

Sonar Range SLAM for Autonomous Underwater Vehicles

A thesis submitted in part fulfilment of the degree of
Bachelor of Engineering (Honours)

by
Arthur Pearce
u7127525

Supervisor: Shawn Ge
Examiner: Prof. Robert Mahony



Australian
National
University

College of Engineering and Computer Science
The Australian National University

November 2024

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university. To the best of the author's knowledge, it contains no material previously published or written by another person, except where due reference is made in the text.

Arthur Pearce
November 1, 2024

Acknowledgements

I would like to first thank my supervisor Shawn Ge. He provided continuous help throughout the project, particularly in the equivariant filter design and related mathematics. I would also like to thank him for helping during the range measurement experiments.

I would like to thank Professor Robert Mahony for giving me the opportunity to work on this project. The high level advice he has provided has made a significant impact on the outcomes of my work. I would particularly like to thank him for pursuing the contact with Blue Robotics, and purchasing their acoustic transducer hardware without which the entire second half of this project would not be possible. I must also thank Andrew Tridgell for first providing the contact with Blue Robotics

I would like to thank Rustom Jehangir and Brian Hoover from Blue Robotics for taking interest in the project, modifying the acoustic transducers to fit our needs, and providing support for the hardware. Finally, I would like to thank Iain Guilliard and Andrew Tridgell for providing advice regarding the microcontroller software and hardware during the difficult stages of development.

Abstract

Autonomous Underwater Vehicles have a growing number of applications for research and commercial use, including ocean studies and maintaining underwater infrastructure. A method of range-only Simultaneous Localisation and Mapping for these underwater robots is desirable due to the absence of GPS and low visibility in marine environments. In this project we develop an Equivariant filter for range-only SLAM, and demonstrate its effectiveness on a synthetic dataset. The filter is intended for a mobile robot equipped with an Inertial Measurement Unit and a range sensor which measures the distances to a number of landmarks, the location of which is not known in advance. Existing commercial technology to enable such acoustic range measurements underwater is prohibitively expensive and locked down. We address this issue by presenting a prototype acoustic range measurement system capable of centimetre level precision.

Contents

Acknowledgements	i
Abstract	ii
Nomenclature	v
1 Introduction	1
2 Literature Review	3
2.1 Sensors Technology for AUV SLAM	3
2.2 Algorithms for AUV SLAM	4
2.3 Work on Rang-Only SLAM for AUVs	5
2.4 Equivariant Filters	6
3 Equivariant Range SLAM Filter	7
3.1 Preliminary Notation	7
3.1.1 Smooth Manifolds	7
3.1.2 Lie Groups	7
3.1.3 Special Orthogonal Group	8
3.1.4 Extended Pose Group	8
3.1.5 Scaled Orthogonal Transformation	8
3.2 Problem Definition	9
3.2.1 System State	9
3.2.2 System Input	9
3.2.3 System Dynamics	9
3.2.4 System Output	10
3.3 System Symmetry	11
3.4 Origin Choice and Local Coordinates	12
3.5 Lift	13
3.6 Equivariant Filter Derivation	14
3.6.1 State Matrix	14
3.6.2 Output Matrix	18
3.6.3 Input Matrix	19
3.7 Filter Simulation Experiment	22
3.7.1 Simulation and Filter set-up	22
3.7.2 Results	22
4 Acoustic Range Sensor	25
4.1 Hardware Overview	25
4.2 Acoustic Range Measurement Protocol	27
4.3 Peripheral and Software Design	29
4.4 Peak Detection	32
4.5 Range Measurement Experiments and Performance	34
4.5.1 Fish-tank Experiments	34

CONTENTS

4.5.2	Experiments in Lake Burley Griffin	36
4.5.3	Noise Characterization	38
4.6	Implementation Challenges	39
5	Conclusions and Future Work	41
References		42

Nomenclature

ADC	Analogue to Digital Converter
ARR	Auto Reload Register
AUV	Autonomous Underwater Vehicle
CPU	Central Processing Unit
DAC	Digital to Analogue Converter
DMA	Direct Memory Access
DVL	Doppler Velocity Log
EIF	Extended Information Filter
EKF	Extended Kalman Filter
EqF	Equivariant Filter
FOG	Fibre Optic Gyroscope
GPIO	General Purpose Input Output
GPS	Global Positioning System
ID	Identity
IMU	Inertial Measurement Unit
LBL	Long Baseline
LiDar	Light Detection and Ranging
MEMS	Micro Electronic Mechanical Systems
MSE	Mean Square Error
PGA	Programmable Gain Amplifier
RAM	Random Access Memory
RX	Receive
SEIF	Sparse Extended Information Filter
SLAM	Simultaneous Localisation And Mapping
Sonar	Sound Navigation and Ranging
TOF	Time Of Flight
TX	Transmit
USBL	Ultra Short Baseline

Chapter 1

Introduction

Simultaneous Localization and Mapping (SLAM) is a process used in robotics where a mobile robot builds a map of an unknown environment while simultaneously keeping track of its own location within that map. The SLAM problem has been a significant research area in robotics for the last few decades [1]. This project focusses on the application of SLAM to Autonomous Underwater Vehicles (AUVs). As human infrastructure and research interests moves into large water environments, there is a need for AUVs to navigate in these environments to avoid the expense and risk of human divers. Applications for AUVs include ocean studies and maintaining underwater equipment such as network cables, ship hulls, fish farms and oil rigs.

Robotic navigation underwater is an inherently difficult task, due to the absence of GPS [2]. This means any navigation reference points (i.e. landmarks) cannot be easily surveyed in advance. Other surveying methods, such as theodolites, laser range-finders or physical tapes, are also impractical for use underwater mainly due to human accessibility problems. The absence of accurate information about the environment that a robot will operate in is a good reason to use SALM as a navigation approach for underwater vehicles.

There are existing solutions to AUV navigation that rely on a surface vessel with GPS connection, so called Long Baseline (LBL) and Ultra Short Baseline (USBL) [3]. Both methods use a surface vessel that has at least four acoustic transceivers submerged in the water, each with a GPS derived location. These acoustic transceivers are able to make range measurements to the AUV, and by having at least four ranges from four known positions, the location of the AUV can be unambiguously determined.

While these solutions are widely used, they have several problems. Firstly, it is undesirable to have a surface support vessel for a robotic system that is supposed to be autonomous. Furthermore, these systems result in the position estimate being known at the support vessel, not the AUV. Hence if the robot is to make control decisions based on its location, this information needs to be transmitted to the robot via a cable or acoustic communication methods, both of which may be undesirable. Finally, the fundamental problem with these systems is that they provide a GPS position of the AUV, which is not useful if the AUV's mission environment is not GPS mapped. As mentioned, it is not possible to map landmarks underwater in advance with GPS. A better approach to the navigation problem is to place landmarks in semantic meaningful places in the environment, for example near a object that needs to be inspected, and have the robot map the landmarks and perform its tasks by navigating relative to them, using the SLAM process.

Traditional approaches to SLAM involving equipping the robot with a calibrated camera and using it to make measurement of the landmarks in the robot frame [4]. The landmarks are visually recognisable objects that can be located using various computer vision techniques. This is not a good option for general underwater SLAM as underwater environments often have poor visibility, either

due to turbidity or darkness [2]. This means cameras and other light based sensors such as LiDar are unreliable for making observations beyond a few meters around the robot. An alternative to vision based sensors are acoustic based sensors, where sound waves are emitted by the robot and reflect off the environment. Since the focus is on landmark SLAM, we are interested in an acoustic range measurement device that measures the range to a transponder device located at each landmark. This can be achieved by measuring the time-of-flight of a signal sent to and from the transponder. Such a system is only capable of producing range measurements, with no bearing information. This is the motivation to produce a SLAM filter that relies only on range measurements to the landmarks as its exteroceptive sensing.

For all the reasons outlined above, necessary to design a range-only SLAM filter for underwater navigation. Typical filter based approaches to SLAM involve use of the Extended Kalman Filter (EKF). Unfortunately, the range-only SLAM problem is highly non-linear and an EKF solution is unlikely to perform well due to linearisation error. The EqF is a particular extension of the EKF that has shown good resistance to linearisation error in SLAM and IMU bias estimation problems [5] [6] [7]. Therefore, the first question this project seeks to answer is “*Can an effective Equivariant Filter be designed for Range-only SLAM?*”.

The practical application of a range-only SLAM filter will require appropriate acoustic landmark range measurement devices. Appropriate technologies for this are very prevalent in the commercial domain. In particular, commercial LBL and USBL systems all require acoustic range measurements. So in principle this technology could be repurposed to make landmark range measurements. Unfortunately, these products are unsuited for the research domain for two reasons. They are too expensive for most academic pursuits (tens of thousands of dollars), and they are not open source, and cannot easily be repurposed, extended or improved. Hence it undesirable to use commercial systems, and there is motivation to produce a acoustic range measurement system that can be understood and improved by the academic research community.

The second part of the project seeks to answer “*Can a range measurement system be designed using affordable hardware?*” The scope for this part of the project is to write firmware for an existing acoustic device hardware. The result should be a prototype range measurement system, capable of measuring the distance between two underwater devices.

The rest of this document is laid out as follows. Chapter 2 gives a review of the existing literature for solutions to the SLAM problem in underwater environments. We then focus on works specific to range-only SLAM in an underwater environment. Finally we provide an overview of the Equivariant filter. Chapter 3 focuses on the theory of the proposed range-only SLAM Equivariant filter. It covers the problem formulation and filter derivation, and evaluates the filter performance on a synthetic dataset. Chapter 4 presents a prototype acoustic landmark range measurement system. We discuss the protocol and firmware design for the system. We present the results of range measurement experiments in a realistic lake environment, and discuss the effectiveness and shortcomings of the sensor. Chapter 5 provides the conclusion and suggests the future direction of the work.

Chapter 2

Literature Review

2.1 Sensors Technology for AUV SLAM

Navigation and localisation underwater is an inherently challenging task, and is made especially non-trivial by the absence of GPS, the signals of which are attenuated even in shallow water. The problem is complicated further by poor visibility and diffraction effects that limit the effectiveness of Cameras and LiDar systems. While acoustic based sensors are improving, the technology still suffers shortcomings such as low bandwidth and low data-rate, and also high latency due to the slow speed of sound in water (compared to that of light). Underwater settings also usually introduce multipath effects, due to the water bed and surface, especially in shallow water.

While not technically SLAM, navigation can be performed with proprioceptive sensors only, known as dead-reckoning. On AUVs these sensors typically include accelerometers, gyroscopes, compass, and Doppler Velocity Log (DVL), which measure velocity relative to the ocean floor. In practice, the effectiveness of dead-reckoning is limited as the position error will grow unbounded over time [2]. Usable dead-reckoning approaches will require the use of a DVL since it can provide position information with only one integration, rather than the two required for obtaining position from accelerometers. In addition, an expensive Fibre Optic Gyroscope (FOG) is necessary as MEMS gyroscopes tend to exhibit significant bias and are not useful for dead-reckoning.

Optical Cameras attached to the AUV can be used as the exteroceptive sensors for SLAM. Using computer vision techniques, landmark features need to be extracted from images taken by the camera/s. Then the location in the image of the landmark provides some indication of its position relative to the robot. For cameras, the landmarks can either be distinctive man-made objects placed in the environment in advance, or surrounding man-made infrastructure, such as legs of piers, as in [8]. Natural objects can also be used as landmarks, such as rocks or reefs on the sea floor as in [9] [10] [11] [12] [13]. Stereo Cameras can be used to estimate depth information about the landmarks in a single two image capture as demonstrated in [11]. A significant downside for using cameras for SLAM underwater is the reduced visibility. Only about 5% of surface light reaches a depth of 100m [4], and turbidity can reduce visibility to a few meters in water much shallower than that.

Sonar Imaging is one solution to poor optical visibility. Single-beam echosounders work by emitting a narrow beam (a few degrees) of sonar energy into the water, then receiving any reflections (echoes) off objects in the environment. Based on the time delay, the ranges of objects within the beam can be inferred. To build up a picture, the beam is repeatedly swept over the environment, to get range measurements over an area, as is used in [14]. Multi-beam echosounders are more sophisticated, receiving the return signal in multiple small receivers and inferring direction from them. Multi-beam echosounders are generally better than single-beam echosounders because they can image a large area

at once. As with cameras, the landmarks need to be extracted as features from the sonar images. This can be harder with sonar than with cameras, as there is generally less resolution. But sonar has the advantage that depth information is collected as a side effect.

To build a map for the SLAM problem, artificial landmarks can be used as in [15], where as [16] [14] [17] use various man-made underwater infrastructure. In [18] good results are achieved just by using the features of the sea-bed. It is worth noting that a combination of optical and sonar imaging can be used to achieve the desirable aspects of both technologies, as demonstrated in [9] [19].

Acoustic transponders are a direct way of providing landmarks for SLAM. The transponder is a active device that listens for sonar signals from the AUV, and responds with its own signal back. The AUV can then infer the range to the transponder based on the total time from pinging to receiving the response. SLAM based on range measurements to transponders has been successfully conducted by [21] [22] [23] [24]. A combination of Sonar imaging and transponder range measurements is used in [25]. Using more than one receiver on the AUV can give information about the bearing of the beacon as well as its range making the SLAM task simpler, as demonstrated by [26] [27].

These acoustic range measurement devices are the bases for common commercial navigation solutions from companies like Advance Navigation, SonarDyne, and WaterLinked. The navigation solutions come in two broad categories - Long Base Line (LBL) and Ultra-Short Base Line (USBL). LBL involves four acoustic transceivers, each with individual GPS-derived positions, spread meters apart in the water. The AUV also has an acoustic transceiver. Four ranges are measured from each surface transceiver to the AUV, which gives enough information to calculate the AUV's position. USBL uses a similar principle, except the beacons are only centimetres apart, and are given one GPS location. USBL requires the addition of GPS heading. USBL and LBL are usually deployed from support vessels (ships), but can also be attached individually to buoys with GPS, and wirelessly connected to each other, as shown in [20].

2.2 Algorithms for AUV SLAM

The general approach to the SLAM problem for AUVs is a filter or optimisation technique over a state space that includes the AUV's pose and the landmark positions. The state space may also include information about the underwater environment that effect acoustic measurements. For example, water current [28], the speed of sound in water [29], tide level [30], can all be estimated to improve navigation.

Extended Kalman Filters (EKF) are widely used in SLAM applications, and apply a linearisation of the system model via Taylor expansion. They then apply a recursive predict–update cycle to estimate the system state - i.e. the robot pose and landmark positions. EKF approaches are by far the most numerous, including works [14] [31] [32] [33] [34] [35] [36] [37]. The EKF effectiveness can be limited underwater if landmark features cannot be extracted from images because objects are indistinct due to turbidity or low-light. For SLAM with only range measurements to transponders, the EKF linearises badly and does not produce good results [38]. Generally filtering can be performed in real time, but has quadratic complexity in the number of landmarks.

The Sparse Extended Information Filter (SEIF) is a well-known approach for SLAM using the information filter, and is an alternative to the EKF. It is designed to reduce the computational complexity of matrix multiplication and inversion operations through the use of sparse matrices. Rather than the state covariance form used in EKF, the SEIF uses an information matrix, which can be kept sparse as many state variables are conditionally independent within a small subset of other

variables. Successful implementations of this filter for AUV SLAM include [39] [40] [41].

The Particle Filter is a non-linear filtering solution. It has the advantage that the system model is not approximated. In a Particle Filter, poses and features are represented by discrete particles (points) in the state space, with associated weights. It solves the full SLAM problem, estimating the current pose and full trajectory. Particle Filters have been successfully used for AUV SLAM in [42] [19] [43] [12], however its practical use limited as it can only be used retrospectively off-line, due to high computation demands. FastSLAM is based on the Particle Filter, but each particle maintains its own estimate of the system state (pose and features). Each particles estimate is updated with a separate EKF. FastSLAM can be made to work in real-time for some applications. It has been used effectively in [44] [42] [45] [46] [47].

Acoustic propagations underwater are relatively slow compared Electro-Magnetic based communication on ground. In some cases this means that landmark measurements are delayed by a factor that cannot be ignored. In [48] a delayed state Extended Information Filter (EIF) is implemented to deal with this issue. Alternatively [49] implements a delayed-state Extended Kalman Filter.

GraphSLAM is a method where the entire trajectory and landmarks are estimated, at the cost of high computational complexity. It has been implemented for AUVs in [25] [50]. Similar to EKF, it uses approximation by Taylor expansion, but it accumulates information and is computationally intensive, and is thus only suitable as an off-line algorithm. The environment and the robot's trajectory are represented as a graph, the nodes of which represent the robot's poses and landmarks, and the edges represent constraints between nodes. These constraints can come from odometry measurements, sensor observations, or loop closures (when the robot revisits a previously known location). This construction allows the SLAM problem to be formulated as a nonlinear optimization problem where the objective is to find the configuration of the nodes that minimize the error in the constraints.

2.3 Work on Rang-Only SLAM for AUVs

The first significant work on Range-Only SLAM with fixed transponders is in 2003 by Newman *et. al.* at MIT [21]. Here an AUV is equipped with a conventional LBL transceiver which is used to measure time-of-fights to small submerged transponders. The only information of the transponders are ranges and some approximate pre-known water column depth. Their approach to range-only SLAM is to express the problem as a non-linear optimization over a search space including the beacon locations and the AUV's position at each point in time. They Employ a Least-Squares optimisation to solve for the AUV trajectory and the transponder locations. For ground truth purposes, the transponders have been surveyed, and an EKF is used to obtain position from a LBL arrangement, compass and DVL. Here the data associativity problem is eliminated by having the transponders operate at different acoustic frequencies.

The filter is tested on data gathered in the so-called “GOATS” experiment conducted in 2002 [51] in shallow water off the coast of Italy. The ranging data needed to be heavily processed, with something approximating a low-pass filter, before being fed into Least-Squares optimisation. The final results are promising, with the transponder positions measured to within 5% accuracy of their surveyed location. Unfortunately, the vehicle trajectory was significantly less accurate. A number of other issues are highlighted, particularly regarding the importance of suitable initial transponder location estimates, as well as initial vehicle trajectory direction.

Three years later, Leonard *et. al.* continued this work [22]. Their second iteration focussed on

a powerful outlier rejection algorithm that allowed the optimisation to perform well when the range measurements were degraded by severe outliers and noise. They also developed an effective method to initialise the transponder positions in the optimisation. Again, the GOATS 2002 data is used, and while the final results are dramatically better than in [21], the improvement is mainly to the algorithm robustness.

2.4 Equivariant Filters

Equivariant Filters (EqF) are a type of observer design used for modern robotics systems that admit natural symmetries due to their inherent Lie-group state-space structure. These systems are known as Equivariant. In the past few years this theory has been developed further to systems on manifolds on which a Lie-group acts rather than systems on the Lie-group itself [5] [52]. Equivariant Filters have been successful for estimating IMU biases during a navigation task to improve accuracy [7].

In works by van Goor *et. al.* an EqF is designed for Visual SLAM application [6] [53], and the Lie Group $\text{SLAM}_n^V(3)$ is introduced. In a previous Honours student project [54], it was shown that this group can also be used in the design of a range-only SLAM EqF.

Chapter 3

Equivariant Range SLAM Filter

3.1 Preliminary Notation

3.1.1 Smooth Manifolds

For a smooth manifold \mathcal{M} , let $T_\xi \mathcal{M}$ denote the tangent space of \mathcal{M} at ξ . Given a differentiable function between smooth manifolds $h : \mathcal{M} \rightarrow \mathcal{N}$, the linear map

$$\begin{aligned} D_\xi|_{\xi'} h(\xi) : T_{\xi'} \mathcal{M} &\rightarrow T_{h(\xi')} \mathcal{N} \\ v &\mapsto D_\xi|_{\xi'} h(\xi)[v] \end{aligned}$$

denotes the differential of h with respect to the argument ξ evaluated at ξ' .

3.1.2 Lie Groups

A general Lie group is denoted \mathbf{G} and has Lie algebra \mathfrak{g} . The identity element is written $\text{id} \in \mathbf{G}$. For any $X \in \mathbf{G}$, the left and right translations by X are denoted L_X and R_X and are defined by

$$L_X(Y) := XY, \quad R_X(Y) := YX$$

where XY denotes the group product between X and Y . The adjoint map $\text{Ad} : \mathbf{G} \times \mathfrak{g} \rightarrow \mathfrak{g}$ is defined by

$$\text{Ad}_X[U] = DL_X \cdot DR_{X^{-1}}[U]$$

for every $X \in \mathbf{G}$ and $U \in \mathfrak{g}$. Wedge notation is used to denote the isomorphism between a Lie-algebra element $U \in \mathfrak{g}$ and its corresponding vector-space element $r \in \mathbb{R}^n$:

$$U^\vee = r \quad r^\wedge = U$$

A right action of a Lie group \mathbf{G} on a manifold \mathcal{M} is a smooth map $\phi : \mathbf{G} \times \mathcal{M} \rightarrow \mathcal{M}$ that satisfies

$$\begin{aligned} \phi(Y, \phi(X, \xi)) &= \phi(XY, \xi) \\ \phi(\text{id}, \xi) &= \xi \end{aligned}$$

for any $X, Y \in \mathbf{G}$ and any $\xi \in \mathcal{M}$. The functions

$$\begin{aligned} \exp : \mathfrak{g} &\rightarrow \mathbf{G} \\ \log : \mathbf{G} &\rightarrow \mathfrak{g} \end{aligned}$$

denote the exponential and logarithmic maps for the Lie group, noting that the latter may not be defined for the whole group. Since all Lie groups used here are matrix Lie groups, these functions are the matrix exponential and logarithm.

3.1.3 Special Orthogonal Group

The Special Orthogonal group $\mathbf{SO}(3)$ is the set of all 3D rotation matrices:

$$\mathbf{SO}(3) = \{R \in \mathbb{R}^{3 \times 3} \mid R^T R = I_3, \det(R) = 1\}$$

Its Lie Algebra is represented by $\mathfrak{so}(3)$, and is the set of all skew-symmetric matrices:

$$\mathfrak{so}(3) = \{\Omega^\times \in \mathbb{R}^{3 \times 3} \mid \Omega \in \mathbb{R}^3\}$$

where for any $\Omega = (\Omega_1, \Omega_2, \Omega_3) \in \mathbb{R}^3$, Ω^\times is given by:

$$\Omega^\times := \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix}$$

Note that in the context of $\mathbf{SO}(3)$, the \times operator is equivalent to the \wedge operator:

$$\Omega^\times = \Omega^\wedge, \quad (\Omega^\times)^\vee = \Omega, \quad \Omega \in \mathbb{R}^3, \quad \Omega^\times \in \mathfrak{so}(3)$$

We have the Adjoint for $\mathbf{SO}(3)$ with its Lie algebra:

$$\text{Ad}_R \Omega^\times = (R\Omega)^\times = R\Omega^\times R^\top$$

3.1.4 Extended Pose Group

We use $\mathbf{SE}_2(3)$ to represent the set of all poses with linear velocity in 3D:

$$\mathbf{SE}_2(3) = \left\{ P = \begin{bmatrix} R_P & x_P & v_P \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mid R_P \in \mathbf{SO}(3), x_P, v_P \in \mathbb{R}^3 \right\}$$

Its Lie Algebra is represented by $\mathfrak{se}_2(3)$:

$$\mathfrak{se}_2(3) = \left\{ V = \begin{bmatrix} \Omega_V^\times & s_V & a_V \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mid \Omega_V, s_V, a_V \in \mathbb{R}^3 \right\}$$

Note that the inverse for any $P \in \mathbf{SE}_2(3)$ can be shown to be:

$$P^{-1} = \begin{bmatrix} R_P & x_P & v_P \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} R_P^\top & -R_P^\top x_P & -R_P^\top v_P \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3.1.5 Scaled Orthogonal Transformation

The Scaled Orthogonal Transformation group $\mathbf{SOT}(3)$ is the set of 3D rotation matrices with a scaling factor:

$$\mathbf{SOT}(3) = \{Q = c_Q R_Q \mid R_Q \in \mathbf{SO}(3), c_Q \in \mathbb{R} \setminus \{0\}\}$$

Its Lie Algebra is represented by $\mathfrak{sot}(3)$:

$$\mathfrak{sot}(3) = \{Z = s_Z I_3 + \Omega_Z^\times \mid \Omega_Z \in \mathbb{R}^3, s_Z \in \mathbb{R}\}$$

3.2 Problem Definition

Consider a mobile robot equipped with an on-board IMU and a range measurement sensor that measures the distances to some landmarks. The arbitrary inertial reference frame is denoted $\{0\}$ and the robot's body-fixed frame is denoted $\{b\}$. For simplicity, we assume that both the IMU and range sensor are rigidly attached to the origin of $\{b\}$ and share the same orientation.

3.2.1 System State

We denote the position, orientation and velocity of the robot as x_P , R_P and v_P respectively, all with respect to $\{0\}$. These navigation states are grouped into the extended pose $P \in \mathbf{SE}_2(3)$. The position of the i^{th} landmark can be defined in $\{0\}$, denoted by $p_i \in \mathbb{R}^3$, or in $\{b\}$, denoted by $q_i \in \mathbb{R}^3$. These two representations are related by the robot's state:

$$\begin{aligned} p_i &= R_P q_i + x_P \\ q_i &= R_P^\top (p_i - x_P) \end{aligned} \quad (3.1)$$

The total system state space is the robot's state with respect to $\{0\}$, and each of the landmark positions in $\{b\}$. We denote this state space as \mathcal{T}_n , with state $\xi = (P, q_i)$:

$$\mathcal{T}_n = \mathbf{SE}_2(3) \times \mathbb{R}^3 \times \dots \times \mathbb{R}^3$$

$$\mathcal{T}_n = \{\xi = (P, q_i) := (P, q_1, \dots, q_n) \mid P \in \mathbf{SE}_2(3), q_1, \dots, q_n \in \mathbb{R}^3\}$$

3.2.2 System Input

The system input is some linear acceleration and angular velocity of the robot. Both these quantities are with respect to $\{0\}$, but measured in $\{b\}$ coordinates. The input space is denoted as \mathcal{U} , defined below:

$$\mathcal{U} = \left\{ U = \begin{bmatrix} \Omega_U^\times & 0 & a_U \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mid \Omega_U, a_U \in \mathbb{R}^3 \right\}$$

3.2.3 System Dynamics

The dynamics of each of the navigation states is given in Table 3.1.

Table 3.1: Summary of the robot state and system inputs

Parameter	Description	Dynamics
R_P	Attitude of the robot in $\{0\}$	$\dot{R}_P = R_P \Omega_U^\times$
x_P	Position of the robot in $\{0\}$	$\dot{x}_P = v_P$
v_P	Velocity of the robot w.r.t. $\{0\}$, measured in $\{0\}$	$\dot{v}_P = R_P a_U + g$
a_U	Acceleration of the robot w.r.t. $\{0\}$, measured in $\{b\}$	-
Ω_U	Angular Velocity of the robot w.r.t $\{0\}$, measured in $\{b\}$	-

To achieve the dynamics in Table 3.1, we use the dynamics below:

$$\dot{P} = P(U + D) + G - D \quad (3.2)$$

with

$$G = \begin{bmatrix} 0 & 0 & g \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

and $g \in \mathbb{R}^3$ the acceleration due to gravity.

Lemma 1. *The dynamics of the landmarks q_i in $\{b\}$ can be expressed with*

$$\dot{q}_i = -\Omega_U^\times q_i - R_P^\top v_P$$

Proof. Begin with Equation 3.1 and differentiate both sides:

$$\begin{aligned} q_i &= R_P^\top p_i - R_P^\top x_P \\ \dot{q}_i &= R_P^\top \dot{p}_i + \dot{R}_P^\top p_i - R_P^\top \dot{x}_P - \dot{R}_P^\top x_P \\ &= \dot{R}_P^\top p_i - R_P^\top v_P - \dot{R}_P^\top x_P \\ &= -R_P^\top \dot{R}_P R_P^\top p_i - R_P^\top v_P + R_P^\top \dot{R}_P R_P^\top x_P \\ &= -\Omega_U^\times R_P^\top p_i - R_P^\top v_P + \Omega_U^\times R_P^\top x \\ &= -\Omega_U^\times R_P^\top (p_i - x_P) - R_P^\top v_P \\ &= -\Omega_U^\times q_i - R_P^\top v_P \end{aligned}$$

□

The total state dynamics can be defined by the function f , based on Lemma 1 and Equation 3.2

$$f : \mathcal{T}_n \times \mathcal{U} \rightarrow T_\xi \mathcal{T}_n$$

$$(\dot{P}, \dot{q}_i) = f((P, q_i), U) := (P(U + D) + G - D, -\Omega_U^\times q_i - R_P^\top v_P) \quad (3.3)$$

3.2.4 System Output

The robot makes range measurements to each of the landmarks. The output space is denoted by \mathcal{N}_n .

$$\mathcal{N}_n := \mathbb{R}_+ \times \dots \times \mathbb{R}_+$$

The output function h produces the magnitudes (Euclidean norm) of each of the landmark positions in $\{b\}$.

$$\begin{aligned} h : \mathcal{T}_n &\rightarrow \mathcal{N}_n \\ h(P, q_i) &:= (|q_1|, \dots, |q_n|) \end{aligned} \quad (3.4)$$

3.3 System Symmetry

The implementation of the EqF requires the identification of a transitive Lie group symmetry for the system. We use the **SLAM** group developed in [6]:

$$\mathbf{SLAM}_n(3) = \mathbf{SE}_2(3) \times \mathbf{SOT}(3) \times \dots \times \mathbf{SOT}(3)$$

$$\mathbf{SLAM}_n(3) = \{X = (A, Q_i) := (A, Q_1, \dots, Q_n) \mid A \in \mathbf{SE}_2(3), \quad Q_1, \dots, Q_n \in \mathbf{SOT}(3)\}$$

The Group product, identity and inverse are given by:

$$\begin{aligned} (A^1, Q_i^1) \cdot (A^2, Q_i^2) &:= (A^1 A^2, Q_i^1 Q_i^2) \\ \text{id} &:= (\mathbf{I}_5, (\mathbf{I}_4)_i) \\ (A, Q_i)^{-1} &= (A^{-1}, Q_i^{-1}) \end{aligned}$$

The Group action used in the EqF derivation is dealt with in Lemma 2.

Lemma 2. *The map $\phi : \mathbf{SLAM}_n(3) \times \mathcal{T}_n \rightarrow \mathcal{T}_n$ defined as*

$$\phi((A, Q_i), (P, q_i)) = \left(P A, c_Q^{-1} R_Q^\top q_i \right)$$

is a (right) group action of $\mathbf{SLAM}_n(3)$ on \mathcal{T}_n .

Proof. We require that $\phi((P, q_i), \text{id}) = (P, q_i)$ and that $\phi((A^2, Q_i^2), \phi((A^1, Q_i^1), (P, q_i))) = \phi((A^1, Q_i^1) \cdot (A^2, Q_i^2), (P, q_i))$

The identity part:

$$\begin{aligned} \phi((P, q_i), \text{id}) &= \left(P \mathbf{I}_5, 1^{-1} \mathbf{I}_3^\top q_i \right) \\ &= (P, q_i) \end{aligned}$$

The associativity part:

$$\begin{aligned} \phi((A^2, Q_i^2), \phi((A^1, Q_i^1), (P, q_i))) &= \phi\left((A^2, Q_i^2), (P A^1, c_{Q_i^1}^{-1} R_{Q_i^1}^\top q_i)\right) \\ &= \left(P A^1 A^2, c_{Q_i^2}^{-1} c_{Q_i^1}^{-1} R_{Q_i^2}^\top R_{Q_i^1}^\top q_i \right) \\ &= \left(P A^1 A^2, \left(c_{Q_i^1} c_{Q_i^2}\right)^{-1} \left(R_{Q_i^1} R_{Q_i^2}\right)^\top q_i \right) \\ &= \phi((A^1, Q_i^1) \cdot (A^2, Q_i^2), (P, q_i)) \end{aligned}$$

□

Lemma 3 shows that the output is equivariant.

Lemma 3. *The function $\rho : \mathbf{SLAM}_n(3) \times \mathcal{N}_n \rightarrow \mathcal{N}_n$ defined as:*

$$\rho((A, Q_i), h(P, q_i)) = c_{Q_i}^{-1} |q_i|$$

satisfies the condition

$$\rho((A, Q_i), h(P, q_i)) = h(\phi((A, Q_i), (P, q_i)))$$

and as such the output is equivariant.

Proof. Show RHS is equal to LHS:

$$\begin{aligned}
 \text{RHS} &= h(\phi(A, Q_i), (P, q_i)) \\
 &= h(PA, c_{Q_i}^{-1} R_{Q_i}^\top q_i) \\
 &= |c_{Q_i}^{-1} R_{Q_i}^\top q_i| \\
 &= c_{Q_i}^{-1} |q_i| \\
 &= \rho((A, Q_i), h(P, q_i)) \\
 &= \text{LHS}
 \end{aligned}$$

□

3.4 Origin Choice and Local Coordinates

The EqF design requires a choice of origin configuration and local coordinates about the origin. The local coordinates of the state are denoted by ϵ , and can be broken down into the robot (ϵ_P) and landmark (ϵ_{qi}) parts:

$$\epsilon = (\epsilon_P, \epsilon_{q1}, \dots, \epsilon_{qn}) \in \mathbb{R}^{9+3n}$$

The robot part can be further broken into attitude (ϵ_Ω), position (ϵ_α) and velocity (ϵ_β) parts:

$$\epsilon_P = (\epsilon_\Omega, \epsilon_\alpha, \epsilon_\beta) \in \mathbb{R}^9$$

Let $(P^\circ, q_i^\circ) \in \mathcal{T}_n$ be the fixed origin that the local coordinates are with reference to. The map from the state space to the local coordinates is defined by θ :

$$\theta : \mathcal{T}_n \rightarrow \mathbb{R}^{9+3n}$$

$$\theta(P, q_i) := \left(\log(P^{\circ-1} P)^\vee, q_i - q_i^\circ \right) \quad (3.5)$$

Lemma 4. *The inverse of the mapping $\theta : \mathcal{T}_n \rightarrow \mathbb{R}^{9+3n}$ defined in Equation 3.5 can be expressed as:*

$$\theta^{-1} : \mathbb{R}^{9+3n} \rightarrow \mathcal{T}_n$$

$$\theta^{-1}(\epsilon) = (P^\circ \exp(\epsilon_P^\wedge), \epsilon_{qi} + q_i^\circ)$$

Proof. We require that $\theta(\theta^{-1}(\epsilon)) = \epsilon$:

$$\begin{aligned}
 \theta(\theta^{-1}(\epsilon)) &= \theta((P^\circ \exp(\epsilon_P^\wedge), \epsilon_{qi} + q_i^\circ)) \\
 &= \left(\log(P^{\circ-1} P^\circ \exp(\epsilon_P^\wedge))^\vee, \epsilon_{qi} + q_i^\circ - q_i^\circ \right) \\
 &= (\epsilon_P, \epsilon_{qi}) \\
 &= \epsilon
 \end{aligned}$$

□

3.5 Lift

We lift the system dynamics from the state space to the symmetry Lie Group with the map Λ . Lemma 5 proposes the lift and shows that it satisfies the lift condition from [5].

Lemma 5. *The lift $\Lambda : \mathcal{T}_n \times \mathcal{U} \rightarrow \mathfrak{slam}_n(3)$ defined as*

$$\Lambda((P, q_i), U) := \left((U + D) + P^{-1}(G - D), \left(\Omega_U + \frac{q_i^\times R_P^\top v_P}{q_i^\top q_i}, \frac{q_i^\top R_P^\top v_P}{q_i^\top q_i} \right) \right)$$

satisfies the lift condition

$$D_{(A, Q_i)|\text{id}} \phi_{(P, q_i)}(A, Q_i)[\Lambda((P, q_i), U)] = f((P, q_i), U)$$

with dynamics f defined in Equation 3.3.

Proof. Show LHS is equal to RHS:

$$\begin{aligned} \text{LHS} &= D_{(A, Q_i)|\text{id}} \phi_{(P, q_i)}(A, Q_i)[\Lambda((P, q_i), U)] \\ &= D_{(A, Q_i)|\text{id}} \phi_{(P, q_i)}(A, Q_i) \left[\left((U + D) + P^{-1}(G - D), \left(\Omega_U + \frac{q_i^\times R_P^\top v_P}{q_i^\top q_i}, \frac{q_i^\top R_P^\top v_P}{q_i^\top q_i} \right) \right) \right] \\ &= \frac{d}{dt} \Big|_{t=0} \left(P \left(I_5 + t(U + D) + tP^{-1}(G - D) \right), \left(1 + \frac{q_i^\top R_P^\top v_P}{q_i^\top q_i} t \right)^{-1} \right. \\ &\quad \cdot \left. \left(I_3 + t \left(\Omega_U + \frac{q_i^\times R_P^\top v_P}{q_i^\top q_i} \right)^\times \right)^\top q_i \right) \\ &= \left(P(U + D) + (G - D), \frac{-q_i^\top R_P^\top v_P}{q_i^\top q_i} q_i + \left(\left(\Omega_U + \frac{q_i^\times R_P^\top v_P}{q_i^\top q_i} \right)^\times \right)^\top q_i \right) \\ &= \left(P(U + D) + (G - D), \frac{-q_i^\top R_P^\top v_P}{q_i^\top q_i} q_i + q_i^\times \left(\Omega_U + \frac{q_i^\times R_P^\top v_P}{q_i^\top q_i} \right) \right) \\ &= \left(P(U + D) + (G - D), \frac{-q_i^\top R_P^\top v_P}{q_i^\top q_i} q_i + q_i^\times \Omega_U + q_i^\times \frac{q_i^\times R_P^\top v_P}{q_i^\top q_i} \right) \\ &= \left(P(U + D) + (G - D), \frac{-q_i^\top R_P^\top v_P}{q_i^\top q_i} q_i - \Omega_U^\times q_i + \frac{(q_i^\top R_P^\top v_P) q_i - (q_i^\top q_i) R_P^\top v_P}{q_i^\top q_i} \right) \\ &= \left(P(U + D) + (G - D), -\Omega_U^\times q_i - R_P^\top v_P \right) \\ &= f((P, q_i), U) \\ &= \text{RHS} \end{aligned}$$

□

3.6 Equivariant Filter Derivation

We define the filter observer state estimate for the EqF to be $\hat{X} = (\hat{A}, \hat{Q}_i) \in \text{SLAM}_n(3)$, and define the initial state estimate as $\hat{X}(0) = \text{id}$. At any point in time, the estimated system state $\hat{\xi} = (\hat{P}, \hat{q}_i) \in \mathcal{T}_n$ can be obtained with:

$$(\hat{P}, \hat{q}_i) = \phi((\hat{A}, \hat{Q}_i), (P^\circ, q_i^\circ))$$

The filter dynamics are defined in Equations 3.6, 3.7 and 3.8, following the definitions laid out by van Goor *et. al.* in [5]. Note also that this particular set up is similar to the Visual Odometry and SLAM in [6] and [53].

$$(\dot{\hat{A}}, \dot{\hat{Q}}_i) = (\hat{A}, \hat{Q}_i) \Lambda \left(\phi((\hat{A}, \hat{Q}_i), (P^\circ, q_i^\circ)), U \right) - \Delta(\hat{A}, \hat{Q}_i) \quad (3.6)$$

$$\Delta = D_{E|\text{id}}\phi(P^\circ, q_i^\circ)(E)^\dagger \cdot D_{(P, q_i)|((P^\circ, q_i^\circ)}\theta(P, q_i)^{-1} \Sigma C^{\circ\top} N^{-1} (y - h(\hat{P}, \hat{q}_i)) \quad (3.7)$$

$$\dot{\Sigma} = A^\circ \Sigma + \Sigma A^{\circ\top} + B_t^\circ M B_t^{\circ-1} - \Sigma C^{\circ\top} N^{-1} C^\circ \Sigma \quad (3.8)$$

Where:

- $A^\circ \in \mathbb{R}^{(9+3n) \times (9+3n)}$, $B_t^\circ \in \mathbb{R}^{(3n+9) \times 6}$ and $C^\circ \in \mathbb{R}^{n \times (9+3n)}$ are the State, Input, and Output matrices derived in Lemmas 6, 8 and 7 respectively,
- $\Sigma \in \mathbb{S}_+(9+3n)$ is the Riccati matrix,
- $\Delta \in \mathfrak{slam}_n(3)$ is the EqF correction term,
- $M \in \mathbb{S}_+(6)$ is the input gain matrix,
- $N \in \mathbb{S}_+(n)$ is the output gain matrix,
- $y = h(P, q_i)$ is the measurement of the true system state,
- $D_{E|\text{id}}\phi(P^\circ, q_i^\circ)(E)^\dagger$ is a fixed right inverse of $D_{E|\text{id}}\phi(P^\circ, q_i^\circ)(E)$.

We use $\mathbb{S}_+(k)$ to denote the set of positive-definite $k \times k$ matrices.

3.6.1 State Matrix

Lemma 6. *The Matrix $A_t^\circ \in \mathbb{R}^{(9+3n) \times (9+3n)}$ defined as*

$$A_t^\circ = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & I_3 & 0 & \cdots & 0 \\ g^\times & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & A_{q_1 v} & A_{q_1 q_1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & A_{q_n v} & 0 & \cdots & A_{q_n q_n} \end{bmatrix}$$

with

$$A_{q_i v} = \hat{Q}_i \left(\hat{q}_i^\times \frac{\hat{q}_i^\times \hat{R}_P^\top}{\hat{q}_i^\top \hat{q}_i} - \hat{q}_i \frac{\hat{q}_i^\top \hat{R}_P^\top}{\hat{q}_i^\top \hat{q}_i} \right) \in \mathbb{R}^{3 \times 3}$$

$$A_{q_i q_i} = -\hat{Q}_i \left(\hat{q}_i^\times \left(\frac{(\hat{R}_P^\top \hat{v}_p)^\times}{\hat{q}_i^\top \hat{q}_i} + \frac{2\hat{q}_i^\times \hat{R}_P^\top \hat{v}_P \hat{q}_i^\top}{(\hat{q}_i^\top \hat{q}_i)^2} \right) + \hat{q}_i \left(\frac{(\hat{R}_P^\top \hat{v}_p)^\top}{\hat{q}_i^\top \hat{q}_i} - \frac{2\hat{q}_i^\top \hat{R}_P^\top \hat{v}_P \hat{q}_i^\top}{(\hat{q}_i^\top \hat{q}_i)^2} \right) \right) \hat{Q}_i^{-1} \in \mathbb{R}^{3 \times 3}$$

fulfils the definition of the state matrix proposed by van Goor et. al. in [5] :

$$A_t^\circ = D_{e|\xi^\circ} \theta(e) \cdot D_{\xi|\hat{\xi}} \phi_{\hat{X}}^{-1}(\xi) \cdot D_{E|\text{id}} \phi_{\hat{\xi}}(E) \cdot D_{\xi|\hat{\xi}} \Lambda(\xi, U) \cdot D_{e|\xi^\circ} \phi_{\hat{X}}(e) \cdot D_{\epsilon|0} \theta^{-1}(\epsilon)$$

when taking $P^\circ = I_5$.

Proof. Let $w = (w_P, w_{qi}) \in \mathbb{R}^{9+3n}$ represent a system state in local coordinates.

Starting with the inverse local coordinate map:

$$\begin{aligned} & D_{\epsilon|0} \theta^{-1}(\epsilon)[(w_P, w_{qi})] \\ &= \frac{d}{dt} \Big|_{t=0} (P^\circ \exp(0 + tw_P^\wedge), q_i^\circ + 0 + tw_{qi}) \\ &= (P^\circ w_P^\wedge, w_{qi}) \\ &= (w_P^\wedge, w_{qi}) \end{aligned}$$

The group action $\phi_{\hat{X}}$ part:

$$\begin{aligned} & D_{e|\xi^\circ} \phi_{\hat{X}}(e) [(w_P^\wedge, w_{qi})] \\ &= \frac{d}{dt} \Big|_{t=0} ((P^\circ + tw_P^\wedge) \hat{A}, \hat{c}_{Qi}^{-1} \hat{R}_{Qi}^\top (q_i^\circ + tw_{qi})) \\ &= (w_P^\wedge \hat{A}, \hat{Q}_i^{-1} w_{qi}) \end{aligned}$$

Let the result of the lift part be defined as $k \in \mathfrak{slam}_n(3)$, with its components $k_A \in \mathfrak{se}_2(3)$ and $(k_{\Omega i}, k_{ci}) \in \mathfrak{tot}(3)$.

$$D_{\xi|\hat{\xi}} \Lambda(\xi, U) \left[(w_P^\wedge \hat{A}, \hat{Q}_i^{-1} w_{qi}) \right] := (k_A, k_{\Omega i}, k_{ci})$$

Starting with the $\mathfrak{se}_2(3)$ part:

$$\begin{aligned} k_A &= D_{P|\hat{P}} ((U + D) + P^{-1}(G - D)) \\ &= \frac{d}{dt} \Big|_{t=0} \left((U + D) + \left(\hat{P} + tw_P^\wedge \hat{A} \right)^{-1} (G - D) \right) \\ &= -\hat{P}^{-1} w_P^\wedge \hat{A} \hat{P}^{-1} (G - D) \\ &= -\hat{P}^{-1} w_P^\wedge (G - D) \end{aligned}$$

Note the last line uses the fact that $\hat{P} = P^\circ \hat{A}$ and hence $\hat{A} = \hat{P}$.

Then the $\text{sot}(3)$ part of the lift:

$$\begin{aligned}
 k_{\Omega i} &= D_{(P, q_i) | (\hat{P}, \hat{q}_i)} \left(\Omega_U + \frac{q_i^\times R_P^\top v_P}{q_i^\top q_i} \right) \left[\left(w_P^\wedge \hat{A}, \hat{Q}_i^{-1} w_{qi} \right) \right] \\
 &= \frac{d}{dt} \Big|_{t=0} \left(\Omega_U + \frac{(\hat{q}_i + t\hat{Q}_i^{-1} w_{qi})^\times (\hat{R}_P + tw_\Omega^\times \hat{R}_A)^\top (\hat{v}_P + t(w_\Omega^\times \hat{v}_A + w_v))}{(\hat{q}_i + t\hat{Q}_i^{-1} w_{qi})^\top (\hat{q}_i + t\hat{Q}_i^{-1} w_{qi})} \right) \\
 &= \frac{\left((\hat{Q}_i^{-1} w_{qi})^\times \hat{R}_P^\top v_P + \hat{q}_i^\times (w_\Omega^\times \hat{R}_A)^\top \hat{v}_P + \hat{q}_i^\times \hat{R}_P^\top (w_\Omega^\times \hat{v}_A + w_v) \right) (\hat{q}_i^\top \hat{q}_i) - \left(\hat{q}_i^\times \hat{R}_P^\top \hat{v}_P \right) \left(2\hat{q}_i^\top \hat{Q}_i^{-1} w_{qi} \right)}{(\hat{q}_i^\top \hat{q}_i)^2} \\
 &= \frac{-\left(\hat{R}_P^\top \hat{v}_P \right)^\times \hat{Q}_i^{-1} w_{qi} + \hat{q}_i^\times \hat{R}_A^\top \hat{v}_P^\times w_\Omega - \hat{q}_i^\times \hat{R}_P^\top \hat{v}_A^\times w_\Omega + \hat{q}_i^\times \hat{R}_P^\top w_v}{(\hat{q}_i^\top \hat{q}_i)} - \frac{2\hat{q}_i^\times \hat{R}_P^\top \hat{v}_P \hat{q}_i^\top \hat{Q}_i^{-1}}{(\hat{q}_i^\top \hat{q}_i)^2} w_{qi} \\
 &= \frac{-\left(\hat{R}_P^\top \hat{v}_P \right)^\times \hat{Q}_i^{-1} w_{qi} + \hat{q}_i^\times \hat{R}_P^\top w_v}{(\hat{q}_i^\top \hat{q}_i)} - \frac{2\hat{q}_i^\times \hat{R}_P^\top \hat{v}_P \hat{q}_i^\top \hat{Q}_i^{-1}}{(\hat{q}_i^\top \hat{q}_i)^2} w_{qi}
 \end{aligned}$$

Again, we use the fact that $\hat{v}_A = \hat{v}_P$ and $\hat{R}_A = \hat{R}_P$ since $P^\circ = I_5$.

Then the scalar part of the $\text{sot}(3)$ part follows similarly:

$$\begin{aligned}
 k_{ci} &= D_{(P, q_i) | (\hat{P}, \hat{q}_i)} \left(\frac{q_i^\top R_P^\top v_P}{q_i^\top q_i} \right) \left[\left(w_P^\wedge \hat{A}, \hat{Q}_i^{-1} w_{qi} \right) \right] \\
 &= \frac{d}{dt} \Big|_{t=0} \left(\frac{(\hat{q}_i + t\hat{Q}_i^{-1} w_{qi})^\top (\hat{R}_P + tw_\Omega^\times \hat{R}_A)^\top (\hat{v}_P + t(w_\Omega^\times \hat{v}_A + w_v))}{(\hat{q}_i + t\hat{Q}_i^{-1} w_{qi})^\top (\hat{q}_i + t\hat{Q}_i^{-1} w_{qi})} \right) \\
 &= \frac{\left((\hat{Q}_i^{-1} w_{qi})^\top \hat{R}_P^\top v_P + \hat{q}_i^\top (w_\Omega^\times \hat{R}_A)^\top \hat{v}_P + \hat{q}_i^\top \hat{R}_P^\top (w_\Omega^\times \hat{v}_A + w_v) \right) (\hat{q}_i^\top \hat{q}_i) - \left(\hat{q}_i^\top \hat{R}_P^\top \hat{v}_P \right) \left(2\hat{q}_i^\top \hat{Q}_i^{-1} w_{qi} \right)}{(\hat{q}_i^\top \hat{q}_i)^2} \\
 &= \frac{\left(\hat{R}_P^\top \hat{v}_P \right)^\top \hat{Q}_i^{-1} w_{qi} + \hat{q}_i^\top \hat{R}_A^\top \hat{v}_P^\times w_\Omega - \hat{q}_i^\top \hat{R}_P^\top \hat{v}_A^\times w_\Omega + \hat{q}_i^\top \hat{R}_P^\top w_v}{(\hat{q}_i^\top \hat{q}_i)} - \frac{2\hat{q}_i^\top \hat{R}_P^\top \hat{v}_P \hat{q}_i^\top \hat{Q}_i^{-1}}{(\hat{q}_i^\top \hat{q}_i)^2} w_{qi} \\
 &= \frac{\left(\hat{R}_P^\top \hat{v}_P \right)^\top \hat{Q}_i^{-1} w_{qi} + \hat{q}_i^\top \hat{R}_P^\top w_v}{(\hat{q}_i^\top \hat{q}_i)} - \frac{2\hat{q}_i^\top \hat{R}_P^\top \hat{v}_P \hat{q}_i^\top \hat{Q}_i^{-1}}{(\hat{q}_i^\top \hat{q}_i)^2} w_{qi}
 \end{aligned}$$

Moving on to the group action $\phi_{\hat{\xi}}$ part:

$$\begin{aligned}
 D_{E|\text{id}} \phi_{\hat{\xi}}(E) [(k_A, k_{\Omega i}, k_{ci})] &= \frac{d}{dt} \Big|_{t=0} \left(\hat{P} (I_5 + tk_A), (1 + tk_{ci})^{-1} (I_3 + tk_{\Omega i}^\times)^\top \hat{q}_i \right) \\
 &= \left[\left(\hat{P} k_A, \left((-1)k_{ci} (1 + tk_{ci})^{-2} (I_3 + tk_{\Omega i}^\times)^\top + (1 + tk_{ci})^{-1} (k_{\Omega i}^\times)^\top \right) \hat{q}_i \right) \right]_{t=0} \\
 &= \left(\hat{P} k_A, (-k_{ci} I_3 - k_{\Omega i}^\times) \hat{q}_i \right) \\
 &= (-w_P^\wedge (G - D), -k_{ci} \hat{q}_i + \hat{q}_i^\times k_{\Omega i})
 \end{aligned}$$

The group action $\phi_{\hat{X}^{-1}}$ part:

$$\begin{aligned} & D_{\xi|\xi} \phi_{\hat{X}^{-1}}(\xi) [(-w_P^\wedge(G - D), -k_{ci}\hat{q}_i + \hat{q}_i^\times k_{\Omega i})] \\ &= \frac{d}{dt} \Big|_{t=0} \left(\left(\hat{P} - tw_P^\wedge(G - D) \right) \hat{A}^{-1}, \hat{c}_{Qi} \hat{R}_{Qi} (\hat{q}_i + t (-k_{ci}\hat{q}_i + \hat{q}_i^\times k_{\Omega i})) \right) \\ &= \left(-w_P^\wedge(G - D) \hat{A}^{-1}, -\hat{Q}_i \hat{q}_i k_{ci} + \hat{Q}_i \hat{q}_i^\times k_{\Omega i} \right) \end{aligned}$$

Lastly, the local coordinate map (θ):

$$\begin{aligned} & D_{e|\xi} \theta(e) \left[\left(-w_P^\wedge(G - D) \hat{A}^{-1}, -\hat{Q}_i \hat{q}_i k_{ci} + \hat{Q}_i \hat{q}_i^\times k_{\Omega i} \right) \right] \\ &= \frac{d}{dt} \Big|_{t=0} \left(\log \left(P^{\circ-1} \left(P^\circ + t - w_P^\wedge(G - D) \hat{A}^{-1} \right) \right)^\vee, \left(q_i^\circ + t \left(-\hat{Q}_i \hat{q}_i k_{ci} + \hat{Q}_i \hat{q}_i^\times k_{\Omega i} \right) \right) - q_i^\circ \right) \\ &= \left(\left(-w_P^\wedge(G - D) \hat{A}^{-1} (P^{\circ-1} P^\circ)^{-1} \right)^\vee, -\hat{Q}_i \hat{q}_i k_{ci} + \hat{Q}_i \hat{q}_i^\times k_{\Omega i} \right) \\ &= \left(\left(-w_P^\wedge(G - D) \hat{A}^{-1} \right)^\vee, -\hat{Q}_i \hat{q}_i k_{ci} + \hat{Q}_i \hat{q}_i^\times k_{\Omega i} \right) \end{aligned}$$

Computing the robot state part further we obtain:

$$\begin{aligned} \left(-w_P^\wedge(G - D) \hat{A}^{-1} \right)^\vee &= \left(\begin{bmatrix} -w_\Omega^\times & -w_x & -w_v \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & g \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \hat{R}_{A^{-1}} & \hat{x}_{A^{-1}} & \hat{v}_{A^{-1}} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right)^\vee \\ &= \left(\begin{bmatrix} -w_\Omega^\times & -w_x & -w_v \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & g \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \right)^\vee \\ &= \begin{bmatrix} 0 & w_v & -w_\Omega^\times g \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}^\vee \\ &= \begin{bmatrix} 0 \\ w_v \\ g^\times w_\Omega \end{bmatrix} \end{aligned}$$

Computing the landmark part further we obtain:

$$\begin{aligned} -\hat{Q}_i \hat{q}_i k_{ci} + \hat{Q}_i \hat{q}_i^\times k_{\Omega i} &= -\hat{Q}_i \hat{q}_i \left(\frac{\left(\hat{R}_P^\top \hat{v}_P \right)^\top \hat{Q}_i^{-1} w_{qi} + \hat{q}_i^\top \hat{R}_P^\top w_v}{(\hat{q}_i^\top \hat{q}_i)} - \frac{2\hat{q}_i^\top \hat{R}_P^\top \hat{v}_P \hat{q}_i^\top \hat{Q}_i^{-1}}{(\hat{q}_i^\top \hat{q}_i)^2} w_{qi} \right) \\ &\quad + \hat{Q}_i \hat{q}_i^\times \left(\frac{-\left(\hat{R}_P^\top \hat{v}_P \right)^\times \hat{Q}_i^{-1} w_{qi} + \hat{q}_i^\times \hat{R}_P^\top w_v}{(\hat{q}_i^\top \hat{q}_i)} - \frac{2\hat{q}_i^\times \hat{R}_P^\top \hat{v}_P \hat{q}_i^\top \hat{Q}_i^{-1}}{(\hat{q}_i^\top \hat{q}_i)^2} w_{qi} \right) \\ &= -\hat{Q}_i \left(\hat{q}_i^\times \left(\frac{(\hat{R}_P^\top \hat{v}_p)^\times}{\hat{q}_i^\top \hat{q}_i} + \frac{2\hat{q}_i^\times \hat{R}_P^\top \hat{v}_P \hat{q}_i^\top}{(\hat{q}_i^\top \hat{q}_i)^2} \right) + \hat{q}_i \left(\frac{(\hat{R}_P^\top \hat{v}_p)^\top}{\hat{q}_i^\top \hat{q}_i} - \frac{2\hat{q}_i^\top \hat{R}_P^\top \hat{v}_P \hat{q}_i^\top}{(\hat{q}_i^\top \hat{q}_i)^2} \right) \right) \hat{Q}_i^{-1} w_{qi} \\ &\quad + \hat{Q}_i \left(\hat{q}_i^\times \frac{\hat{q}_i^\times \hat{R}_P^\top}{\hat{q}_i^\top \hat{q}_i} - \hat{q}_i \frac{\hat{q}_i^\top \hat{R}_P^\top}{\hat{q}_i^\top \hat{q}_i} \right) w_v \end{aligned}$$

So we require a matrix A_t° a such that:

$$A_t^\circ \begin{bmatrix} w_\Omega \\ w_x \\ w_v \\ w_{q1} \\ \vdots \\ w_{qn} \end{bmatrix} = \begin{bmatrix} 0 \\ w_v \\ g^\times w_\Omega \\ A_{q_1 v} w_v + A_{q_1 q_1} w_{q1} \\ \vdots \\ A_{q_n v} w_v + A_{q_n q_n} w_{qn} \end{bmatrix}$$

with:

$$\begin{aligned} A_{q_i v} &= \hat{Q}_i \left(\hat{q}_i^\times \frac{\hat{q}_i^\times \hat{R}_P^\top}{\hat{q}_i^\top \hat{q}_i} - \hat{q}_i \frac{\hat{q}_i^\top \hat{R}_P^\top}{\hat{q}_i^\top \hat{q}_i} \right) \\ A_{q_i q_i} &= -\hat{Q}_i \left(\hat{q}_i^\times \left(\frac{(\hat{R}_P^\top \hat{v}_p)^\times}{\hat{q}_i^\top \hat{q}_i} + \frac{2\hat{q}_i^\times \hat{R}_P^\top \hat{v}_P \hat{q}_i^\top}{(\hat{q}_i^\top \hat{q}_i)^2} \right) + \hat{q}_i \left(\frac{(\hat{R}_P^\top \hat{v}_p)^\top}{\hat{q}_i^\top \hat{q}_i} - \frac{2\hat{q}_i^\top \hat{R}_P^\top \hat{v}_P \hat{q}_i^\top}{(\hat{q}_i^\top \hat{q}_i)^2} \right) \right) \hat{Q}_i^{-1} \end{aligned}$$

By inspecting the proposed matrix, this can be seen to be the case.

□

3.6.2 Output Matrix

Lemma 7. *The Output Matrix $C_t^\circ \in \mathbb{R}^{n \times (9+3n)}$ defined as*

$$C_t^\circ = \begin{bmatrix} 0 & C_{q1} & 0 & \cdots & 0 \\ 0 & 0 & C_{q2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & C_{qn} \end{bmatrix}$$

with

$$C_{qi} := \frac{q_i^{\circ\top}}{\hat{c}_{Qi} |q_i^\circ|} \in \mathbb{R}^{1 \times 3}$$

fulfils the definition of the input matrix proposed by van Goor et. al. in [5] :

$$C_t^\circ = D_{\xi|\hat{\xi}} h(\xi) \cdot D_{e|\xi^\circ} \phi_{\hat{X}}(e) \cdot D_{\epsilon|0} \theta^{-1}(\epsilon)$$

when taking $P^\circ = I_5$.

Proof. Let $w = (w_P, w_{qi}) \in \mathbb{R}^{9+3n}$ represent a system state in local coordinates. From Lemma 6 we have that:

$$D_{\epsilon|0} \theta^{-1}(\epsilon)[w_P, w_{qi}] = (w_P^\wedge, w_{qi})$$

The group action part:

$$\begin{aligned} &D_{e|\xi^\circ} \phi_{\hat{X}}(e)[w_P^\wedge, w_{qi}] \\ &= \frac{d}{dt} \Big|_{t=0} \left((P^\circ + tw_P^\wedge) \hat{A}, \hat{c}_{Qi}^{-1} R_{Qi}^\top (q_i^\circ + tw_{qi}) \right) \\ &= (w_P^\wedge \hat{A}, \hat{c}_{Qi}^{-1} R_{Qi}^\top w_{qi}) \end{aligned}$$

Now let $k_i = \hat{c}_{Qi}^{-1} R_{Qi}^\top w_{qi}$. The output part:

$$\begin{aligned}
 & D_{\xi|\hat{\xi}} h(\xi)[w_P^\wedge, k_i] \\
 &= D_{q_i|\hat{q}_i}(|q_i|)[k_i] \\
 &= \frac{d}{dt} \Big|_{t=0} |\hat{q}_i + tk_i| \\
 &= \frac{d}{dt} \Big|_{t=0} \sqrt{(\hat{q}_i + tk_i)^\top (\hat{q}_i + tk_i)} \\
 &= \left[\frac{1}{2\sqrt{(\hat{q}_i + tk_i)^\top (\hat{q}_i + tk_i)}} \left(k_i^\top (\hat{q}_i + tk_i) + (\hat{q}_i + tk_i)^\top k_i \right) \right]_{t=0} \\
 &= \frac{1}{2\sqrt{\hat{q}_i^\top \hat{q}_i}} (k_i^\top \hat{q}_i + \hat{q}_i^\top k_i) \\
 &= \frac{\hat{q}_i^\top}{\sqrt{\hat{q}_i^\top \hat{q}_i}} k_i \\
 &= \frac{\hat{q}_i^\top}{|\hat{q}_i|} k_i \\
 &= \frac{\hat{q}_i^\top}{|\hat{q}_i|} \hat{c}_{Qi}^{-1} R_{Qi}^\top w_{qi} \\
 &= \frac{\left(\hat{c}_{Qi}^{-1} R_{Qi}^\top q_i^\circ \right)^\top}{\hat{c}_{Qi}^{-1} |q_i^\circ|} \hat{c}_{Qi}^{-1} R^\top w_{qi} \\
 &= \frac{q_i^\circ{}^\top}{|q_i^\circ|} \hat{c}_{Qi}^{-1} w_{qi}
 \end{aligned}$$

The last two lines come from the group action on the landmark: $\hat{q}_i = \hat{c}_{Qi}^{-1} R_{Qi}^\top q_i^\circ$ and thus $|\hat{q}_i| = \hat{c}_{Qi} |q_i^\circ|$. We require a matrix C_t° such that:

$$C_t^\circ \begin{bmatrix} w_P \\ w_{q1} \\ \vdots \\ w_{qn} \end{bmatrix} = \begin{bmatrix} \frac{q_1^\circ{}^\top}{\hat{c}_{Q1} |q_1^\circ|} w_{q1} \\ \vdots \\ \frac{q_n^\circ{}^\top}{\hat{c}_{Qn} |q_n^\circ|} w_{qn} \end{bmatrix}$$

It is easy to see that the proposed matrix will achieve this. \square

3.6.3 Input Matrix

Lemma 8. *The Input Matrix $B_t^\circ \in \mathbb{R}^{(3n+9) \times 6}$ defined as*

$$B_t^\circ = \begin{bmatrix} \hat{R}_A & 0 \\ \hat{x}_A^\times \hat{R}_A & 0 \\ \hat{v}_A^\times \hat{R}_A & \hat{R}_A \\ q_1^\circ{}^\times \hat{R}_{Q1} & 0 \\ \vdots & \vdots \\ q_n^\circ{}^\times \hat{R}_{Qn} & 0 \end{bmatrix}$$

fulfils the definition of the input matrix proposed by van Goor et. al. in [5]:

$$B_t^\circ = D_{e|\xi^\circ} \theta(e) \cdot D_{\xi|\hat{\xi}} \phi_{\hat{X}^{-1}}(\xi) \cdot D_{E|\text{id}} \phi_{\hat{\xi}}(E) \cdot D_{u|(\Omega, a)} \Lambda(\hat{\xi}, u)$$

taking $P^\circ = I_5$.

Proof. Let $w = (\Omega_w, a_w) \in \mathbb{R}^6$ represent an input to the system. Begin with the lift part:

$$\begin{aligned} & D_{u|(\Omega, a)} \left((U + D) + \hat{P}^{-1}(G - D), \Omega_U + \frac{\hat{q}_i^\times \hat{R}_P \hat{v}_P}{\hat{q}_i^\top \hat{q}_i}, \frac{\hat{q}_i^\top \hat{R}_P \hat{v}_P}{\hat{q}_i^\top \hat{q}_i} \right) [w] \\ &= \frac{d}{dt} \Big|_{t=0} \left(((U + tW) + D) \hat{P}^{-1}(G - D), \Omega_U + t\Omega_w + \frac{\hat{q}_i^\times \hat{R}_P \hat{v}_P}{\hat{q}_i^\top \hat{q}_i}, \frac{\hat{q}_i^\top \hat{R}_P \hat{v}_P}{\hat{q}_i^\top \hat{q}_i} \right) \\ &= (w, \Omega_w, 0) \in \mathfrak{slam}_n(3) \end{aligned}$$

The group action parts:

$$\begin{aligned} & D_{E|\text{id}} \phi_{\hat{\xi}}(E) [(w^\wedge, \Omega_w^\times, 0)] \\ &= \frac{d}{dt} \Big|_{t=0} \left(\hat{P}(\mathbf{I}_5 + tw^\wedge), (1+0t)^{-1} (\mathbf{I}_3 + t\Omega_w^\times)^\top \hat{q}_i \right) \\ &= \left[\left(\hat{P}w^\wedge, 0(-1)(1+0t)^{-2} (\mathbf{I}_3 + t\Omega_w^\times)^\top \hat{q}_i + (1+0t)^{-1} (\Omega_w^\times)^\top \hat{q}_i \right) \right]_{t=0} \\ &= \left(\hat{P}w^\wedge, (\Omega_w^\times)^\top \hat{q}_i \right) \\ &= \left(\hat{P}w^\wedge, \hat{q}_i^\times \Omega_w \right) \end{aligned}$$

$$\begin{aligned} & D_{\xi|\hat{\xi}} \phi_{\hat{X}^{-1}}(\xi) \left[\left(\hat{P}w^\wedge, \hat{q}_i^\times \Omega_w \right) \right] \\ &= \frac{d}{dt} \Big|_{t=0} \left(\left(\hat{P} + t\hat{P}w^\wedge \right) \hat{A}^{-1}, (\hat{c}_{Qi}^{-1})^{-1} \left(\hat{R}_{Qi}^\top \right)^\top (\hat{q}_i + t\hat{q}_i^\times \Omega_w) \right) \\ &= \left(\hat{P}w^\wedge \hat{A}^{-1}, \hat{c}_{Qi} \hat{R}_{Qi} \hat{q}_i^\times \Omega_w \right) \\ &= \left(\hat{A}w^\wedge \hat{A}^{-1}, \hat{c}_{Qi} \hat{R}_{Qi} \left(\hat{c}_{Qi}^{-1} \hat{R}_{Qi}^\top q_i^\circ \right)^\times \Omega_w \right) \\ &= \left(\hat{A}w^\wedge \hat{A}^{-1}, \hat{c}_{Qi} \hat{c}_{Qi}^{-1} \hat{R}_{Qi} \hat{R}_{Qi}^\top q_i^\circ \times \hat{R}_{Qi} \Omega_w \right) \\ &= \left(\hat{A}w^\wedge \hat{A}^{-1}, q_i^\circ \times \hat{R}_{Qi} \Omega_w \right) \end{aligned}$$

The last few lines use the group action and the Adjoint for $\mathbf{SO}(3)$. The local coordinate map part:

$$\begin{aligned} & D_{e|\xi^\circ} \theta(e) \left(\left[\hat{A}w^\wedge \hat{A}^{-1}, q_i^\circ \times \hat{R}_{Qi} \Omega_w \right] \right) \\ &= \frac{d}{dt} \Big|_{t=0} \left(\log \left(P^\circ{}^{-1} \left(P^\circ + t\hat{A}w^\wedge \hat{A}^{-1} \right) \right)^\vee, \left(q_i^\circ + t q_i^\circ \times \hat{R}_{Qi} \Omega_w \right) - q_i^\circ \right) \\ &= \left(\left(\hat{A}w^\wedge \hat{A}^{-1} (P^\circ{}^{-1} P^\circ) {}^{-1} \right)^\vee, q_i^\circ \times \hat{R}_{Qi} \Omega_w \right) \\ &= \left(\left(\hat{A}w^\wedge \hat{A}^{-1} \right)^\vee, q_i^\circ \times \hat{R}_{Qi} \Omega_w \right) \end{aligned}$$

Computing the robot pose part further:

$$\begin{aligned}
 (\hat{A}w^\wedge \hat{A}^{-1})^\vee &= \left(\begin{bmatrix} \hat{R}_A & \hat{x}_A & \hat{v}_A \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Omega_w^\times & 0 & a_w \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{R}_A^\top & -\hat{R}_A^\top \hat{x}_A & -\hat{R}_A^\top \hat{v}_A \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right)^\vee \\
 &= \left(\begin{bmatrix} \hat{R}_A \Omega_w^\times \hat{R}_A^\top & -\hat{R}_A \Omega_w^\times \hat{R}_A^\top \hat{x}_A & -\hat{R}_A \Omega_w^\times \hat{R}_A^\top \hat{v}_A + \hat{R}_A a_w \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^\vee \\
 &= \left(\begin{bmatrix} (\hat{R}_A \Omega_w)^\times & \hat{x}_A^\times \hat{R}_A \Omega_w & \hat{v}_A^\times \hat{R}_A \Omega_w + \hat{R}_A a_w \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^\vee \\
 &= \begin{bmatrix} \hat{R}_A \Omega_w \\ \hat{x}_A^\times \hat{R}_A \Omega_w \\ \hat{v}_A^\times \hat{R}_A \Omega_w + \hat{R}_A a_w \end{bmatrix}
 \end{aligned}$$

We require a matrix B_t° such that:

$$B_t^\circ \begin{bmatrix} \Omega_w \\ a_w \end{bmatrix} = \begin{bmatrix} \hat{R}_A \Omega_w \\ \hat{x}_A^\times \hat{R}_A \Omega_w \\ \hat{v}_A^\times \hat{R}_A \Omega_w + \hat{R}_A a_w \\ q_1^\circ \times \hat{R}_{Q1} \Omega_w \\ \vdots \\ q_n^\circ \times \hat{R}_{Qn} \Omega_w \end{bmatrix}$$

And clearly the proposed matrix will achieve this:

$$\begin{bmatrix} \hat{R}_A & 0 \\ \hat{x}_A^\times \hat{R}_A & 0 \\ \hat{v}_A^\times \hat{R}_A & \hat{R}_A \\ q_1^\circ \times \hat{R}_{Q1} & 0 \\ \vdots & \vdots \\ q_n^\circ \times \hat{R}_{Qn} & 0 \end{bmatrix} \begin{bmatrix} \Omega_w \\ a_w \end{bmatrix} = \begin{bmatrix} \hat{R}_A \Omega_w \\ \hat{x}_A^\times \hat{R}_A \Omega_w \\ \hat{v}_A^\times \hat{R}_A \Omega_w + \hat{R}_A a_w \\ q_1^\circ \times \hat{R}_{Q1} \Omega_w \\ \vdots \\ q_n^\circ \times \hat{R}_{Qn} \Omega_w \end{bmatrix}$$

□

3.7 Filter Simulation Experiment

3.7.1 Simulation and Filter set-up

The filter is tested on a synthetic trajectory. The trajectory consists of a circular path in the x-y plane with a diameter 2 meters. The robot completes several laps of this circle path, at a constant velocity of 2m/s. In addition to the translation, the robot undergoes some random wandering in its rotation. Synthetic IMU acceleration and angular velocity measurements are calculated from the time derivatives of the trajectory. Zero-mean Gaussian noise is added to the IMU measurements, as shown in Table 3.2. Note that there are no biases here.

Four landmarks are placed above the trajectory in a grid formation. Synthetic range measurements to the landmarks are calculated, again with zero-mean Gaussian noise and no bias. The rate of range measurements is slower than that of the IMU, to reflect the real world nature of such measurement devices.

Table 3.2: Measurement noise parameters for the filter simulation.

Measurement	Noise Standard Deviation	Rate
Acceleration	0.01 m/s ²	200 Hz
Angular Velocity	0.005 Rad	200 Hz
Landmark Range	0.01 m	5 Hz

The goal of the experiment is for the robot to locate itself and estimate the relative positions of the landmarks. The filter must complete the full SLAM task; there is no prior knowledge of the landmark positions, and no pre filtering. The initial filter state covariance matrix is set to $I_3 * 5.0(m)$ to provide the filter with sufficient robustness so that it can converge even with no landmark prior. Since there is no prior information on the landmarks, the result of the filter will not be aligned to the global reference frame. To calculate this alignment error, the Umeyama algorithm is used to find the $\text{SE}(3)$ transformation that best maps the estimated landmarks to the ground truth landmarks.

3.7.2 Results

Figure 3.1 shows a 3-dimensional representation of the simulation results. The true and estimated circle trajectories are shown by the solid lines. As can be seen, there is some initial transient error in the estimated robot position, but it soon converges close to the true circular trajectory.

The initial landmark estimates are each set to the first range measurement in the positive z direction. The trajectory of the landmark estimates as they converge can be seen as the dashed lines in figure 3.1. This corresponds to the filter working to estimate the rotation part of the $\text{SOT}(3)$ landmark states. The scale part of the $\text{SOT}(3)$ landmark state is much easier for the filter to deduce from the range measurements.

Note that alignment is not applied in the plot as it is visually insignificant. This is because the synthetic data is well behaved with no biases, so the alignment error is minimal.

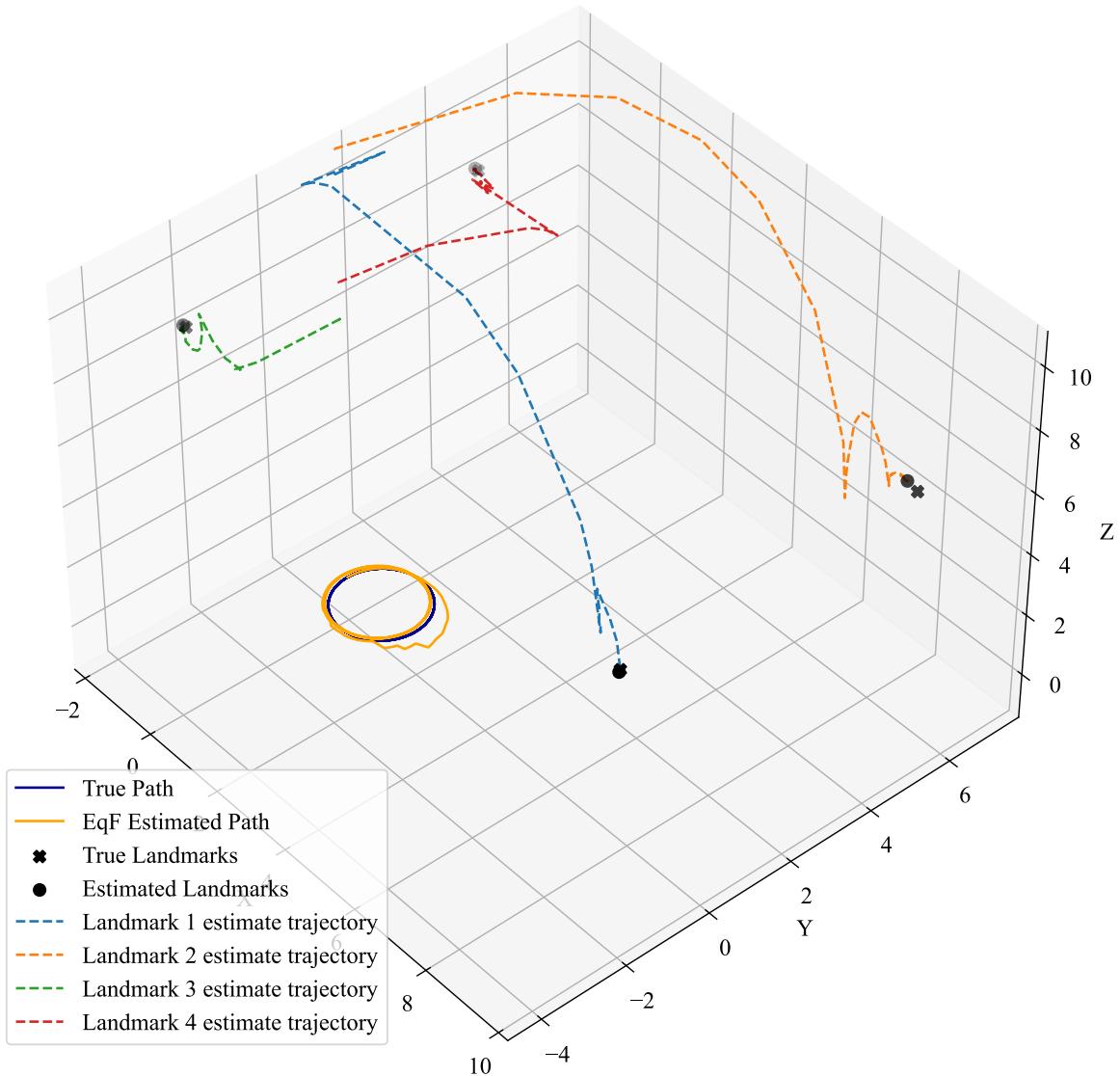


Figure 3.1: The true and estimated trajectories of the robot and the landmark position (without alignment).

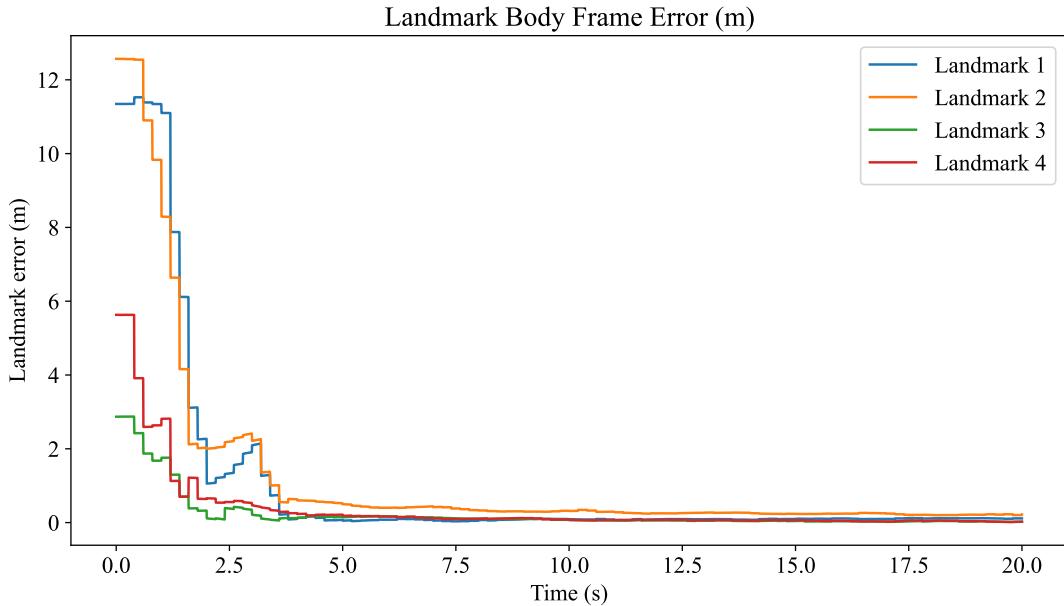


Figure 3.2: Estimation error of landmark positions.

Figure 3.2 shows the evolution of the error of the landmark estimation (in the body frame) over time. After some transient period, the filter is able to converge on the true body frame landmark value, as the error goes towards zero. Within the first 10 seconds the magnitude of the landmark errors goes below 0.4m. Note that the stepped nature of the plot is due to the low frequency range measurements.

Table 3.3 shows the landmark estimate position error after applying alignment. Most landmarks are able to achieve an accuracy of around 10cm, which is understandable given the general size of the simulation. Notice that landmarks 1 and 2 have notably more error after alignment than landmarks 3 and 4. This is because landmarks 1 and 2 are both further away. When a landmark is further away, less information can be obtained from only range measurements as movement induces less relative change in the range measurement.

Table 3.3: The landmark error after alignment.

Landmark Error Magnitude After Alignment (m)				
Landmark 1	Landmark 2	Landmark 3	Landmark 4	Average
0.093	0.136	0.072	0.074	0.094

Chapter 4

Acoustic Range Sensor

4.1 Hardware Overview

This chapter documents the design of a prototype landmark range measurement system, using hardware provided by the underwater robotics company Blue Robotics. The decision to use hardware from Blue Robotics comes after investigation into multiple commercial sonar transponder suppliers. Companies considered include Nantong Saiyang Electronics, Raymarine, Advanced Navigation, WaterLinked, EvoLogics and SonarDyne. While they all offer sonar transponder based technology, their products suffer from two problems. The systems are too expensive (upwards of tens of thousands of dollars), and they use proprietary software and hardware that cannot be easily modified for experiments in this project. Blue Robotics is more framed towards Universities and individuals putting together semi-modular devices to create navigation solutions. Many of their products are open source, and they encourage modification and adaptation of their products. This makes them a more suitable company to work with. Upon being contacted, they showed interest in the research effort for underwater navigation using SLAM.



Figure 4.1: Blue Robotics altimeter device. Enclosed within the aluminium housing are the electronics, battery and microcontroller. The closer end is the diaphragm, made of a rubber material. The piezoelectric transducer is embedded in the protrusion of the diaphragm.

Unfortunately, Blue Robotics does not have an off-the-shelf acoustic transponder device that they can provide. Instead, they provided four of their altimeter devices, which are intended to be used to measure water depth by reflecting acoustic signals off the ocean floor. One of the provided devices is depicted in Figure 4.1

While altimeter device is not exactly the landmark range measurement device required for the navigation filter presented in Chapter 3, it has all of the hardware components necessary for such a device. The design task documented in this chapter is the design of a range measurement protocol for measuring ranges to landmarks, and firmware that implements the protocol to run on the hardware provided by Blue Robotics.

The hardware provided by blue robotics includes:

- Waterproof aluminium enclosure
- Pezo-electric transducer and diaphragm
- Battery power supply
- Pre-amplifiers and analogue filtering electronics for receiving
- Transducer driver electronics for transmitting
- STM32 F303re microcontroller

In water, the transducer resonates at 115kHz. Blue Robotics provided a modified diaphragm that has an approximately cylindrical power profile as in Figure 4.2. It provides a reasonably omnidirectional transmission when the devices are held vertically all on a similar horizontal plane, as was case for the experiments here. The diaphragm used for the devices intended purpose as a altimeter is highly directional (15 degree beam-width), and would not have been suitable.

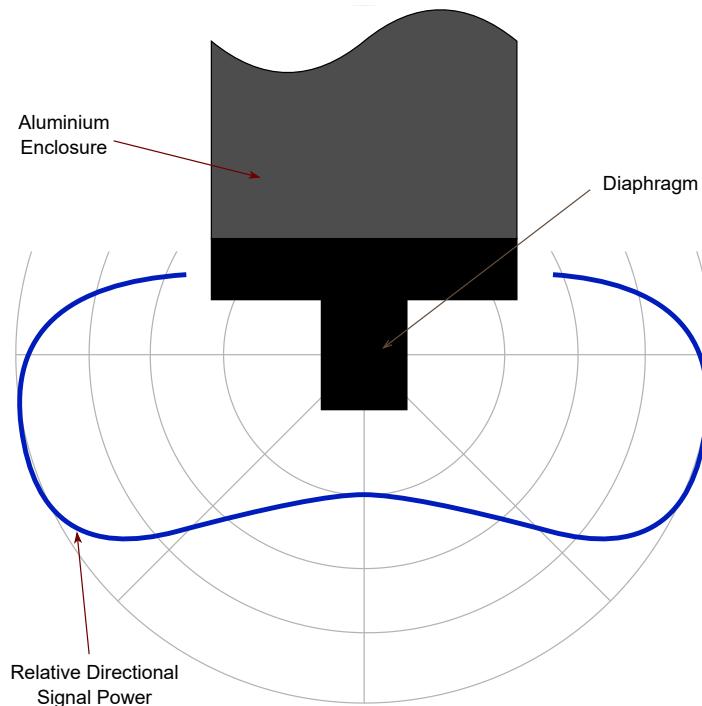


Figure 4.2: Approximate transducer cylindrical power distribution

4.2 Acoustic Range Measurement Protocol

The work here is to design a protocol for range measurements from a master device to a transponder device. When used for SLAM, the master device would be attached to the AUV, and the transponder would be co-located with a landmark. Figure 4.3 shows the basic principle of the protocol. The master unit emits a short acoustic signal in all directions which propagates through the water at the speed of sound. The transponder device will receive this signal, and respond with its own signal, which the master will receive. The master calculates the total time between sending and receiving and infers the range based on the speed of sound in water. For now we assume that this “signal” sent between the two devices is a perfect unit impulse.

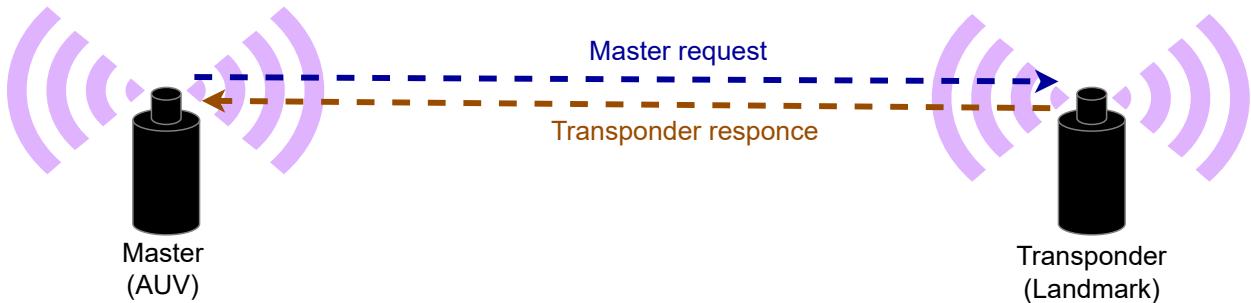


Figure 4.3: Principle of Time of Flight range measurement to a landmark.

The relationship of the range and TOF in its simplest form is shown in Equation 4.1:

$$d = \frac{t_{TOF} \cdot c_w}{2} \quad (4.1)$$

where t_{TOF} is the total Time-Of-Flight (TOF) calculated by the master, c_w is the speed of sound in water, and d is the inferred range to the transponder.

In practice, the master unit cannot record the TOF directly, but must instead record “timestamps” of when the first transmit and first receive events occur. These timestamps are the elapsed time since the master device was switched on, measured in some arbitrary unit known to the master device. We denote these timestamps as t_{TX} and t_{RX} , noting that here TX and RX represent transmit and receive respectively on the master unit. The TOF is then $t_{TOF} = t_{RX} - t_{TX}$.

The master unit does not make measurements at random times, but takes periodic measurements continually. This corresponds to repeatedly transmitting the request pulse to the transponder with a constant delay between each request. We denote the period between each request as T_{meas} . We take advantage of this periodicity to avoid having to record the transmit timestamp t_{TX} .

We now introduce the concept of a system timestamp, denoted k_{TX} and k_{RX} . Both system timestamps are rounded to positive whole numbers, i.e. $k_{TX}, k_{RX} \in \mathbb{N}$. We denote the set of natural numbers as \mathbb{N} and we include 0 as a natural number. The smallest unit of time between two consecutive system timestamps is denoted Δt . To have sufficient timing resolution for the TOF, we assume that Δt is very small. In other words, the system timestamps are incremented sufficiently frequently so that the rounding error caused by this discretization of time is negligible. We further assume that the measurement period is a perfect integer multiple of the timestamp resolution, i.e. $T_{meas} = n\Delta t, n \in \mathbb{N}$. The first request to the transponder should occur at $k_{TX} = 0$. Finally, we assume that the TOF is always less than the measurement period, $t_{TOF} < T_{meas}$. Practically, this means that the true range between the master and transponder never exceeds some limit.

Now the TOF can be calculated by only measuring the RX system timestamp, as in Equation 4.2:

$$t_{TOF} = \Delta t \cdot \text{mod} \left(k_{RX}, \frac{T_{meas}}{\Delta t} \right) \quad (4.2)$$

where the function $\text{mod} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ is the modulo operation that returns the remainder after integer division of its arguments.

In practice, the transponder cannot respond to the master immediately. There is some constant delay between when it receives and retransmits the signal. This delay should be roughly programmable (to avoid multipath as discussed in section 4.4), however, its exact value is unknown and must be found through calibration. It is also important to note that neither the master or transponder can know precisely when a transmission occurs with respect to their system timestamps. There will be some constant but unknown delay, due to the dynamics of the sonar transducer, for example. We lump both these effects into the calibration parameter α , in Equation 4.3. The precise value for the speed of sound in water is also unknown, so the calibration parameter β is used to correct this. This parameter will also correct any inaccuracy of the master device clock frequency.

We now have the complete equation that the master unit uses to perform a range measurement by recording only the time that the response arrives from the transponder (Equation 4.3).

$$d = \Delta t \cdot \text{mod} \left(k_{RX}, \frac{T_{meas}}{\Delta t} \right) \cdot \beta - \alpha \quad (4.3)$$

Where:

- d is the range measurement (m)
- T_{meas} is the period between transponder requests (s)
- Δt is the resolution of the system timestamps (s)
- k_{RX} is the master device system timestamp when it receives the transponder response (unitless)
- α is the offset calibration parameter (m)
- β is the scale calibration parameter (m/s)

The calibration procedure to find α and β involves taking at least two measurements at known distances. Then a straight line is fit to the (d, t_{TOF}) points. The gradient of this line is $\beta \cdot \Delta t$, and the y intercept is α .

4.3 Peripheral and Software Design

The hardware provided by blue robotics includes a STM32 F303RE microcontroller that can monitor the received signals after analogue processing, and can control the transducer driver to emit signals. One task in this project is to write firmware with appropriate peripheral settings to direct the microcontroller to follow the range measurement protocol laid out in section 4.2. Here “peripherals” refers to the auxiliary devices on the microcontroller die outside on the CPU, mainly the timers, ADC (Analogue to Digital Converter) and DMA (Direct Memory Access) channels. Two sets of firmware are required for the master and transducer units.

The key performance requirement for the firmware on both devices is the resolution for the receive timestamps, from Equation 4.3 in section 4.2. This will determine the precision of the range measurements. The target range precision is $\pm 1\text{cm}$, so we require a timing precision of $\pm 13\mu\text{s}$. This is the primary driver for the peripheral design towards the use of DMA and not timer interrupts.

Figure 4.4 shows how the peripherals are configured on the microcontroller. There are two timers driven off the same system clock. These control the two DMA channels for the receive and transmit functions. The received signal is filtered and amplified by the analogue electronics, before being digitised by the ADC. The first DMA channel is used to then pipe this data directly into the RX buffer in memory. The goal is to preserve the timing accuracy of the signal in the buffer. The RX buffer is used in a circular double buffer configuration. Data is continually written to the buffer and once the buffer is filled the DMA jumps back to the beginning and starts overwriting the buffer again.

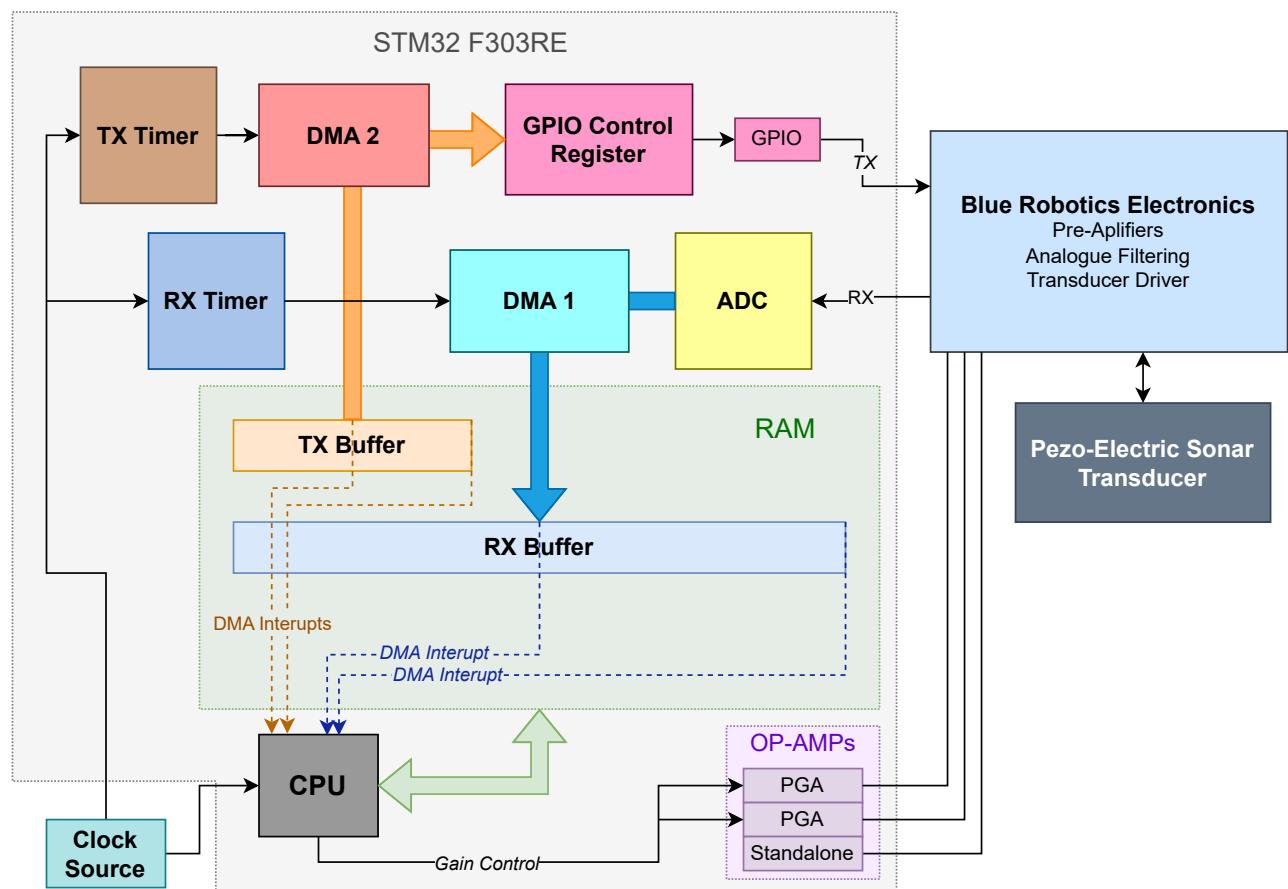


Figure 4.4: Hardware architecture for the Master and Transponder devices.

An interrupt is generated when the DMA finishes writing to the first half of the buffer, and when finishing writing to the second half of the buffer. These interrupts are just used to direct the CPU to process the data in the buffer, they are not the basis for the timing. The CPU must be able to process the half buffer fast enough before the DMA starts overwriting it, but any jitter in this execution is not an issue. Processing the buffer refers to performing peak detections as described in section 4.4.

The transmit buffer is set up similarly, but in reverse. The second DMA channel continually writes out the contents of the TX buffer to the GPIO control register, which is connected to the signal for the transducer driver. When a waveform is to be transmitted, it is written to the TX buffer at the appropriate index (which corresponds to time), while the DMA is writing out the other half of the buffer. Then the DMA will sweep across the waveform and set the GPIO pin voltage with precise timing. The waveform must then be removed from the buffer to avoid repeated transmission. Note that only digital (square wave) signals can be sent to the transducer driver since it is connected to a GPIO pin. Ideally this would be a DAC so that any analogue waveform could be transmitted, but unfortunately there is no DAC available on the microcontroller pin that the transducer driver is connected to. We are not able to modify the circuit board to change this.

The range measurement protocol (section 4.2) requires that a system timestamp is recorded for the time that each transponder response is received at the master. This timestamp is based on the index of the signal location in the master RX buffer (we assume that the exact signal reference location can be found as described in section 4.4). Thus the system timestamp is effectively incremented at the ADC sample rate. Of course the RX buffer is continually overwritten, so the index will not be unique. Each time the whole RX buffer is filled, a counter value, known as the timestamp prefix, is incremented by the CPU. Thus the full system timestamp is given by:

$$k_{RX} = p_{RX} \cdot l_{RX} + i_{RX}$$

Where k_{RX} is the receive system timestamp, i_{RX} is the index in the RX buffer of the received signal, l_{RX} is the length (in ADC samples) of the RX buffer, and p_{RX} is the current prefix.

Table 4.1 shows the parameters for the timers. Both timers are used in count up mode, with no internal clock division and auto-reload preload enabled. An update event is created each time the timer value reaches the ARR (Auto Reload Register) value (and the value then resets to 0). The frequency of the update events can be calculated with

$$\text{Update Event Frequency} = \text{Clock Input} / (\text{ARR} + 1)$$

The ADC is 12-bit and single ended, configured in synchronous clock mode with regular conversions and DMA continuous requests enabled.

Table 4.1: The timer parameters used to drive the receive and transmit functionality.

Timer	Clock Input	ARR	Update Event Freq.	Function
RX Timer	System Clock (72 MHz)	52-1	1.38 MHz	Trigger ADC Conversion
TX Timer	System Clock (72 MHz)	312-1	231 kHz	Trigger DMA 2 Request

The ARR value for the TX timer is chosen to divide the system clock frequency as close as possible into twice the transducer resonant frequency. Note it is twice the resonant frequency because we need to set the transducer driver high and low each carrier period. The ARR value for the RX timer is then chosen as a perfect integer divisor of the TX ARR value, so that the sample frequency is about

1 MHz. The sample frequency is necessary to ensure accurate discretization of the signal. The RX timer must divide the TX timer perfectly so that the timers remain synchronised to allow a common system timestamp for TX and RX.

Table 4.2 shows some of the protocol and microcontroller timing parameters for the implementation in this project expressed in the system timestamp units and SI units. Note that “System Units” refers to the unit of time between two consecutive system timestamps, and is the same as the ADC sampling period. Notice again how the TX timer is half the signal carrier period, and that in system units the TX timer period is an integer multiple of the ADC sample period. The RX and TX buffer have the same temporal length, and the measurement period and transponder delays are both multiples of the buffer length. These are design choices to simplify the implementation.

The transponder delay is chosen to allow multipath effects to die down before retransmitting at the transponder. Then the measurement period must be chosen so that it is greater than the TOF including the transponder delay for the operating distances, as mentioned in section 4.2.

Table 4.2: The specific values for the various time related parameters in the microcontroller and range measurement protocol implementation. Note that “System Units” refers to the unit of time between two consecutive system timestamps, and is the same as the sampling period.

Parameter	Period (System units)	Period	Frequency
ADC Sampling	1	725 ns	1.38 MHz
Signal Carrier	12	8.70 μ s	115 kHz
TX Timer	6	4.35 μ s	231 kHz
RX Buffer	18000	13.0 ms	76.7 Hz
TX Buffer	18000	13.0 ms	76.7 Hz
Repeat Measurements	11×18000	143 ms	6.97 Hz
Transponder Delay	2×18000	26.1 ms	38.3 Hz

The design overview so far has been oriented towards the master unit. The transponder unit operates with all the same principles, except that it must respond only when it receives a signal. It receives a signal when it detects a peak in its RX buffer. It then records the system timestamp of the event based on the index of the peak in the RX buffer. It calculates the system timestamp for the response by adding the programmed transponder delay to the received timestamp. It will then write the response waveform in the TX buffer, at the appropriate index dictated response timestamp. Note that in practice the DMA in the RX buffer and the DMA in the TX buffer cannot be perfectly synchronised at start-up, and will have some small constant offset for the duration of operation. This constant offset can be corrected through calibration. Note also that the transponder should not respond to echoes of its own response, so we use a “cool-down period” after responding, during which the transponder will not react to anything it receives.

4.4 Peak Detection

The range measurement protocol in section 4.2 refers to a “signal” which is transmitted between the master and transponder. This signal is treated as an instantaneous event, like a unit impulse. In practice this is not achievable; any acoustic signal will occupy some space in time. When stimulated at the resonant frequency, the transducer will oscillate with an increasing amplitude. To be able to have enough signal volume to be received across some distance, the transducer must be given time to reach that volume.

In this implementation, the signal that is sent between the master and transponder is called a “pulse”. At the microcontroller, this pulse has an envelope of a single square wave, and is modulated at the carrier frequency of 115 kHz. When passed through the transducer, the pulse degrades to the signal in Figure 4.5. The figure shows a real recorded pulse signal in Lake Burley Griffin.

Since the pulse is not instantaneous, we require a reference point on the pulse. The location in time of this reference point is the RX timestamp. Note that from the microcontroller’s perspective, the system timestamp is related to the index at which this reference point is located in the RX buffer (see section 4.3).

In Figure 4.5, the pulse clearly has an approximately symmetrical shape. The amplitude rises to some maximum then decreases. The implementation here exploits this by using the peak of the pulse as the reference point.

There is a requirement that the peak is located consistently, which means that the time between when the microcontroller initiates a pulse and when the peak of the pulse occurs is consistent. Any jitter will result in imprecision in the range measurements. Here we assume that the response of the transducer is deterministic, so any inconsistency in the location of the peak will occur when detecting it in the RX buffer of the microcontroller. The shape of the pulse will be distorted by noise while propagating between the devices.

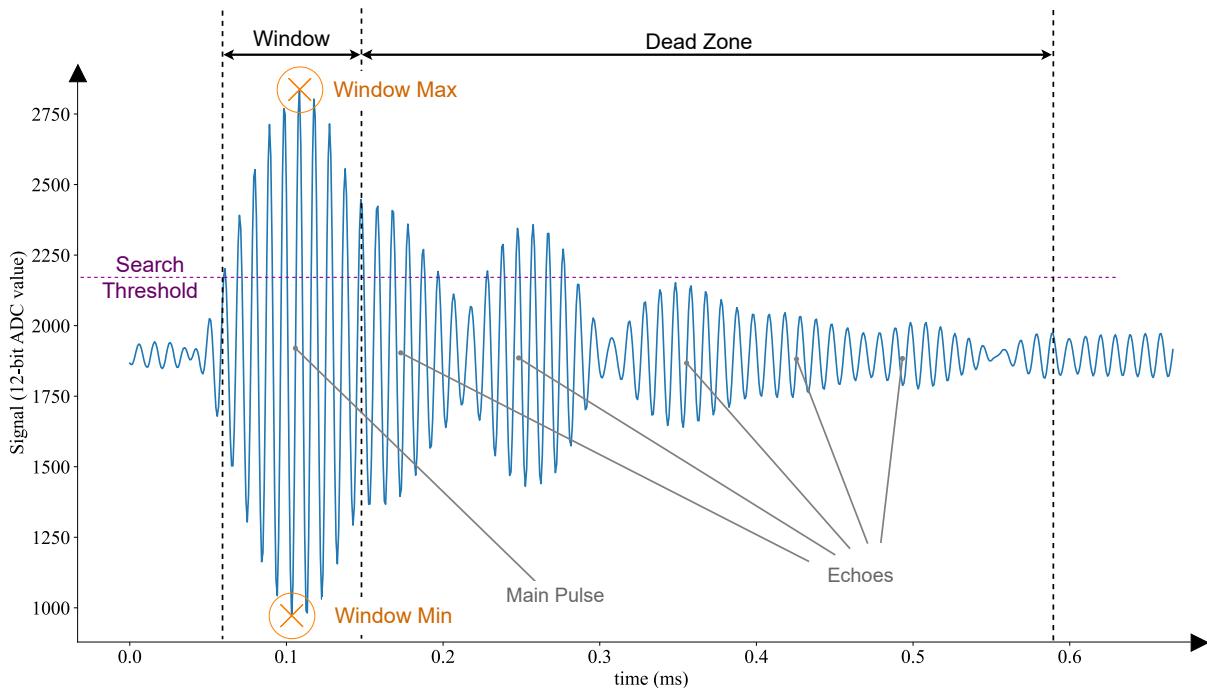


Figure 4.5: Peak detection at the Master and Transponder devices.

To detect a pulse peak, we first wait for some a sufficiently loud signal. Most of the time there will be silence, so we keep searching forwards in time until we find a signal value above some threshold, shown in Figure 4.5. The threshold should be above the noise floor, but low enough to be trigger by weaker pulses that are attenuated across longer ranges. Ideally this threshold would be dynamically adjusted to the conditions, however, in this implementation it is static.

The point at which the signal exceeds the threshold determines the start of the “pulse window”. The end of the window is some constant length of time later. We find the maximum and minimum signal values over the window. The peak is then estimated as the average in time of these maximum and minimum locations. The failure mode for this detection is when the noise distorts the maximum and minimum values enough that a neighbour carrier period becomes the maximum/minimum. Given the timing precision requirement of $\pm 13 \mu\text{s}$ and the carrier period of $8.7 \mu\text{s}$, a jump of the peak between two neighbouring carrier periods is tolerable. By taking the average in time of both the maximum and minimum values, we halve the variation if one of these values jumps between two neighbouring carrier periods.

Once the end of the window is reached, we ignore a large part of the immediately following signal. This area is termed the “dead-zone” and is labelled in Figure 4.5. This is required to avoid erroneously detecting any echoes of the main pulse that have been reflected off other parts of the environment. These echoes may well be louder than the search threshold. The length of the dead zone must be long enough to be confident that the echoes have reduced in volume to below the noise floor. Note that the length of this dead-zone is one factor that determines how frequently pulses can be sent between the devices.

Once we have passed the dead-zone, we resume searching for signal values above the threshold that belong to the next pulse.

Table 4.3 gives context on the real values of the parameters for the peak searching algorithm described above. The values are given in terms of the ADC sample rate and carrier period, as well as in SI units.

Table 4.3: The specific values for the pulse and peak search parameters in the microcontroller and range measurement protocol implementation. Note that “System Units” refers to the unit of time between two consecutive system timestamps, and is the same as the ADC sampling period.

Parameter	System units	Units of carrier period	World units
Carrier Period	12	1	$8.70 \mu\text{s}$
Pulse Width	96	8	$69.6 \mu\text{s}$
Window Width	192	16	$139 \mu\text{s}$
Dead-Zone	4200	350	3.04 ms

Note that Table 4.3 shows the pulse width as generated at the microcontroller GPIO and fed to the driver. Since the transducer continues oscillating after the input is removed, the actual pulse width is about twice as long. This is why the window width is twice the pulse width in the table.

4.5 Range Measurement Experiments and Performance

4.5.1 Fish-tank Experiments

The range measurement devices are tested in a lab environment inside of a re-purposed fish tank, depicted in Figure 4.6. The tank has 182 Litre capacity, and length, width and depth of 120, 37 and 47 cm. Two devices, the master and the transponder, are suspended in the tank, about 20cm below the water surface, separated by some distance (note only one device is depicted in the image). The devices are held in the vertical position to utilise the cylindrical power distribution discussed in section 4.1. Since the glass walls reflect the sound waves with very little attenuation, echoes in the fish tank are severe. To combat this effect, 30mm acoustic foam is added to all five internal surfaces of the fish tank. The foam is standard commercial acoustic foam, which is unlikely to be perfectly suited to the purpose, but performs well enough here.

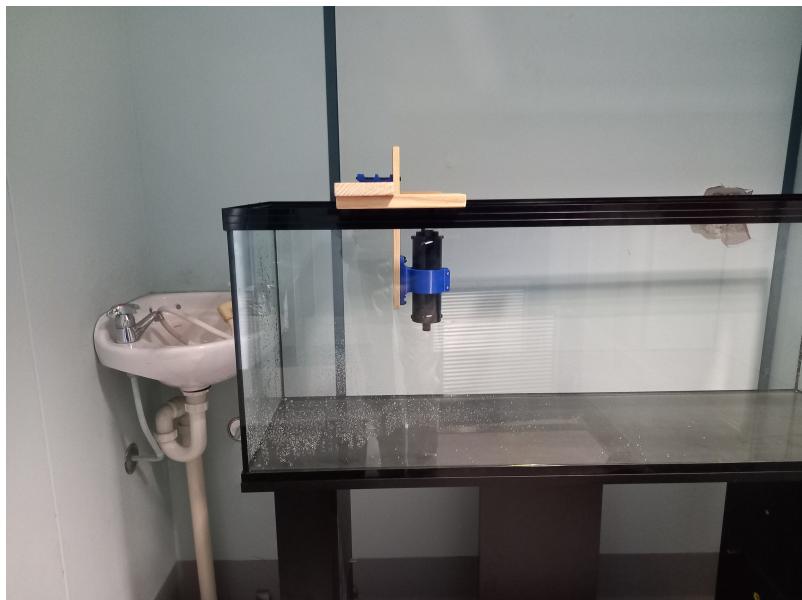


Figure 4.6: The fish tank used for lab testing. Two transducer devices are suspended in the fish tank (only one shown here). Range measurements are made using the Master/Transponder protocol, while varying the distances between the two devices. Not depicted here is the acoustic foam lining the fish tank to reduce echoes.

In the experiment, we continuously record range measurements from the master device, while changing the distance between the two devices between known values. The specific distance that is actually being measured is the distance between the centers of the two device diaphragms. The diaphragm of the device in Figure 4.6 is the small protrusion at the bottom of the cylindrical device.

The goal of the experiment is to establish that the precisions of the range measurements is within the target ± 1 cm. The second objective is to calibrate the Master device and show that accurate range measurements can be made. The calibration here involves estimating the constant range offset parameter (α in section 4.2). The scale parameter is not estimated, as we cannot obtain diverse enough range sample points within the confines of the tank. We instead assume the speed of sound in water is 1500 m/s and that the clock source of the microcontroller oscillates at exactly the nominal value of 72MHz.

The calibration procedure in this experiment is to take one range measurement with the devices at known separation (20cm). The offset distance is calculated as the difference between the true and

measured range. It is added to all subsequent range measurements, before they are stored. The results below were recorded about 15 minutes and several reboot cycles of both devices after the calibration was performed. No adjustments to the results have been made off-line.

Figure 4.7 shows the results from the tank experiment. The plot shows the range measurements as they are made by the master device over time. The devices are first placed 10cm apart, then moved to 20 and 30cm apart, pausing at each distance. Then the distance is set to 20cm and varied by 2.5 cm, to test the fidelity of the measurements.

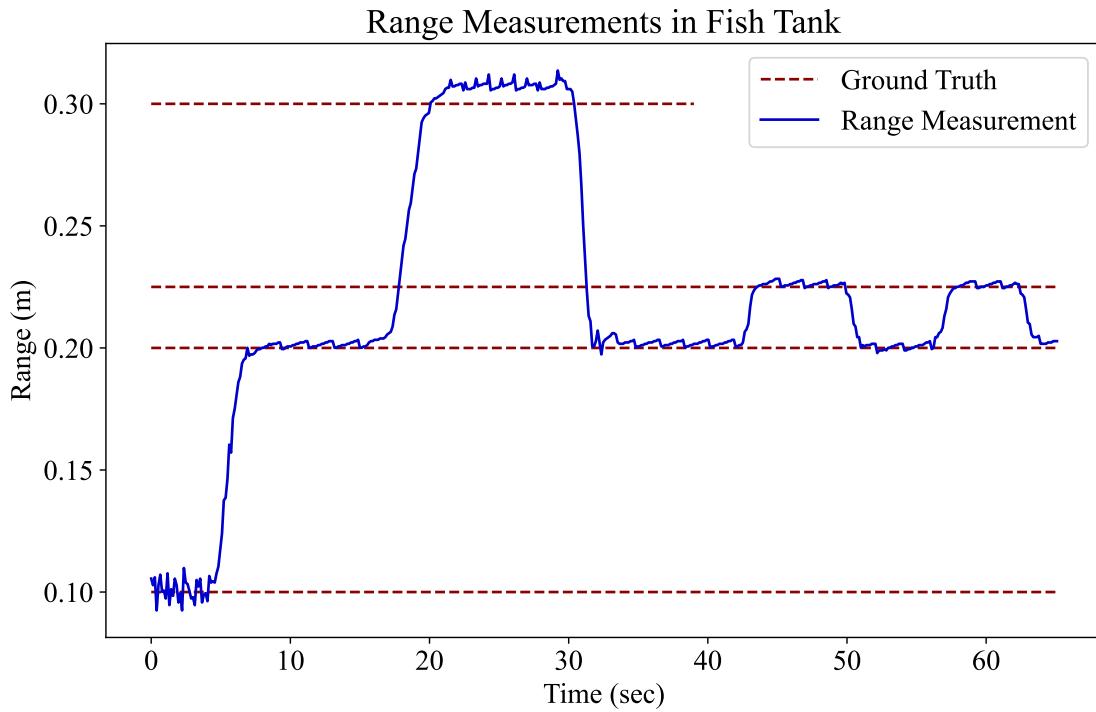


Figure 4.7: Range measurements taken under lab conditions. The Master and Transponder are placed 10cm apart, then moved to 20cm apart, then 30cm. Then they are placed 20cm apart and moved back and forth by 2.5cm. The range measurement frequency is 7 Hz.

Clearly, the system is able to take meaningful measurements throughout the experiment. The precision of the measurements remains within the target ± 1 cm, although only just at 10 cm distance. This precision means the small movement of only 2.5 cm is clearly detectable. The accuracy looks promising except at the 30 cm distance, where there is clearly some error in the measurement that is worse than everywhere else.

There are two factors that may have contributed to the range error at 30 cm. The first is the absence of scaling calibration, but given the accuracy at 10 and 20 cm this could not fully explain error. The second cause is the severe multipath in the small tank. The acoustic pulse does not exclusively take the shortest distance between the devices, but will take many different paths by reflecting off the walls of the fish tank and surface of the water. This means multiple pulses will be seen at the receiving end. Normally this is not a problem, as the peak detection algorithm in section 4.4 attempts to ignore these echoes. The failure observed here requires two degenerative properties of this multipath. The echo must be louder than the direct path, and occur close enough to the main pulse so that it distorts the signal inside the “window” described in section 4.4. Thus the maximum and minimum values detected will occur significantly after the true pulse peak, and result in an increase

in the range measurement as seen here. Early experiments recording pulses in the tank revealed that both these conditions occur often for particular placements of the devices. The placements of the two devices that result in this effect are unpredictable and chaotic. In principle, this failure mode would be prohibitive of the devices effectiveness, however, the multipath in the tank is worse than in a more open water environment where the devices are intended to be used. The device did not obviously fail in this manner in the later tests discussed in section 4.5.2.

The range measurements produce a saw-tooth wave at 20 and 25 cm. There are two combined factors that cause this - firstly, the clocks sources in the master and transponder oscillate at slightly different frequencies, and secondly, the TX buffer in the transponder has a temporal resolution of half the carrier period, not the sample frequency (for implementation simplification reasons). We are seeing the range measurement increase as the temporal rounding error from the RX to TX buffer increases, then sharply drop as the transponder then rounds to the next index in the TX buffer. We can infer the difference in the master and transponder clocks from the frequency of the saw-tooth wave. They appear to diverge by about 5 μ s per second.

The saw-tooth does not occur at 10 cm because signal noise is affecting the peak detection more at that distance. How signal noise affects the range measurements is discussed in section 4.5.3. It should be acknowledged that the fact that this rounding effect is visible is a testament to the precision of the microcontroller timing when not disturbed by noise or multipath effects.

4.5.2 Experiments in Lake Burley Griffin



Figure 4.8: Experimental set-up in Lake Burley Griffin. The Master is secured to the jetty, attached to a fence paling and submerged at about 1 meter depth. The Transponder is similarly submerged, but is left free to be moved by hand, along the paths indicated by the arrows.

To test the device at longer ranges in a more realistic environment, range measurements were taken in Lake Burley Griffin. The master device was secured to a jetty as shown in Figure 4.8, submerged in the water by about one meter. The transponder is similarly submerged, but is left free to be moved by hand, along the paths indicated by the arrows. Note that the intended use is to have the master

unit on a moving AUV, and the transponder fixed to a landmark. Here doing the reverse is more convenient, and of course there is no difference from the devices' perspective.

Range measurements are continually recorded from the master. The same calibration offset parameter is used as in the tank experiment, with no scale calibration applied.

Figure 4.9 shows three sets of range measurements for three different movements of the transponder. The first is incrementally moving the transponder in half meter steps along the jetty. The range measurements show the movement, and even the inaccuracies of the physical movements of the transponder. Note the last step is significantly larger than the others because the last step was to the end of the jetty, not half a meter.

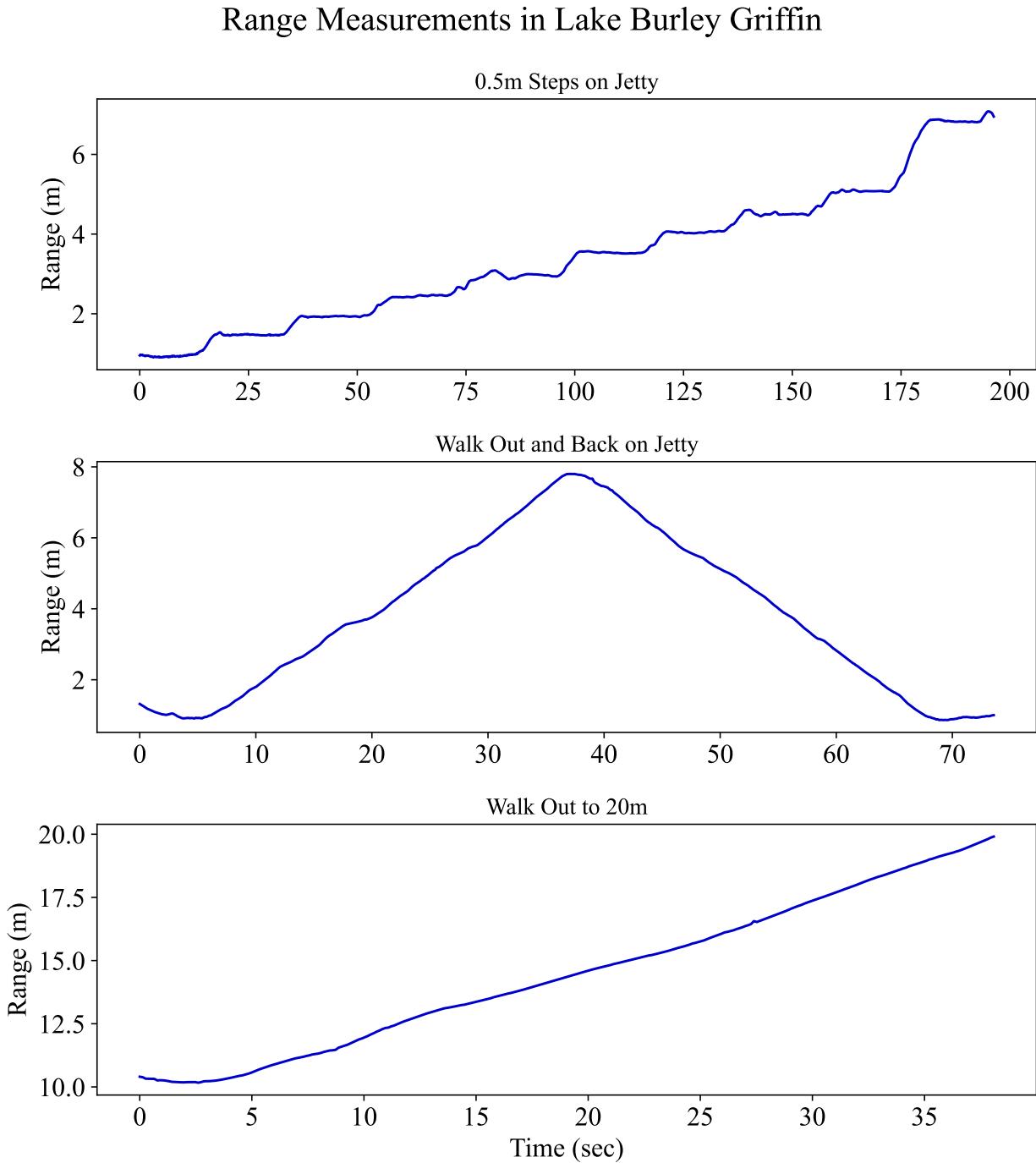


Figure 4.9: Range measurements taken off a jetty in Lake Burley Griffin. The range measurement frequency is 7 Hz.

The second movement is walking the transponder along the jetty away from the master, and then back towards the master. As expected, the range measurements increase at a constant rate and then decrease again.

The third movement is to establish the maximum range this implementation can operate at. The transponder is walked out along the peninsula (the single ended arrow in Figure 4.8) until the master no longer receives responses from the transponder. We were able to make range measurements up to 20 m before losing signal.

Obtaining a ground truth for this experiment was not attempted, as measuring distances underwater by hand is not easy. In fact that is the very problem that this technology is attempting to overcome. The results here are thus mostly anecdotal and the accuracy of the measurements cannot be confirmed. There is probably significant scaling error since the speed of sound in water has not been calibrated. The results do show, however, that the devices appear to be working correctly in a more realistic environment than the tank, and they are not obviously failing in any way. The motion implied by the range measurements is feasible - there are no absurd velocities or sudden changes in range.

4.5.3 Noise Characterization

To estimate the precision of the range measurements in the lake environment, we use the same jetty as in the previous section. Both the transponder and the master are fixed to the jetty at a depth of about 1 m and separated by about 5.8 m. We then record the range measurements for about a minute. Figure 4.10 shows the results.

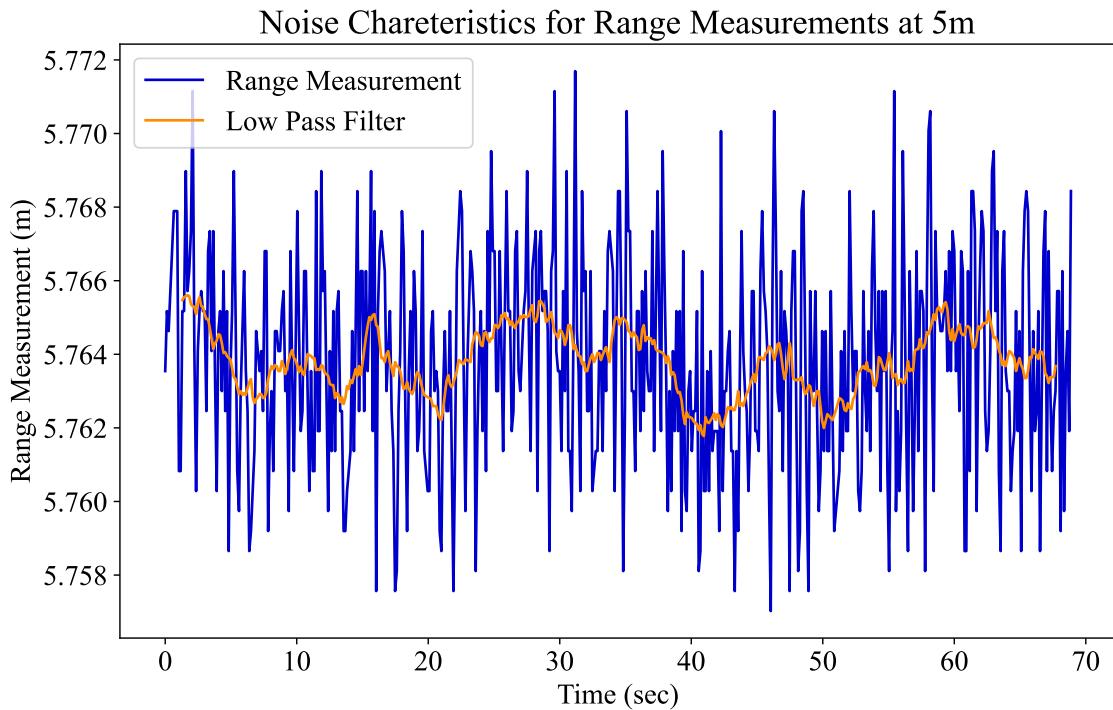


Figure 4.10: Range measurements taken off a jetty in Lake Burley Griffin. The master and Transpon-
der are fixed at about 5.8m apart, at a depth of about 1m. The range measurement frequency is 7 Hz.

In the ideal case we would expect the range measurement to remain constant, as neither device is moving. We observe a deviation in the range within ± 1 cm, which is within the precision target.

The orange line in Figure 4.10 shows the result of applying an aggressive averaging filter to the measurements. It seems that the average range measurement wanders over time. This is most likely due to that fact that the master and transponder cannot be held perfectly still, and will oscillate somewhat with the movement of the water.

By inspecting the Figure 4.10 closely, it is clear that the noise in the measurements is not Gaussian. At about 14 seconds, the range seems to be the same value several times in a row. In fact all over the plot there are places where the range measurement is fluctuating between fixed levels. This is due to the way in which imprecision enters into the pulse peak detection algorithm from section 4.4. When noise is added to the acoustic signal, the peak and trough of each period of carrier is increased or decreased slightly. This may result in the maximum or minimum of the pulse occurring at a different time within the pulse, but they will always be aligned with one of the carrier periods. Thus the timing reference for the peak always changes in integer multiples half carrier period steps. These translate to the distinct levels in range measurements. The wandering of the signal average makes the levels more difficult to see across the whole plot.

4.6 Implementation Challenges

The prototype presented here has two key deficiencies that make it unsuitable for use in landmark range-only SLAM. Firstly, the system can only make range measurements between two devices - i.e. the robot and one landmark. Range-only SLAM requires that the robot makes range measurements to many landmarks. The second deficiency is that the range is limited because the receiver gains on both devices are fixed, and are not increased as the range increases.

Table 4.4: The requirements for any landmark range measurement system that uses the time-of-flight of short signals passed between a master and transponder unit.

Requirement	Description
Timing Precision	The master device must be able to measure the TOF within some acceptable precision to provide sufficiently precise range measurements. The transponders must be able to respond to the master's signals at a constant delay with jitter less than the required precision.
Multipath Robustness	The system must be immune to loud echoes of any part of the signal from the operating environment. The delay until the first echo and the attenuation rate of the echoes will depend on the operating environment.
Landmark Identification	The master device must be able to determine which response it receives belongs to which transponder.
Collision Avoidance	The master device must be robust to multiple responses from multiple transponders arriving at the master simultaneously and interfering with each other. Alternatively, the system may prevent multiple responses from multiple transponders arriving at the master simultaneously.
Dynamic Gain Control	The master device must maintain an internal model of the distance to each transponder, and use this model to set its receiving gains before receiving a response from each transponder (to avoid loss of signal or saturation). Each transponder must maintain an internal model of the distance to the master, and set its receiver gains accordingly.

Table 4.4 outlines the key non-trivial requirements for a range measurement system involving time-of-flight calculations between one robot and multiple landmarks. The requirements apply for cases where the master transmits short signals to the transponder which then replies. The system presented here has fulfilled only the first two listed requirements. Note that many of these requirements are the same as for any wireless communication protocol.

We will now offer some suggestions for how to extend the present prototype system to fulfil all the requirements above. Landmark Identification and Collision avoidance can both be solved by augmenting the request signal emitted by the master. Instead of one pulse, the signal would consist of several pulses close together, which encode a number representing a landmark ID. Each transponder is assigned a unique ID in advance, and will only respond to request signals with their ID. Their response will also contain their own ID. Thus the master can associate responses to landmarks, and also coordinate the landmark responses so that there are no collisions. There are many options for encoding an ID with several pulses, but one idea is to take advantage of the precise timing developed here. By slightly varying the time between each consecutive pulses in the signal, we can encode more than one bit per pulse. This is advantageous as we cannot place consecutive pulses in a signal arbitrarily close together due to multipath effects, and hence a request signal may become undesirably long for large numbers of IDs.

Chapter 5

Conclusions and Future Work

The work in this project can be divided into two main parts. The first part presents the derivation of an EqF for range-only SLAM. The filter has been verified on synthetic data, and worked successfully to localize itself and map four landmarks. The second contribution of this work is the development of a prototype landmark range measurement system. The protocol and software design has been presented, and experiments show it has a precision of around $\pm 1\text{cm}$, and that it can be calibrated to achieve accuracy. While the system has limitations, it has been demonstrated that it is possible to perform range measurements with affordable hardware.

The motivation for this work was to advance the efforts in making range-only SLAM possible for AUVs. In light of this, the next steps would be to attach an IMU to the master acoustic unit, have it move in some trajectory in a large body of water, and collect a dataset of IMU measurements and range measurements to multiple beacons. Then the filter could be tested and verified on a real dataset. As outlined in section 4.6, some extensions would have to be made to the range measurement system before it could be used for multiple landmarks.

More broadly in the domain of underwater robotics, there is work to be done in developing the technology and knowledge that is currently tied up in commercial entities. This is not just the case for sonar range measurement systems, but for a variety of other underwater robotics sensors and equipment. It is critical that these devices, and their principles of operation, are available and affordable to researchers; not only so they can build technology on top of these systems, but also so the systems themselves can be improved. I hope I have made progress in this direction by presenting the design of a range measurement system here so that others can use it and improve it.

References

Bibliography

- [1] T. Bailey and H. Durrant-Whyte, “Simultaneous localization and mapping (slam): Part ii,” *IEEE robotics & automation magazine*, vol. 13, no. 3, pp. 108–117, 2006.
- [2] F. Guth, L. Silveira, S. Botelho, P. Drews, and P. Ballester, “Underwater slam: Challenges, state of the art, algorithms and a new biologically-inspired approach,” in *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2014, pp. 981–986.
- [3] Q. Luo, X. Yan, C. Ju, Y. Chen, and Z. Luo, “An ultra-short baseline underwater positioning system with kalman filtering,” *Sensors*, vol. 21, no. 1, p. 143, 2020.
- [4] L. Paull, S. Saeedi, M. Seto, and H. Li, “Auv navigation and localization: A review,” *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131–149, 2014.
- [5] P. van Goor, T. Hamel, and R. Mahony, “Equivariant filter (eqf),” *IEEE Transactions on Automatic Control*, vol. 68, no. 6, pp. 3501–3512, 2023.
- [6] P. van Goor and R. Mahony, “Eqvio: An equivariant filter for visual-inertial odometry,” *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3567–3585, 2023.
- [7] A. Fornasier, Y. Ng, R. Mahony, and S. Weiss, “Equivariant filter design for inertial navigation systems with input measurement biases,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 4333–4339.
- [8] G. B. Zaffari, M. M. Dos Santos, P. L. Drews, and S. S. Botelho, “Effects of water currents in a continuous attractor neural network for slam applications,” in *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)*. IEEE, 2016, pp. 328–333.
- [9] M. Massot-Campos, G. Oliver, A. Bodenmann, and B. Thornton, “Submap bathymetric slam using structured light in underwater environments,” in *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*. IEEE, 2016, pp. 181–188.
- [10] D. Bandara, Z. Leong, H. Nguyen, S. Jayasinghe, and A. L. Forrest, “Technologies for under-ice auv navigation,” in *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*. IEEE, 2016, pp. 108–114.
- [11] I. Mahon, S. B. Williams, O. Pizarro, and M. Johnson-Roberson, “Efficient view-based slam using visual loop closures,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1002–1014, 2008.
- [12] S. Augenstein and S. M. Rock, “Improved frame-to-frame pose tracking during vision-only slam/sfm with a tumbling target,” in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3131–3138.
- [13] F. Ferreira, G. Veruggio, M. Caccia, and G. Bruzzone, “Real-time optical slam-based mosaicking for unmanned underwater vehicles,” *Intelligent Service Robotics*, vol. 5, no. 1, pp. 55–71, 2012.
- [14] D. Ribas, P. Ridao, J. D. Tardós, and J. Neira, “Underwater slam in man-made structured environments,” *Journal of Field Robotics*, vol. 25, no. 11-12, pp. 898–921, 2008.

- [15] B. Alves, A. Pascoal, and T. Hormigo, “A navigation algorithm for under-the-ice robotic operations,” in *2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)*. IEEE, 2020, pp. 1–6.
- [16] A. Maldonado-Ramírez and L. A. Torres-Mendez, “Learning ad-hoc compact representations from salient landmarks for visual place recognition in underwater environments,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5739–5745.
- [17] A. Mallios, P. Ridao, D. Ribas, F. Maurelli, and Y. Petillot, “Ekf-slam for auv navigation under probabilistic sonar scan-matching,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 4404–4411.
- [18] F. Menna, E. Nocerino, M. M. Nawaf, J. Seinturier, A. Torresani, P. Drap, F. Remondino, and B. Chemisky, “Towards real-time underwater photogrammetry for subsea metrology applications,” in *OCEANS 2019-Marseille*. IEEE, 2019, pp. 1–10.
- [19] M. M. Dos Santos, P. Ballester, G. B. Zaffari, P. Drews, and S. Botelho, “A topological descriptor of acoustic images for navigation and mapping,” in *2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)*. IEEE, 2015, pp. 289–294.
- [20] M. Liu, J. Zhu, X. Pan, G. Wang, J. Liu, Z. Peng, and J.-H. Cui, “A distributed intelligent buoy system for tracking underwater vehicles,” *Journal of Marine Science and Engineering*, vol. 11, no. 9, 2023. [Online]. Available: <https://www.mdpi.com/2077-1312/11/9/1661>
- [21] P. Newman and J. Leonard, “Pure range-only sub-sea slam,” in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 2. Ieee, 2003, pp. 1921–1926.
- [22] E. Olson, J. J. Leonard, and S. Teller, “Robust range-only beacon localization,” *IEEE Journal of Oceanic Engineering*, vol. 31, no. 4, pp. 949–958, 2006.
- [23] N. R. Rypkema, E. M. Fischel, and H. Schmidt, “Closed-loop single-beacon passive acoustic navigation for low-cost autonomous underwater vehicles,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 641–648.
- [24] M. Real, P. Vial, N. Palomeras, and M. Carreras, “Underwater acoustic localization using pose-graph slam,” in *OCEANS 2023 - Limerick*, 2023, pp. 1–6.
- [25] M. F. Fallon, M. Kaess, H. Johannsson, and J. J. Leonard, “Efficient auv navigation fusing acoustic ranging and side-scan sonar,” in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 2398–2405.
- [26] J. Choi, Y. Lee, T. Kim, J. Jung, and H.-T. Choi, “Ekf slam using acoustic sources for autonomous underwater vehicle equipped with two hydrophones,” in *OCEANS 2016 MTS/IEEE Monterey*, 2016, pp. 1–4.
- [27] J. Choi, J. Park, J. Jung, and H.-T. Choi, “Gaussian sum filter based slam for autonomous navigation of underwater vehicles using acoustic sources,” in *OCEANS 2019 - Marseille*, 2019, pp. 1–4.
- [28] B. Garau, A. Alvarez, and G. Oliver, “Auv navigation through turbulent ocean environments supported by onboard h-adcp,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 3556–3561.
- [29] J. J. Leonard and A. Bahr, “Autonomous underwater vehicle navigation,” *Springer handbook of ocean engineering*, pp. 341–358, 2016.
- [30] G. T. Donovan, “Position error correction for an autonomous underwater vehicle inertial navigation system (ins) using a particle filter,” *IEEE Journal of Oceanic Engineering*, vol. 37, no. 3, pp. 431–445, 2012.

- [31] R. M. Eustice, "Large-area visually augmented navigation for autonomous underwater vehicles," Ph.D. dissertation, Massachusetts Institute of Technology, 2005.
- [32] R. M. Eustice, O. Pizarro, and H. Singh, "Visually augmented navigation for autonomous underwater vehicles," *IEEE Journal of oceanic Engineering*, vol. 33, no. 2, pp. 103–122, 2008.
- [33] A. Elibol, N. Gracias, and R. Garcia, "Augmented state–extended kalman filter combined framework for topology estimation in large-area underwater mapping," *Journal of Field Robotics*, vol. 27, no. 5, pp. 656–674, 2010.
- [34] J. Salvi, Y. Petillot, and E. Batlle, "Visual slam for 3d large-scale seabed acquisition employing underwater vehicles," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 1011–1016.
- [35] J. Salvi, Y. Petillo, S. Thomas, and J. Aulinás, "Visual slam for underwater vehicles using video velocity log and natural landmarks," in *OCEANS 2008*. IEEE, 2008, pp. 1–6.
- [36] D. Ribas, P. Ridao, J. Neira, and J. D. Tardos, "Slam using an imaging sonar for partially structured underwater environments," in *2006 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2006, pp. 5040–5045.
- [37] D. Ribas, P. Ridao, J. D. Tardós, and J. Neira, "Underwater slam in a marina environment," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 1455–1460.
- [38] G. Vallicrosa, P. Ridao, D. Ribas, and A. Palomer, "Active range-only beacon localization for auv homing," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 2286–2291.
- [39] R. M. Eustice, H. Singh, J. J. Leonard, and M. R. Walter, "Visually mapping the rms titanic: Conservative covariance estimates for slam information filters," *The international journal of robotics research*, vol. 25, no. 12, pp. 1223–1242, 2006.
- [40] R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly sparse delayed-state filters for view-based slam," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1100–1114, 2006.
- [41] M. Walter, F. Hover, and J. Leonard, "Slam for ship hull inspection using exactly sparse extended information filters," in *2008 IEEE international conference on robotics and automation*. IEEE, 2008, pp. 1463–1470.
- [42] S. Barkby, S. B. Williams, O. Pizarro, and M. V. Jakuba, "A featureless approach to efficient bathymetric slam using distributed particle mapping," *Journal of Field Robotics*, vol. 28, no. 1, pp. 19–39, 2011.
- [43] A. Concha, P. Drews-Jr, M. Campos, and J. Civera, "Real-time localization and dense mapping in underwater environments from a monocular sequence," in *OCEANS 2015-Genova*. IEEE, 2015, pp. 1–5.
- [44] S. Barkby, S. B. Williams, O. Pizarro, and M. V. Jakuba, "Bathymetric particle filter slam using trajectory maps," *The International journal of robotics research*, vol. 31, no. 12, pp. 1409–1430, 2012.
- [45] S. Barkby, S. Williams, O. Pizarro, and M. Jakuba, "Incorporating prior maps with bathymetric distributed particle slam for improved auv navigation and mapping," in *OCEANS 2009*. IEEE, 2009, pp. 1–7.
- [46] ——, "An efficient approach to bathymetric slam," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 219–224.

- [47] N. Fairfield and D. Wettergreen, “Active localization on the ocean floor with multibeam sonar,” in *OCEANS 2008*. IEEE, 2008, pp. 1–10.
- [48] D. Ribas, P. Ridao, A. Mallios, and N. Palomeras, “Delayed state information filter for usbl-aided auv navigation,” in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 4898–4903.
- [49] S. E. Webster, R. M. Eustice, H. Singh, and L. L. Whitcomb, “Advances in single-beacon one-way-travel-time acoustic navigation for underwater vehicles,” *The International Journal of Robotics Research*, vol. 31, no. 8, pp. 935–950, 2012.
- [50] M. VanMiddlesworth, M. Kaess, F. Hover, and J. J. Leonard, “Mapping 3d underwater environments with smoothed submaps,” in *Field and Service Robotics: Results of the 9th International Conference*. Springer, 2015, pp. 17–30.
- [51] J. Edwards, H. Schmidt, and K. LePage, “Bistatic synthetic aperture target detection and imaging with an auv,” *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 690–699, 2001.
- [52] R. Mahony, T. Hamel, and J. Trumpf, “Equivariant systems theory and observer design,” *arXiv preprint arXiv:2006.08276*, 2020.
- [53] P. v. Goor, R. Mahony, T. Hamel, and J. Trumpf, “A geometric observer design for visual localisation and mapping,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 2543–2549.
- [54] W. Yao, “Equivariant filter for range-only slam,” *College of Engineering and Computer Science, The Australian National University*, 2021.