion formula* defined as:

$$\boldsymbol{R} = (q_w^2 - \boldsymbol{q}_v^\top \boldsymbol{q}_v)\boldsymbol{I} + 2\boldsymbol{q}_v \boldsymbol{q}_v^\top + 2q_w[\boldsymbol{q}_v]_\times \tag{2.12}$$

Further, the quaternion equivalent to a specific rotation vector denoted by $\boldsymbol{q} \triangleq \boldsymbol{q}\{\boldsymbol{\phi}\}$ can be obtained through the *rotation vector to quaternion conversion formula* defined as:

$$\boldsymbol{q} \triangleq \text{Exp}(\phi \boldsymbol{u}) = e^{\phi \boldsymbol{u}/2} = \cos\frac{\phi}{2} + \boldsymbol{u}\sin\frac{\phi}{2} = \begin{bmatrix} \cos\phi/2 \\ \boldsymbol{u}\sin\phi/2 \end{bmatrix} \tag{2.13}$$

It should be noted that unit quaternions as a representation define a double cover of $SO(3)$, where a quaternion and its negative encode the same rotation. As such, any provided conversion formula going from $SO(3)$ representations to quaternions will only give one of two possible quaternion representations [13].

2.3 Matrix Lie Groups

To better understand the Special Orthogonal group $SO(3)$ formed by rotations, a small detour via the realm of Lie theory will be made. More specifically, this section will be limited to *matrix Lie groups* of which $SO(3)$ is an example. Note that Lie groups are highly abstract constructions and the theory is by no means simple. This section serves as an exceptionally brief introduction to the topic, intended to accentuate the cross-relations between rotation vectors and parameterizations of higher dimensionality. It is based on the work of Solà et al. [24] and the summary provided by T. Haavardsholm [16].

Lie theory, concerning the theory of continuous symmetry applied to geometry and differential equations, was largely created by the Norwegian mathematician S. Lie. A crucial mathematical object within this theory is *the Lie group*, defined as a smooth differentiable manifold whose elements satisfy the group axioms. A group $(\mathcal{G}, \circ)$ is a set $\mathcal{G}$ with a composition operation $\circ$ that satisfies the axioms

$$\text{Closure under } \circ: \quad \mathcal{X} \circ \mathcal{Y} \in \mathcal{G} \tag{2.14}$$

$$\text{Identity } \mathcal{E}: \quad \mathcal{E} \circ \mathcal{X} = \mathcal{X} \circ \mathcal{E} = \mathcal{X} \tag{2.15}$$

$$\text{Inverse } \mathcal{X}^{-1}: \quad \mathcal{X}^{-1} \circ \mathcal{X} = \mathcal{X} \circ \mathcal{X}^{-1} = \mathcal{E} \tag{2.16}$$

$$\text{Associativity } \circ: \quad (\mathcal{X} \circ \mathcal{Y}) \circ \mathcal{Z} = \mathcal{X} \circ (\mathcal{Y} \circ \mathcal{Z}) \tag{2.17}$$

for elements $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \in \mathcal{G}$. For matrix Lie groups, the elements are matrices, $\circ$ is matrix multiplication, and the inversion operation is equivalent with the matrix inverse. The group action of matrix Lie groups on vectors is $\boldsymbol{R} \cdot \boldsymbol{v} \triangleq \boldsymbol{R}\boldsymbol{v}$ for rotations $\boldsymbol{R} \in SO(3)$.

The smooth manifold of matrix Lie groups is a topological space that locally resembles linear space, and the manifold looks the same at every point. Thus, the manifold has a unique tangent space at each point which is a vector space. In Figure 2.4, the manifold $\mathcal{M}$ is visualized as a curved, smooth surface with a tangent vectors space $\mathcal{TM}_{\mathcal{X}}$ at the point $\mathcal{X}$. The tangent space at the identity $\mathcal{TM}_{\mathcal{E}}$ is called the *Lie algebra* of $\mathcal{M}$, and its elements can be identified with vectors in $\mathbb{R}^m$, where $m = \mathbf{dim}_{\mathbb{R}}(\mathcal{M})$. Further, elements of the Lie algebra can be mapped into elements of the

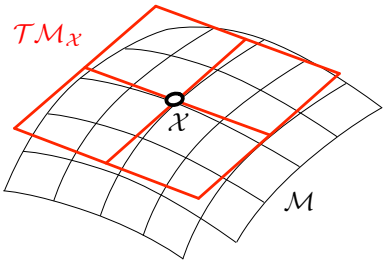


Figure 2.4: Visualization of a manifold $\mathcal{M}$ with a tangent vectors space $\mathcal{TM}_\mathcal{X}$ at the point $\mathcal{X}$ [24]. (Cropped; licensed under CC BY-NC-SA 4.0)

group using the *exponential map* $\exp : \mathcal{TM}_\mathcal{E} \to \mathcal{M}$. The log map is the inverse operation. This relation between the matrix Lie groups and the Lie algebra is visualized in Figure 2.5. A great incentive for the use of matrix Lie groups is found in the fact that local properties of the manifold allow for the use of linear algebra and calculus while maintaining the global properties of groups [16].

The reader might recall from Section 2.2 how the skew operator is known as the Lie algebra of $SO(3)$. This relation can be demonstrated through the orthogonality condition from Equation (2.2). The tangent space of $SO(3)$ can be found by taking the time derivative of this constraint, resulting in the following:

$$\mathcal{T}SO(3)_{\boldsymbol{R}} = \boldsymbol{R}^\top \dot{\boldsymbol{R}} = -(\boldsymbol{R}^\top \dot{\boldsymbol{R}})^\top \tag{2.18}$$

Being the negative of its transpose, $\mathcal{T}SO(3)_{\boldsymbol{R}}$ is by definition a skew-symmetric matrix. Thus, evaluating the tangent space at the identity rotation matrix $\boldsymbol{I}$ verifies how $[\boldsymbol{a}]_\times$ is in the Lie algebra of $SO(3)$. Further, since the tangent space vector $\boldsymbol{\phi} = \phi \boldsymbol{u}$ corresponds to a rotation vector, the exponential map $\exp([\boldsymbol{\phi}]_\times)$ mapping such vectors into $SO(3)$ is simply the Rodrigues rotation formula from Equation (2.7).

Another example of a matrix Lie group is the *Special Unitary group* $SU(2)$ which topologically is the 3-sphere identifying as the unit quaternions group S^3. The 3-dimensional manifold embedded in 4-dimensional space is defined by the unit norm constraint, and the S^3 group resembles the linear space locally, not globally. The Lie algebra is the space of pure imaginary quaternions, isomorphic to the hyperplane $\mathbb{R}^3$ visualized as a red grid in Figure 2.5. Tangent vectors $\tau_1, \tau_2, \boldsymbol{\nu} t$ of the Lie algebra wrap the manifold along the corresponding great arc through the exponential map. The exponential map is defined through the rotation vector to quaternion conversion formula from Equation (2.13).

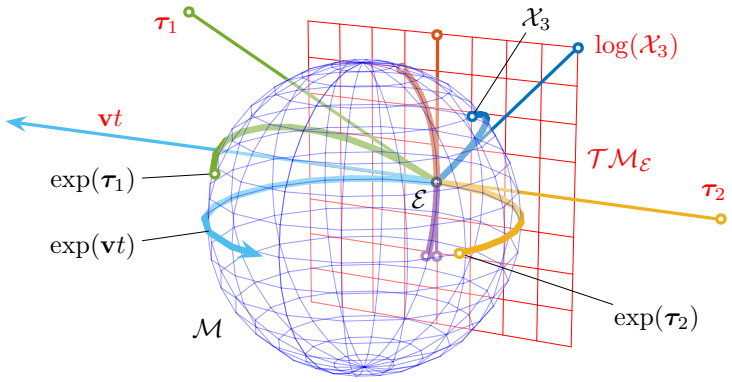


Figure 2.5: Visualization of the relation between the matrix Lie group and the Lie algebra. The Lie algebra $\mathcal{TM}_\mathcal{E}$ is the tangent space to the manifold $\mathcal{M}$ at the identity $\mathcal{E}$. The exponential map wraps tangent vectors on $\mathcal{TM}_\mathcal{E}$ over the manifold $\mathcal{M}$ along the corresponding great arc. Contrarily, the logarithmic map unwraps elements on the manifold onto the Lie algebra [24]. (Licensed under CC BY-NC-SA 4.0)

2.4 Evolution of States

With a better understanding of how translation and rotation can be represented mathematically, such representations can be combined to form the geometric configuration of poses. However, being able to represent the state of a vehicle constitutes only one half of the story; the other half being the description of how the state evolves. The evolution of states can commonly be modeled as a set of nonlinear differential equations of the form

$$\dot{\boldsymbol{x}} = f(t, \boldsymbol{x}), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x_0} \tag{2.19}$$

where $\boldsymbol{x}$ is the state and $f(t, \boldsymbol{x})$ is a nonlinear function of time and state. Finding exact or approximate solutions for such systems implies the use of integration methods, of which a selection will be presented in this section.

Closed-form Integration

In certain cases it is possible to do integration through closed-form expressions, that is formulas in terms of named functions as opposed to approximations using numerical techniques. Closed-form solutions are often desirable, as they tend to be computationally and analytically superior. However, such solutions can be difficult to derive, if they exist at all. As an example provided by J. Solá [13], consider the first-order linear differential

equation,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A} \cdot \boldsymbol{x}(t) \tag{2.20}$$

where the relation is linear and constant over the interval of integration $[t_k, t_k + \Delta t]$. In such cases, integration results in

$$\boldsymbol{x}_{k+1} = \mathrm{e}^{\boldsymbol{A}\cdot\Delta t}\boldsymbol{x}_k = \Phi \boldsymbol{x}_k \tag{2.21}$$

Here, Φ is the transition matrix with its Taylor expansion defined as

$$\Phi = \mathrm{e}^{\boldsymbol{A}\cdot\Delta t} = \boldsymbol{I} + \boldsymbol{A}\Delta t + \frac{1}{2}\boldsymbol{A}^2\Delta t^2 + \frac{1}{3!}\boldsymbol{A}^3\Delta t^3 + \cdots = \sum_{k=0}^{\infty}\frac{1}{n!}\boldsymbol{A}^n\Delta t^n \tag{2.22}$$

Certain instances of $\boldsymbol{A}$ may allow for writing the integration in closed form. Other means of integration must be used in the cases of inaccessible closed-form solutions.

Runge-Kutta Integration

One of the most utilized families of implicit and explicit iterative methods are the *Runge-Kutta (RK) methods* stemming from the work of C. Runge [25] and W. Kutta [26]. These methods estimate changes over the interval of integration, before summing up these changes over the interval through integration. The general RK method is formulated as follows

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta t \sum_{i=1}^{s} b_i K_i \tag{2.23}$$

$$K_i = g(t_k\ \Delta \cdot c_i, \boldsymbol{x}_k + \Delta t \sum_{j=1}^{s} a_{ij} K_j) \tag{2.24}$$

where the coefficients a_{ij}, b_i, c_j define the RK method, and s is the number of stages of the method. A specific method can be considered either explicit or implicit depending on the the terms a_{ij}. Embedded in the RK formulation there are several integration methods; the Euler method, the midpoint method, the RK4 method to mention some.

The Euler method constitutes the simplest Runge-Kutta method with one stage, where derivatives are assumed constant over the integration interval. Given a nonlinear differential equation as Equation (2.19), the forward Euler method for integration is given by the formula

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta t f(t_k, \boldsymbol{x}_k) \tag{2.25}$$

Although the Euler method can be numerically unstable and has a slow convergence of error, it does function as a simple integration technique [13]. More advanced RK methods can improve on efficiency, accuracy and stability.

Monte Carlo Integration

For high-dimensional problems, solving multidimensional integrals can become a computationally exhaustive task when using deterministic approaches. The method of *Monte Carlo integration* addresses this through approximating solutions employing a non-deterministic approach relying on the law of large numbers. From a statistical point of view, an integral of the form

$$I = \int \boldsymbol{f}(\boldsymbol{x})\boldsymbol{\pi}(\boldsymbol{x})\, d\boldsymbol{x} \tag{2.26}$$

can be approximately evaluated through drawing samples $\boldsymbol{x}^i$ from the Probability Density Function (PDF) $\boldsymbol{\pi}(\boldsymbol{x})$ taking the average of $\boldsymbol{f}(\boldsymbol{x}^i)$. Letting $\{\boldsymbol{x}^i\}_{i=1}^N$ be N independent and identically distributed samples from $\boldsymbol{\pi}(\bullet)$, defining the corresponding sample mean of $\boldsymbol{f}(\boldsymbol{x})$ as

$$I_N = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{f}(\boldsymbol{x}^i) \tag{2.27}$$

Then the approximation I_N will converge to I as the number of samples goes towards infinity [23]. As such, the error scales dependent on the number of samples and is considered independent of the number of dimensions. Furthermore Monte Carlo integration works for both smooth and discontinuous integrands.

Chapter 3

State Estimation

To accurately navigate underwater environments is a challenging task which requires multiple sensors and performing state estimation. The properties of such environments constrain the set of sensors applicable. Moreover, the chosen sensor suite will ultimately influence what state estimation approach to choose. This chapter will present a selection of sensors used in this project, alternative solutions to the state estimation problem, and the quaternion based error-state Kalman filter (ESKF) formulation implemented and tested in this project.

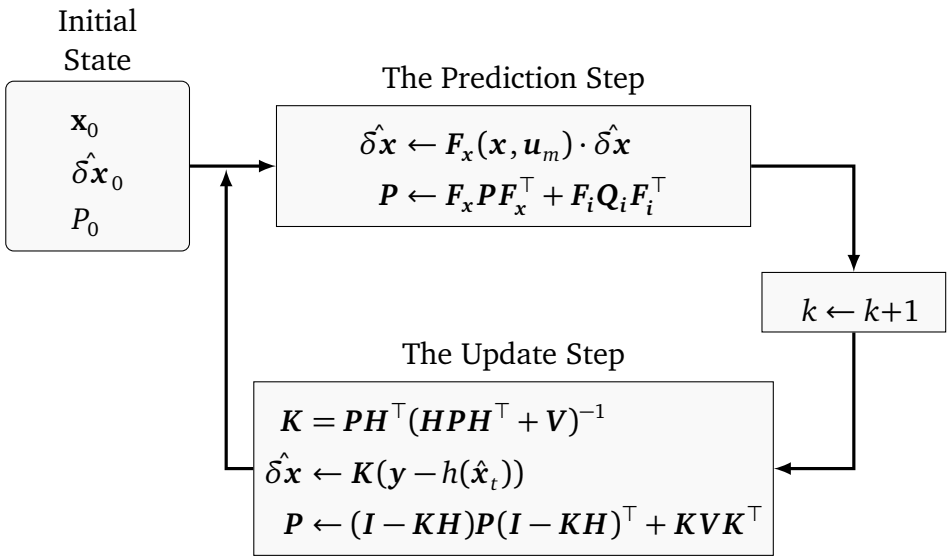


Figure 3.1: Visualization of the error-state Kalman filter (ESKF) formulation.

3.1 Sensors in Underwater Applications

Attenuation of radio signals combined with poor visibility greatly constrains the set of sensors suited for underwater applications, making sensors such as RAdio Detection And Ranging (RADAR), GPS and cameras unfit. However, sound waves travel significantly faster in water than they do in air, and this property justifies the use of acoustic sensors. As a result, acoustic sensors are commonly used in underwater applications. This section will present a selection of important sensors used in this project, and explain the working principles and usage for these. The Side-Scan Sonar (SSS) sensor will be omitted from this section as it is covered in detail in Chapter 7.

Inertial Measurement Unit (IMU)

An Inertial Measurement Unit (IMU) is a device for determining movements in terms of acceleration, angular velocity, and rotation using a combination of several sensors. The IMU usually combines a linear acceleration sensor with a gyroscope measuring along three axes each, resulting in a 6 DOF device. Depending on the IMU, orientation estimates may be maintained and supplied through integration of the gyroscope measured rotation rates. In such cases, long term pitch and roll stability is achieved through selective use of earth's gravity vector measured by the accelerometer. Further, a magnetometer can be added to provide heading reference and maintain the long-term heading accuracy [5], resulting in a three sensor IMU with 9 degrees of freedom.

As with any sensor, the various measurements from an IMU will to a certain degree be corrupted by noise. However, due to temperature, time, and mechanical stress the accelerometer and gyrometer are inherently prone to drift; a slow-changing random process accumulating over time. As a result, measurements will also be corrupted by bias noise, which should be taken into consideration when utilizing IMU measurements. Upon adding a magnetometer, the presence of magnetic variation, magnetic anomalies, and electromagnetic interference (EMI) will also have an impact on the IMU's performance.

Pressure Sensor

A pressure sensor is a device for measuring pressure, and it is commonly used in maritime applications involving submersed vehicles. Utilizing pressure measurements p, the depth z at which a vehicle is positioned can be

estimated. This can be done using the least squares fitted UNESCO formula [27], included here for completeness:

$$\begin{aligned} z &= \frac{C_1 p + C_2 p^2 + C_3 p^3 + C_4 p^4}{g(\phi) + \frac{1}{2}\gamma' p} + \frac{\Delta D}{9.8} \\ \gamma' &= 2.184 \cdot 10^{-6}\ \mathrm{m/s^2/dbar} \\ g(\phi) &= 9.780318(1.0 + 5.2788 \cdot 10^{-3} \sin^2\phi + 2.36 \cdot 10^{-5} \sin^4\phi) \end{aligned} \tag{3.1}$$

Here, $\gamma\prime$ is the mean vertical gradient of gravity, $g(\phi)$ is gravity at the ocean surface as a function of latitude, ΔD is the geopotential anomaly in the water column, and the constants are $\{C_i\} = \{9.73, -2.25 \cdot 10^{-5}, 2.28 \cdot 10^{-10}, -1.82 \cdot 10^{-15}\}$ assuming pressure is given in decibar. The resulting depth z is given in meters. Further, salinity, temperature, and fluid density are factors affecting depth estimates. If pressure measurements are sufficiently accurate, a pressure sensor resolves position along the vertical axis due to the steep pressure gradient underwater.

Doppler Velocity Log (DVL)

A Doppler Velocity Log (DVL) is an acoustic device commonly used in maritime applications to provide information about the velocity of a vehicle relative to the sea floor. Typically, it has four transceivers mounted at an angle, emitting acoustic pulses. When the DVL is sufficiently close to the bottom, the transceivers will receive reflected pulses and acquire *bottom lock*, meaning that a sufficient number of beam measurements are available. If the vehicle and thus the DVL is moving with regard to the sea floor, the reflected pulses will be subject to a Doppler shift [5].

The Doppler shift measured from one transceiver is directly proportional to velocity. As the Doppler shift only works with radial motion, and not angular motion, a velocity component parallel to the transceiver can be calculated. Using multiple transceivers pointed in different directions allows for calculation of different velocity components. With three oriented transceivers one can obtain velocity in three dimensions with the assumption of uniform (homogeneous) currents across layers of constant depth. An additional fourth transceiver can be used to evaluate data quality, namely errors due to inhomogeneties in the water and errors caused by malfunctioning equipment [28].

By combining the velocity components in three dimensions with built-in roll, pitch and heading sensors, a DVL can compute the vehicle's speed vector in a world-referenced frame [5]. The quality of measurements can be

prohibited by bubbles inhibiting the transmission of sound. Bubbles may reduce profiling range, and can in extreme cases block the signal completely [28].

3.2 Alternative State Estimation Solutions

Complex problems often tend to generate a myriad of good and not-so-good solutions, and state estimation is no exception. When operating in challenging and constraining underwater environments, the act of choosing the right tool for the job becomes ever so important. However, making a choice is just as much about what not to choose. This section will present a selection of alternative solutions to the state estimation problem visualized in Figure 3.3, while Section 3.3 will address the error-state Kalman filter (ESKF) solution used in this project.

The Bayes Filter

Functioning as the basis for the majority of state estimate solutions mentioned here is the *Bayes filter*. This general probabilistic approach seeks to estimate a Probability Density Function (PDF) to represent the likelihood of the state using measurements up to and including the current time. The filtering is done in a cyclic manner consisting of a *prediction step* and an *update step* as visualized in Figure 3.2. In the prediction step, the current estimate is propagated forward in time using a process model describing how the state evolves in time. In the update step, the predicted estimate is then corrected using a measurement alongside with a measurement model relating said measurements to the state vector. Performing these steps as measurements arrive results in the exact recursive filter that is Bayes filter [29]. Regardless, the Bayes filter is nothing more than a mathematical artifact and it cannot be implemented in practice. As mentioned by T. Barfoot [15], there are two primary reasons for this:

1. Probability density functions live in an infinite-dimensional space. Thus, an infinite amount of memory would be needed to completely represent the likelihood of the state.
2. Exact evaluation of integrals involved in the prediction step would be computationally very expensive requiring infinite computing resources.

As a result, the Bayes filter is considered intractable, and limiting assumptions and approximations must be made to provide utilizable solutions.

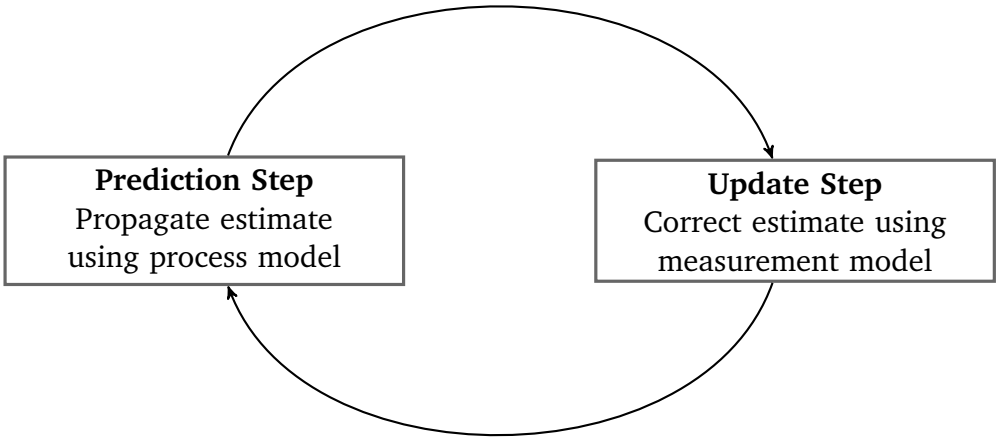


Figure 3.2: Visualization of the cyclic filtering process of the Bayes filter.

The Particle Filter

An approximation of the Bayes filter can be found the technique of *particle filters*. The particle filter approximates PDFs through random sampling using a finite number of samples, thereby handling the memory-related issue of the Bayes filter. Each sample can be passed through nonlinearities and be recombined to create an approximation of the transformed PDF. To overcome the computational resource issue, the prediction step can be approximated through employing Monte Carlo integration as presented in Section 2.4. Within the context of localization, the particle representation comes with a set of advantages as mentioned by S. Thrun [30]:

- Particle filters can accommodate a large range of arbitrary sensor characteristics, motion dynamics, and noise distributions.
- Particle filters focus computational resources in areas that are most relevant.
- Finally, particle filters are surprisingly easy to implement, which makes them an attractive paradigm for mobile robot localization.

However, there are also disadvantages related to particle filters to be mentioned. Most importantly, the number of samples needed increases exponentially with the dimension of the state space. As a result, the particle filter can become computationally intractable for high-dimensional state estimation problems[1]. Further, and somewhat counter-intuitive, is the fact

[1]Although rarely a problem when performing 6 DOF estimation.

that performance degrades when sensors are too accurate. This is a result of the cluster-generating effect of *particle depletion*, where the particle cloud gradually collapses with all the probability mass centered around one single cluster of particles [23]. There exists many flavors of particle filters handling the various pitfalls arising from their stochastic nature, nevertheless this line of thought will not be pursued further.

High-Gain Observers

Given the nonlinear nature of the state estimation problem, the class of nonlinear observers should naturally be taken into account. However, the realm of nonlinear observers does cover a wide range of solutions such as sliding mode observers, bounding observers, and extended observers to mention some. Prior to viewing the Kalman filter and some of its variations, a slight digression in the direction of *high-gain observers* are in order. As stated by H. Khalil [31]:

> *The technique, known as high-gain observers, works for a wide class of nonlinear systems and guarantees that the output feedback controller recovers the performance of the state feedback controller when the observer gain is sufficiently high.*

High gains translates to fast observer dynamics, and sufficiently high gains will result in observer dynamics that are considerably faster than the dynamics of the system to control, thus attenuating uncertainty in estimates. High-gain observers are well suited for systems where large perturbations occurs on account of being nonlinear converging observers of which stability can be proven. In addition, they are typically less computationally expensive than alternative methods. However, the high gain ensuring convergence will in the presence of measurement noise also increase noise effects. Consequently, there will be a trade-off between steady-state errors and noise-induced errors when operating with such observers. Worth mentioning, sufficiently high gains could give rise to the *peaking phenomenon*, introducing impulsive behavior into the system which could destabilize the closed-loop system in the lack of global growth conditions [31].

The Kalman Filter and Its Many Flavors

Another approach to overcome the issues of the Bayes filter can be found in the legacy of R. Kalman, namely the *Kalman filter* [32] and its many variations. Given normally distributed variables and linear transitions, the

Bayes filter becomes equal to the Kalman filter which provides the best linear unbiased estimate. Nevertheless, in its original form the Kalman filter is not applicable to nonlinear problems as the optimality of Kalman filtering will only hold under linear and Gaussian systems. Fortunately, since its conception the Kalman filter has been the subject of extensive research and application. Today there exists numerous generalizations of the method allowing the state estimation of nonlinear systems, although the sheer number of abbreviations alone can push one's mind into the right half-plane. Out of concern[2], this section will limit the variations being viewed to the *EKF*, *IEKF*, *UKF*, and *EnKF*.

The original Kalman filter can be extended to use on nonlinear systems, resulting in the famous *extended Kalman filter (EKF)*. Like the original formulation, the EKF approximates the likelihood of the state as a Gaussian PDF, assuming noise terms to be Gaussian. However, through linearizing the process and measurement models of the Bayes filter, the EKF allows for evaluation of integrals involved in the prediction and update steps in closed form [33]. By doing so, both primary issues of the Bayes filter are handled at the cost of losing optimality. The validity of the assumption of Gaussian noise terms can be questioned, as the noise affecting the system will typically vary a lot with time. However, it is desirable from a perspective of simplicity to describe noise using fixed standard deviation [23]. The performance of the EKF for a general nonlinear system depends on the nonlinearity close to the operating point of the linearization. The fact that the operating point is the mean of the state estimate, and not the true state, may often cause the estimate to being biased and inconsistent [15]. This can be improved by iteratively recomputing the operating point, allowing linearization about increasingly better estimates. Doing so results in the iterated extended Kalman filter (IEKF), although the improved performance do come at the cost of increased computational requirements.

Another approach to improve on the EKF is found in the *unscented Kalman filter (UKF)*, first proposed by S. Julier et al. [34]. Rather than linearizing, the UKF make use of a sigmapoint transformation to pass PDFs through the nonlinear prediction and measurement models. The method is not restricted to assuming Gaussian noise, somewhat reminiscent of Monte Carlo-type methods such as the particle filter. However, samples are drawn according to a specific, deterministic algorithm, avoiding the problems of statistical convergence. Because of this fundamental difference, high order information about distributions of the state can be captured using a significantly small number of points [34]. Consequently, the UKF provides a balance between low computational effort and high performance. The

[2]For both the author and the reader.

method is computationally expensive compared to the EKF, although it does not require analytical derivatives.

For high-dimensional estimation problems the ensemble Kalman filter (EnKF), introduced by G. Evensen [35], is a sought-after solution as the number of ensemble members used can be much smaller than the state dimension. The EnKF solution is closely related to particle filters; much like particles, the state representation and propagation is done through an ensemble of vectors. However, the EnKF updates ensembles by a linear shift based on the assumption of Gaussian PDFs, thus avoiding reweighting, resampling, and the issue of particle depletion related to such methods [36]. In theory, as the number of ensembles tends to infinity, the EnKF converges to the exact Kalman filter for linear Gaussian models. Due to its properties, the EnKF is mainly used for very high-dimensional problems as found in e.g. meteorology. Compared to such problems, the estimation problem of underwater navigation is considered low-dimensional, and the EnKF is visited for the sake of completeness.

To summarize, there exist a great number of state estimation solutions out there, and only a limited selection of them has been covered here. Note, a statistical Bayesian approach has been taken, explaining the lopsidedness of Figure 3.3. Nevertheless, all of the above-mentioned solutions have been cast aside in favor of the solution selected for this project, the *error-state Kalman filter (ESKF)*.

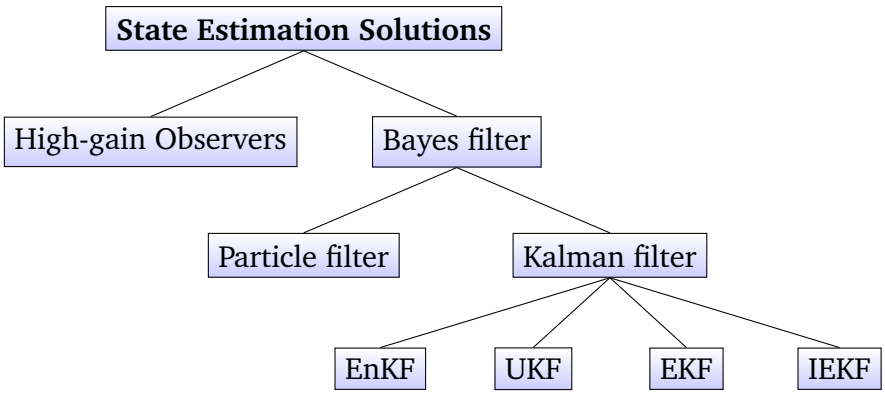


Figure 3.3: Visualization of the presented alternative state estimation solutions. Children of the Bayes filter node correspond to respective approximations.

3.3 Quaternion based error-state Kalman filter (ESKF)

In this project, the *quaternion based error-state Kalman filter (ESKF)* as presented by J. Solà [13] is adopted and implemented for the purpose of data acquisition. The ESKF has been selected on account of the filter's assets as mentioned by Solà,

- The orientation error-state is minimal, avoiding issues related to over parametrization and the consequent risk of singularity of the involved covariance matrices, resulting typically from enforcing constraints.
- The error-state system is always operating close to the origin, and therefore far from possible parameter singularities
- The error-state is always small, meaning that all second-order products are negligible and the computation of Jacobians easy and fast.
- The error dynamics are slow because all the large-signal dynamics have been integrated in the nominal-state. This means that we can apply filter corrections at a lower rate than the predictions.

in addition to the essential property of allowing bias estimation. As emphasized in Section 3.1, the bias noise corrupting IMU measurements should be taken into consideration in order to minimize drift in the state estimates. Other formulations of the ESKF include the work of K. Gade [37], as well as the work of S. Roumeliotis et al. [38, 39]. However, due to use of a minimalist notation, and a quaternion-based, passive body-to-world Hamilton convention coinciding with the conventions of software libraries such as ROS and Eigen, the ESKF as presented by Solà is adopted in this project.

The error-state Kalman filter (ESKF) is formulated such that the true state of the system can be decomposed into a nominal state and an error-state. The nominal state arises from integrating high-frequency IMU data without considering noise and model imperfections. As a consequence, the nominal state will accumulate errors which are attributed to the error-state. In parallel with integration of the nominal state, the ESKF predicts a Gaussian estimate of the error-state which has a evolution function defined by a time-variant linear dynamic system. Integration and prediction is performed when new input through IMU measurements is available. Further, filter correction is performed upon the arrival of other information such as velocity or depth measurements rendering the error-state observable. This correction provides a posterior Gaussian estimate of the error-state, before the error-state is injected into the nominal state and reset to zero [13]. The workflow of the error-state Kalman filter (ESKF) is visualized in

Table 3.1: All variables in the error-state Kalman filter (ESKF) [13].

Magnitude	True	Nominal	Error	Composition	Measured	Noise
Full state	$\boldsymbol{x}_t$	$\boldsymbol{x}$	$\delta\boldsymbol{x}$	$\boldsymbol{x}_t = \boldsymbol{x} \oplus \delta\boldsymbol{x}$		
Position	$\boldsymbol{p}_t$	$\boldsymbol{p}$	$\delta\boldsymbol{p}$	$\boldsymbol{p}_t = \boldsymbol{p} + \delta\boldsymbol{p}$		
Velocity	$\boldsymbol{v}_t$	$\boldsymbol{v}$	$\delta\boldsymbol{v}$	$\boldsymbol{v}_t = \boldsymbol{v} + \delta\boldsymbol{v}$		
Quaternion	$\boldsymbol{q}_t$	$\boldsymbol{q}$	$\delta\boldsymbol{q}$	$\boldsymbol{q}_t = \boldsymbol{q} \oplus \delta\boldsymbol{q}$		
Rotation matrix	$\boldsymbol{R}_t$	$\boldsymbol{R}$	$\delta\boldsymbol{R}$	$\boldsymbol{R}_t = \boldsymbol{R}\delta\boldsymbol{R}$		
Angles vector			$\delta\boldsymbol{\theta}$	$\delta\boldsymbol{q} = e^{\delta\boldsymbol{\theta}/2}$ $\delta\boldsymbol{R} = e^{[\delta\boldsymbol{\theta}]_\times}$		
Acc. bias	$\boldsymbol{a}_{bt}$	$\boldsymbol{a}_b$	$\delta\boldsymbol{a}_b$	$\boldsymbol{a}_{bt} = \boldsymbol{a}_b + \delta\boldsymbol{a}_b$		$\boldsymbol{a}_w$
Gyro. bias	$\boldsymbol{\omega}_{bt}$	$\boldsymbol{\omega}_b$	$\delta\boldsymbol{\omega}_b$	$\boldsymbol{\omega}_{bt} = \boldsymbol{\omega}_b + \delta\boldsymbol{\omega}_b$		$\boldsymbol{\omega}_w$
Gravity vector	$\boldsymbol{g}_t$	$\boldsymbol{g}$	$\delta\boldsymbol{g}$	$\boldsymbol{g}_t = \boldsymbol{g} + \delta\boldsymbol{g}$		
Acceleration	$\boldsymbol{a}_t$				$\boldsymbol{a}_m$	$\boldsymbol{a}_n$
Angular rate	$\boldsymbol{\omega}_t$				$\boldsymbol{\omega}_m$	$\boldsymbol{\omega}_n$

Figure 3.1.

The definitions of all variables used in the error-state filter formulations are summarized in Table 3.1. It is worth noting that both the angular rates $\boldsymbol{\omega}$ and the angular error $\delta\boldsymbol{\theta}$ are defined *locally* with respect to the nominal orientation following the so-called classical approach. Orientation is represented through quaternions using the *Hamilton* convention, which is *right-handed* and describes rotation in a *local-to-global* fashion. Following common practice, the Earth's rotation rate $\boldsymbol{\omega}_\epsilon$ has been neglected in the rotational kinematics described in Equation (3.2c), which would otherwise be $\dot{\boldsymbol{q}}_t = \frac{1}{2}\boldsymbol{q}_t \otimes (\boldsymbol{\omega}_m - \boldsymbol{R}_t^\top \boldsymbol{\omega}_\epsilon - \boldsymbol{\omega}_{bt} - \boldsymbol{\omega}_n)$. This is considered valid due to the magnitude of the noises and biases associated with the IMU sensor.

3.3.1 The True-State Kinematics in Continuous Time

The true-state kinematics in continuous time are described by the following system

$$\dot{\boldsymbol{p}}_t = \boldsymbol{v}_t \tag{3.2a}$$
$$\dot{\boldsymbol{v}}_t = \boldsymbol{R}_t(\boldsymbol{a}_m - \boldsymbol{a}_{bt} - \boldsymbol{a}_n) - \boldsymbol{g} \tag{3.2b}$$
$$\dot{\boldsymbol{q}}_t = \frac{1}{2}\boldsymbol{q}_t \otimes (\boldsymbol{\omega}_m - \boldsymbol{\omega}_{bt} - \boldsymbol{\omega}_n) \tag{3.2c}$$
$$\dot{\boldsymbol{a}}_{bt} = \boldsymbol{a}_w \tag{3.2d}$$
$$\dot{\boldsymbol{\omega}}_{bt} = \boldsymbol{\omega}_w \tag{3.2e}$$
$$\dot{\boldsymbol{g}}_t = \boldsymbol{0} \tag{3.2f}$$

which can be summarized as $\dot{\boldsymbol{x}}_t = f_t(\boldsymbol{x}_t, \boldsymbol{u}, \boldsymbol{w})$. This system has state $\boldsymbol{x}_t$, takes noisy IMU readings $\boldsymbol{u}_m$ as inputs, and is perturbed by white Gaussian noise $\boldsymbol{w}$, all defined by:

$$\boldsymbol{x}_t = \begin{bmatrix} \boldsymbol{p}_t \\ \boldsymbol{v}_t \\ \boldsymbol{q}_t \\ \boldsymbol{a}_{bt} \\ \boldsymbol{\omega}_{bt} \\ \boldsymbol{g}_t \end{bmatrix} \qquad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{a}_m - \boldsymbol{a}_n \\ \boldsymbol{\omega}_m - \boldsymbol{\omega}_n \end{bmatrix} \qquad \boldsymbol{w} = \begin{bmatrix} \boldsymbol{a}_w \\ \boldsymbol{\omega}_w \end{bmatrix} \tag{3.3a}$$

3.3.2 The Nominal State Kinematics in Discrete Time

As previously mentioned, the true state $\boldsymbol{x}_t$ will be estimated through the decomposed nominal and error-state. This estimation is performed in discrete time and both states must be formulated as difference equations accounting for discrete time intervals $\Delta t > 0$. Following Solà's approach, the Euler method of integration described in Section 2.4 is applied to acquire the discrete time kinematics of both the nominal and the error-state. In doing so, the difference equations of the nominal-state are written as

$$\boldsymbol{p} \leftarrow \boldsymbol{p} + \boldsymbol{v}\Delta t + \frac{1}{2}(\boldsymbol{R}(\boldsymbol{a}_m - \boldsymbol{a}_b) - \boldsymbol{g})\Delta t^2 \tag{3.4a}$$

$$\boldsymbol{v} \leftarrow \boldsymbol{v} + (\boldsymbol{R}(\boldsymbol{a}_m - \boldsymbol{a}_b) - \boldsymbol{g})\Delta t \tag{3.4b}$$

$$\boldsymbol{q} \leftarrow \boldsymbol{q} \otimes \boldsymbol{q}\{(\boldsymbol{\omega}_m - \boldsymbol{\omega}_b)\Delta t\} \tag{3.4c}$$

$$\boldsymbol{a}_b \leftarrow \boldsymbol{a}_b \tag{3.4d}$$

$$\boldsymbol{\omega}_b \leftarrow \boldsymbol{\omega}_b \tag{3.4e}$$

$$\boldsymbol{g} \leftarrow \boldsymbol{g} \tag{3.4f}$$

where $x \leftarrow f(x, \bullet)$ constitute a time update of the type $x_{k+1} = f(x_k, \bullet_k)$. As defined in Equation (2.12), $\boldsymbol{R} \triangleq \boldsymbol{R}\{\boldsymbol{q}\}$ is the rotation matrix associated with the current nominal orientation $\boldsymbol{q}$, and $\boldsymbol{q}\{v\}$ is the quaternion associated with the rotation v, according to Equation (2.13).

3.3.3 The Error-State Kinematics in Discrete Time

The error-state kinematics are composed of both deterministic and stochastic elements, where integration of stochastic elements results in random impulses. Thus, the discrete time kinematics of the error-state are written

as

$$\delta \boldsymbol{p} \leftarrow \delta \boldsymbol{p} + \delta \boldsymbol{v} \Delta t \tag{3.5a}$$
$$\delta \boldsymbol{v} \leftarrow \delta \boldsymbol{v} + (-\boldsymbol{R}[\boldsymbol{a}_m - \boldsymbol{a}_b]_\times \delta \boldsymbol{\theta} - \boldsymbol{R}\delta \boldsymbol{a}_b)\Delta t + \boldsymbol{v}_i \tag{3.5b}$$
$$\delta \boldsymbol{\theta} \leftarrow \boldsymbol{R}^\top\{(\boldsymbol{\omega}_m - \boldsymbol{\omega}_b)\Delta t\}\delta \boldsymbol{\theta} - \delta \boldsymbol{\omega}_b \Delta t + \boldsymbol{\theta}_i \tag{3.5c}$$
$$\delta \boldsymbol{a}_b \leftarrow \delta \boldsymbol{a}_b + \boldsymbol{a}_i \tag{3.5d}$$
$$\delta \boldsymbol{\omega}_b \leftarrow \delta \boldsymbol{\omega}_b + \boldsymbol{\omega}_i \tag{3.5e}$$
$$\delta \boldsymbol{g} \leftarrow \delta \boldsymbol{g} \tag{3.5f}$$

where $\boldsymbol{v}_i$, $\boldsymbol{\theta}_i$, $\boldsymbol{a}_i$ and $\boldsymbol{\omega}_i$ are the random impulses applied to the velocity, orientation, and bias estimates, modeled by white Gaussian processes with zero mean and covariance matrices obtained through integration of the covariances of $\boldsymbol{a}_n$, $\boldsymbol{\omega}_n$, $\boldsymbol{a}_w$ and $\boldsymbol{\omega}_w$ over the time step Δt. The resulting covariance matrices become

$$\boldsymbol{V}_i = \sigma^2_{\tilde{a}_n} \Delta t^2 \boldsymbol{I} \tag{3.6a}$$
$$\boldsymbol{\Theta}_i = \sigma^2_{\tilde{\omega}_n} \Delta t^2 \boldsymbol{I} \tag{3.6b}$$
$$\boldsymbol{A}_i = \sigma^2_{\tilde{a}_w} \Delta t \boldsymbol{I} \tag{3.6c}$$
$$\boldsymbol{\Omega}_i = \sigma^2_{\tilde{\omega}_w} \Delta t \boldsymbol{I} \tag{3.6d}$$

where the values of the standard deviation of accelerometer and gyrometer noise and bias terms $\sigma_{\tilde{a}_n}$, $\sigma_{\tilde{\omega}_n}$, $\sigma_{\tilde{a}_w}$ and $\sigma_{\tilde{\omega}_w}$ must be specified. The definition of the skew operator $[\bullet]_\times$ used in Equation (3.5b) is found in Equation (2.8).

3.3.4 The ESKF Prediction Step

Making use of a more compact form considering the nominal state vector $\boldsymbol{x}$, the error-state vector $\delta \boldsymbol{x}$, the input vector $\boldsymbol{u}_m$, and the perturbation impulse vector $\boldsymbol{i}$ as follows

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{v} \\ \boldsymbol{q} \\ \boldsymbol{a}_b \\ \boldsymbol{\omega}_b \\ \boldsymbol{g} \end{bmatrix} \qquad \delta \boldsymbol{x} = \begin{bmatrix} \delta \boldsymbol{p} \\ \delta \boldsymbol{v} \\ \delta \boldsymbol{q} \\ \delta \boldsymbol{a}_b \\ \delta \boldsymbol{\omega}_b \\ \delta \boldsymbol{g} \end{bmatrix} \qquad \boldsymbol{u}_m = \begin{bmatrix} \boldsymbol{a}_m \\ \boldsymbol{\omega}_m \end{bmatrix} \qquad \boldsymbol{i} = \begin{bmatrix} \boldsymbol{v}_i \\ \boldsymbol{\theta}_i \\ \boldsymbol{a}_i \\ \boldsymbol{\omega}_i \end{bmatrix} \tag{3.7a}$$

the error-state system becomes

$$\delta \boldsymbol{x} \leftarrow f(\boldsymbol{x}, \delta \boldsymbol{x}, \boldsymbol{u}_m, \boldsymbol{i}) = \boldsymbol{F}_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{u}_m) \cdot \delta \boldsymbol{x} + \boldsymbol{F}_{\boldsymbol{i}} \cdot \boldsymbol{i} \tag{3.8}$$

Further, the ESKF prediction equations are defined as

$$\hat{\delta \boldsymbol{x}} \leftarrow \boldsymbol{F}_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{u}_m) \cdot \hat{\delta \boldsymbol{x}} \tag{3.9}$$

$$\boldsymbol{P} \leftarrow \boldsymbol{F}_{\boldsymbol{x}} \boldsymbol{P} \boldsymbol{F}_{\boldsymbol{x}}^{\top} + \boldsymbol{F}_{\boldsymbol{i}} \boldsymbol{Q}_{\boldsymbol{i}} \boldsymbol{F}_{\boldsymbol{i}}^{\top} \tag{3.10}$$

where $\delta \boldsymbol{x} \sim \mathcal{N}(\hat{\delta \boldsymbol{x}}, \boldsymbol{P})$; $\boldsymbol{F}_{\boldsymbol{x}}$ and $\boldsymbol{F}_{\boldsymbol{i}}$ are the Jacobians of $f()$ with respect to the error and perturbation vectors; and $\boldsymbol{Q}_{\boldsymbol{i}}$ is the covariance matrix of the perturbation impulses [13]. The Jacobians and the covariance matrix are extracted from the filter kinematics and stated as follows

$$\boldsymbol{F}_{\boldsymbol{x}} = \frac{\partial f}{\partial \delta \boldsymbol{x}}\Big|_{\boldsymbol{x}, \boldsymbol{u}_m} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{I}\Delta t & 0 & 0 & 0 & 0 \\ 0 & \boldsymbol{I} & -\boldsymbol{R}[\boldsymbol{a}_m - \boldsymbol{a}_b]_{\times}\Delta t & -\boldsymbol{R}\Delta t & 0 & -\boldsymbol{I}\Delta t \\ 0 & 0 & \boldsymbol{R}^{\top}\{(\boldsymbol{\omega}_m - \boldsymbol{\omega}_b)\Delta t\} & 0 & -\boldsymbol{I}\Delta t & 0 \\ 0 & 0 & 0 & \boldsymbol{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & \boldsymbol{I} & 0 \\ 0 & 0 & 0 & 0 & 0 & \boldsymbol{I} \end{bmatrix}$$

$$\boldsymbol{F}_{\boldsymbol{i}} = \frac{\partial f}{\partial \boldsymbol{i}}\Big|_{\boldsymbol{x}, \boldsymbol{u}_m} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \boldsymbol{I} & 0 & 0 & 0 \\ 0 & \boldsymbol{I} & 0 & 0 \\ 0 & 0 & \boldsymbol{I} & 0 \\ 0 & 0 & 0 & \boldsymbol{I} \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{Q}_{\boldsymbol{i}} = \begin{bmatrix} \boldsymbol{V}_{\boldsymbol{i}} & 0 & 0 & 0 \\ 0 & \boldsymbol{\Theta}_{\boldsymbol{i}} & 0 & 0 \\ 0 & 0 & \boldsymbol{A}_{\boldsymbol{i}} & 0 \\ 0 & 0 & 0 & \boldsymbol{\Omega}_{\boldsymbol{i}} \end{bmatrix}$$

The prediction step alongside with integration of the nominal state will be performed upon arrival of new IMU measurements.

3.3.5 The ESKF Update Step

Accompanying the IMU measurements used for prediction are orientation estimates from the IMU itself, as well as velocity and depth measurements from the DVL and the pressure sensor. These will be used to correct the filter and thus observe the bias errors. The ESKF update step consist of observation of the error-state, injection of errors into the nominal state, and reset of the error-state.

Supplementary measurements from a sensor delivering information dependent on the state are modeled as

$$\boldsymbol{y} = h(\boldsymbol{x}_t) + \boldsymbol{v} \tag{3.11}$$

where $h()$ is a general nonlinear function of the true state, and v is a white Gaussian noise with covariance $\boldsymbol{V}$. In the case of orientation estimates,

velocity measurements, and depth measurements observations are respectively modeled as

$$\boldsymbol{y}_{IMU} = \boldsymbol{q}_t + \boldsymbol{v}_{IMU} \qquad \boldsymbol{v}_{IMU} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{V}_{IMU}\boldsymbol{V}_{IMU}^{\top}) \tag{3.12a}$$

$$\boldsymbol{y}_{DVL} = \boldsymbol{v}_{xy\_t} + \boldsymbol{v}_{DVL} \qquad \boldsymbol{v}_{DVL} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{V}_{DVL}\boldsymbol{V}_{DVL}^{\top}) \tag{3.12b}$$

$$y_{PS} = p_{z\_t} + v_{PS} \qquad v_{PS} \sim \mathcal{N}(0, V_{PS}^2) \tag{3.12c}$$

where $\boldsymbol{q}_t$ is the true orientation, $\boldsymbol{v}_{xy\_t}$ is the true velocity along the x-axis and y-axis in $\boldsymbol{\mathcal{F}}_{NED}$, p_{z_t} is the true position along the z- in $\boldsymbol{\mathcal{F}}_{NED}$, $\boldsymbol{V}_{IMU}$ is the IMU standard deviation, $\boldsymbol{V}_{DVL}$ is the DVL standard deviation, and V_{PS} the standard deviation related to pressure sensor noise. The Jacobian matrix $\boldsymbol{H}$ with respect to the error-state evaluated at the nominal state $\boldsymbol{x}$ is defined as

$$\boldsymbol{H} \triangleq \frac{\partial h}{\partial \delta \boldsymbol{x}}\bigg|_{\boldsymbol{x}} = \frac{\partial h}{\partial \boldsymbol{x}_t}\bigg|_{\boldsymbol{x}} \frac{\partial \boldsymbol{x}_t}{\partial \delta \boldsymbol{x}}\bigg|_{\boldsymbol{x}} = \boldsymbol{H}_{\boldsymbol{x}} \boldsymbol{X}_{\delta \boldsymbol{x}} \tag{3.13}$$

where the chain rule has been utilized to compute $\boldsymbol{H}$. The Jacobian of $h()$ with respect to its own argument, $\boldsymbol{H}_{\boldsymbol{x}}$, has the following form for orientation estimates, velocity measurements and depth measurements respectively

$$\boldsymbol{H}_{\boldsymbol{x},IMU} \triangleq \frac{\partial \boldsymbol{q}_t}{\partial \boldsymbol{x}_t}\bigg|_{\boldsymbol{x}} = \begin{bmatrix}\boldsymbol{0}_{4\times3} & \boldsymbol{0}_{4\times3} & \boldsymbol{I}_{4\times4} & \boldsymbol{0}_{4\times3} & \boldsymbol{0}_{4\times3} & \boldsymbol{0}_{4\times3}\end{bmatrix} \tag{3.14a}$$

$$\boldsymbol{H}_{\boldsymbol{x},DVL} \triangleq \frac{\partial \boldsymbol{v}_t}{\partial \boldsymbol{x}_t}\bigg|_{\boldsymbol{x}} = \begin{bmatrix}\boldsymbol{0}_{2\times3} & \boldsymbol{I}_{2\times3} & \boldsymbol{0}_{2\times4} & \boldsymbol{0}_{2\times3} & \boldsymbol{0}_{2\times3} & \boldsymbol{0}_{2\times3}\end{bmatrix} \tag{3.14b}$$

$$\boldsymbol{H}_{\boldsymbol{x},PS} \triangleq \frac{\partial p_{z\_t}}{\partial \boldsymbol{x}_t}\bigg|_{\boldsymbol{x}} = \begin{bmatrix}0 & 0 & 1 & \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times4} & \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3}\end{bmatrix} \tag{3.14c}$$

while the Jacobian of the true state with respect to the error-state, $\boldsymbol{X}_{\delta \boldsymbol{x}}$, is defined as

$$\boldsymbol{X}_{\delta \boldsymbol{x}} \triangleq \frac{\partial \boldsymbol{x}_t}{\partial \delta \boldsymbol{x}}\bigg|_{\boldsymbol{x}} = \begin{bmatrix}\boldsymbol{I}_{6\times6} & \boldsymbol{0}_{6\times3} & \boldsymbol{0}_{6\times9} \\ \boldsymbol{0}_{4\times6} & \boldsymbol{Q}_{\delta\boldsymbol{\theta}} & \boldsymbol{0}_{4\times9} \\ \boldsymbol{0}_{9\times6} & \boldsymbol{0}_{9\times3} & \boldsymbol{I}_{9\times9}\end{bmatrix} \tag{3.15}$$

$$\boldsymbol{Q}_{\delta\boldsymbol{\theta}} \triangleq \frac{\partial \delta(\boldsymbol{q} \otimes \delta\boldsymbol{q})}{\partial \delta \boldsymbol{\theta}}\bigg|_{\boldsymbol{q}} = \frac{1}{2}\begin{bmatrix}-q_x & -q_y & -q_z \\ q_w & -q_z & q_y \\ q_z & q_w & -q_x \\ -q_y & q_x & q_w\end{bmatrix} \tag{3.16}$$

Utilizing the previously defined Jacobian $\boldsymbol{H}$, the filter correction equations can at last be written as

$$\boldsymbol{K} = \boldsymbol{P}\boldsymbol{H}^{\top}(\boldsymbol{H}\boldsymbol{P}\boldsymbol{H}^{\top} + \boldsymbol{V})^{-1} \tag{3.17}$$

$$\hat{\delta \boldsymbol{x}} \leftarrow \boldsymbol{K}(\boldsymbol{y} - h(\hat{\boldsymbol{x}}_t)) \tag{3.18}$$

$$\boldsymbol{P} \leftarrow (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\boldsymbol{P}(\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})^{\top} + \boldsymbol{K}\boldsymbol{V}\boldsymbol{K}^{\top} \tag{3.19}$$

Here, both the Jacobian $\boldsymbol{H}$ and the covariance matrix $\boldsymbol{V}$ to be used will depend on the type of measurement received. The symmetric and positive Joseph form is chosen over the classic covariance update function on account of being less sensitive to round-off errors [40].

Having observed the error-state, the nominal state must be updated through injection of the error:

$$\boldsymbol{p} \leftarrow \boldsymbol{p} + \hat{\delta\boldsymbol{p}} \tag{3.20a}$$

$$\boldsymbol{v} \leftarrow \boldsymbol{v} + \hat{\delta\boldsymbol{v}} \tag{3.20b}$$

$$\boldsymbol{q} \leftarrow \boldsymbol{q} \otimes \boldsymbol{q}\{\hat{\delta\boldsymbol{\theta}}\} \tag{3.20c}$$

$$\boldsymbol{a}_b \leftarrow \boldsymbol{a}_b + \hat{\delta\boldsymbol{a}_b} \tag{3.20d}$$

$$\boldsymbol{\omega}_b \leftarrow \boldsymbol{\omega}_b + \hat{\delta\boldsymbol{\omega}_b} \tag{3.20e}$$

$$\boldsymbol{g} \leftarrow \boldsymbol{g} + \hat{\delta\boldsymbol{g}} \tag{3.20f}$$

Lastly, a reset of the error-state mean $\hat{\delta\boldsymbol{x}}$ and covariance $\boldsymbol{P}$ must be performed to conclude the ESKF update step:

$$\hat{\delta\boldsymbol{x}} \leftarrow 0 \tag{3.21}$$

$$\boldsymbol{P} \leftarrow \boldsymbol{G}\boldsymbol{P}\boldsymbol{G}^\top \tag{3.22}$$

where the Jacobian matrix $\boldsymbol{G}$ is defined by

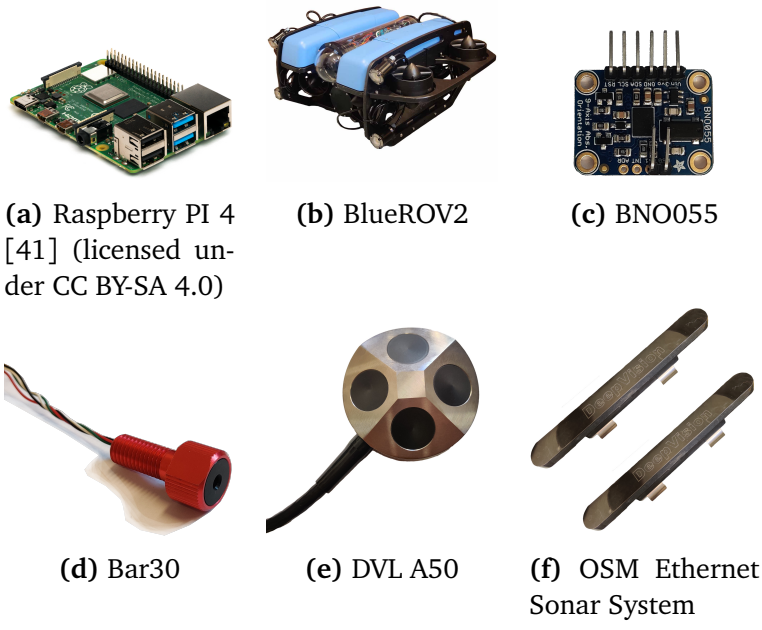
$$\boldsymbol{G} \triangleq \left.\frac{\partial(\delta\boldsymbol{x} \ominus \hat{\delta\boldsymbol{x}})}{\partial\,\delta\boldsymbol{x}}\right|_{\hat{\delta\boldsymbol{x}}} = \begin{bmatrix} \boldsymbol{I}_{6\times6} & \boldsymbol{0}_{6\times3} & \boldsymbol{0}_{6\times9} \\ \boldsymbol{0}_{3\times6} & \boldsymbol{I}_{3\times3} - [\frac{1}{2}\hat{\delta\boldsymbol{\theta}}]_\times & \boldsymbol{0}_{3\times9} \\ \boldsymbol{0}_{9\times6} & \boldsymbol{0}_{9\times3} & \boldsymbol{I}_{9\times9} \end{bmatrix} \tag{3.23}$$

The complete ESKF update step is executed for every valid orientation estimate from the IMU, and velocity and depth measurement arriving respectively from the DVL and the pressure sensor.

Chapter 4

Hardware and Implementation

Implementation of the error-state Kalman filter (ESKF) formulation presented in the previous chapter is dependent on the hardware configuration. To this extent, various methods and solutions have been utilized in an attempt to minimize process noise and properly tune the filter. This chapter will present the relevant parts of the hardware configuration used in this project, alongside with sensor specific solutions, Welford's method for computing running variance, and the filter consistency metrics.



(a) Raspberry PI 4 [41] (licensed under CC BY-SA 4.0)

(b) BlueROV2

(c) BNO055

(d) Bar30

(e) DVL A50

(f) OSM Ethernet Sonar System

Figure 4.1: Visualization of the relevant parts of the hardware configuration used in this project.

4.1 Hardware Configuration

The UUV configuration used in this project consists of a repurposed ROV equipped with a Raspberry PI (RPI) 4 model B and sensor suite. This section presents the essential parts of the hardware-setup which has been used to perform field tests and experiments. See Figure 4.1 for a visualization of all separate parts.

Blue Robotics BlueROV2

The ROV used is the BlueROV2 in heavy configuration from Blue Robotics Inc set up with four lumen subsea lights and 200 meter Fathom ROV tether. The BlueROV2 is rated for a maximum depth of 100 meters, and it has a maximum forward speed of 1.5 meters per second [42]. The ROV is compact and lightweight compared to a general AUV setup, making it deployable for a single researcher.

Raspberry PI 4 Model B

At the core, tying hardware and software together, is the single-board computer Raspberry PI 4 model B with 32GB memory. For the purpose of this project, the Raspberry PI is set up with Ubuntu 20.04 focal fossa, ROS2 Galactic, enabled I2C and necessary libraries.

Adafruit Fusion Breakout Board - BNO055

The IMU used to measure angular velocity, linear acceleration, and estimate absolute orientation is the 9-DOF fusion breakout board from Adafruit with a BNO055 sensor from Bosch. The BNO055 provides linear acceleration measurements, angular velocity measurements, and absolute orientation estimates at a frequency of 100 Hz [43].

Water Linked DVL A50

The DVL used to measure linear velocity is the DVL A50 from Water Linked AS. The A50 has an operational range from 0.05 to 50 meters with performance possibly degrading over 35 meters depending on the conditions. It communicates velocity, transducer, and position updates at varying frequencies. However, the ping rate of velocity and transducer updates are commonly in the range of 4 to 15 Hz, as the sensor is adaptive to altitude [44].

Blue Robotics Bar30

The pressure sensor used to measure depth is the Bar30 from Blue Robotics Inc. The Bar30 can measure up to 30 Bar (300 meters depth) with a depth resolution of 2 millimeters. It communicates depth measurements at a frequency of 10 Hz [45].

DeepVision OSM Ethernet Sonar System

The side-scan sonar used to perform sonar imaging is the OSM Ethernet sonar system with 680 kHz transducers from DeepVision AB. The sonar system has a max operating depth of 100 meters, operating with a chirp signal. Depending on settings, the sonar system may deliver images with resolution down to 1 centimeter [46]. The transducers have a vertical beam width of $\alpha = 60^\circ$ and horizontal beam width of $\phi = 0.7^\circ$, and they are symmetrically mounted facing down with an angular sensor placement of $\theta = 45^\circ$.

4.2 Sensor Placement in Body Frame

Both state estimation and control of the BlueROV2 are performed using the body frame as the reference frame. The various sensor frames are all aligned with the body frame. However, the DVL, the IMU and the pressure sensor are all placed at various locations. Figure 4.2 shows how the various sensor frames are located with respect to the body frame, and the offset vectors are summarized in Table 4.1. The placement of each sensor is of interest because the positional offset from body origin will ultimately affect the measurements coming from the respective sensor. Thus, these positional offsets are taken into account when the measurements are used in the error-state Kalman filter (ESKF).

Table 4.1: Table of positional offset vectors of the origin of each sensor frame with respect to the body frame origin in meters.

Sensor	$x_{\mathcal{F}_{body}}$	$y_{\mathcal{F}_{body}}$	$z_{\mathcal{F}_{body}}$
$\boldsymbol{r}_{IMU}$	0.057	0.027	-0.025
$\boldsymbol{r}_{DVL}$	-0.020	-0.095	0.133
$\boldsymbol{r}_{PS}$	-0.175	-0.015	-0.050

For the IMU, the offset $\boldsymbol{r}_{IMU}$ give rise to a linear acceleration component when the vehicle experiences a change in orientation. This linear acceler-

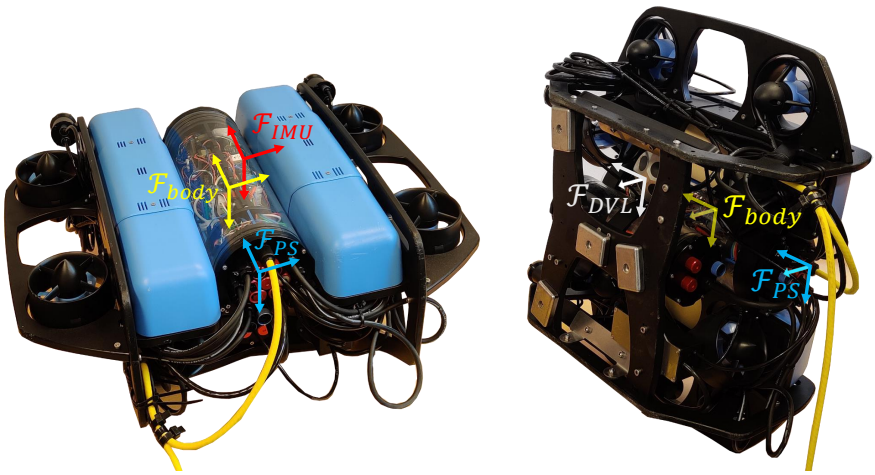


Figure 4.2: Visualization of the various sensor frames with respect to the body frame, $\mathcal{F}_{body}$.

ation component $\boldsymbol{a}_{offset,IMU}$ can be described as

$$\boldsymbol{a}_{offset,IMU} = \frac{d}{dt}\boldsymbol{v}_{offset,IMU} = \frac{d}{dt}(\boldsymbol{\omega} \times \boldsymbol{r}_{IMU}) = \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_{IMU}) \tag{4.1}$$

where $\boldsymbol{v}_{offset,IMU}$ is the resulting linear velocity component which can be analogous written as $\boldsymbol{\omega} \times \boldsymbol{r}_{IMU}$, and $\boldsymbol{\omega}$ is the measured angular velocity with the bias component removed, $\boldsymbol{\omega} = \boldsymbol{\omega}_m - \boldsymbol{\omega}_b$. Thus, the linear acceleration component can be computed and subtracted from the linear acceleration measurements coming from the IMU.

In a similar fashion, the offset of the DVL $\boldsymbol{r}_{DVL}$ give rise to a linear velocity component upon changes in the vehicles orientation. This linear velocity component can be described as

$$\boldsymbol{v}_{offset,DVL} = \boldsymbol{\omega} \times \boldsymbol{r}_{DVL} \tag{4.2}$$

where the same relationship between angular velocity and linear velocity has been used to define the resulting linear velocity component $\boldsymbol{v}_{offset,DVL}$. Again, this linear velocity component can be computed and subtracted from the linear velocity measurements coming from the DVL.

Lastly, the positional offset of the pressure sensor $\boldsymbol{r}_{PS}$ will cause an offset between the depth measurements and the position along the z-axis of the body frame. This offset can be described as

$$\boldsymbol{p}_{offset,PS} = \boldsymbol{R}\boldsymbol{r}_{PS} \tag{4.3}$$

where $\boldsymbol{R} \triangleq \boldsymbol{R}\{\boldsymbol{q}\}$ is the rotation matrix associated with the current nominal orientation $\boldsymbol{q}$. Note that only the element corresponding to the z-axis

$p_{z\_offset,PS}$ is used when subtracting the offset in the measurement function of the pressure sensor. An interesting detail is found in the fact that orientation is part of the state, and consequently the subtraction must be accounted for in the Jacobian $\boldsymbol{H}_{\boldsymbol{x},PS}$. Thus, the Jacobian changes from Equation (3.14c) to the following

$$\boldsymbol{H}_{\boldsymbol{x},PS} \triangleq \left.\frac{\partial p_{z\_t}}{\partial \boldsymbol{x}_t}\right|_{\boldsymbol{x}} = \begin{bmatrix} 0 & 0 & 1 & \mathbf{0}_{1\times 3} & \left.\frac{\partial p_{z\_offset,PS}}{\partial \boldsymbol{q}_t}\right|_{\boldsymbol{q}} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \end{bmatrix} \tag{4.4}$$

where the offset differentiated with respect to orientation $\left.\frac{\partial p_{z\_offset,PS}}{\partial \boldsymbol{q}_t}\right|_{\boldsymbol{q}}$ becomes

$$\left.\frac{\partial p_{z\_offset,PS}}{\partial \boldsymbol{q}_t}\right|_{\boldsymbol{q}} = \begin{bmatrix} 2q_y r_{x\_PS} - 2q_x r_{y\_PS} + 2q_w r_{z\_PS} \\ 2q_z r_{x\_PS} - 2q_w r_{y\_PS} - 2q_x r_{z\_PS} \\ 2q_w r_{x\_PS} + 2q_z r_{y\_PS} - 2q_y r_{z\_PS} \\ 2q_x r_{x\_PS} + 2q_y r_{y\_PS} + 2q_z r_{z\_PS} \end{bmatrix}^\top \tag{4.5}$$

4.3 Axis Convention of the IMU

The BNO055 IMU can be reconfigured to new reference axes to match the orientation of the body frame. However, the orientation estimate provided by the IMU is given in East, North, Up (ENU) and can not be changed onboard the sensor [47]. To solve this matter, the orientation estimate from the IMU $\boldsymbol{q}_{ENU}$ is transformed into North, East, Down (NED) through the following quaternion multiplication:

$$\boldsymbol{q}_{NED} = [0, -\sqrt{1/2}, -\sqrt{1/2}, 0] \otimes \boldsymbol{q}_{ENU} \tag{4.6}$$

This is equivalent to a 90 degree rotation around the z axis followed by a 180 degree rotation around the x axis. Further, the re-oriented estimate $\boldsymbol{q}_{NED}$ gets fused into the error-state Kalman filter (ESKF) as described in Section 3.3.5.

4.4 Erroneous DVL Velocity Information along the Z-axis

The Doppler Velocity Log (DVL) provides velocity measurements along all three axes of its local frame. However, due to the working principles of the DVL and the fact that the sea floor is not flat, the DVL will inevitably output erroneous velocity information along the z-axis of the world frame. The case of the vehicle moving deeper towards the sea floor is from the DVL's

point of view equivalent with the sea becoming shallower as visualized in Figure 4.3. Bathymetric data could be fused into the state estimate to adjust for the erroneous measurements. However, detailed bathymetric maps are not necessarily available and depth measurements from the barometer are reasonably accurate. For these reasons the velocity information along the z-axis of the world frame are not used. In practice, the DVL measurement $\boldsymbol{v}$ and its covariance matrix $\boldsymbol{V}$ are transformed from the local frame to NED frame as follows:

$$\boldsymbol{v}_{NED} = \boldsymbol{R}\boldsymbol{v} \tag{4.7}$$

$$\boldsymbol{V}_{NED} = \boldsymbol{R}\boldsymbol{V}\boldsymbol{R}^{\top} \tag{4.8}$$

Here, $\boldsymbol{R}$ is the rotation matrix associated with the current nominal orientation, which is described using the Hamilton convention. Note that only the information along the x- and y-axis are kept and fused into the error-state Kalman filter (ESKF).

4.5 Computing Running Variance

Knowledge about process and measurement noise statistics is required when using the error-state Kalman filter (ESKF), and physical considerations are considered mandatory. Depending on the sensor, such information can be acquired from the respective sensor's data sheet. However, noise statistics are not necessarily available, and even if they are, environmental factors such as vibrations, electromagnetic interference (EMI) and build quality can impact the respective sensor's performance. Because of this, the various noise components of the ESKF should be measured during operation, ensuring the use of statistics which reflect the respective sensor performance during real-world employment.

As mentioned in Sections 3.3.3 and 3.3.5, the process and measurement noise are modeled as white Gaussian noise with zero mean. These can be measured through the respective sample variance, which mathematically can be computed as:

$$\sigma^2 = \frac{1}{n(n-1)}(n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2) \tag{4.9}$$

For the purpose of noise estimation x_i represents measurement i from a sensor, while for bias noise estimation x_i represents the difference between two successive measurements i and $i-1$. The naive way to compute variance would be to accumulate both the sum of measurements and the sum

of the squares of the measurements for a given number of measurements n. Large measurements with minor differences would require computing a small number as the difference of two large numbers, evidently leading to loss of precision. In extreme cases, Equation (4.9) may evaluate to negative sample variance values, thus emphasizing the need for an alternate method.

In 1962 B. Welford presented a method for accurately computing running variance, guaranteeing positive sample variance [48]. The algorithm is summarized in Algorithm 1, and the reader may observe how the variance is computed in a running fashion as consecutive measurements arrive. This method is adopted for the computation of noise statistics used in the ESKF on account of its superior numerical properties, as well as being well suited for iterative use.

Algorithm 1 Computing running variance estimate, $\hat{\sigma}^2$.

Initialize:
  $M_1 \leftarrow x_1$
  $S_1 \leftarrow 0$
while *True* **do**
  if *New measurement* x_i *arrives* **then**
    $M_i = M_{i-1} + (x_i - M_{i-1})/i$
    $S_i = S_{i-1} + (x_i - M_{i-1})(x_i - M_i)$
  if $i \geq 2$ **then**
    $\hat{\sigma}^2 = S_i/(i-1)$

4.6 Filter Consistency

The process of tuning an error-state Kalman filter (ESKF) can be both time consuming and challenging, are physical considerations are not always sufficient to reach the desired performance. However, a more systematic way approach to tuning is by the means of *filter consistency*. As stated by E. Brekke [23], a filter is usually considered consistent if its errors on average are well described by the filters output, leading to the following requirements:

1. The state errors should be acceptable as zero mean.
2. The state errors should have magnitude commensurate with the state covariance yielded by the filter.
3. The innovations should be acceptable as zero mean.

4. The innovations should have magnitude commensurate with the innovation covariance yielded by the filter.
5. The innovations should be acceptable as white.

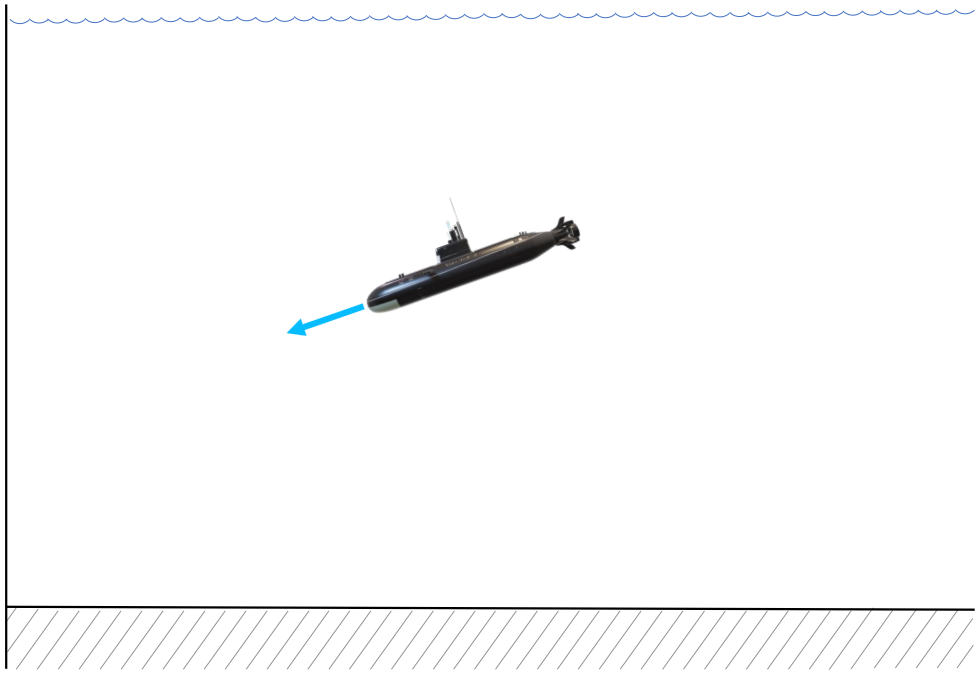
Tools for testing criteria 2 and 4 can are respectively normalized estimation error squared (NEES)

$$\epsilon_{NEES} = (\hat{\boldsymbol{x}}_t - \boldsymbol{x_t})^\top \boldsymbol{P}^{-1} (\hat{\boldsymbol{x}}_t - \boldsymbol{x_t}) \tag{4.10}$$

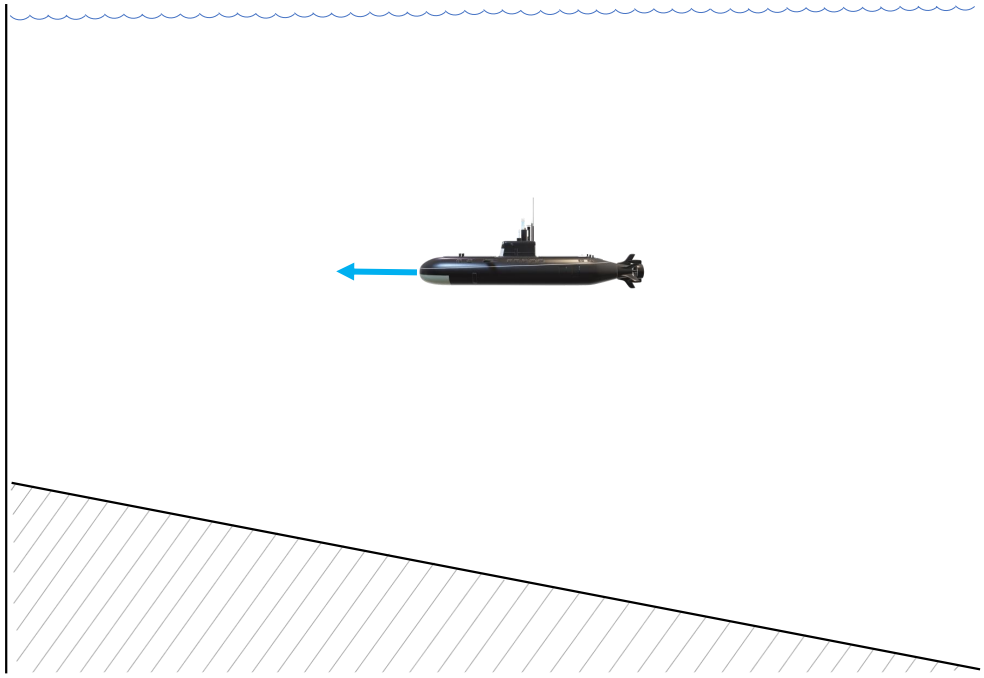
and normalized innovation squared (NIS)

$$\epsilon_{NIS} = (\boldsymbol{y} - h(\hat{\boldsymbol{x}}_t))^\top \boldsymbol{V}^{-1} (\boldsymbol{y} - h(\hat{\boldsymbol{x}}_t)) \tag{4.11}$$

Nonetheless, in the absence of a ground truth only NIS can be utilized to evaluate the filter's state of consistency. High NIS-values may indicate an overconfident filter, where the magnitude of innovations are not proportional with the innovation covariance.



(a) Vehicle moving deeper towards the sea floor.



(b) Vehicle moving in the xy-plane while the sea becomes shallower.

Figure 4.3: Case (a) and (b) are considered equivalent with respect to the DVL velocity measurements.

Chapter 5

Evaluation of the ESKF

The accuracy of state estimates affects the ROV's ability to perform. To this extent, evaluation of the error-state Kalman filter (ESKF) is imperative to know what performance can be expected. Further, quantifying drift will help accentuate the need for limiting the unbounded uncertainty growth, motivating Side-Scan Sonar (SSS) related work.

The error-state Kalman filter (ESKF) presented has been implemented, tuned, and tested using indoor facilities at Gløshaugen and Tyholt, as well as the fjord of Trondheim. Moreover, the filter has been used to gather sonar data sets from various locations such as Trondhjem Biological Station (TBS), Ilsvika, Kyvannet, and Dora. This chapter will present and evaluate results related to the measurement of ESKF noise statistics, drift, and filter consistency. In addition, a benchmark evaluation of the RPI will be presented.

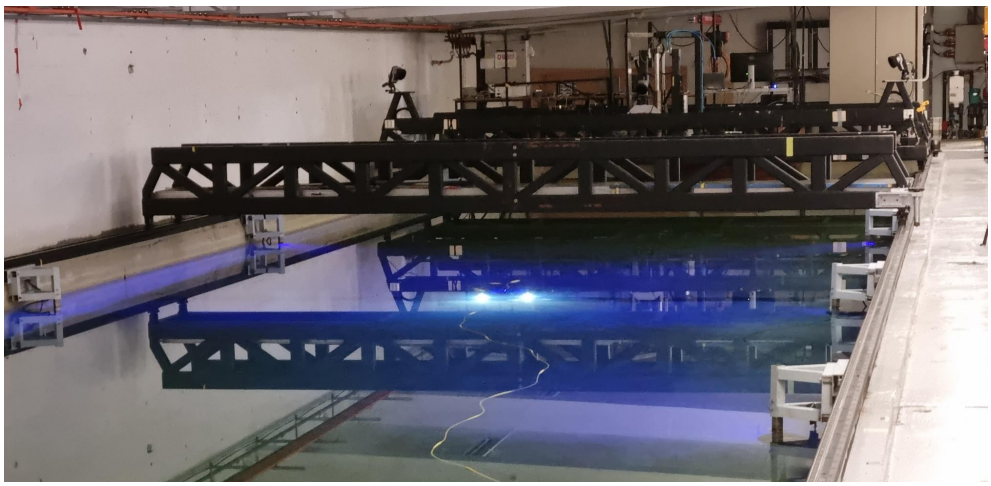


Figure 5.1: Testing of the ROV at the MC-lab utilizing the Qualisys real-time positioning system.

5.1 ESKF Noise Statistics

Using the error-state Kalman filter (ESKF) formulation requires knowledge of both measurement and process noise. When operating, the ESKF is set up with a combination of values from both data sheets and experimental measurements, whichever provides the largest uncertainty value. Note how the DVL is not considered here, since the sensor itself provides uncertainty estimates accompanying its velocity measurements.

Experimental measurements of the uncertainties related to the IMU and the pressure sensor were taken using Welford's method for computing running variance as presented in Section 4.5. Due to the lack of ground truth during the tuning process, the experimental measurements were taken while keeping the ROV motionless. Nonetheless, tests were conducted both with and without the thrusters running. All noise measurements were gathered in 10 second runs to avoid the influence of drift, while bias measurements were gathered in 2 minute runs. Results from experiments with and without thrusters running is presented in Table 5.1. The statistics from the data sheet [47] and the largest values from experimental measurements are presented in Table 5.2.

Table 5.1: Table of noise statistics from experimental measurements with and without thrusters running.

Statistic	Variable	No thrust	Full thrust
Acceleration noise	$\sigma_{\tilde{a}_n}$	$1.077 \cdot 10^{-2}$	$6.973 \cdot 10^{-1}$
Angular rate noise	$\sigma_{\tilde{\omega}_n}$	$6.739 \cdot 10^{-4}$	$1.139 \cdot 10^{-2}$
Orientation noise	V_{IMU}	0	$9.167 \cdot 10^{-4}$
Accelerometer bias	$\sigma_{\tilde{a}_w}$	$1.643 \cdot 10^{-2}$	$6.766 \cdot 10^{-2}$
Gyrometer bias	$\sigma_{\tilde{\omega}_w}$	$9.436 \cdot 10^{-4}$	$1.426 \cdot 10^{-2}$
Pressure sensor noise	V_{PS}	$9.529 \cdot 10^{-4}$	$2.468 \cdot 10^{-3}$

Comparing the runs with and without thrusters running reveals how all noise statistics[1] presumably are affected by the thrusters. Table 5.1 reveals how all statistics were measured to higher values with the thrusters running. Small movements may have occurred during testing despite the efforts to keep the ROV stationary, which may have affected the variance measurements. However, since the observed trend is consistent for all measurements it is assumed to be a result of vibrations and electromagnetic interference (EMI) introduced by power electronics and thrusters. Comparing measured and data sheet values in Table 5.2 shows how the accel-

[1] Excluding the linear velocity noise of the DVL.

Table 5.2: Table of noise statistics from data sheet [47] and experimental measurements. The largest value used in the ESKF formulation is marked as bold.

Statistic	Variable	Data sheet value	Computed value
Acceleration noise	$\sigma_{\tilde{a}_n}$	$1.864 \cdot 10^{-2}$	$\mathbf{6.973 \cdot 10^{-1}}$
Angular rate noise	$\sigma_{\tilde{\omega}_n}$	$\mathbf{5.236 \cdot 10^{-2}}$	$1.139 \cdot 10^{-2}$
Orientation noise	V_{IMU}	$\mathbf{5.236 \cdot 10^{-2}}$	$9.167 \cdot 10^{-4}$
Accelerometer bias	$\sigma_{\tilde{a}_w}$	$1.400 \cdot 10^{-3}$	$\mathbf{6.766 \cdot 10^{-2}}$
Gyrometer bias	$\sigma_{\tilde{\omega}_w}$	$3.800 \cdot 10^{-3}$	$\mathbf{1.426 \cdot 10^{-2}}$
Pressure sensor noise	V_{PS}	-	$\mathbf{2.468 \cdot 10^{-3}}$

eration noise and bias terms are in fact larger when measured, emphasizing the necessity of performing such measurements.

5.2 ESKF Performance

Proper assessment of the performance of the error-state Kalman filter (ESKF) requires access to some sort of ground truth. The lack of GPS fix makes assessment in field challenging. Because of this, the system was tested using the indoor test facilities of the Marine cybernetics laboratory. The MC-lab contains a water tank equipped with a real-time positioning system provided by Qualisys AB, see Figure 5.1. The tank itself measures 40 by 6 meters with a depth of 1.5 meters. However, the Qualisys system provides motion measurements only for a restricted area of the pool, thus limiting the area of operation. The ESKF's performance will be evaluated through comparing the state estimates to ground truth measurements from the Qualisys system. Note, one of the cameras was inoperative during testing, explaining the occasional loss of track. All testing has been performed using the hysteretic controller from Basso et al. [49][2].

To accommodate the limited area of operation, the trajectory visualized in Figure 5.2 was generated. The trajectory consists of two set-points and three consecutive rounds in a circle with a radius of 1 meter, where surge speed is set to 0.2 meters per second. Pitch and roll is set to 0, while the heading is 0 for the set-points and follows the direction of travel for the circle. By mistake, the filter was initialized using the square of the noise statistics in Table 5.2 when executing the trajectory[3]. As a result, the ESKF was initialized to be overly confident in acceleration, angular velocity and

[2] Consult the project thesis for more information [50]

[3] See Appendix A.

depth measurements. However, this was not discovered in time to perform new experiments, and the filter performance will be assessed using these values.

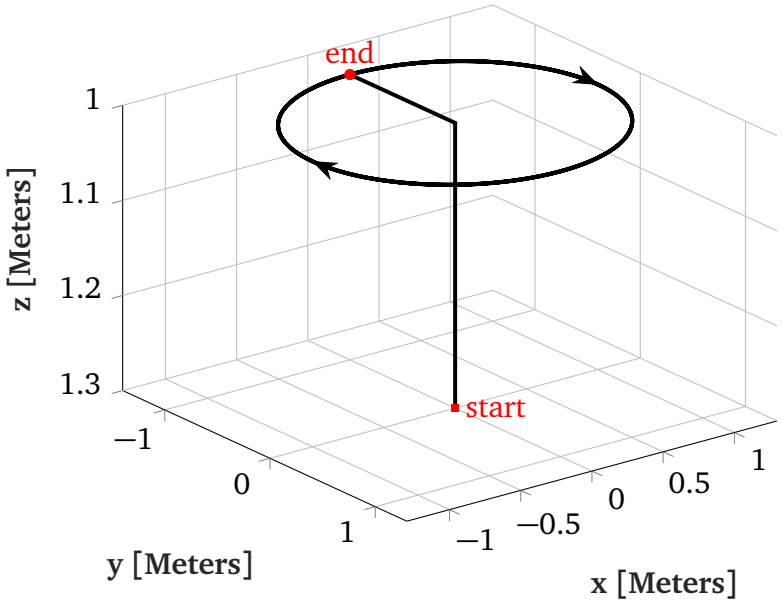


Figure 5.2: Trajectory consisting of two set-points and 3 overlapping circles in the xy-plane.

Comparing the state estimates to the corresponding ground truth measurements for the executed trajectory reveals the error of the ESKF. Figure 5.6 shows the position along each axis plotted against time for both the ESKF estimates and the Qualisys measurements. The depth estimates are rather noisy with occasional severe spikes, which can be attributed to the wrongfully initialized pressure sensor noise statistic amplifying noise from the pressure sensor. Nevertheless, the underlying trend of the estimate seem to be matching well with the ground truth. Further, the positional estimates in the xy-plane are also fairly accurate, as can be seen in Figure 5.3. Visual inspection shows some error in position along the x-axis. After completing the trajectory, the ROV suffered a positional drift of 0.15 meters. Per hour, this amounts to a positional error of 3.8 meters. This positional error is attributed to the biased measurements of the IMU, as well as the imprecise integration technique of Euler integration.

Throughout execution of the trajectory, the ROV also suffered drift in heading. Figure 5.5 shows the heading plotted against time for both the ESKF and the Qualisys measurements. Visual inspection shows how the estimate appears to follow the ground truth accurately. However, by the end of the trajectory the ROV had drifted 4.9 degrees. Over an hour, this

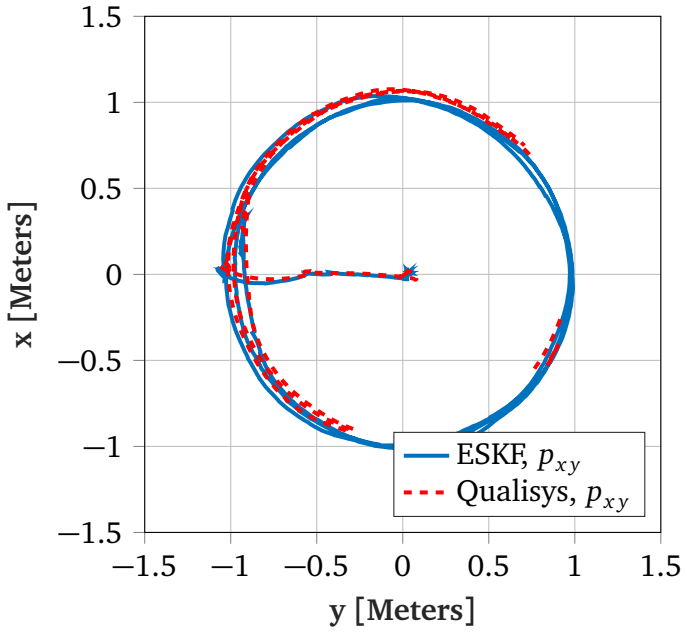


Figure 5.3: State estimate vs. ground truth: Cross-sectional result of x and y from execution of planned trajectory.

amounts to a drift of 123 degrees. There are several factors contributing to the accumulation of drift in orientation. Most apparent is the accuracy of the IMU, which is by far the most inexpensive sensor of the sensor suite[4]. Furthermore, the loosely coupled solution of fusing orientation estimates from the IMU may degrade the performance of the ESKF. Lastly, the imprecise integration technique of Euler integration may also contribute to the error observed. The severe drift in heading is concerning as it will cause the ROV to veer off course over time.

The accuracy of the state estimate is heavily dependent on the availability of DVL measurements. In the absence of such measurements, the filter relies entirely on acceleration measurements from the IMU to provide positional estimates in the xy-plane. Figure 5.4 shows a snippet of a data set gathered during a field test at Trondhjem Biological Station (TBS), where the DVL experienced a temporary loss of fix. Within 10 seconds, the state estimate drifted 45 meters along the x-axis before fix was acquired again. Similar drift was also observed along the y-axis. This severe drift in position can be attributed to the act of integrating biased acceleration measurements twice. Because of this, the ROV will veer significantly off track when experiencing continuous absence of DVL measurements.

[4]The BNO055 is priced at 350 NOK.

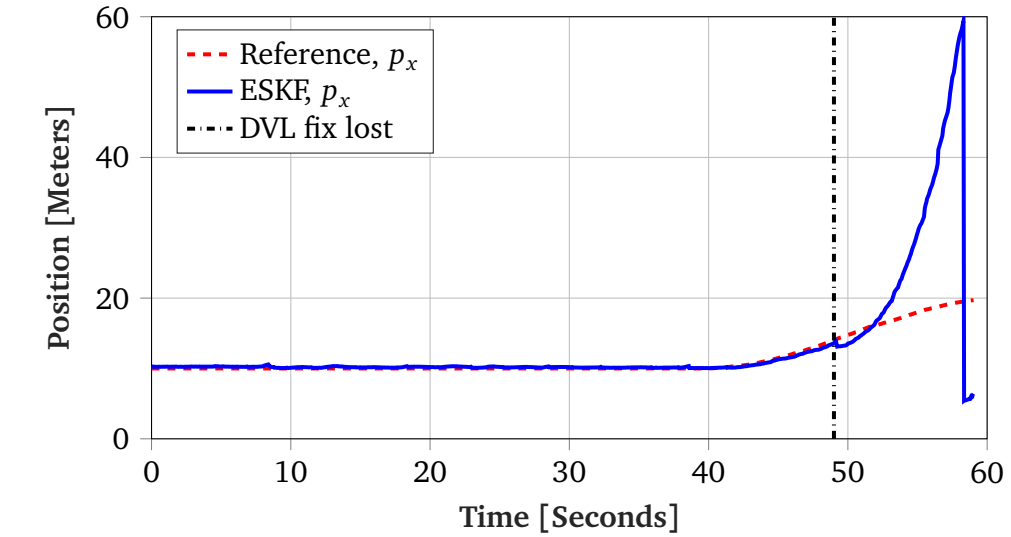


Figure 5.4: Loss of fix: Positional drift along x-axis after the DVL lose fix. Section of data set from Trondhjem Biological Station (TBS).

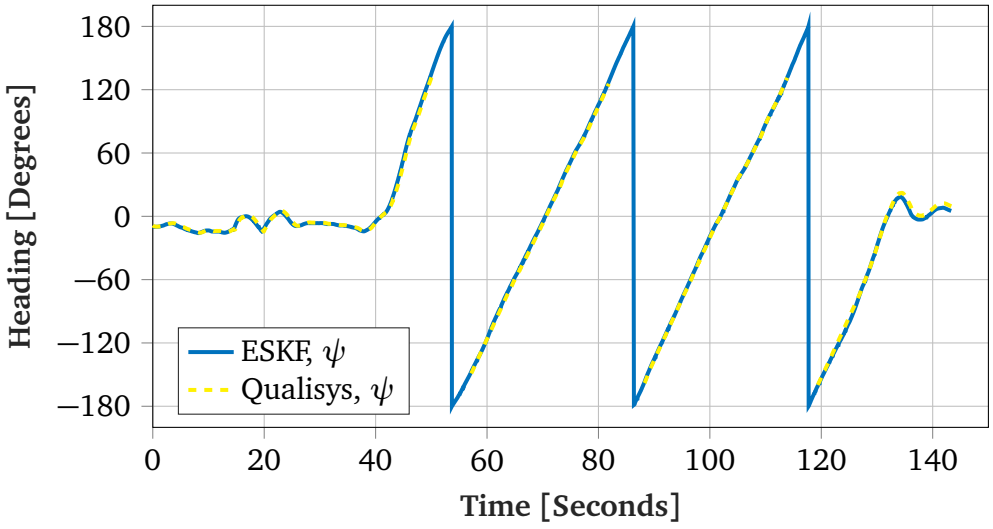


Figure 5.5: State estimate vs. ground truth: Resulting heading from execution of planned trajectory.

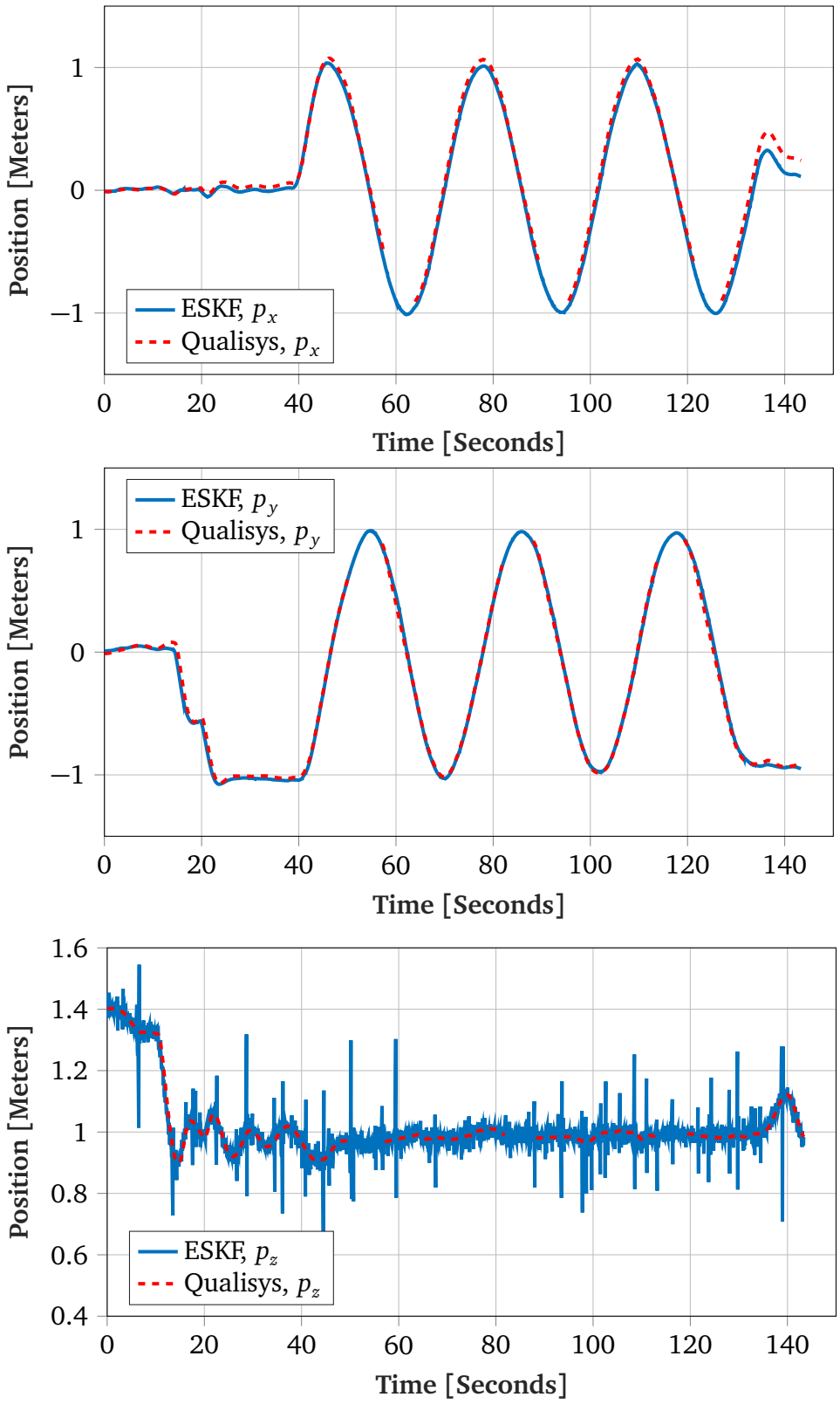


Figure 5.6: State estimate vs. ground truth: Resulting position in x, y, and z from execution of planned trajectory.

5.3 ESKF Consistency

To better understand the behavior of the error-state Kalman filter (ESKF), a filter consistency analysis is in order. For this purpose, normalized innovation squared (NIS) values related to measurements have been stored during a 2 minute run along with the respective innovations. Since the orientation estimates, velocity measurements, and depth measurements are all modeled as previously shown in Equation (3.12), the respective innovation should be normally distributed. Figures 5.7 to 5.9 shows histograms of all elements of the innovations related to orientation estimates, velocity measurements, and depth measurements. The histograms reveal how all innovations can be interpreted as normally distributed with a mean of approximately zero. However, the variance of the orientation and depth innovation distributions does not coincide with the measurement variance from Table 5.2. The standard deviation of the depth innovation was calculated to be $1.235 \cdot 10^{-2}$, which is higher than the expected measurement standard deviation of $2.468 \cdot 10^{-3}$. Meanwhile, the standard deviation of the orientation innovations appear to be significantly lower with values for q_w, q_x, q_y, q_z of $1.779 \cdot 10^{-6}$, $3.780 \cdot 10^{-5}$, $4.250 \cdot 10^{-5}$, $3.756 \cdot 10^{-5}$ compared to the expected measurement standard deviation of $5.236 \cdot 10^{-2}$. Consequently, the distributions do not exhibit the expected variance.

Squaring the assumed normally distributed innovations to form NIS metrics will result in χ^2 distributed random variables with degrees of freedom equivalent to the dimension s of the innovation. For orientation innovations, velocity innovations, and depth innovations, s is equal to 4,2, and 1, respectively. Adding N different realizations of the NIS metric will result in a new χ^2 distributed random variable with Ns degrees of freedom. Hence, the average NIS should obey a scaled χ^2 distribution. Figure 5.10 shows histograms of the respective NIS metrics related to orientation estimates, velocity measurements, and depth measurements. Once again, the histograms can more or less be interpreted as the expected χ^2 distributions. However, the averaged NIS for all cases are exceptionally large. Table 5.3 shows the averaged NIS, alongside with the upper and lower bound given by a confidence interval of 90% for each distribution. In all cases the average NIS is exceeding the upper bound, indicating a overconfident filter. All information coming from measurement updates are heavily prioritized, and the magnitude of the innovations are not proportional with their respective innovation covariances. Further tuning may improve the situation, however the inconsistency may indicate a faulty implementation and/or a poor filter model.

Table 5.3: Averaged normalized innovation squared (NIS) metrics, and lower/upper bound defined by 90% confidence interval for the respective distribution of NIS.

Averaged NIS	Lower bound	Upper bound	Computed value
Orientation, $\boldsymbol{q}$	3.906	4.095	38.788
Linear velocity, $\boldsymbol{v}$	3.902	4.098	149.087
Position, p_z	3.907	4.093	14.305

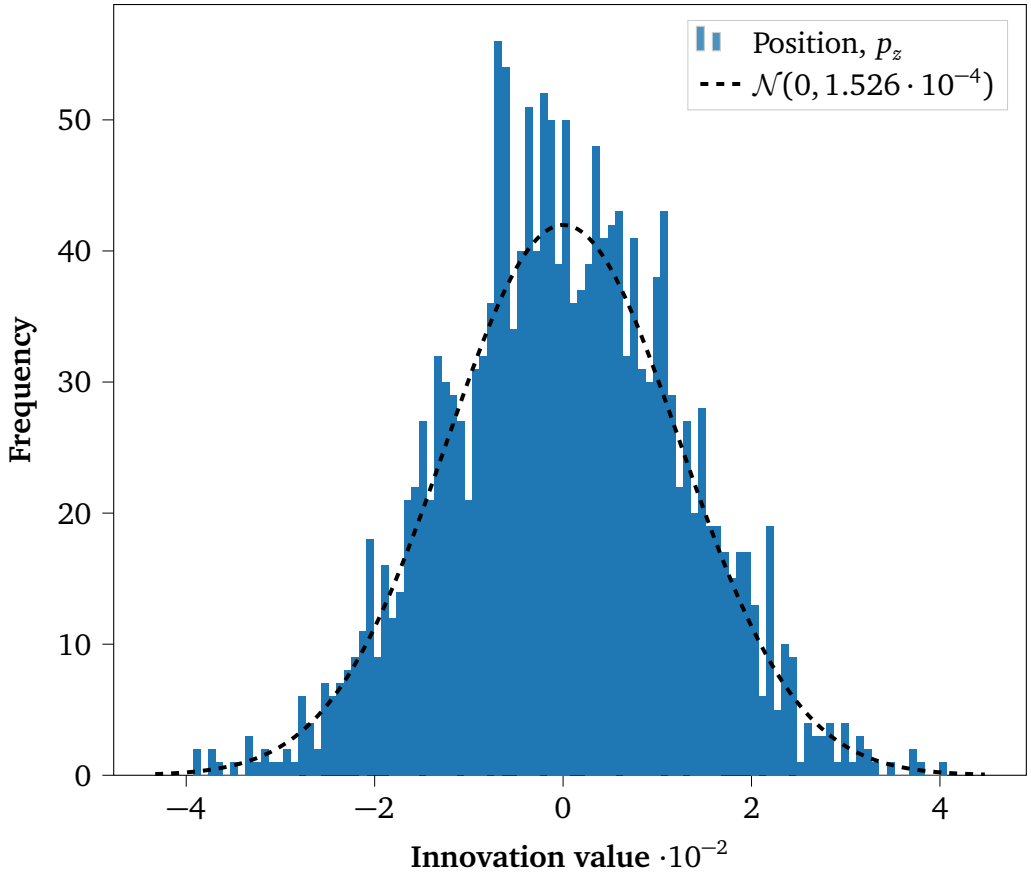


Figure 5.7: Histogram of innovation related to depth measurement containing 1766 innovations, alongside with the fitted normal distribution.

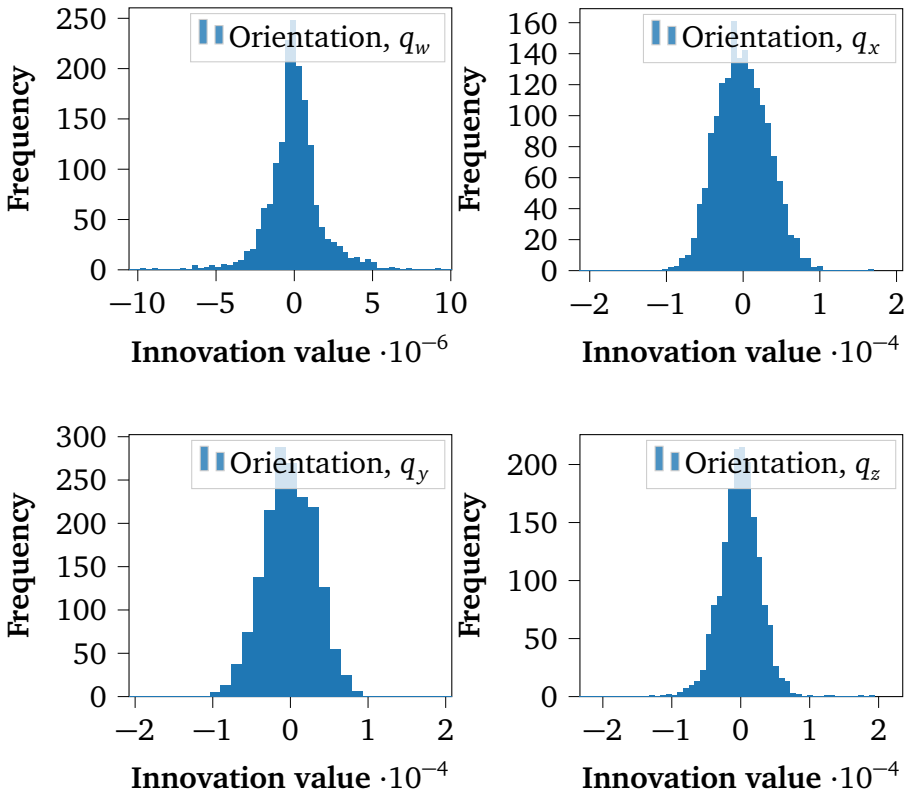


Figure 5.8: Histograms of innovations related to orientation elements q_w, q_x, q_y, q_z, containing 1708 innovations.

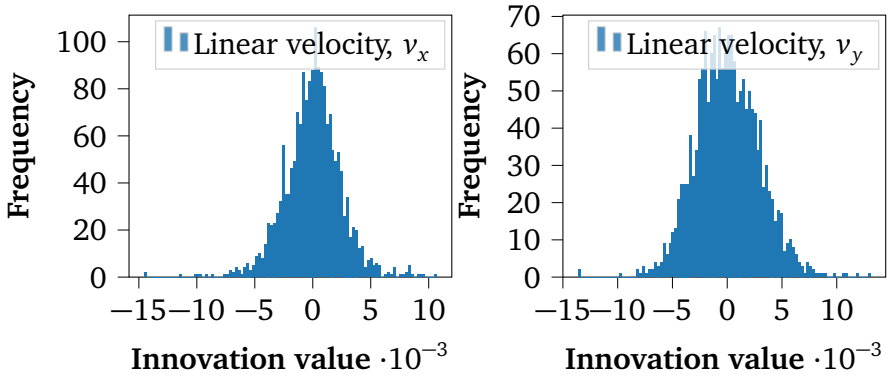


Figure 5.9: Histograms of innovations related to velocity elements v_x, v_y, containing 1664 innovations.

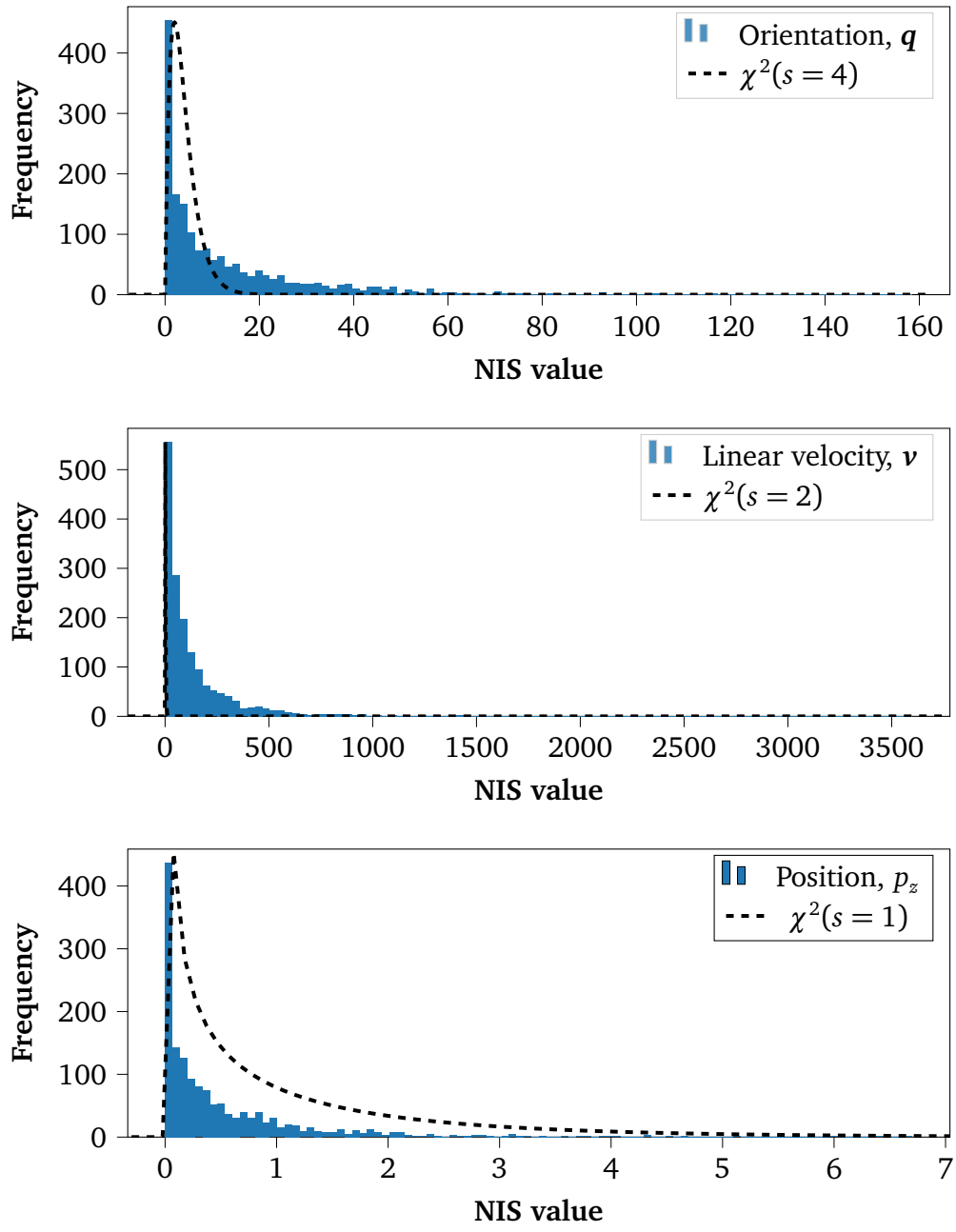


Figure 5.10: Histograms of normalized innovation squared (NIS) metrics related to orientation estimates, velocity measurements, and depth measurements, alongside with the fitted χ^2 distributions.

5.4 Benchmarking the Raspberry PI (RPI)

Operating underwater and pushing in the direction of autonomy, the Raspberry PI (RPI) must be able to run all separate parts of the system simultaneously. When performing field experiments this includes

- various sensor drivers (DVL, sonar, IMU, GPS, pressure sensor),
- the ESKF,
- the trajectory publisher,
- the control code (hysteretic controller and actuator driver),
- the camera stream,
- not to mention the logging of data from all separate parts of the system.

The RPI's system-on-chip may under sustained heavy workload reach dangerously high temperatures, at which point the operating speed will be reduced. This action of lowering the CPU clock's frequency due to high temperatures goes by the name of *thermal throttling*, and it indicates an overload of the RPI. Figure 5.11 shows the RPI's clock frequency and CPU temperature during a 1 hour and 15 minute long run from field testing in Ilsvika. The clock frequency never deviates from 1500 MHz, indicating no thermal throttling. However, the CPU reaches temperatures over $80°C$, and thermal throttling will occur when passing $85°C$.

To circumvent the temperature build-up, a better heat-sink solution can be employed. For now, the RPI rests against the enclosure made of cast acrylic plastic, although aluminium enclosures providing increased transferring of excess heat are available. However, the problems of high temperatures and thermal throttling are only to be considered symptoms of the sickness. Figure 5.12 shows the total CPU usage[5] through the same run, revealing that the RPI is being pushed to the limit of its processing abilities. The averaged total CPU usage throughout the full run is 96.53%, explaining the resulting high CPU temperatures. This problem can be addressed through optimizing the system, namely rewriting Python-implementations to achieve better efficiency. However, note that the RPI performs no processing related to sonar data for now. As computer vision is known to be computationally demanding, the final system may require more computing power than a single RPI can deliver alone. Introducing multiple computers can solve this, in addition to opening for the possibility of separating performance critical tasks such as control from other tasks.

[5]Total CPU usage constituting the average of all 4 cores at a given time.

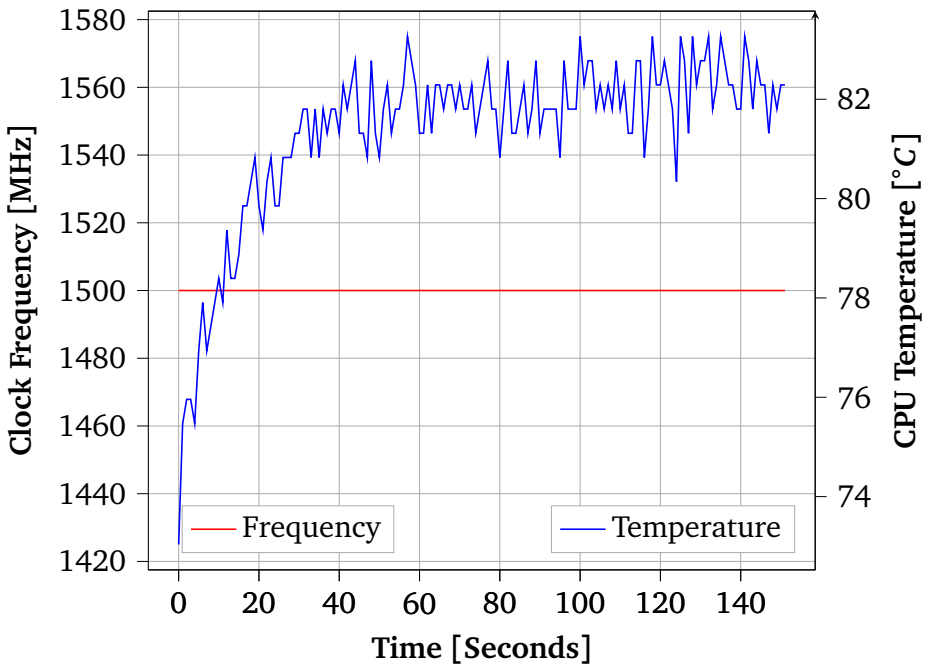


Figure 5.11: CPU clock frequency and CPU temperature of the RPI during field testing.

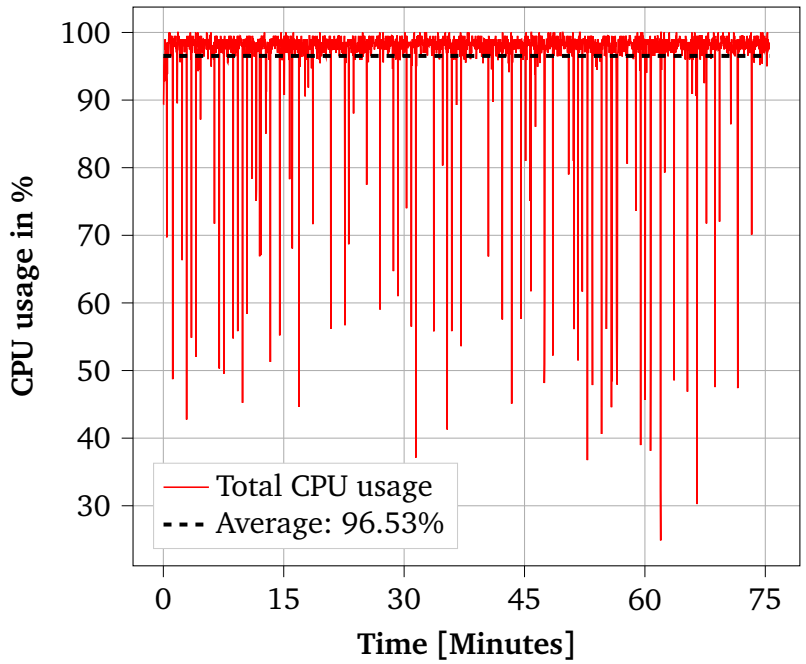


Figure 5.12: Total CPU usage of the RPI and its average during field testing.

Part III

To Hear Is To See!

Chapter 6

Primer on Acoustic Imaging

Acoustic imaging is the act of 'seeing with sound', where acoustic waves traveling freely[1] through fresh or salt water, tissue, and a variety of other materials are utilized to construct *acoustic images*. Acoustic waves will reflect from boundaries between different materials and provide information about changes in sound speed, density, and object shape. Consequently, acoustic images essentially becomes maps of density variations [51]. The technique of acoustic imaging remain particularly useful in contexts where imaging with electromagnetic waves is not possible or desirable. As such, the technique has been applied extensively to a wide range of fields including medical imaging, non-destructive testing, geophysical exploration, and underwater imaging [52].

This chapter will provide a brief introduction to acoustic waves and physical phenomenons related to sound propagation and attenuation, and the manipulation of sound to perform acoustic imaging.

6.1 Acoustic Waves

The description of acoustic waves and how such waves propagate in a medium constitute the premise of acoustic imaging. From a physics perspective, acoustic waves are considered mechanical vibrations transferring energy through the medium in which it passes. Stemming from the work of J. d'Alembert [53], general waves can be mathematically described by the two-way wave equation:

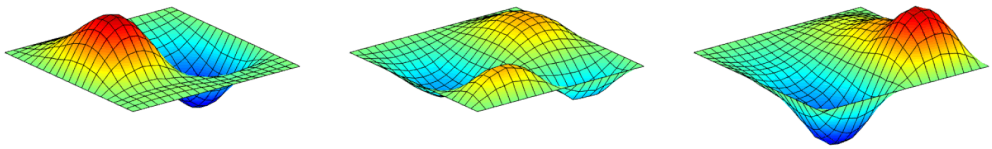
$$\frac{\partial^2 u}{\partial t^2} = k^2\left(\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \cdots + \frac{\partial^2 u}{\partial x_n^2}\right) \tag{6.1}$$

[1] Relative to electromagnetic waves.

Here, $u = u(x_1, x_2, \cdots, x_n; t)$ describes scalar functions of time and space, and k is a fixed non-negative real coefficient. A solution to the 2-dimensional case to the wave equation is visualized in Figure 6.1. However, physical solutions to the wave equation are usually obtained by specifying further conditions, such as initial and boundary conditions. The basic equations of acoustics in fluids can be formulated by considering the equations for an inviscid[2] and compressible fluid, using the principles of the momentum equation, the continuity equation, and the equation of state. Doing so while assuming constant entropy, no mean flow, linear elasticity[3], and constant density in space[4] results in the acoustic wave equation

$$\nabla^2 p' - \frac{1}{c_0^2} \frac{\partial^2 p'}{\partial t^2} = 0 \tag{6.2}$$

where ∇^2 is the Laplacian operator, p' is the pressure term dependent on time and spatial positions, and c_0 is the sound speed at the ambient conditions [54]. Note that changes to the operating conditions such as pressure, temperature, and salinity will affect the speed of sound. As such, varying conditions have a great impact on how acoustic waves interact with its medium.



(a) $t = t_1$ **(b)** $t = t_2$ **(c)** $t = t_3$

Figure 6.1: Visualization of a 2-dimensional solution to the wave equation at different time steps [55]. (Picturized; licensed under CC BY-SA 4.0)

6.2 Propagation and Attenuation

As acoustic waves propagate through a medium they will inevitably be subjected to various phenomenons affecting propagation energy and direc-

[2]Viscosity equal to zero.
[3]Linear relationship between the changes in pressure and volume.
[4]Not in time, and allowing for a small time and spatial dependent fluctuation term.

tion. The varying speed of sound mentioned in the previous section is such a phenomenon. Given two mediums with distinct properties resulting in different sound speeds, these will act as an interface. This interface would cause the direction of propagation to change, since acoustic waves follow Snell's law of reflection and refraction [56]. Having the angle of incidence θ_i, the reflection angle θ_R, and the angle of the through-transmitted wave θ_T, it follows from Snell's law that

$$\begin{cases} \theta_i = \theta_R \\ \dfrac{C_1}{\sin\theta_i} = \dfrac{C_2}{\sin\theta_T} \end{cases} \tag{6.3}$$

Here C_1 and C_2 are the speed of sound in the first and second mediums, respectively [57]. Snell's law and the change of propagation direction it describes is visualized in Figure 6.2. In practice, the angles can be calculated using pressure amplitudes of the incident, reflected, and through-transmitted wave, as well as acoustic impedances for the mediums. The varying sound speed is consequently something to be aware of when designing acoustic systems.

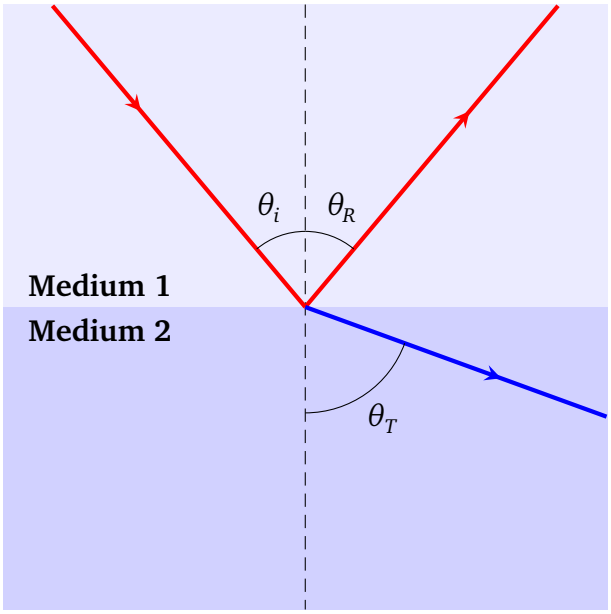


Figure 6.2: Visualization of Snell's law: A planar wave impinging upon the interface separating two mediums for which the speed of sound is different will be partially reflected and partially through-transmitted [57].

In addition to being subject of direction changes, the energy of acoustic waves are also attenuated as they propagate through a medium. Given a

point source, acoustic waves will spread out uniformly in all directions under the assumptions of an unbounded, iso-velocity, non-absorbing medium. Due to the conservation of energy, the intensity times the surface area at any range would have to be constant as visualized in Figure 6.3, resulting in the following expression

$$I(R) = \frac{I_0 4\pi R_0^2}{4\pi R^2} \tag{6.4}$$

Here, I_0 is the intensity at the reference range R_0, and $I(R)$ the intensity at range R. This is the so-called *inverse square law* for spherical sound waves [58]. Using the equation for transmission loss

$$TL = -10\log(\frac{I(R,D)}{I_0}) \tag{6.5}$$

and a reference range of $R_0 = 1$ meter results in the inverse square law spreading loss given by

$$TL = 20\log(R) \tag{6.6}$$

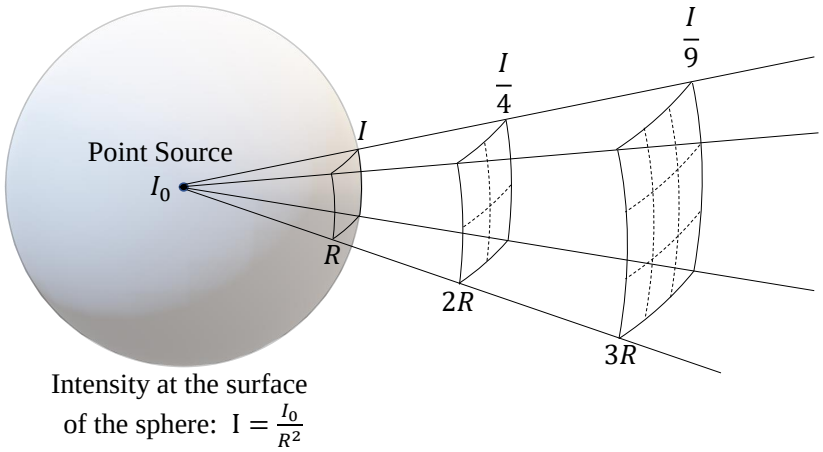


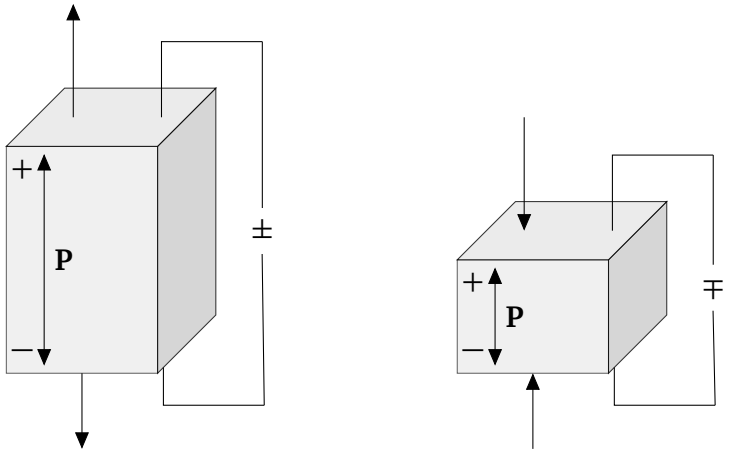
Figure 6.3: Visualization of the inverse square law for spherical sound: Given a point source, acoustic waves will spread out uniformly in all directions under the assumptions of an unbounded, iso-velocity, non-absorbing medium [58].

Another factor contributing to the attenuation is the dissipation of acoustic energy to viscous losses, heat conduction losses, and losses associated with internal molecular processes. Inhomogenities in the medium such as fog droplets, suspended particles, bubbles, thermal microcells, or regions of turbulence may also cause additional attenuation arising from

absorption and scattering [59]. Dependent on the situation the dissipation can be so slight that it may be neglected, as was done when stating the acoustic wave equation in Equation (6.2). However, ultimately all acoustic energy will be converted into random thermal energy. This heating of the medium as acoustic waves propagate is equivalent with an increase in entropy. Note how the assumption of constant entropy of which the acoustic wave equation depend does not hold in general [54].

6.3 Transducers and Imaging

Essential to acoustic imaging is the ability to generate, transmit, and receive suitable acoustic signals. Most acoustic sensing systems are commonly based on transducers made by piezoelectric materials. A piezoelectric element will produce an electric charge in response to mechanical stress. Furthermore, when subject to electric voltage a piezoelectric element will exert force on its surroundings. This effect is visualized in Figure 6.4. Consequently, this property allows for the transmission and reception of acoustic signals [54].



(a) Subject to electric voltage. **(b)** Subject to mechanical stress.

Figure 6.4: Visualization of the piezoelectric effect: A piezoelectric element will produce an electric charge in response to mechanical stress, and exert force on its surroundings in response to an applied electrical field [54].

The number and arrangement of transducers in an acoustic sensing system will depend on the application, as will the acoustic pulse selected to acquire information. Often, multiple transducers are organized in arrays with specific directivity patterns, offering flexibility in beam steering

and dynamic focusing. The frequency, amplitude and length of the acoustic pulse governs the range and resolution achievable for an acoustic sensing system. In general, high frequencies and short spatial pulse lengths will yield narrow echoes which can be better resolved resulting in higher resolution. However, long spatial pulse lengths tend to yield better range. The compromise of choosing a suitable acoustic signal will thus depend on the application [57].

Transmitting and receiving acoustic signals can be done using both pulse-echo and through-transmission techniques. However, for practical reasons pulse-echo imaging is often preferred. Once an acoustic signal has been both emitted and received by the transducers, the raw measurement can be used to generate an acoustic image. The basic process of converting raw measurements into displayable images consist of three fundamental steps [57]: Firstly, the raw data should be filtered to remove noise components not stemming from the acoustic process. Secondly, the signal's envelope is to be extracted through the use of the *Hilbert transform*. Through shifting all phase angles by 90 degrees and taking the absolute value, the signal will be substantially smoothed allowing for a better visual representation. Lastly, the processed amplitude signal must be converted into pixels utilizing spatial coordinates and dynamic range to construct the acoustic image. Here, spatial coordinates are found using travel time of the signal, Time Of Flight (TOF), combined with approximations of the sound speed. The resulting image will contain information about the medium of which the acoustic signal has passed through.

To summarize This chapter has provided a brief introduction to the behavior of sound, and how it can be manipulated to perform acoustic imaging. However, the field of acoustic theory is vast and this chapter is only scratching the surface of a few selected topics with the main takeaway being that the real world is *complex* and challenging to accurately model. This complexity warrants the use of approximations and simplifications when performing acoustic imaging, as will be the case with the technology of the Side-Scan Sonar (SSS).

Chapter 7

Side-Scan Sonar (SSS)

Performing acoustic imaging in underwater environments is commonly done using SOund Navigation And Ranging (SONAR) technology. Through the means of sound propagation, SONAR can be applied to navigation, ranging, communication, and detection of objects on or under the surface. The first successful underwater transducer was developed by R. Fessenden in 1912, and through continuous research incentivized by both war and exploration the technology has matured into the acoustic systems of today [60]. There are commonly two categories of sonar systems; passive systems made to only receive signals, and active systems emitting signals and listening for echoes. Among the active systems there exist a wide assortment of solutions, such as forward facing sonar, synthetic aperture sonar, towed array sonar to mention a few. For efficient mapping of large areas of the sea floor, the *Side-Scan Sonar (SSS)* stands out as a convenient tool.

This chapter will introduce the technology of Side-Scan Sonar (SSS), alongside some of its working principles and signal types. In addition, a basic acoustic image will be constructed from SSS data.

7.1 Working principles

A standard Side-Scan Sonar (SSS) setup consist of two sensing heads commonly mounted symmetrically on port and starboard side of a vehicle or a tow fish. The angle at which the sensing heads are mounted, θ, is fixed. Each sensing head is made up of multiple transducers functioning as transmitters and receivers of acoustic signals. Relying on sound propagation, the sensing heads periodically emit acoustic pulses to reach objects or the sea floor, before listening for backscatter energy. Consequently, each sensing head will record echo intensities into a data vector at fixed time intervals. This data vector is often referred to as a *swath*, and its elements contain-

ing information about object and sea floor reflectivity are so-called *bins*. The shape of swaths acquired by each sensing head are direct results of the sensor openings, α and ϕ, respectively modeling how sound expand in the YZ and XY planes [61]. Commonly, swaths are considered thin lines along-track covering large areas across-track due to a relatively large α and equally small ϕ. The characteristics of a SSS sensing head is visualized in Figure 7.1.

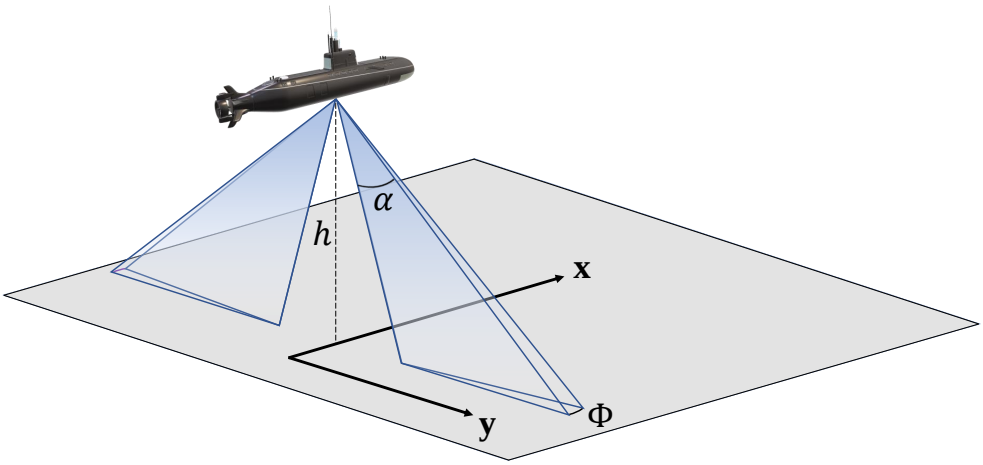


Figure 7.1: Visualization of the SSS sensing heads: The angles α and ϕ are known as the sensor openings respectively modeling how sound expand in the YZ and XY planes [61].

Under the assumption of sound propagating along straight paths, effectively disregarding the possible effect of Snell's law, the Time Of Flight (TOF) corresponding to each bin will determine the *slant range* r_s. Moreover, the SSS range depends on time elapsed between emitted pulses and the sampling period of the acoustic signal determines the slant range resolution δ_s. Note how each bin ultimately correspond to the diagonal distance of r_s rather than a direct horizontal distance on the sea floor.

An important aspect of SSS systems is the separation of energy originating from the signal of interest from energy related to unwanted noise. In general, the portion of the acoustic energy reflected back will be of several orders of magnitude lower. The attenuation of acoustic signals undeniably affect the amount of backscatter. However, the geometry of the sensor-target system is of greater importance [56]. Dependent on the angle of incidence, most of the acoustic energy will reflect in a specular direction as visualized in Figure 7.2. Moreover, energy may be lost in the sea floor, as well as reflected along other angles. Relating received signal power to transmitted signal power for two-way propagation is done through the *ac-*

tive sonar equation. The signal-to-noise ratio often measured in decibel is formulated as

$$SNR = SL - 2TL - (NL - DI) + TS \tag{7.1}$$

where SL is the transmitter's source level, $2TL$ is the two-way transmission loss, NL is the noise level, DI is the directivity index[1], and TS is the target strength [54].

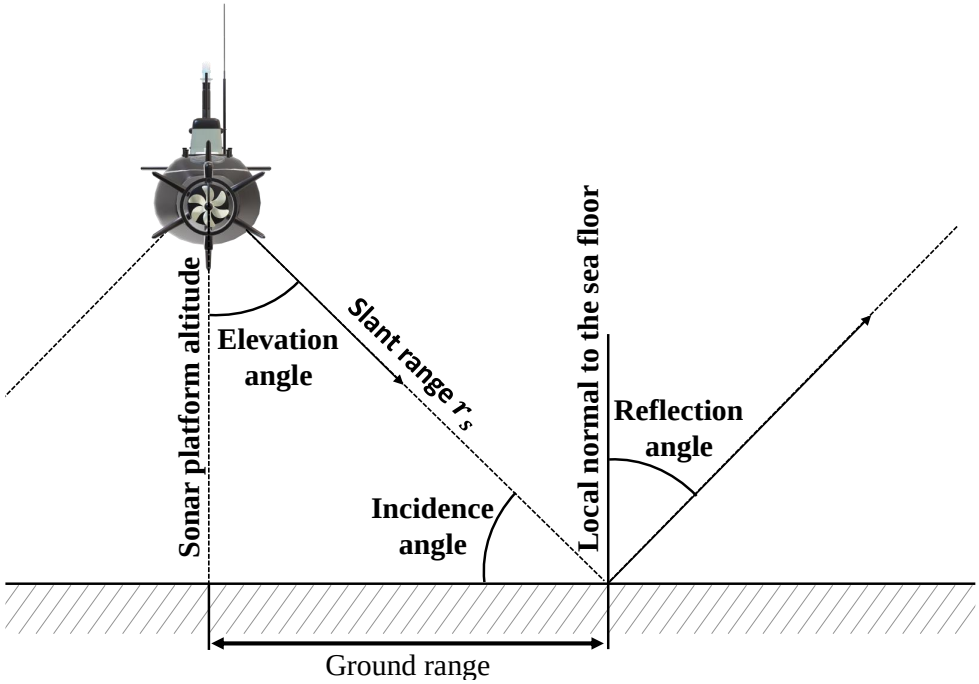


Figure 7.2: Visualization of specular scattering of Side-Scan Sonar (SSS) signal. Only a small portion of acoustic energy will scatter back and be received by the SSS [56].

7.2 Continuous and Pulsed Signals

The sonar equation accentuates how active sonar systems are designed to improve the signal-to-noise ratio, in consequence improving the imaging capabilities. To this extent, the properties of the signal that is used in the sonar system will be defining for the system's performance. Active sonars may use either a continuous or a pulsed signal, where the frequency, amplitude and shape of the signal can differ depending on the use case. Traditionally, pulsed signals have been used when designing sonar systems,

[1]Ratio of the total noise power emitted and received along the main response axis.

although the scientific interest in the use of continuous signals has been growing along with advances in hardware and computing power. Continuous signals allow for maximization of pulse energy through a 100% duty cycle [62]. However, the concept of continuous active sonars are mostly motivated by the use case of target tracking in anti-submarine warfare, and will not be investigated further.

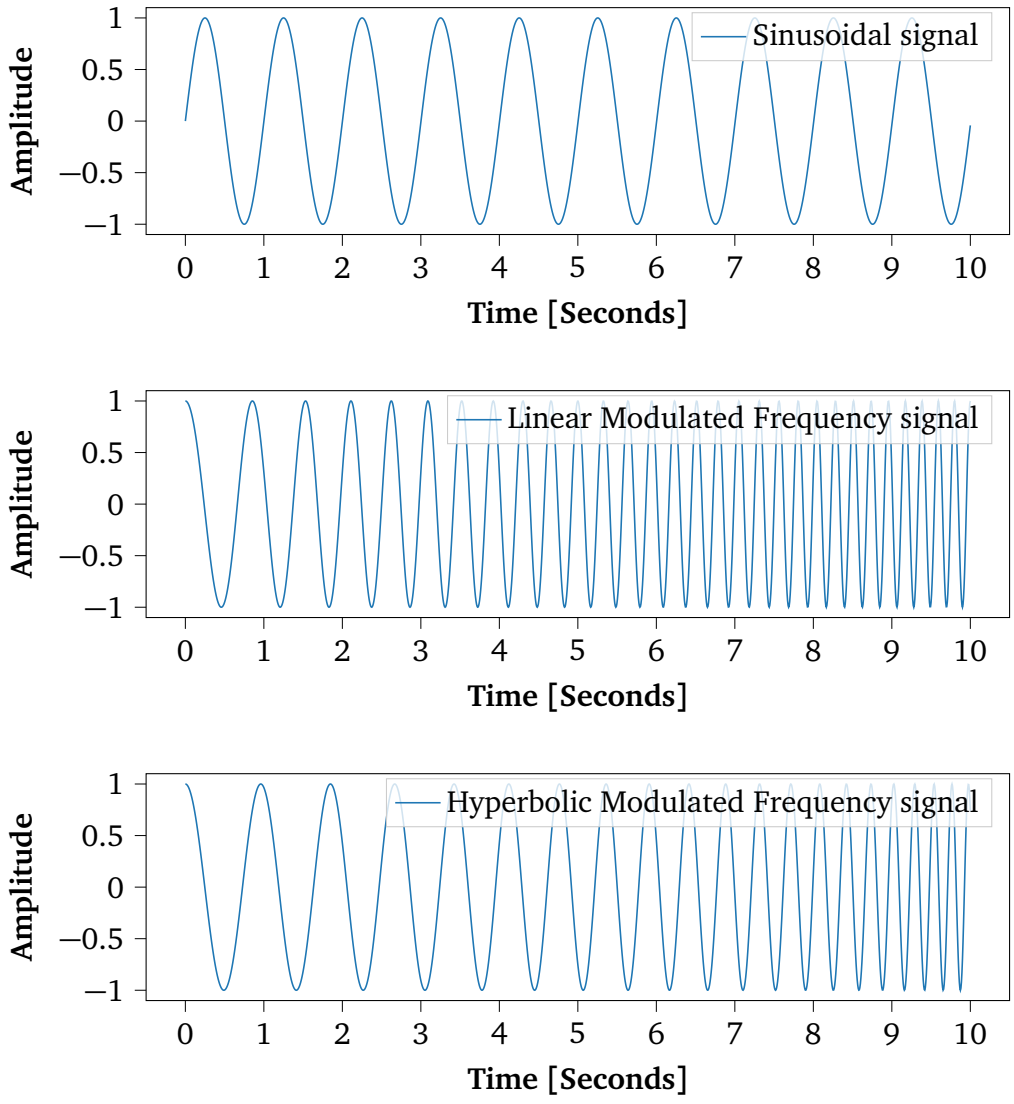


Figure 7.3: Visualization of sinusoidal, linear modulated frequency, and hyperbolic modulated frequency signals.

Among pulsed signals, most active sonar systems rely on either short sinusoidal or frequency modulated signals, and the choice is application

dependent. A simple example can be found in the *constant amplitude sinusoidal waveform function*, which suppresses reverberations well in environments with Doppler induced scatter [63]. However, sinusoidal signals provide relatively poor range resolution relative to frequency modulated signals. The *linear frequency modulated pulse*, better known as *chirp*, provides long range and high resolution. Furthermore, the chirp has a strong resistance to white Gaussian noise on account of its self-correlation characteristics [64]. As such, it is well suited for SSS applications. The *hyperbolic frequency modulated pulse* has similar properties to the chirp, in addition to the ability to also suppress Doppler induced scatter. See Figure 7.3 for a visualization of all three signals. Other signals such as the pseudo random noise BPSK[2] coded and the ricker are also available. Nevertheless, the chirp is often the signal of choice in SSS systems.

7.3 Imaging using Side-Scan Sonar (SSS)

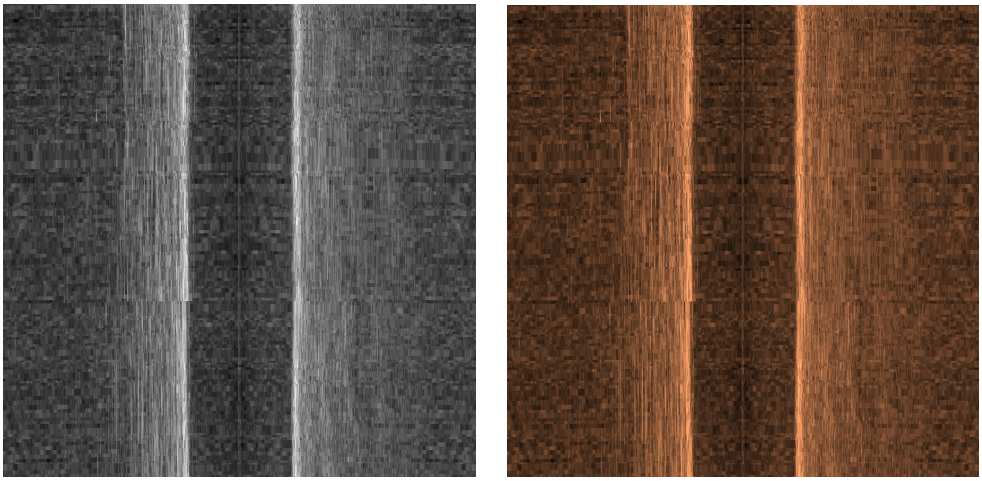
So far, physical phenomenons, working principles, underlying assumptions, and acoustic senors and signals related to SSS technology has been addressed. Henceforth, this foundation will be used to interpret acoustic images constructed from SSS data, and explicate the need for processing of these images for them to usable for navigational purposes.

A basic acoustic image can be constructed through simply stacking consecutively gathered swaths. Figure 7.4 shows such acoustic images where each row correspond to a single swath and the echo intensities are mapped to a gray scale and a copper colored scale, respectively. Further, dark and bright pixels indicate low and high echo intensities. Traditionally, the copper colored scale have been utilized to visualize echo intensity in the literature. However, the gray scale mapping provides higher contrast and more detailed images[3], and is therefore the preferred mode of visualization. The black strip along the center line of the acoustic image is referred to as the *blind zone*. This region corresponds to distances from the transducers where no sea floor is detected, and it should not be confused with the *nadir*; the region directly beneath the ROV not ensonified by the sonar.

The basic acoustic images constructed by consecutive swaths are not suited for navigational purposes. This is because the image without further processing does not accurately represent the sea floor due to intensity variations and geometric distortions. The pipeline addressing these problems is presented and evaluated in Chapter 8 and Chapter 9, respectively.

[2]Binary Phase Shift Keying

[3]In the eyes of the colorblind author.



(a) Gray scale. **(b)** Copper scale.

Figure 7.4: Basic acoustic image of consecutively gathered swaths. Each row correspond to a single swath, and the echo intensities are mapped to two separate color scales.

Chapter 8

Processing of Sonar Data

The Side-Scan Sonar (SSS) provides weak and ambiguous sensory information, where data easily becomes altered as a result of multiple environmental and situational factors. To enable the use of computer vision methods for navigational purposes, processing of sonar data is imperative. In this project, the image construction pipeline consists of performing intensity normalization, blind zone removal, slant range correction and geometric correction, as visualized in Figure 8.1. This chapter will present each step of the pipeline, addressing the respective problems to solve, alternative solutions and the specific methods implemented.

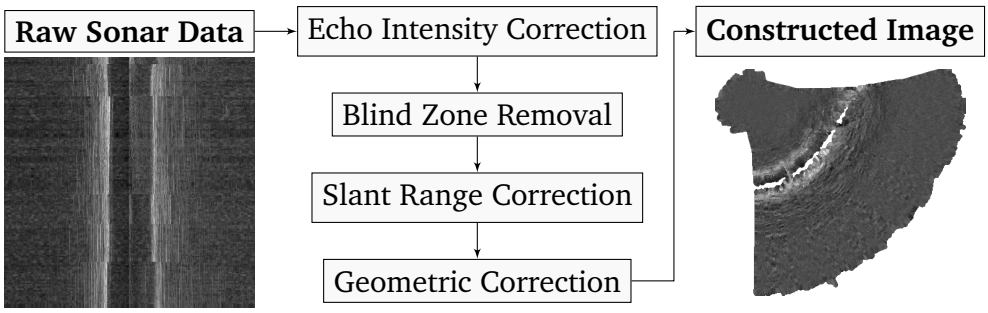


Figure 8.1: Visualization of the image construction pipeline transforming raw sonar data into a processed acoustic image.

8.1 Echo Intensity Correction

The phenomenon of uneven ensonification patterns and brightness variations in sonar images are referred to as echo decay. The most noticeable

visual artifact arising from echo decay is the presence of an artificial gradient of intensity values, as seen in Figure 7.4. This intensity variation poses a challenge as landmark detection relies exclusively on the backscatter energy to recognize landmarks. Depending on range and direction of the sonar head relative to the sea floor, a single landmark may appear different in the acoustic image, further complicating the task at hand [65].

The majority of the echo decay can be attributed to the high absorption rate of sound by the water combined with the spreading loss of sound described by the inverse square law for spherical sound waves, as described in Section 6.2. The attenuation of acoustic signals will naturally become more severe the further the signal travels. Although not as prominent, the gain of the receiver's sensor is also a factor affecting all swaths, inducing brightness variations in the acoustic images. The highest values are achieved when the received echo is perpendicular to the sensor, while echos received at an angle will appear with less strength. Consequently, the strength of the received echo will depend on the angular sensor placement θ and the grazing angle. All previously mentioned factors combined render single swaths to have a distribution close to a Rayleigh distribution, as can be seen in Figure 8.2.

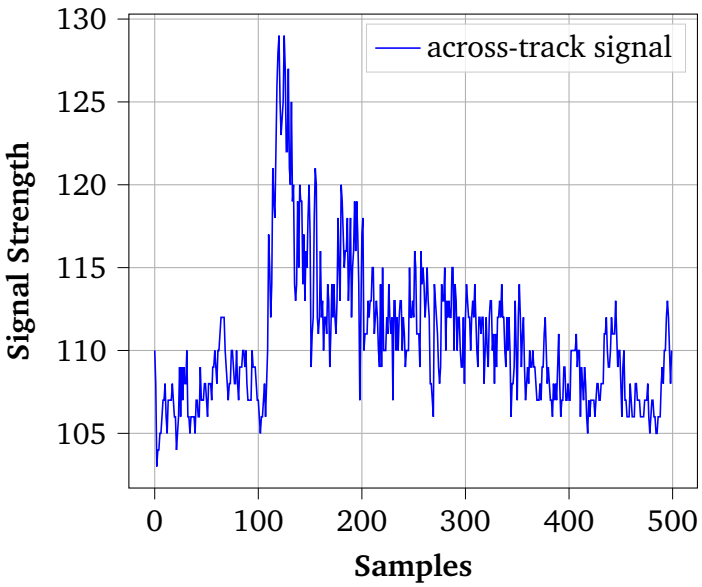


Figure 8.2: Swath from right transducer with a shape reminiscent of a Rayleigh distribution.

The problem of echo decay is well known, and several solutions have been proposed to deal with the resulting brightness variations in the acoustic images. A simple solution is to compress the dynamic range of the in-

tensity values through applying a logarithmic transform, $log(x)$. As mentioned by Al-Rawi et al. [65], log transformed images will have a higher across-track coefficient of variation indicating lower image quality. In the literature, normalization is often handled through the technique of Time Variable Gain (TVG), where the receiver's amplification is adjusted according to the transmission loss expected as a function of distance or time after pulse transmission. Needless to say, accuracy will depend on whether the actual transmission loss is equal to the loss included in the gain control [54].

In addition to TVG, a rich selection of normalization techniques is available when handling the echo decay. Burguera et al. [61] model the sea floor as a Lambertian surface and normalize acoustic images using a sensitivity pattern model. Significant work has also been carried out as part of the SWARMs project (`http://www.swarms.eu/`), where normalization is handled using methods such as Dark Channel Prior (DCP) [66], MIxed exponential Regression Analysis (MIRA) [66], and non-uniformity correction techniques [67].

In this project, the approach of image enhancement through cubic spline regression, as proposed by Al-Rawi et al. [65], is adopted to perform the echo intensity correction. Through the process of polynomial fitting, the underlying distribution of intensity values in single left and right swath are estimated separately. This is done using a smoothing spline f minimizing

$$p \sum_{j=1}^{n} |y_j - f(x_j)|^2 + (1-p) \int |D^2 f(t)|^2 dt \tag{8.1}$$

where the first term is the *error measure* and the second term is the *roughness measure*. Here, n is the number of entries in x and y describing the samples and the intensity values in a swath, respectively. The integral is over the smallest interval containing all the entries of x, while $D^2 f$ denotes the second derivative of the function f. Lastly, p is the smoothing parameter ranging from 0 to 1 weighting the error measure versus the roughness measure where the bounds are:

- 0: The smoothing spline is the least-squares straight line fit to the data
- 1: The natural cubic spline interpolant

As such, the cubic smoothing spline can be used as an effective data modeling tool for noisy data and is well suited for estimating the swath distribution. Further, samples from this distribution are used as weights to normalize the across-track signal profile using the general enhancement

model

$$\hat{I}(z) = \frac{I(z)}{w(z)} \tag{8.2}$$

where $I(z)$ is the signal value (at location z) that has been acquired by the sonar, $w(z)$ is the weight value estimated through polynomial fitting for each location z, and $\hat{I}(z)$ is the normalized signal value [65].

The main advantage of the cubic spline regression based enhancement is how normalization can be performed in-situ, making the technique applicable for online processing as swath measurements arrive. Furthermore, normalization techniques based on across-track signal profiles may be preferred when processing side-scan sonar images, since the across-track signals are highly correlated compared to along-track samples. As previously mentioned, swaths are close to Rayleigh distributed and across-track signals are consequently not suitably modeled by exponential shapes as done by MIRA. Thus, utilizing cubic spline regression to estimate the weights $w(z)$ will improve correction of non-uniformities, as well as more accurately address the effect of the sensor gain [65].

8.2 Blind Zone Removal

Intensity values in the blind zone attribute to sensor noise, and the presence of suspended particles and other possible objects in the water column. Changes in the width of the blind zone reflect changes in the altitude of the sonar head. Due to the working mechanics of the side-scan sonar, higher altitudes results in a wider blind zone. As the blind zone does not hold any useful information for navigational purposes it is of interest to detect and remove it, thus performing the process of *blind zone removal* [61].

Detecting the edge of the blind zone is equivalent to detecting the first significant echo corresponding to the first sea floor point producing an echo. This specific echo is the so-called First Bottom Return (FBR). Thus, all echoes arriving before the FBR in a swath will be part of the blind zone, making detection of the FBR an essential part of blind zone removal. There exists numerous of methods for doing bottom detection, to mention a few: Al-Rawi et al. [68] proposed detection through the use of both cubic spline regression and moving window average filtering, Yu et al. [69] proposed detection through a combination of threshold methods, denoising algorithms, spatial-temporal matching and extreme value detection, while more recently Zheng et al. [70] proposed automatic detection based on semantic segmentation.

If the platform altitude h is available at each time step, the need for advanced bottom detection schemes can be alleviated. As presented by Bur-

guera et al. [61], the altitude h and the ground range r_{FBR} corresponding to the First Bottom Return (FBR) are related through the angular sensor placement θ and the sensor opening α as follows

$$r_{FBR} = \frac{h}{\tan(\theta + \frac{\alpha}{2})} \tag{8.3}$$

under the flat-floor assumption as visualized in Figure 8.3. Given a symmetrical set-up of the separate sonar heads, the blind zone removal can be performed through rejection of data points whose ground ranges lie in the interval of $[-r_{FBR}, r_{FBR}]$ as these contain no useful information. This approach is adopted to perform blind zone removal due to its simplicity, and on the account of altitude measurements being available from the DVL. The specific sensor placement and sensor opening values used are specified in Section 4.1.

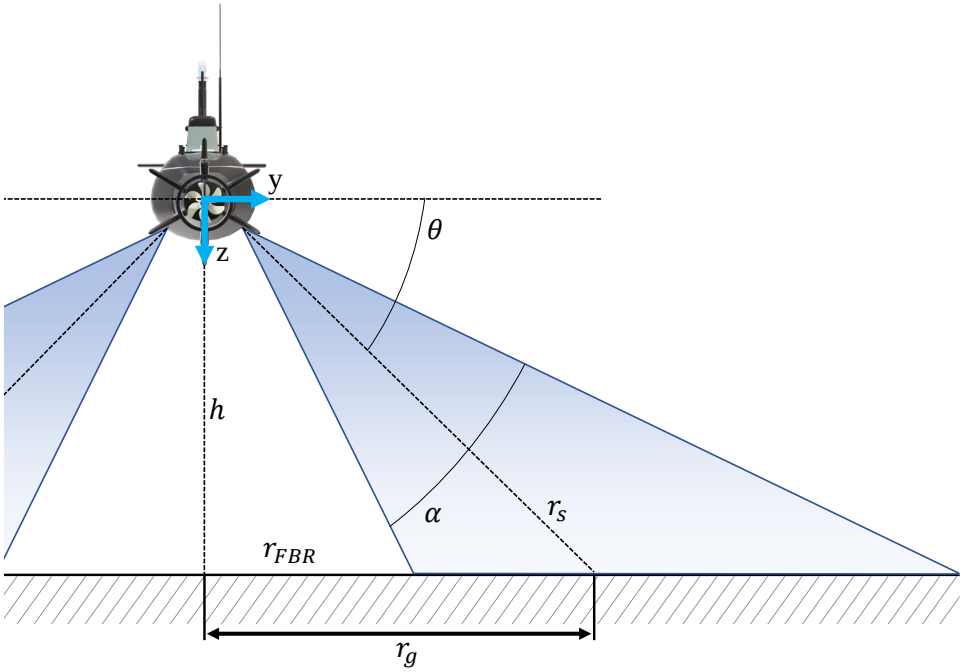


Figure 8.3: Visualization of the First Bottom Return (FBR) alongside with other SSS characteristics: h constitutes platform altitude, θ the angular sensor placement, α the sensor opening, r_s the slant range, and r_g ground range.

8.3 Slant Range Correction

Inherent to the side-scan sonar is the slant range effect which can be considered a geometric distortion. This effect is a consequence of the fact that across-track coordinates originate from the sonar slant range rather than the horizontal distance on the bottom as emphasized in Chapter 7. The process of computing the ground range for every slant range signal is referred to as *slant range correction*. Under the flat-floor assumption, the slant range correction becomes a simple geometric problem as visualized in Figure 8.3. The ground range r_g can be computed for every slant range r_s in a single swath through

$$r_g(r_s, h) = \sqrt{r_s^2 - h^2} \tag{8.4}$$

where h is the platform altitude at the moment of emission [71]. However, as pointed out by Burguera et al. [61] it is preferable to have slant range as a function of ground range to avoid gaps in the corrected swath. Consequently, slant range correction will be performed through implementation of Algorithm 2.

Algorithm 2 Slant range correction.

Input: `Input swath, corresponding altitude, swath resolution`
Output: `Corrected swath`

procedure SLANTRANGECORRECTION(I, h, σ_s)
  for `all bins` b `in swath` I **do**
    $\hat{b} \leftarrow \frac{\sqrt{(\sigma_s \cdot b)^2 + h^2}}{\sigma_s}$
    $w_1 \leftarrow \hat{b} - \lfloor \hat{b} \rfloor$
    $w_2 \leftarrow 1 - w_1$
    $I_g(b) \leftarrow w_2 \cdot I(\lfloor \hat{b} \rfloor) + w_1 \cdot I(\lceil \hat{b} \rceil)$
  return I_g

8.4 Geometric Correction

Another source of geometric distortion is variation in trajectory, speed, or orientation of the sonar head. Such variations produce discrepancies between the relative location of features in the acoustic image and the true feature locations on the sea floor. Ideally, the sonar head should be moving above the sea floor at a constant speed, on a straight path, with the heading aligned with the trajectory. These are not reasonable constraints for an UUV performing various underwater operations, thus the sonar head

will be exposed to certain motion instabilities. As specified by Cobra et al. [72], motion instabilities may be divided into two types: translational and rotational.

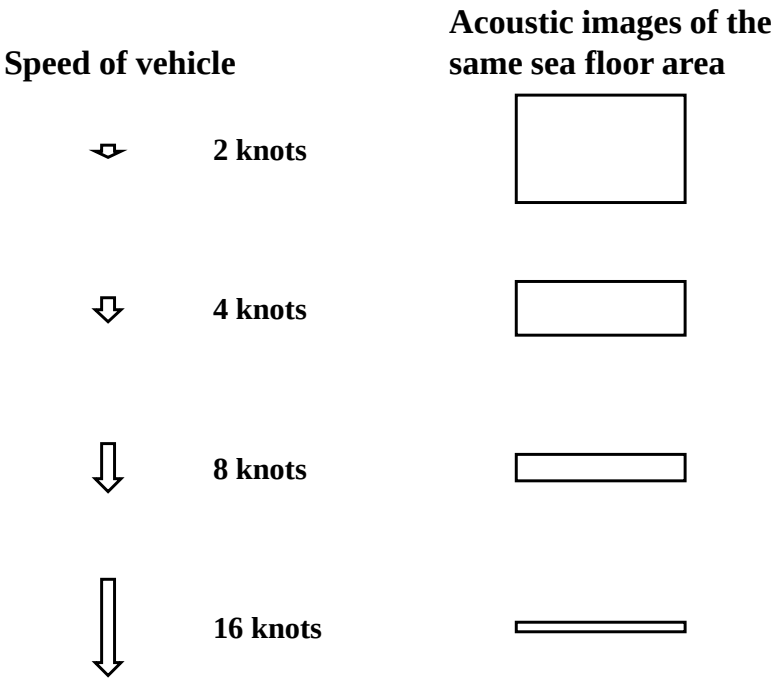


Figure 8.4: Visualization of how speed variations cause changes to the aspect ratio of the resulting acoustic image.

Translational instabilities correspond to speed variations, as well as lateral and vertical displacements from the ideal straight path. Forming an acoustic image by stitching consecutive swaths may in the presence of such instabilities cause large-scale distortions. Further, speed variations will cause changes to the aspect ratio of the resulting image, as seen in Figure 8.4, due to the constant scanning rate of the side-scan sonar. Thus, small objects will appear wide and large upon speed reduction, while speeding up may shrink or even conceal objects. In addition, high speeds may cause black gaps in the processed image corresponding to unscanned regions. Such regions indicate a mismatch between the sonar scanning rate and the speed of the vehicle.

Rotational instabilities correspond to deviations from the ideal orientation of the sonar head, where the heading of the vehicle is aligned with the trajectory and the transducers are pointing toward the sea floor with no variations in pitch, roll, or yaw. Pitching and yawing may cause geometric distortions as a result of double scanning, while rolling may cause

intensity distortions due to the beam rotating within its own plane. Objects in a backscanned area will appear in triplicate in the acoustic image, consequently complicating the use of computer vision. Rotational instabilities are illustrated in Figure 8.5.

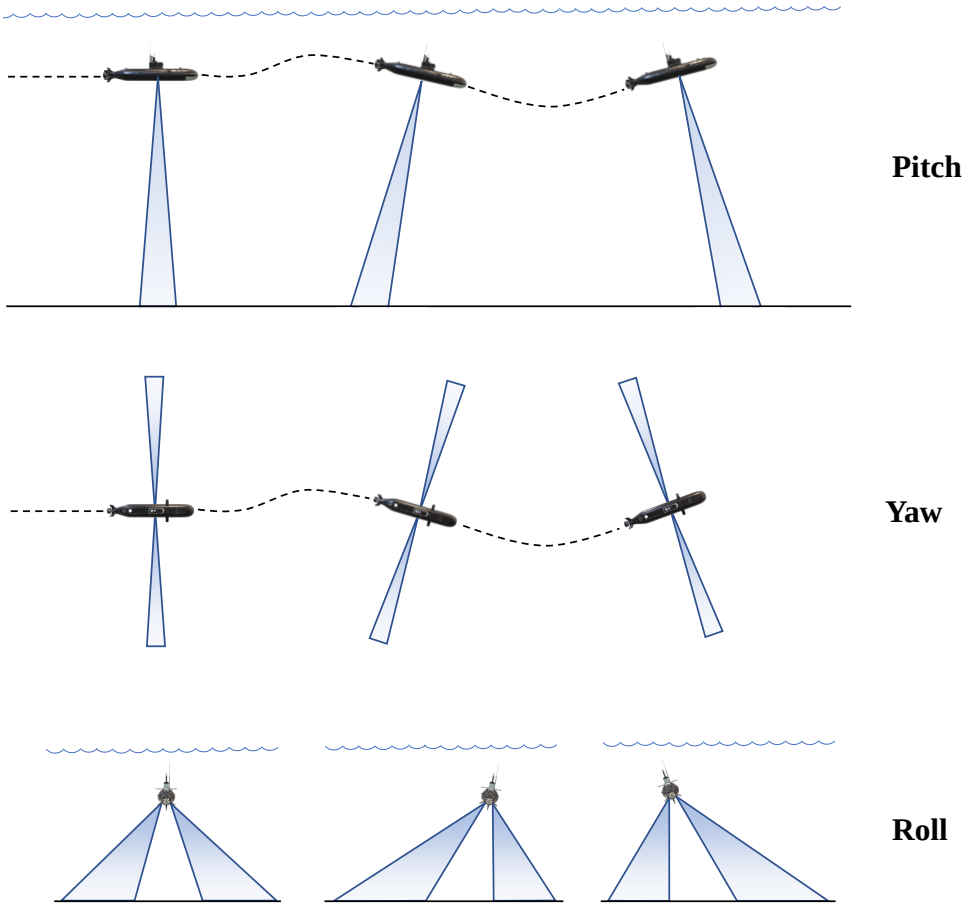


Figure 8.5: Rotational instabilities of the sonar head: pitching and yawing produce geometric distortions by causing the beams to scan ahead or back, either simultaneously on both sides, in the case of pitching, or alternately on the port and starboard sides, in the case of yawing; rolling produces intensity distortions due to the beam rotating within its own plane [72].

Depending on the amount of motion instabilities, the severity of the resulting geometric distortions may render the use of computer vision infeasible. Hence, it is of interest to perform geometric correction as part of the processing pipeline. Performing along-track corrections to account for speed variations can be done through the process of *anamorphosis*. Striving to produce an acoustic image where inter-pixel spacing is equal in both along-track and across-track, the along-track spacing is determined by ei-

ther the width of the horizontal beam on the ground or the distance traveled by the transducer during the reception interval, whichever is smallest. Successive lines are thus replicated or sub-sampled depending on the vehicle speed, resulting in an image with a 1:1 aspect ratio. Other solutions include UUV localization or SLAM techniques to account for motion between swaths: Ye et al. [73] perform speed correction using GPS information, while Teixeira et al. [74] model the complete reconstruction using factor graphs in tandem with the SLAM problem.

However, the BlueROV2 utilized in this project is far less stable in orientation in relation to surge speed compared to an Autonomous Underwater Vehicle (AUV). Furthermore, when accounting for rotational instabilities the system is not restricted to straight sections which enables extraction of more information. Since all geometric distortions resulting from motion instabilities are of a geometric nature, the proposed method from Sheffer et al. [71] of exploiting the geometric relations for correction purposes will be adopted. Hence, correction of geometric distortions and the reconstruction of sonar images will be done using state estimates. By combining information about the location, speed and orientation of the platform, each swath is located in a specific geometrical position in the acoustic image under the flat floor assumption. Thus, 3-dimensional sonar samples are mapped onto a 2-dimensional map creating a geocorrected acoustic image. Note, the geometric correction only accounts for platform movement between separate swaths, consequently disregarding the movement occurring as a single swath is gathered.

Accounting for translational instabilities, the location and speed corrections are performed through projection of the integrated location of the n^{th} swath, $PL_n \in \Omega$, into the processed sonar image. For every swath n, this projection defines the location of the platform in the sonar image at the moment of emission. The integrated location of the n^{th} swath corresponds to the positional estimate $\boldsymbol{p}$ in Equation (3.7a) provided by the error-state Kalman filter (ESKF).

Geometric distortion resulting from rotational instabilities are handled using the inherent geometric relations, where every swath is mapped from the platform's frame of reference to the sonar image through a linear transformation. Under the flat-floor assumption, the altitude coordinates of all samples have the same value in the platform's frame of reference as all samples are perpendicular to the platform [71]. Thus, the coordinates of a sample i in swath n can be written as

$$L_{n,i} = \begin{bmatrix} 0 \\ r_g(i) \\ h_n \end{bmatrix} \tag{8.5}$$

where $r_g(i)$ is the ground range of sample i and h_n is the altitude of the sonar head at the moment of emission of swath n, see Section 8.3. Further, the linear transformation of each swath can be written as

$$T_n = \begin{bmatrix} 0 & \cos(\psi_n) & -\sin(\theta_n)\sin(\psi_n) \\ 0 & \sin(\psi_n) & \sin(\theta_n)\cos(\psi_n) \\ 0 & 0 & 0 \end{bmatrix} \tag{8.6}$$

Here, θ_n and ψ_n are the pitch and the yaw of the platform at the emission of swath n, respectively. Following the argument of Sheffer et al. [71], the first column of T_n is set to zero as motion along the roll axis is not reflected as a geometric distortion in the acoustic image. Mapping 3-dimensional samples onto a 2-dimensional map is done setting the third row of T_n to zero.

Combining the proposed correction solutions for both translational and rotational instabilities results in the complete geometric correction step of the image construction pipeline. The location of sample i in swath n of the processed acoustic image, adopting the formulations of Sheffer et al. [71], is defined as

$$I_{n,i} = PL_n + T_n * L_{n,i} \tag{8.7}$$

where

$I_{n,i}$ is the image coordinate and intensity of sample i of swath n in the acoustic image;
PL_n is the location of the platform in the acoustic image at the moment of emission of swath n;
T_n is the linear transformation of swath n; and
$L_{n,i}$ is the location of sample i of swath n in the platform frame of reference.

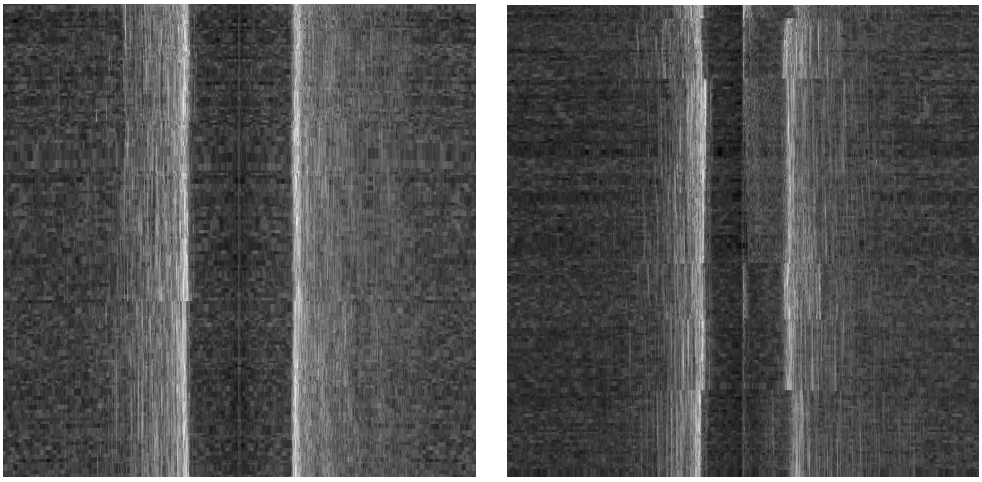
The resulting geocorrected image consists of a set of matching image coordinates and intensity values. However, due to the motion changes experienced by the vehicle, the image will inevitably suffer from gaps and overlapping data. The gaps can be addressed through the use of interpolation, while the overlapping data can be handled through blending techniques. It is of interest to make use of an interpolation/blending scheme capable of preserving the information and details in the processed image. In this project, the approach of storing corrected data in a k-d tree and performing a K-Nearest Neighbor (KNN) search will be adopted. The KNN search finds the k nearest neighbors to each spatial point of a defined grid, and corresponding intensity values are defined based on these neighboring points. This approach provides flexibility with regard to the handling of data, where parameters such as the number of neighbors, and range limits

between neighboring points are adjustable. More importantly, the intensity values may be defined as the mean or percentile of a given amount of neighbors, or removed all together if the computed variance at a specific point in the image is too high. By enabling the computation of variance, the KNN solution allows assessment of conflicting information, as well as the amount of information to be lost due to the KNN processing. Consequently, the approach is capable of handling both gaps and overlapping swaths while sufficiently preserving details in the image.

Chapter 9

Evaluation of the Image Construction Pipeline

The image construction pipeline as presented in Chapter 8 has been implemented and tested on data sets acquired through field experiments. Each data set consist of a recording containing altitude measurements from the DVL, state estimates provided by the error-state Kalman filter (ESKF), and raw sonar data in the form of swaths as they arrived. This allows for playing back a data set to simulate online processing of sonar data from the comfort of the office[1]. Hence, all testing have been conducted locally rather than onboard on the Raspberry PI (RPI).



(a) Straight section. **(b)** Turning section.

Figure 9.1: Basic acoustic images created from data set acquired in Ilsvika, Trondheim. Each section consists 1000 consecutive pings.

[1]Far, far away from seagulls suffering from rectal tenesmus.

This chapter will present results from and evaluate each step of the pipeline, showcasing how raw measurements and estimates are combined to construct processed acoustic images. For this purpose, a data set acquired at Ilsvika in Trondheim has been divided into two separate sections; a section where the ROV follows a straight line and a section where it turns. Serving as a nice visual cue, the acoustic images of consecutively gathered swaths shown in Figure 9.1 will be used in every step of the pipeline. Note that all processing is performed on data from both left and right transducers, although for the purpose of demonstration the following sections will mainly use right transducer data.

9.1 Pipeline Evaluation: Intensity Correction

The cubic spline regression based enhancement technique presented in Section 8.1 constitutes the first step of the pipeline. The technique has been tested using different values for the smoothing parameter p, and evaluation is based on visual inspection of swaths, histograms, and acoustic images, as well as the CV for across-track signals. The Coefficient of Variation (CV) is defined as the ratio of the standard deviation to the mean,

$$c_{CV} = \frac{\sigma}{\mu} \tag{9.1}$$

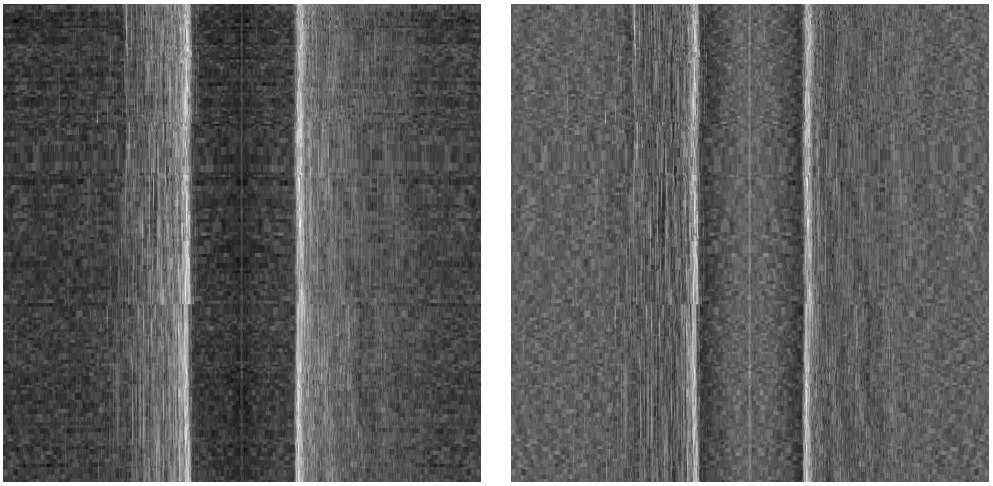
and it will be used to measure across-track intensity changes. As stated by Al-Rawi et al. [65], low CV values indicate less intensity variation and possibly better image quality.

In the search of an appropriate value for the smoothing parameter p, the average CV after normalization, $\bar{c}_{CV,Normalized}$, has been calculated using a data set containing 10 000 successive swaths. The results are summarized in Table 9.1.

Table 9.1: Table of average Coefficient of Variation (CV) after normalization of data set containing 10 000 successive swaths, while varying the smoothing parameter p.

p	$\bar{c}_{CV,Normalized}$	p	$\bar{c}_{CV,Normalized}$
10^{-0}	$1.23251 \cdot 10^{-18}$	10^{-5}	0.02507
10^{-1}	0.01364	10^{-6}	0.02779
10^{-2}	0.01602	10^{-7}	0.03083
10^{-3}	0.01881	10^{-8}	0.03643
10^{-4}	0.02203	10^{-9}	0.03838

In terms of average CV, normalization using cubic spline regression generates lower values regardless of the smoothing parameter, as the average CV prior to normalization was calculated to be 0.04295. In general, setting the smoothing parameter high provides a small average CV, where $p = 1$ results in practically no intensity changes. Figure 9.3 shows swath line from the right transducer of ping 183 for the data set, the respective cubic spline fitted curve with $p = 1$, and the resulting normalized swath. Visual inspection reveals how the technique has removed all information from the swath in an attempt to normalize it, resulting in a constant across-track signal. Here, the fitted curve coincides with the natural cubic spline interpolant. Hence, the CV metric alone can't provide sufficient information when trying to find an appropriate value for the smoothing parameter p, but must be combined with visual inspection to avoid complete loss of all intensity information.



(a) Raw image. **(b)** Normalized image.

Figure 9.2: Normalization of the acoustic image from the straight section using smoothing parameter $p = 1 \cdot 10^{-6}$.

The suggestion from Al-Rawi et al. [65] of using $p = 1 \cdot 10^{-8}$ has been attempted. However, visual inspection indicated that the fitted curve became somewhat linear, not sufficiently representing the underlying distribution of the swath. Reflecting this observation, the smoothing parameter was adjusted to a value of $p = 1 \cdot 10^{-6}$. Figure 9.4 shows the same swath line as previously mentioned, the cubic spline fitted curve, and the resulting normalized swath using the adjusted smoothing parameter. Here, the impact of across-track signal attenuation is visibly reduced while overall preserving the intensity information inherent in the swath line. To substantiate this observation, the CV of the signal has been reduced from 0.03847

to 0.02694. Figure 9.2 shows the resulting image after performing intensity correction of the acoustic image from the straight section. Comparing the two, the reader may observe how the artificial gradient in the images is severely reduced. However, further tuning of the smoothing parameter may result in increased normalization performance.

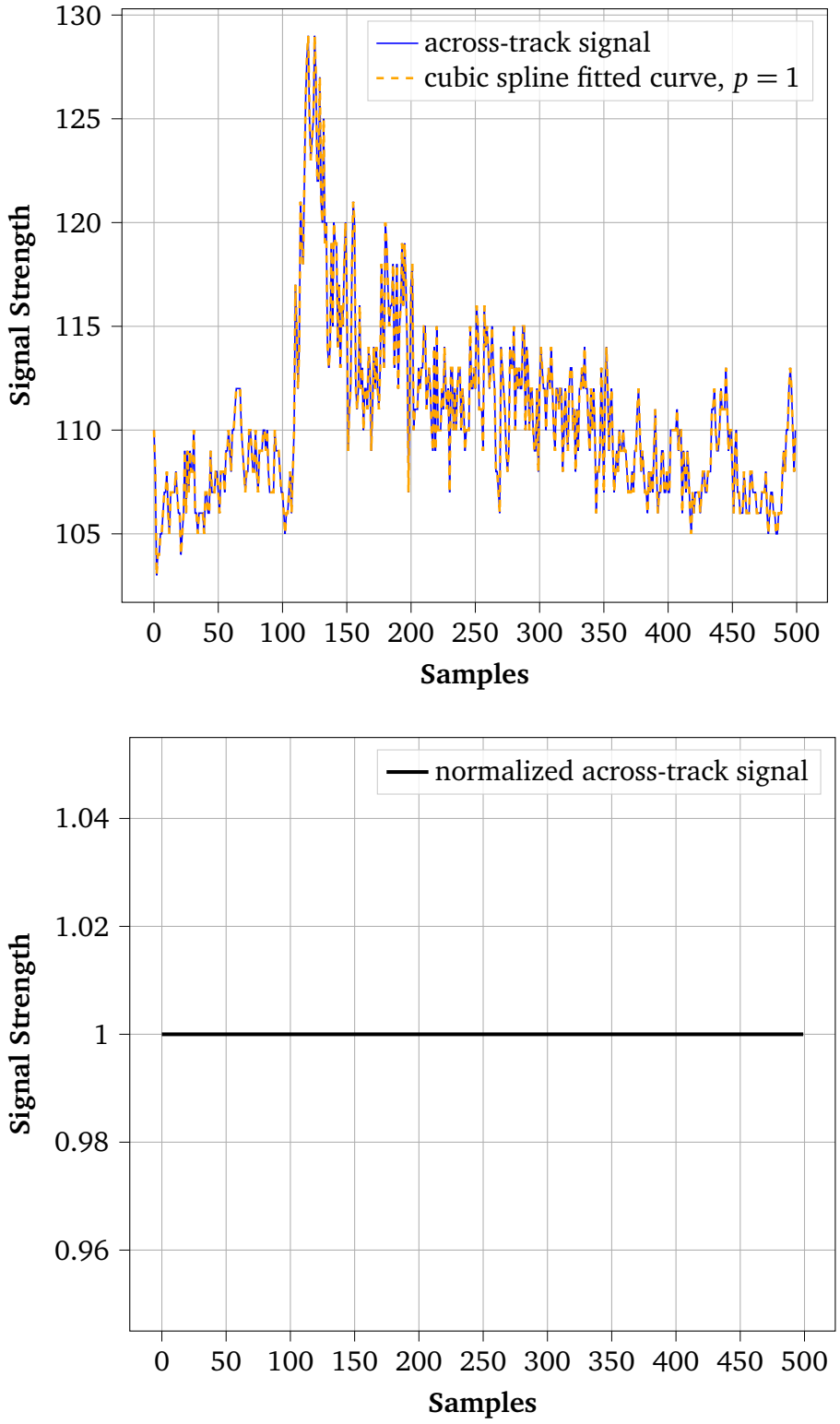


Figure 9.3: Across-track signal from right transducer of ping 183 from the data set, respective cubic spline fitted curve with smoothing parameter $p = 1$, and resulting normalized across-track signal.

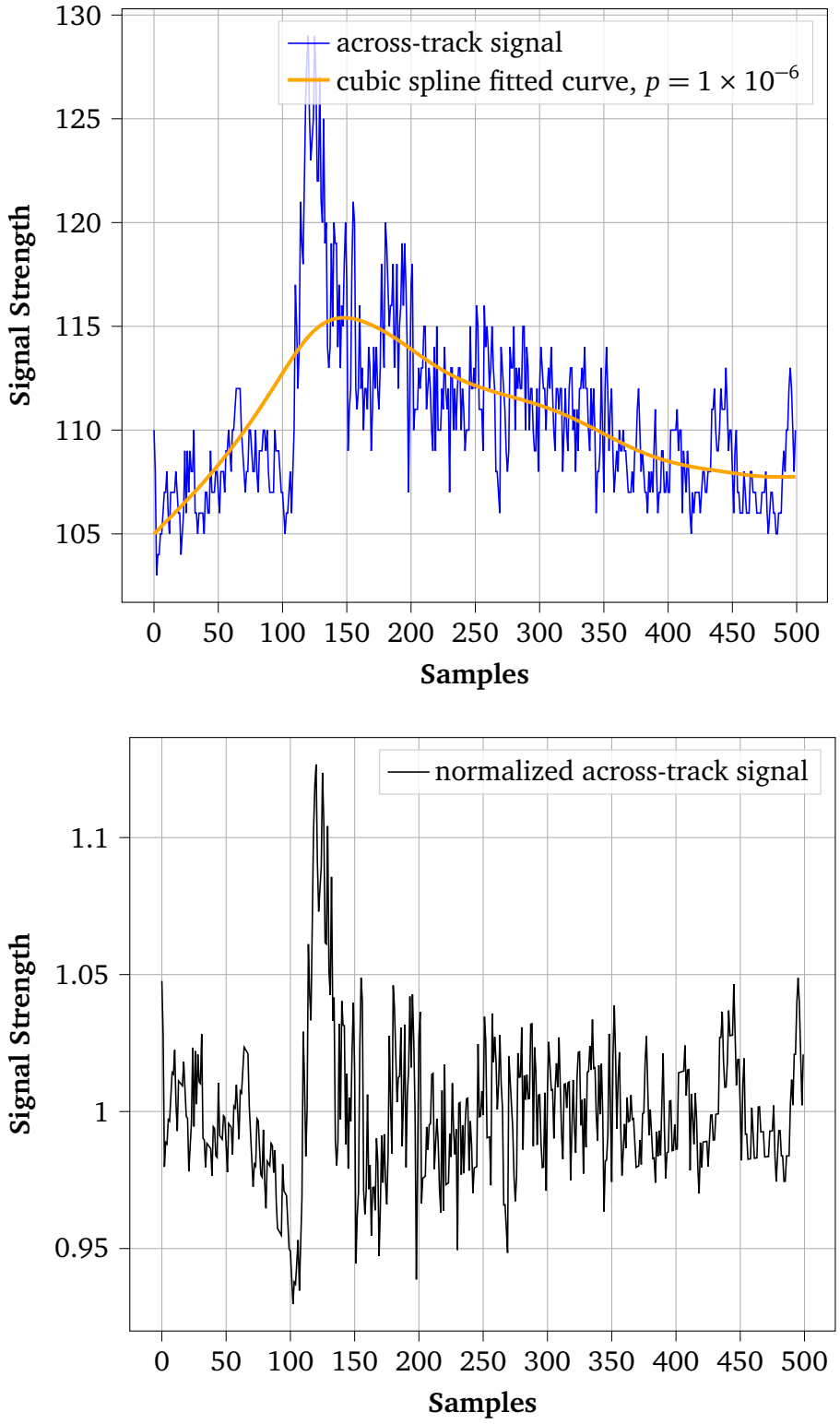
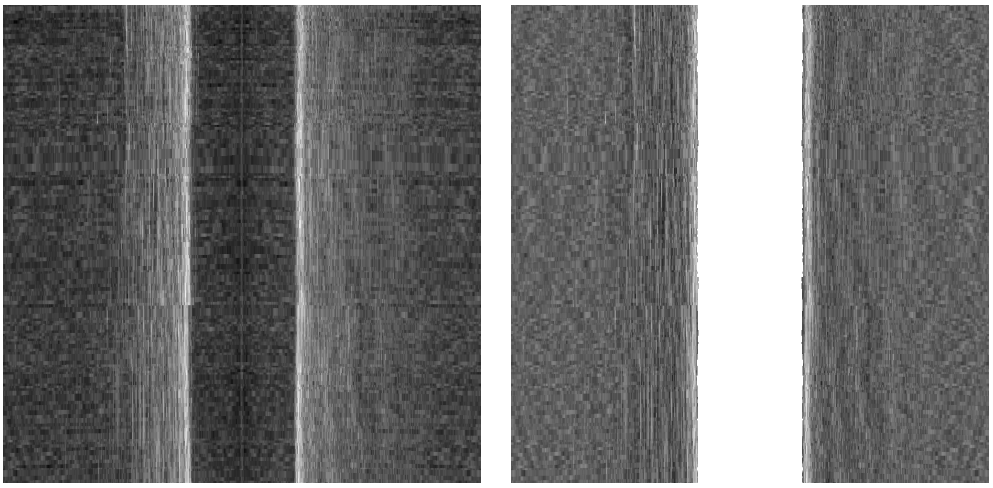


Figure 9.4: Across-track signal from right transducer of ping 183 from the data set, respective cubic spline fitted curve with smoothing parameter $p = 1 \cdot 10^{-6}$, and resulting normalized across-track signal.

9.2 Pipeline Evaluation: Blind Zone Removal

The blind zone removing technique presented in Section 8.2 constitutes the second step of the pipeline. Utilizing altitude measurements from the DVL, the angular sensor placement $\theta = 45°$, and the sensor opening $\alpha = 60°$, the First Bottom Return (FBR) is calculated for every swath line. Figure 9.6 shows the normalized across-track signal of ping 183 from the right transducer alongside with bin corresponding to the calculated FBR, and the resulting signal after removal. The first 110 bins contain intensity values mainly attributed to sensor noise and suspended particles in the water column, thus providing no useful information. Hence, the information is removed accordingly. Figure 9.5 shows the resulting images after performing blind zone removal of the acoustic images from Figure 9.2. Comparing the two, the reader may observe how the black strip along the center line is removed, leaving the intensity information of interest.



(a) Raw image.

(b) Normalized image without blind zone.

Figure 9.5: Normalized image with blind zone removed using the acoustic image from the straight section.

Visual inspection of the resulting acoustic images indicates that this technique removes the blind zone with sufficient accuracy. However, the technique is crucially dependent on accurate altitude measurements, where noisy measurements will result in poor performance. The approach will remain completely useless during continuous absence of altitude measurements, further emphasizing the system's dependency on the DVL sensor.

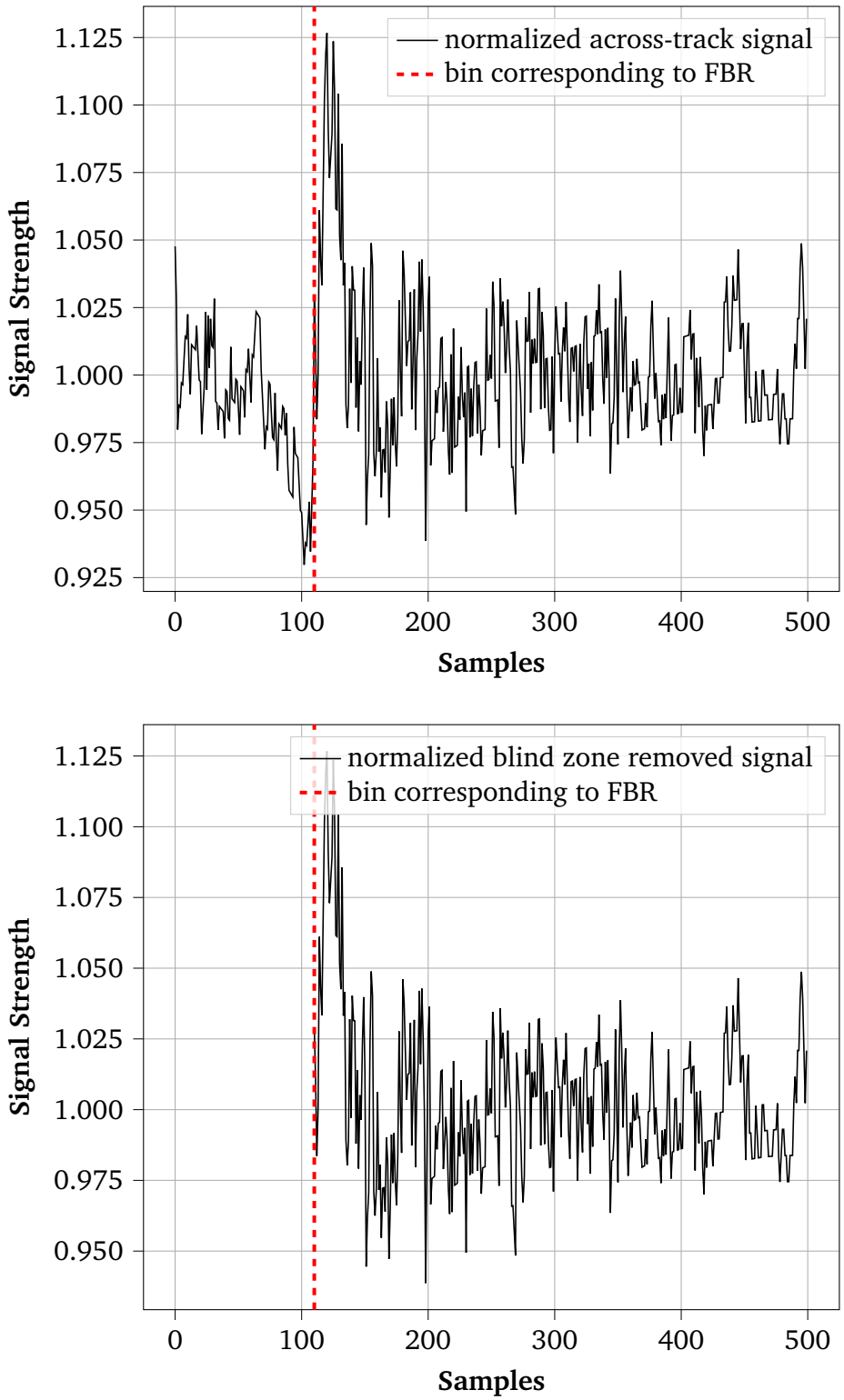
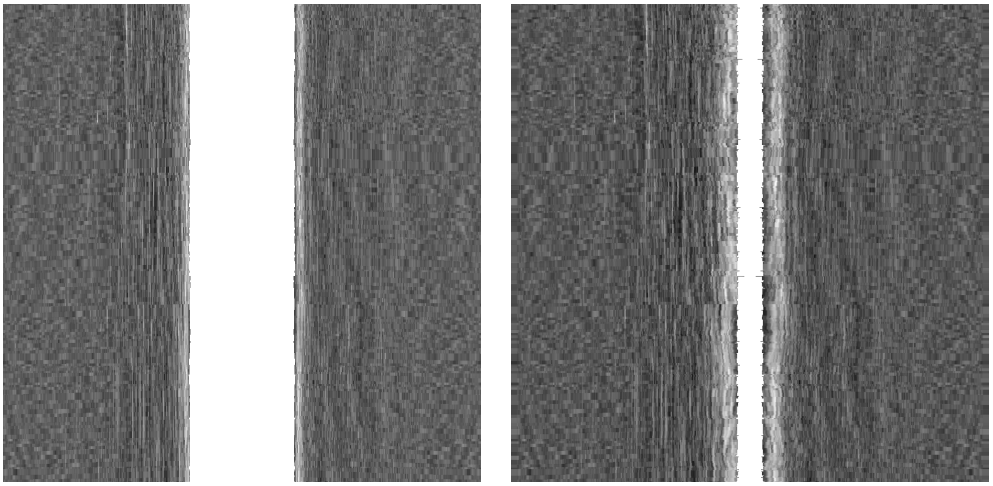


Figure 9.6: Normalized across-track signal from right transducer of ping 183 with bin corresponding to First Bottom Return (FBR), and resulting blind zone removed signal.

9.3 Pipeline Evaluation: Slant Range Correction

The slant range correction technique presented in Section 8.3 constitutes the third step of the pipeline. Utilizing altitude measurements from the DVL and the slant range resolution corresponding to each swath line, the swath line is corrected to represent the horizontal distance on the sea floor. Figure 9.7 shows the resulting images after performing slant range correction on the same straight section of the data set. The most visible effect of the slant range correction step is how the size of the blind zone is reduced. For the straight section, the ROV kept an altitude of 3.2 meters. Prior to slant range correction, the blind zone of the acoustic image had an average magnitude of 6.16 meters. However, at the given altitude and under the flat floor assumption, the horizontal distance expected to be covered by the blind zone can be calculated to be 1.71 meters using Equation (8.3). Thus, the blind zone should be reduced by a factor of approximately 4 such that the across-track signal accurately describes the horizontal distance it covers. The reader may through visual inspection of the resulting image verify how the blind zone is more or less correctly reduced. As it happens, the average magnitude of the slant range corrected blind zone is 1.51 meters.



(a) Normalized image without blind zone.

(b) Slant range corrected, normalized image without blind zone.

Figure 9.7: Slant range corrected, normalized acoustic image with blind zone removed using the acoustic image from the straight section.

Much like the second step in the pipeline, the slant range correction step is also crucially dependent on accurate altitude information. Furthermore,

the performance of the correction technique also depends on the validity of the flat floor assumption. A tilting sea floor along the across-track direction will result in a shift in the actual horizontal distance covered. As such, the performance of the slant range correction will vary depending on conditions.

9.4 Pipeline Evaluation: Geometric Correction

The geometric correction technique presented in Section 8.4 constitutes the last step of the pipeline. Utilizing state estimates from the error-state Kalman filter (ESKF) corresponding to each swath line, the processed data is mapped to the corresponding 2 dimensional location of the sea floor. Figure 9.8 shows the resulting image after performing intensity normalization, blind zone removal, slant range correction, and geometric correction on the full turn in the data set. Although the section visualized in Figure 9.1 only constitutes a fraction of the complete turn, comparing the two goes to show the necessity of processing the sonar data.

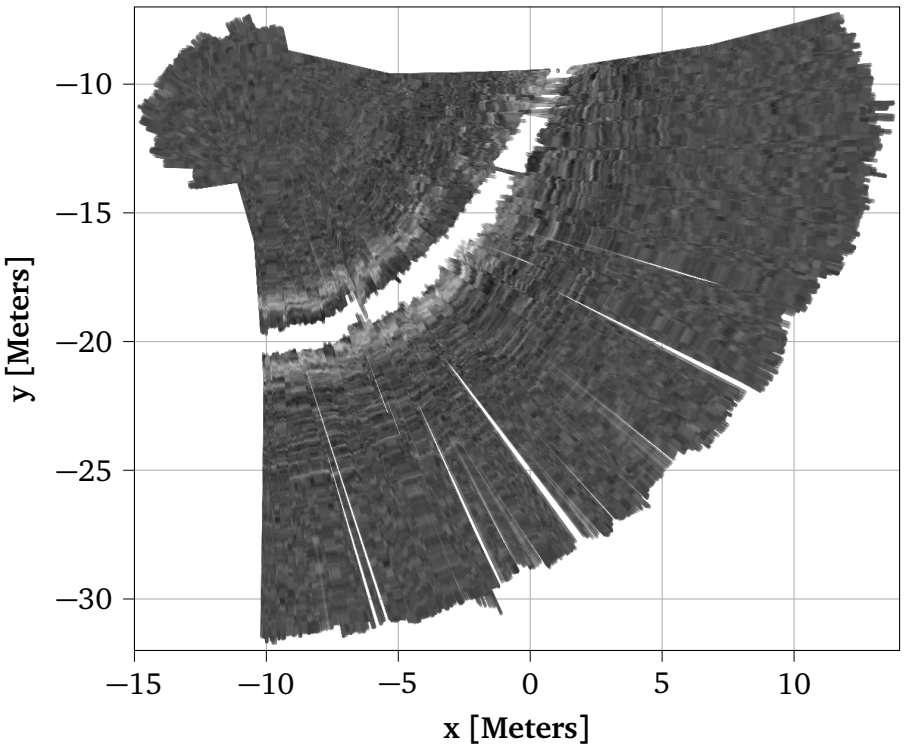


Figure 9.8: Geometrically corrected, slant rage corrected, normalized image without blind zone using the full turn in the Ilsvika data set.

The performance of the geometric correction technique is dependent on the accuracy of the corresponding state estimates, where poor estimates will result in poor mapping of the sonar data. The effect of inaccurate state estimates on the mapping is visible in Figure 9.8, where some data is wrongfully mapped into the blind zone. Similar to the slant range correction, the technique's performance depends on the validity of the flat floor assumption.

Finally, the image is usable for navigational purposes as detection of landmarks now can provide information to help localize the ROV. However, orientation changes of the platform will cause both gaps and overlapping data in the geometrically corrected image. This can be seen in Figure 9.8; several gaps are visible at the outer edge of the turn, while swaths severely overlap each other in the pivoting point of the turn.

To address both issues, the K-Nearest Neighbor (KNN) solution presented in Section 8.4 was implemented and tested. Figure 9.9 shows the KNN filtered image using $k = 4$, a range limit of 0.3 meters, a variance ceiling of $5 \cdot 10^{-3}$, and intensity values defined by the 10% percentile. Visual inspection reveals how the gaps are filled and the overlapping swaths averaged out. Moreover, the filtering process appears to smooth over the blind zone in the areas where data is inaccurately mapped. As such, the KNN filtering is dependent on the performance of the geometric correction solution.

Figure 9.10 shows the computed variance map of the turn using the KNN solution with $k = 4$ and a valid distance of 30 cm. With this specific configuration, the computed variance is rather low ranging from $5 \cdot 10^{-4}$ to $5 \cdot 10^{-3}$ compared to the intensity values ranging from 0.88 to 1.25. Further, the reader may observe how variations are prominent in areas where there are significant returns. This may be a result of pose uncertainty causing inaccurate mapping of intensity values, or conflicting information rooted in the change of view when moving. Regardless, it is a cause for concern; the areas containing significant returns are also the areas where detection is likely to happen when using the images for navigational purposes. If so, a landmark may be detected using conflicting information due to the overlapping swaths. However, as mentioned the KNN solution allows for removing conflicting information, hence possibly avoiding faulty landmarks. Note, the variance may also increase naturally with return strength in which it should not degrade the detector performance.

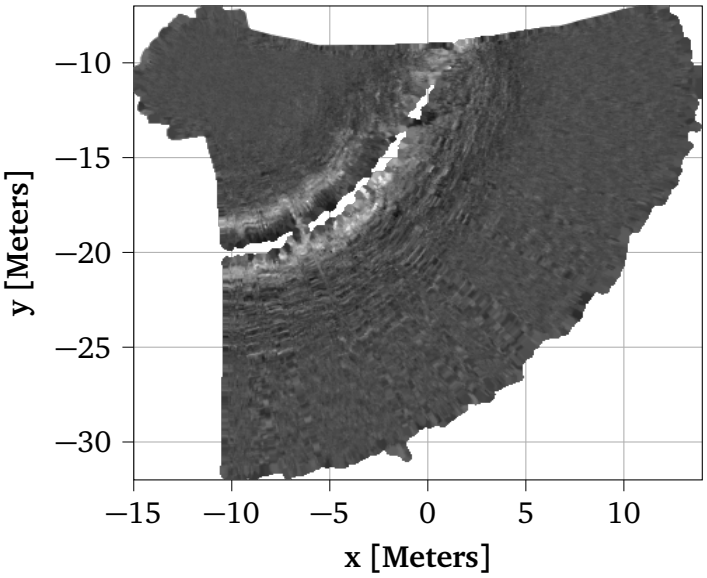


Figure 9.9: Geometrically corrected, KNN filtered acoustic image using $k = 4$, a range limit of 0.3 meters, a variance ceiling of $5 \cdot 10^{-3}$, and a 10% percentile.

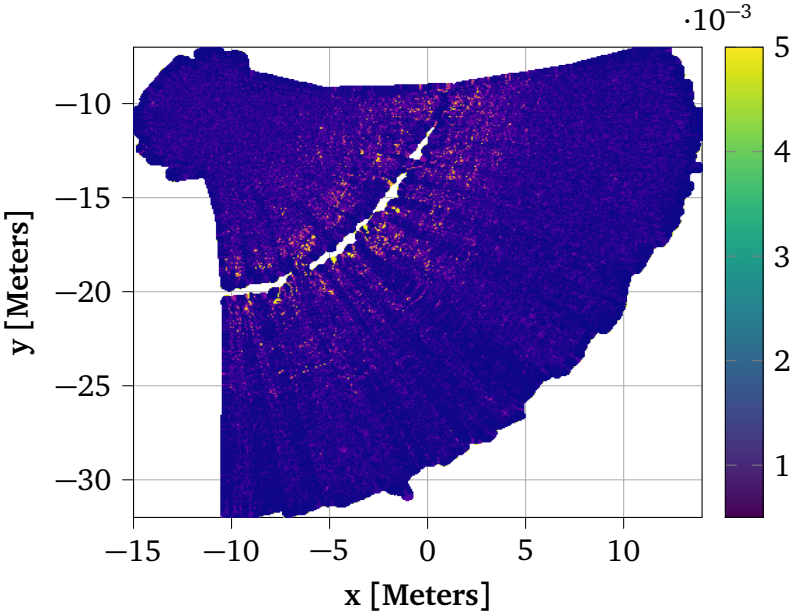


Figure 9.10: Variance map of the geometrically corrected, KNN filtered acoustic image using $k = 4$, a range limit of 0.3 meters, a variance ceiling of $5 \cdot 10^{-3}$, and a 10% percentile.

Part IV

All Good Things Come To An End.

Chapter 10

Improvements for the Future

The error-state Kalman filter (ESKF) and image construction pipeline implementations come with limitations and caveats. This chapter addresses some of these, suggesting solutions and future work to improve the capabilities and performance of the overall configuration.

10.1 Computational Efficiency

Performing all tasks related to control, navigation, and processing of sonar data is computationally demanding. Doing so with the current implementations requires more processing power than the Raspberry PI (RPI) is able to deliver. The benchmarking results presented in Section 5.4 reveals how the RPI is being pushed to the limit of its processing capabilities prior to processing sonar data, not to mention performing detection for navigational purposes. For now, the majority of software implemented in this project is written in Python, and rewriting in more performant languages will obviously improve the situation. As previously mentioned, another solution can be found in switching or complementing the RPI with more capable computers. Future hardware upgrades will include switching to Khadas computers, which provides more computational power. Additional upgrades may also include the adding of NVIDIA Jetson to make use of the parallel processing power of GPUs. Doing so, the image construction pipeline can be significantly sped up, consequently relieving the CPU of significant work.

10.2 DVL Dependency

One major shortcoming of the system is the severe dependency on the DVL, as continuous loss of fix in practice will render both the ESKF and the image construction pipeline incapacitated. As shown in Section 5.2, the ESKF will drift immensely when dead reckoning. Regardless of the filter's effort of tracking the estimate uncertainty, the ROV will quickly veer off course and continue doing so until fix is acquired again. Because of this, the geometric correction step will perform poorly resulting in inaccurate processed sonar images. Moreover, a continuous lack of altitude measurements affect both blind zone removal and slant range correction. In its current state, the system can only handle a momentary loss of fix.

Measures can be taken to address and reduce the system's dependency on the DVL sensor. Enhancing dead reckoning capabilities of the ESKF can be done through an upgrade of the IMU. In the future, the BNO055 will be swapped with the STIM300*[1] providing more reliable measurements, and consequently better filter performance. Doing so will also alleviate the fusing of orientation estimates from the filter. Further, the blind zone removal step can be complemented with alternative signal processing techniques to determine the FBR, adding redundancy to the system. Thus, the altitude can be computed using Equation (8.3), and the slant range correction step can be performed regardless of available altitude measurements from the DVL.

10.3 Improving System Performance

To further improve the performance of the ESKF more tuning is in order. As specified in Section 5.1, the noise statistics used in this project were measured while keeping the ROV motionless due to the lack of ground truth while moving. However, this approach does not necessarily provide statistics reflecting the uncertainty present when the ROV is moving. Because of this, further tuning should be done in facilities capable of providing ground truth measurements. Moreover, in the current implementation the respective statistics are used directly in a naive fashion, where the uncertainties related to sensor offsets[2] and the transformation of DVL measurements are not considered. Nevertheless, there will be uncertainty related to angular velocity measurements and orientation estimates, and these should be accounted for in the filter. Not doing so contributes to the overconfidence

[1] Purchase not confirmed, upgrading to ADIS is also possible.

[2] This does not include the positional offset of the pressure sensor, since it is accounted for through the Jacobian, see Equation (4.4).

observed in the filter consistency analysis. Lastly, the ESKF performance may also be improved through the use of more advanced integration techniques with higher accuracy and better numerical stability.

Improving state estimate accuracy will consequently improve the performance of the image construction pipeline. Further improvements of the pipeline may include tuning of the intensity correction smoothing parameter and the KNN parameters. In this thesis the evaluation of the pipeline has been heavily based on visual inspection. However, further tuning should be done in combination with performing detection in the resulting acoustic images, as this is the reason why processing is necessary in the first place.

Chapter 11

Conclusion

Through this thesis a foundation for an easily deployable Remotely Operated Vehicle (ROV) configuration has been developed. The foundation consist of an error-state Kalman filter (ESKF) performing state estimation, and an image construction pipeline for Side-Scan Sonar (SSS) constructing acoustic images which more accurately represent the sea floor. With the current hardware configuration, the ESKF estimates appear to drift 1.2 meters per hour in the xy-plane and 120 degrees per hour in heading, accentuating the need for detection methods to limit uncertainty in the estimates. Based on visual inspection, the pipeline appears to provide images well suited for detection.

Serving as an Achilles heel of the system, testing has revealed how both the ESKF and the pipeline is severely dependent on measurements from the Doppler Velocity Log (DVL) sensor to function properly. Further, simultaneously performing tasks related to navigation, control, sensing, and processing is not computationally feasible using the current hardware configuration and implementations.

Future work should aim to address these issues through a combination of hardware upgrades, as well as adding redundancy to the pipeline. Accuracy of state estimates can be improved through better uncertainty handling, and more accurate integration techniques. Following this, further research should focus on development of a detector to be used on processed acoustic images. Accordingly, the complete system should be retuned and re-evaluated to accommodate the detector's performance.

Bibliography

[1] S. B. Williams, O. Pizarro, I. Mahon, and M. Johnson-Roberson, "Simultaneous localisation and mapping and dense stereoscopic seafloor reconstruction using an auv," in *Experimental Robotics*, O. Khatib, V. Kumar, and G. J. Pappas, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 407–416.

[2] F. S. Hover, R. M. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, and J. J. Leonard, "Advanced perception, navigation and planning for autonomous in-water ship hull inspection," *The International Journal of Robotics Research*, vol. 31, no. 12, pp. 1445–1464, 2012, ISSN: 0278-3649. DOI: 10.1177/0278364912461059.

[3] M. E. Clarke, N. Tolimieri, and H. Singh, "Using the seabed auv to assess populations of groundfish in untrawlable areas," in pp. 357–372. DOI: 10.1007/978-1-4020-9210-7_20.

[4] A. Murphy, M. Landamore, and R. Birmingham, "The role of autonomous underwater vehicles for marine search and rescue operations," *Underwater Technology*, vol. 27, no. 4, pp. 195–205, 2008, ISSN: 1756-0543. DOI: 10.3723/ut.27.195.

[5] J. J. Leonard and A. Bahr, "Autonomous underwater vehicle navigation," in 2016, pp. 341–358. DOI: 10.1007/978-3-319-16649-0_14.

[6] P. Norgren, "Autonomous underwater vehicles in arctic marine operations: Arctic marine research and ice monitoring," 2018.

[7] K. G. Kebkal and A. I. Mashoshin, "Auv acoustic positioning methods," *Gyroscopy and Navigation*, vol. 8, no. 1, pp. 80–89, 2017, ISSN: 2075-1087. DOI: 10.1134/s2075108717010059.

[8] J. Bao, D. Li, X. Qiao, and T. Rauschenbach, "Integrated navigation for autonomous underwater vehicles in aquaculture: A review," *Information Processing in Agriculture*, vol. 7, no. 1, pp. 139–151, 2020, ISSN: 2214-3173. DOI: 10.1016/j.inpa.2019.04.003.

[9] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016, ISSN: 1552-3098. DOI: 10.1109/tro.2016.2624754.

[10] Y. Cong, C. Gu, T. Zhang, and Y. Gao, "Underwater robot sensing technology: A survey," *Fundamental Research*, vol. 1, no. 3, pp. 337–345, 2021.

[11] M. F. Fallon, M. Kaess, H. Johannsson, and J. J. Leonard, "Efficient auv navigation fusing acoustic ranging and side-scan sonar," in 2011. DOI: 10.1109/icra.2011.5980302.

[12] I. Tena Ruiz, S. De Raucourt, Y. Petillot, and D. Lane, "Concurrent mapping and localization using sidescan sonar," *IEEE Journal of Oceanic Engineering*, vol. 29, no. 2, pp. 442–456, 2004, ISSN: 0364-9059. DOI: 10.1109/joe.2004.829790.

[13] J. Sola, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.

[14] C. Sherrington, *The integrative action of the nervous system*. CUP Archive, 1952.

[15] T. D. Barfoot, *State Estimation for Robotics*, 1st. USA: Cambridge University Press, 2017, ISBN: 1107159393.

[16] T. V. Haavardsholm, "A handbook in visual slam," in pp. 7–15.

[17] T. I. Fossen, "Kinematics," in *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, Ltd, 2011, ch. 2, pp. 15–44, ISBN: 9781119994138. DOI: https://doi.org/10.1002/9781119994138.ch2. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119994138.ch2. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119994138.ch2.

[18] L. Euler, "Du mouvement d'un corps solide quelconque lorsqu'il tourne autour d'un axe mobile," *Mémoires de l'académie des sciences de Berlin*, pp. 176–227, 1767.

[19] L. Euler, "De motu corporum circa punctum fixum mobilum," *Leonhardi Euleri Opera Postuma, Vol. 2*, pp. 43–62, 1862.

[20] G. H. Bryan, *Stability in aviation: an introduction to dynamical stability as applied to the motions of aeroplanes*. Macmillan and Company, limited, 1911.

[21] J. Sempere, *File:taitbrianzyx.svg - wikimedia commons*, https://commons.wikimedia.org/wiki/File:Taitbrianzyx.svg, (Accessed on 05/05/2022), Jul. 2009.

[22] J. Stuelpnagel, "On the parametrization of the three-dimensional rotation group," *SIAM review*, vol. 6, no. 4, pp. 422–430, 1964.

[23] E. Brekke, "Fundamentals of sensor fusion," in pp. 53–76, 174–175.

[24] J. Sola, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," *arXiv preprint arXiv:1812.01537*, 2018.

[25] C. Runge, "Über die numerische auflösung von differentialgleichungen," *Mathematische Annalen*, vol. 46, no. 2, pp. 167–178, 1895.

[26] W. Kutta, "Beitrag zur naherungsweisen integration totaler differentialgleichungen," *Z. Math. Phys.*, vol. 46, pp. 435–453, 1901.

[27] N. Fofonoff and R. Millard Jr, *1983, algorithms for computation of fundamental properties of seawater: Unesco technical papers in marine science, number 44*.

[28] R. D. Instruments, "Acoustic doppler current profiler principles of operation: A practical primer," *RD Instruments*, pp. 3–14 32, 1996.

[29] A. Jazwinski, *Stochastic Processes and Filtering Theory*, ser. Mathematics in science and engineering. Academic Press, 1970, ISBN: 9780123815507. [Online]. Available: https://books.google.no/books?id=zEZHlAEACAAJ.

[30] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1, pp. 99–141, 2001, ISSN: 0004-3702. DOI: https://doi.org/10.1016/S0004-3702(01)00069-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370201000698.

[31] H. K. Khalil, *Nonlinear systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002, The book can be consulted by contacting: PH-AID: Wallet, Lionel. [Online]. Available: https://cds.cern.ch/record/1173048.

[32] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[33] L. A. McGee, S. F. Schmidt, L. A. Mcgee, and S. F. Sc, "Discovery of the kalman filter as a practical tool for aerospace and," in *Industry," National Aeronautics and Space Administration, Ames Research*, Citeseer, 1985.

[34] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, International Society for Optics and Photonics, vol. 3068, 1997, pp. 182–193.

[35] G. Evensen and P. J. Van Leeuwen, "An ensemble kalman smoother for nonlinear dynamics," *Monthly Weather Review*, vol. 128, no. 6, pp. 1852–1867, 2000, ISSN: 0027-0644. DOI: 10.1175/1520-0493(2000)128<1852:aeksfn>2.0.co;2.

[36] M. Katzfuss, J. R. Stroud, and C. K. Wikle, "Understanding the ensemble kalman filter," *The American Statistician*, vol. 70, no. 4, pp. 350–357, 2016, ISSN: 0003-1305. DOI: 10.1080/00031305.2016.1141709.

[37] K. Gade, "Integrering av treghetsnavigasjon i en autonom undervannsfarkost," *Institutt for teknisk kybernetikk, Norwegian University of Science and Technology*, 1997.

[38] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey, "Circumventing dynamic modeling: Evaluation of the error-state kalman filter applied to mobile robot localization," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, IEEE, vol. 2, 1999, pp. 1656–1663.

[39] N. Trawny and S. I. Roumeliotis, "Indirect kalman filter for 3d attitude estimation," *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, vol. 2, p. 2005, 2005.

[40] B. Gerald J, *Factorization Methods for Discrete Sequential Estimation*. Ser. Mathematics in Science and Engineering Series v. 128. Academic Press, 1977, pp. 96–100, ISBN: 9780120973507.

[41] *File:raspberry pi 4 model b - side.jpg - wikimedia commons*, https://commons.wikimedia.org/wiki/File:Raspberry_Pi_4_Model_B_-_Side.jpg, (Accessed on 14/06/2022).

[42] *Bluerov2 - the world's most affordable high-performance rov*, https://bluerobotics.com/store/rov/bluerov2/, (Accessed on 15/06/2022).

[43] K. Townsend, *Adafruit bno055 absolute orientation sensor*, https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor, (Accessed on 10/07/2022), Apr. 2015.

[44] W. Linked, *Water linked a50 dvl - documentation*, https://waterlinked.github.io/dvl/dvl-a50/, (Accessed on 10/07/2022).

[45] BlueRobotics, *Bar30 high-resolution 300m depth/pressure sensor*, https://bluerobotics.com/store/sensors-sonars-cameras/sensors/bar30-sensor-r1/, (Accessed on 10/07/2022).

[46] *Deepvision - oem sonar modules*, https://deepvision.se/products/oem-sonar-modules/, (Accessed on 15/06/2022).

[47] B. Sensortec, *Bno055 - intelligent 9-axis absolute orientation sensor*, https://https://cdn-shop.adafruit.com/datasheets/BST_BNO055_DS000_12.pdf, (Accessed on 29/04/2022), Nov. 2014.

[48] B. Welford, "Note on a method for calculating corrected sums of squares and products," *Technometrics*, vol. 4, no. 3, pp. 419–420, 1962.

[49] E. A. Basso, H. M. Schmidt-Didlaukies, K. Y. Pettersen, and A. J. Sorensen, "Global asymptotic tracking for marine vehicles using adaptive hybrid feedback," *IEEE Transactions on Automatic Control*, pp. 1–1, 2022, ISSN: 0018-9286. DOI: 10.1109/tac.2022.3161372.

[50] B. Hogstad, "Software development for data acquisition with underwater remotely operated vehicles," 2021.

[51] A. Richards, "Alien vision(exploring the electromagnetic spectrum with imaging technology)," SPIE- The International society for optical engineering, 2001.

[52] W. S. Gan, *Acoustical Imaging: Techniques and Applications for Engineers*. John Wiley & Sons, 2012.

[53] A. Haude and C. Spener, *Histoire de l'Academie Royale des Sciences et des Belles Lettres de Berlin... Avec les Mémoires pour la même Année, tirez des Registres de cette Academie*. chez Ambroise Haude, 1746.

[54] J. Hovem, *Marine Acoustics: The Physics of Sound in Underwater Environments*. Peninsula Publishing, 2012, ISBN: 9780932146656. [Online]. Available: https://books.google.no/books?id=6lVPMwEACAAJ.

[55] B. H. Foster, *File:2d wave.gif - wikimedia commons*, https://commons.wikimedia.org/wiki/File:2D_Wave.gif, (Accessed on 01/06/2022), Mar. 2015.

[56] P. Blondel, "The handbook of sidescan sonar," in Jan. 2009, pp. 249–276, ISBN: 978-3-540-42641-7. DOI: 10.1007/978-3-540-49886-5_11.

[57] H. Azhari, J. A. Kennedy, N. Weiss, and L. Volokh, *From Signals to Image*. Springer, 2020, p. 328.

[58] R. P. Hodges, *Underwater acoustics: Analysis, design and performance of sonar*. John Wiley & Sons, 2011.

[59] L. E. Kinsler, A. R. Frey, A. B. Coppens, and J. V. Sanders, *Fundamentals of acoustics*. John wiley & sons, 2000.

[60] A. D'amico and R. Pittenger, "A brief history of active sonar," SPACE and NAVAL WARFARE SYSTEMS CENTER SAN DIEGO CA, Tech. Rep., 2009.

[61] A. Burguera and G. Oliver, "High-resolution underwater mapping using side-scan sonar," *PloS one*, vol. 11, no. 1, e0146396, 2016.

[62] G. Canepa, A. Munafo, M. Micheli, L. Morlando, and S. Murphy, "Real-time continuous active sonar processing," in 2015. DOI: 10.1109/oceans-genova.2015.7271658.

[63] D. W. Ricker, *Echo signal processing*. Springer Science & Business Media, 2003.

[64] Y. Fei, W. Wen-Jun, and C. En, "Characteristics analysis of hfm signal over underwater acoustic channels," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 11, no. 3, pp. 1173–1180, 2013.

[65] M. Al-Rawi, A. Galdran, A. Isasi, F. Elmgren, G. Carbonara, E. Falotico, D. A. Real-Arce, J. Rodriguez, J. Bastos, M. Pinto, and et al., "Cubic spline regression based enhancement of side-scan sonar imagery," in 2017. DOI: 10.1109/oceanse.2017.8084567.

[66] M. S. Al-Rawi, A. Galdran, X. Yuan, M. Eckert, J.-F. Martinez, F. Elmgren, B. Curuklu, J. Rodriguez, J. Bastos, M. Pinto, and et al., "Intensity normalization of sidescan sonar imagery," in 2016. DOI: 10.1109/ipta.2016.7820967.

[67] A. Galdran, A. Isasi, M. Al-Rawi, J. Rodriguez, J. Bastos, F. Elmgren, and M. Pinto, "An efficient non-uniformity correction technique for side-scan sonar imagery," in *OCEANS 2017-Aberdeen*, IEEE, 2017, pp. 1–6.

[68] M. Al-Rawi, F. Elmgren, M. Frasheri, B. Curuklu, X. Yuan, J.-F. Martinez, J. Bastos, J. Rodriguez, and M. Pinto, "Algorithms for the detection of first bottom returns and objects in the water column in sidescan sonar images," in 2017. DOI: 10.1109/oceanse.2017.8084587.

[69] H. Yu, Z. Li, D. Li, and T. Shen, "Bottom detection method of side-scan sonar image for auv missions," *Complexity*, vol. 2020, pp. 1–9, 2020, ISSN: 1099-0526. DOI: 10.1155/2020/8890410.

[70] G. Zheng, H. Zhang, Y. Li, and J. Zhao, "A universal automatic bottom tracking method of side scan sonar data based on semantic segmentation," *Remote Sensing*, vol. 13, no. 10, p. 1945, 2021, ISSN: 2072-4292. DOI: 10.3390/rs13101945.

[71] T. Sheffer and H. Guterman, "Geometrical correction of side-scan sonar images," in *2018 IEEE International Conference on the Science of Electrical Engineering in Israel (ICSEE)*, 2018, pp. 1–5. DOI: 10.1109/ICSEE.2018.8646188.

[72] D. Cobra, A. Oppenheim, and J. Jaffe, "Geometric distortions in side-scan sonar images: A procedure for their estimation and correction," *IEEE Journal of Oceanic Engineering*, vol. 17, no. 3, pp. 252–268, 1992, ISSN: 0364-9059. DOI: 10.1109/48.153442.

[73] X. Ye, H. Yang, Y. Jia, and J. Liu, "Geometric correction method of side-scan sonar image," in *OCEANS 2019 - Marseille*, 2019, pp. 1–7. DOI: 10.1109/OCEANSE.2019.8867463.

[74] P. V. Teixeira, D. Fourie, M. Kaess, and J. J. Leonard, "Dense, sonar-based reconstruction of underwater scenes," in 2019. DOI: 10.1109/iros40897.2019.8968071.

Appendix A

Alternative Preface Remarks

```
$ export PR="Now I know what pain is."
$ export PR="I should probably document this."
$ export PR="I should have documented this."
$ export PR="Unfortunately my computer does exactly what I tell it to do."
$ export PR="Variance ain’t standard deviation."
```

Appendix B

Distinction of Software

The itemized software below is embedded with a hyperlink to the respective repositories. Note, part of the API related to the SSS sensor driver is not publicly available due to proprietary rights.

Software developed by the author:

- error-state Kalman filter (ESKF)
- Image construction pipeline
- Trajectory planner/publisher
- DVL sensor driver
- Pressure sensor driver
- GPS sensor driver
- Text user interface
- Start-up procedures

Software modified by the author:

- IMU sensor driver
- SSS sensor driver

Software unified by the author:

- Hysteretic controller
- PD controller
- Teleoperation solution