2021-12

AN INTEGRATED SIMULATION APPROACH FOR AUV IMAGE-BASED SLAM NAVIGATION

Figueroa, Jesus

Monterey, CA; Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

AN INTEGRATED SIMULATION APPROACH FOR AUV IMAGE-BASED SLAM NAVIGATION

by

Jesus Figueroa

December 2021

Thesis Advisor:	Douglas P. Horner
Second Reader:	Brian S. Bingham

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2021	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE AN INTEGRATED SIMULATION APPROACH FOR AUV IMAGE-BASED SLAM NAVIGATION	**5. FUNDING NUMBERS**
6. AUTHOR(S) Jesus Figueroa	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

This thesis develops a simulation framework for undersea feature-based navigation. Using an autonomous underwater vehicle (AUV) to locate an item of interest on the seafloor is a capability that would greatly benefit the Navy. AUVs provide a gateway toward removing the workforce requirement; however, they are still costly both in acquisition and maintenance. A solution to this problem is using two AUVs, one with increased capability and charged with finding and marking seafloor items with a beacon. An expendable AUV outfitted with cost-effective sensors would relocate, identify and neutralize the threat. Using undersea imaging to correlate seafloor images to an a priori image mosaic together with a ultra short baseline (USBL) beacon allows the AUV to complete challenging mission objectives without traditional navigation systems. Incremental Smoothing and Mapping 2 (iSAM2) is a Simultaneous Localization and Mapping (SLAM) technique that can be used by the AUV for position localization and is an appropriate technique, with image and USBL sensing, for real-time navigation operations. A simulation framework provides the ability to evaluate an AUV's performance while minimizing the risk of real-world operations. The framework is composed of a software architecture that allows for testing using the same software applied in real-world operations. This thesis demonstrates this framework and provides analysis for its usability for image-based SLAM.

14. SUBJECT TERMS SLAM, iSAM2, USBL, AUV	**15. NUMBER OF PAGES** 99
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

AN INTEGRATED SIMULATION APPROACH FOR AUV IMAGE-BASED SLAM NAVIGATION

Jesus Figueroa
Lieutenant, United States Navy
BS, United States Naval Academy, 2014

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2021

Approved by: Douglas P. Horner
Advisor

Brian S. Bingham
Second Reader

Garth V. Hobson
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis develops a simulation framework for undersea feature-based navigation. Using an autonomous underwater vehicle (AUV) to locate an item of interest on the seafloor is a capability that would greatly benefit the Navy. AUVs provide a gateway toward removing the workforce requirement; however, they are still costly both in acquisition and maintenance. A solution to this problem is using two AUVs, one with increased capability and charged with finding and marking seafloor items with a beacon. An expendable AUV outfitted with cost-effective sensors would relocate, identify and neutralize the threat. Using undersea imaging to correlate seafloor images to an a priori image mosaic together with a ultra short baseline (USBL) beacon allows the AUV to complete challenging mission objectives without traditional navigation systems. Incremental Smoothing and Mapping 2 (iSAM2) is a Simultaneous Localization and Mapping (SLAM) technique that can be used by the AUV for position localization and is an appropriate technique, with image and USBL sensing, for real-time navigation operations. A simulation framework provides the ability to evaluate an AUV's performance while minimizing the risk of real-world operations. The framework is composed of a software architecture that allows for testing using the same software applied in real-world operations. This thesis demonstrates this framework and provides analysis for its usability for image-based SLAM.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AVA	advanced autonomous architecture
AUV	autonomous underwater vehicle
CAVR	Center for Autonomous Vehicle Research
COB	center of buoyancy
CIC	close in control
CTT	cross tunnel thruster
DOF	degree of freedom
DoD	Department of Defense
DAE	digital asset exchange
D-USBL	digital ultra-short baseline
EOD	Explosive Ordnance Disposal
EKF	Extended Kalman Filter
GTSAM	Georgia Tech Smoothing and Mapping
GPS	global positioning system
H/SIL	hardware/software-in-the-loop
HCTT	horizontal cross tunnel thrusters
iSAM	Incremental Smoothing and Mapping
iSAM2	Incremental Smoothing and Mapping 2
i.i.d	independent identically distributed

INS internal navigation dystem

LUX lumens per square meter

MATLAB Matrix Laboratory

MASINT measurement and signature intelligence

MCM Mine Countermeasure

NPS Naval Postgraduate School

NED north east down

ODE Open Dynamics Engine

OSRF Open Source Robotics Foundation

PID Proportional Integral Derivative

PNG portable network graphics

PUC position of uncertainty

RIN re-acquire identify neutralize

REMUS Remote Environmental Monitoring UnitS

ROV remotely operated vehicle

ROS Robot Operating System

RPM rotations per minute

SCM search classify map

SCM-RIN search classify map re-acquire identify neutralize

SRWBR short range wide band radio

SLAM Simultaneous Localization and Mapping

SAM Smoothing and Filtering

TLAM Tomahawk Land Attack Missiles

TCP transmission control protocol

USBL ultra-short baseline

USG United States government

UUV-SIM Unmanned Undersea Vehicle Simulator

UDP user datagram protocol

UXO un-exploded ordnance

USN U.S. Navy

VCTT vertical cross tunnel thrusters

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

This thesis could not have been completed without the expert guidance and dedication of my thesis advisor Dr. Douglas Horner. Thank you for your focused attention, advice, and assistance throughout my research.

To my wife Audry, thank you for your endless support and encouragement in completing this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

1.1 Introduction

After Operation Desert Storm, the Naval Sea Systems Command Supervisor of Salvage was tasked with the recovery efforts of several failed Tomahawk Land Attack Missiles (TLAM) fired from the USS Mobile Bay (CG 53), USS Paul F. Foster (DD 964), and USS Fife (DD 991) in the Arabian Gulf. With an average depth of 200-225 feet, the operating area proved impractical for Navy Diver searches. As a result, the salvage team used a remotely operated vehicle (ROV) from 23 March to 7 April of 1991 to develop a debris field map and eventually recover the failed missiles by divers [1]. A autonomous underwater vehicle (AUV), capable of navigating near the sea bottom whilst surveying and creating a map of its findings, would have been a critical asset in executing this salvage operation in a timely and cost-efficient manner.

1.2 Motivation

The underwater domain adds complexity for unmanned naval systems, particularly with respect to navigation and communication. The requirement for a AUV to repeatedly return to the ocean surface to localize position is sub-optimal from both the tactical and energy efficiency standpoints.

The sheer size of the undersea domain, from offshore littoral to deep-sea submergence, requires a AUV presence. To operate potentially independently requires a level of autonomy not currently available. Position estimation without external beacon systems is a key part of that autonomy for these complex, long-duration missions.

A common objective for the AUVs is the ability to detect and locate an undersea object. Locating a submerged object of interest without GPS has many applications in education, the private sector, and defense. The reliable location of an underwater feature or object would help the scientific community in the areas of underwater navigation and deep-sea

exploration. Private sector applications of this ability include surveying, dredging, and salvage operations. The variety of applications of AUVs in all sectors incentivize the development of reliable AUV navigation.

The U.S. Navy, Explosive Ordnance Disposal (EOD) and Mine Countermeasure (MCM) missions would greatly benefit from utilizing autonomous vehicles to conduct operations. It would reduce risk exposure to sailors and marines. Current methods for mine detection, neutralization, and removal of any other type of sea ordnance involve deploying highly trained technicians or using survey ships to perform route clearance for Navy vessels. The manned nature of these current techniques places a substantial risk on the sailors performing the operations. Additionally, route clearing exposes expensive equipment, primarily the MCM naval vessels and their crews. Even prior to the executions of operations, ordnance disposal operations consume large DOD expenses in the training of sailors and maintenance of equipment. AUVs provide a solution to many MCM and EOD missions that result in reduced cost and risk.

1.3 Problem Statement

Current efforts to integrate AUVs in underwater mine countermeasure operations use a technique known as search classify map re-acquire identify neutralize (SCM-RIN). Figure 1.1 shows an example of EOD-MCM Operations.

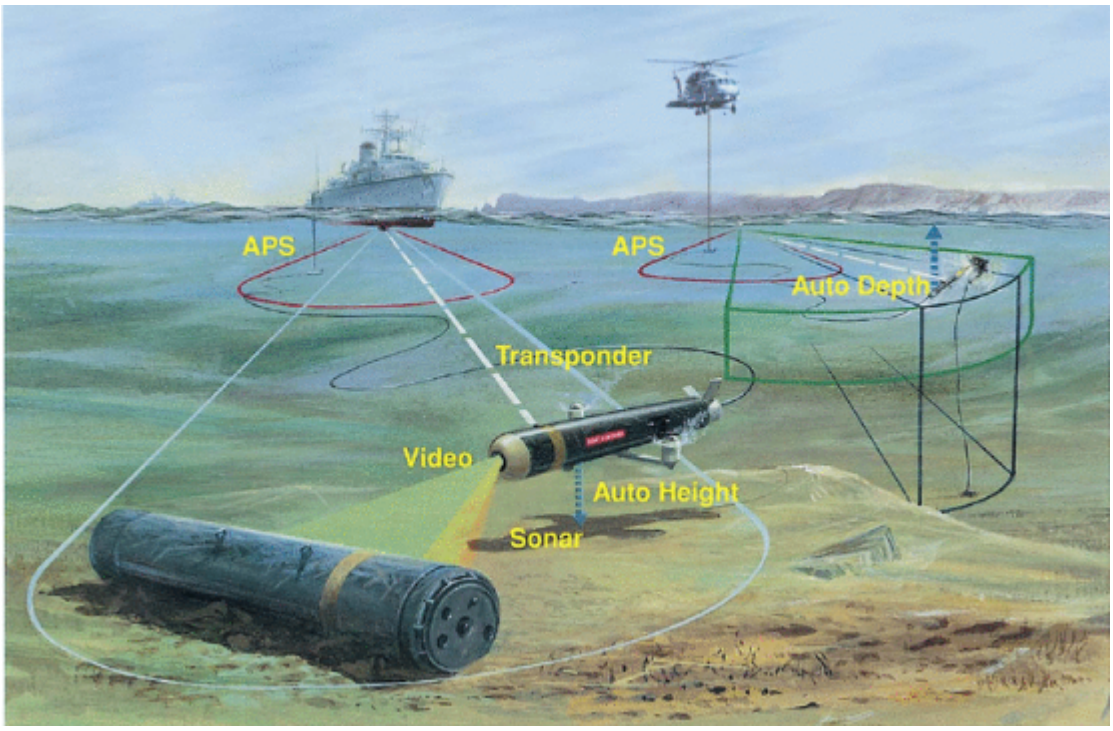


Figure 1.1. SCM-RIN Operations Using an ROV. Source: [2].

As depicted, the search classify map (SCM) portion of the mission is conducted by a tethered ROV which collects the seafloor information. This data is post-processed and used as a mapping tool for the re-acquire identify neutralize (RIN) portion of the mission by EOD.

An autonomous alternative to these costly procedures is to replicate the SCM-RIN mission with two unmanned, un-tethered AUVs. A "smart" AUV would conduct a survey of the seafloor with an onboard camera, identifying potential threats or objects of interest and dropping an ultra-short baseline (USBL) transponder in the vicinity of these objects. This allows for the majority of expenses, development, training, and maintenance to be concentrated on the surveying vehicle. The collected data would then be transferred to a low-cost AUV with the specific purpose of re-acquiring the objects of interest, identifying threats, and neutralizing them. The transfer of information between the two vehicles gives the second vehicle the advantage of gaining access to information in the form of an a priori image mosiac and consistent range and bearing measurements from a known location via the USBL transponder. The AUV in question must be able to conduct accurate position estimation with the inexpensive onboard sensors to execute the second half of the mission.

It must use the range and bearing from the USBL to navigate the target's vicinity and utilize a camera together with the image mosaic for highly accurate in-close navigation to the target.

1.4 Approach

This thesis focuses on a simulation environment to exercise the autonomy onboard the AUV for the RIN portion of the mission. First, I introduce the navigation technique implemented onboard the Remote Environmental Monitoring UnitS (REMUS) vehicle for precise navigation. I then describe the simulation framework developed to demonstrate the solution to the RIN mission using an AUV.

The AUV navigation methodology described includes advanced techniques in sensor processing, navigation, and control. The AUV will combine the two sensor measurements of the USBL and camera for position estimation with a technique called Simultaneous Localization and Mapping (SLAM). The SLAM technique used is called Incremental Smoothing and Mapping 2 (iSAM2). It is a non-linear, least-squared, smoothing approach that permits accurate estimation for the vehicle and designated features. As the vehicle approaches the target area, it needs to reduce speed for matching the camera images with the prior image mosaic. Use of cross tunnel thruster (CTT) control was used for slow-speed maneuvers.

The framework for implementing and analyzing AUV missions in simulation is described and used to demonstrate the RIN scenario. The simulation environment consists of a hardware and software in the loop framework. The simulated underwater environment mathematically represents the hydrodynamic and hydro-static forces using a Fossen model [3]. It is coupled with a model of REMUS that contains hydrodynamic coefficients developed using the same model [4]. Finally, I demonstrate the autonomous solution to the RIN mission by implementing the image-based navigation within the simulation framework.

1.5 SCM-RIN Related Work

Research in this field is largely divided into either the SCM or RIN portions of the mission profile. Reference [5] explores the fusion of magnetic, acoustic, and electro-optic sensors

to successfully confirm the identification of an undersea threat. Reference [6] develops an optimal navigation and reload strategy for a AUV capable of dropping munitions to a set of known potential threats, using a multi-dimensional Markov Process. Reference [6] also addresses the dynamic undersea environment, proposing an advanced autonomous architecture (AVA) that is modular and expandable by using a layered approach to mission management, tasking, and real-time behavior. Reference [7] uses image improvement techniques to existing marker recognition algorithms to improve undersea marker detection. Finally, reference [8] presents a SLAM framework using fiducial markers allowing calibration between an onboard camera and the vehicle odometry.

The hydrodynamic model implemented in this thesis builds on previous work completed by Eric Bermudez [9] and Ian Taylor [4]. Bermudez developed a hydrodynamic model for a cross tunnel thruster-enabled AUV [9]. Taylor further developed this model to include accurate three and six degree of freedom (DOF) models of REMUS 100 that include variable speed CTTs [4].

1.6 Outline

The rest of this thesis is structured in the following manner. Chapter II describes the relevant SLAM position estimation technique given the absence of global positioning system (GPS) but with beacon and potential camera measurements. Chapter III discusses the USBL sensor, undersea imaging, fiducial markers, and the challenges associated with these tools in the undersea environment. Chapter IV introduces the simulation framework and discuss the components of its architecture. Challenges associated with controlling the NPS REMUS vehicle are also addressed. Chapter V gives the results and analysis of the simulation framework and its usability. A demonstration of Image-Based SLAM in simulation is presented and discussed. Chapter VI provides a conclusion of the research and suggestions for improvements and future work.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Simultaneous Localization and Mapping

This chapter first summarizes the history of SLAM and position estimation. I discuss the concept of uncertainty and how it is represented. Next is a description of the SLAM algorithm implemented, iSAM2, and its components. Finally, I present an example demonstrating the inner workings of iSAM2.

2.1 Background

In the late 1980s, the localization and mapping problem was addressed as two distinct processes until Whyte et al. presented their solution in 1995, which fundamentally shifted the approach to this problem. This process combined the mapping and localization problems into a single convergent estimation problem, and includes the correlation between landmarks to further improve the solution. Combining the localization and mapping processes coined the term "Simultaneous Localization and Mapping" [10]. Whyte et al. defines SLAM as "a process by which a mobile robot can build a map of an environment and at the same time use this map to deduce its location" [10].

Autonomous vehicle position estimation has evolved over the years, as developers have created different algorithms and expansions to improve localization. Position estimation originates with Kalman and the Kalman Filter, which combines the system and measurement models using Bayes Theorem [11]. This process assumes a relationship between two successive steps, which is linear, and a Gaussian distribution summarizes the error in the system and measurement models. Additionally, the system is described as a Markov Process where the entirety of prior information can be summarized as a single time-step [11]. Since Kalman's algorithm, several adaptations have been implemented to account for non-linear representations of the measurement and system models. The Extended Kalman Filter (EKF) denotes the state space vector as a collection of functions and uses Taylor Series Expansion to linearize the system at a given time step. Another adaptation is the Rao-Blackwellized or FastSLAM algorithm which shifted the description of vehicle motion to a generalized set of samples with a non-Gaussian probability distribution. The ever-increasing state-space

representation requirement of SLAM makes the application of particle filtering challenging. However, application of Rao-Blackwellization eased computational requirements. Rao-Blackwellization uses probability dependency properties to exploit the fact that only a portion of probability distributions need to be sampled [12]. The fundamental concepts of position estimation revolve around uncertainty and its representation as a stochastic process.

2.2 Uncertainty

Uncertainty manifests in robotics in several ways. Even if manufactured with high precision, any sensor or actuator will not operate at an exact specification. Most systems use calibration to minimize uncertainty prior to performing a task. Uncertainty, however, degrades the confidence in a vehicles position particularly if the operation or evolution takes place over a long time. This section illustrates the factors that contribute to uncertainty in the REMUS vehicle's navigation, explains how stochastic representation helps mitigate that uncertainty, and then provides an example of positional uncertainty and how it is minimized.

2.2.1 Sources of Uncertainty in REMUS

The NPS REMUS vehicle navigation process takes high-level control commands and transforms them into executable low-level control commands. As an example, the high-level control command of velocity will be executed as a low-level control command of rotations per minute (RPM) by the aft thrust propeller. A velocity command that correlates to 3000 RPM will most likely operate at a range of values from 2950 to 3050 RPM. REMUS uses a Kearfott internal navigation dystem (INS), which suffers from uncertainty quantified as 0.5% per distance traveled, further contributing to uncertainty growth. The USBL transponder also suffers from error in its accuracy of range and bearing measurements acoustically transmitted to the REMUS vehicle. Depending on the distance from the USBL, a range error may be off by 1 meter, and bearing can vary up to 2.5 degrees from the true value. Sensor error was modeled as Gaussian noise within the limits of these bearing and range parameters. These sources of uncertainty contribute the REMUS vehicle's overall position of uncertainty (PUC) to grow over time.

2.2.2 Stochastic Representation

Rather than addressing the uncertainty as its separate process or over-expending resources to maximize robot precision and sensor accuracy, the stochastic map includes uncertainty as part of the system representation. Fundamentally, the stochastic map relates the uncertainty of system variables to one another by way of the system covariance matrix. It defines an uncertain spatial relationship as every possible combination of uncertainty between state variables, using mean and standard deviation [13]. Equation 2.1 defines the covariance matrix.

$$C(\mathbf{x}) = E(xx^T) \tag{2.1}$$

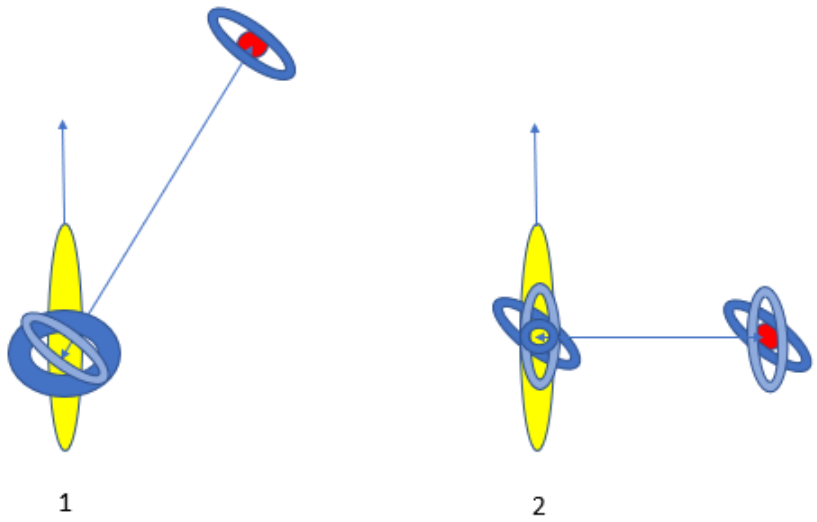
Equation 2.2 shows a three DOF process model, which will later be used as an example. The diagonal entries in the covariance matrix represent the variance of each state, and the off-diagonal entries show the uncertainty relationship between two state variables. Off diagonal entries become important in updating estimates where relational dependencies have changed [13].

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \psi \end{bmatrix}; \qquad C(\mathbf{x}) = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{\psi x} & \sigma_{\psi y} & \sigma_{\psi\psi} \end{bmatrix} \tag{2.2}$$

2.2.3 Uncertainty Example

In this two-dimensional example of navigation via USBL beaconing, REMUS begins at the origin and travels north to a goal of [50,0]. Bearing and range measurements are received by a USBL transponder at a range of 35 meters. The bearing and range measurements are used to reduce the vehicle's PUC. However, since the vehicle has more confidence in the range measurements than the bearing measurements, the vehicle will "favor" the range when

calculating its PUC. Without the USBL beaconing sensor, the uncertainty would grow as a circle (given equal covariance in σ_{xx} and σ_{yy}), but the range and bearing measurements received from the USBL cause the PUC to grow in the shape of an ellipse where the narrow portion of the ellipse favors the measurement of most certainty, in this case range.



(1) REMUS vehicle traveling north uses USBL to resolve bearing and range from transponder and reduce its PUC. (2) After traveling north the REMUS vehicle receives a new measurement from the same transponder to further minimize PUC.

Figure 2.1. REMUS Uncertainty Example

Figure 2.1 displays uncertainty as it relates to position estimations. The REMUS vehicle is at an initial position traveling north using the USBL transponder as an external navigation sensor. It receives an initial bearing and range and uses the uncertainty attributes of the USBL transponder to establish a positional uncertainty of both the transponder and its position. As the vehicle travels north, it will continue to take measurements from the same transponder accounting for the change in uncertainty relative to the transponder changes. As shown in position 2, the original measurement highlighted in the dark blue and the new measurement in the light blue as well as its navigation information via the process model, are used to localize with a smaller positional uncertainty.

2.3 iSAM2

Kaesses iSAM2 algorithm solves the SLAM problem by formulating it as a non-linear least-squares problem. Incremental Smoothing and Mapping (iSAM) is an incremental smoothing process that develops an exact representation of the entire vehicle trajectory and feature map. The use of *QR* decomposition allows for computation efficiency. Since the algorithm maintains covariances for all states, iSAM2 supports data association. Since the development of iSAM2, Prof. Frank Dellaert has extended this algorithm to Georgia Tech Smoothing and Mapping (GTSAM), a C++ based toolbox that develops graphical models or factor graphs of complex estimation problems, with the help of his many students to include Kaess. [14] This section outlines iSAM2 by first introducing the system model and then describes the steps that make up the algorithm: least squares approximation, linearization, and incremental matrix factorization.

2.3.1 System Model

The dynamic motion of the REMUS vehicle is summarized in equation 2.3 as a single non-linear function with control inputs ($\mathbf{u}$) and added noise ($\mathbf{w}$). Similarly, equation 2.4 shows the measurement model which accounts for the identification and measurement of landmarks ($\mathbf{l}$) and has added noise ($\mathbf{v}$). Both additive noise representations are Gaussian, with zero-mean and assumed independent identically distributed (i.i.d) [15] [1]. Note that both the system and measurement models are time-dependent, and there can be more than one landmark.

$$\mathbf{x}_i = f_i(\mathbf{x}_{i-1}, \mathbf{u}_i) + \mathbf{w}_i \tag{2.3}$$

$$\mathbf{z}_k = h_k(\mathbf{x}_{i_k}, \mathbf{l}_{j_k}) + \mathbf{v}_k \tag{2.4}$$

[1] Equations and information for section 2.3 are from [15], unless otherwise cited.

2.3.2 Least Squares

Non-linear least squares approximate a non-linear function as linear and minimize the summation of squared terms. Smoothing and Filtering (SAM) formulates a map estimate of the trajectory X and the map of landmarks L, given the control inputs ($\mathbf{u}$) and landmark measurements ($\mathbf{z}$) from the previously mentioned system and measurement models. The mathematical representation of the non-linear least squares problem is as follows:

$$\mathbf{X}^*, \mathbf{L}^* = \arg\min_{X,L} \left\{ \sum_{i=1}^{M} \| f_i(\mathbf{x}_{i-1}, \mathbf{u}_i) - \mathbf{x}_i \|^2_{\Lambda_i} + \sum_{k=1}^{K} \| h_k(\mathbf{x}_{i_k}, \mathbf{l}_{j_k}) - \mathbf{z}_k \|^2_{\Gamma_k} \right\} \tag{2.5}$$

2.3.3 Linearization

The linearization process is performed by Taylor Series Expansion of the system and measurement models described in equations 2.6, 2.7, and 2.8. This results in three Jacobian matrices. F is the Jacobian of the system model linearized about the point x^0. Similarly, the Jacobians of the measurement model are taken with respect to x^0 and l^0. The resulting three Jacobian matrices are shown below.

$$F_i^{i-1} = \left. \frac{\delta f_i(\mathbf{x}_{i-1}, \mathbf{u}_i)}{\delta \mathbf{x}_{i-1}} \right|_{\mathbf{x}^0_{i-1}} \tag{2.6}$$

$$H_k^{i_k} = \left. \frac{\delta h_k(\mathbf{x}_{ik}, \mathbf{l}_{j_k})}{\delta \mathbf{x}_{ik}} \right|_{\mathbf{x}^0_{i_k}, \mathbf{l}^0_{j_k}} \tag{2.7}$$

$$J_k^{j_k} = \left. \frac{\delta h_k(\mathbf{x}_{i_k}, \mathbf{l}_{j_k})}{\delta \mathbf{l}_{j_k}} \right|_{\mathbf{x}^0_{i_k}, \mathbf{l}^0_{j_k}} \tag{2.8}$$

These matrices are grouped into a large A matrix to include a particular matrix G which properly addresses *dx/dx*. Using these linearized systems and measurement models, 2.5

becomes

$$\delta\boldsymbol{\theta}^* = \arg\min_{\delta\boldsymbol{\theta}} \left\{ \sum_{i=1}^{M} \|F_i^{i-1}\delta\mathbf{x}_{i-1} + G_i^i\delta\mathbf{x}_i - \mathbf{a}_i\|_{\Lambda_i}^2 + \sum_{k=1}^{K} \|H_k^{i_k}\delta\mathbf{x}_{i_k} + J_k^{j_k}\delta\mathbf{l}_{j_k} - \mathbf{c}_k\|_{\Gamma_k}^2 \right\} \quad (2.9)$$

Theta is a single vector of $M + K$ dimensions where M is the number of pose variables, and K is the number of landmark variables. Further algebraic manipulation of eq 2.9 results in the sparse standard least-squares problem, where the A matrix in eq. 2.10 is the augmented Jacobian matrix of the process, and measurement models and the remaining terms are grouped into the b vector as shown in eq. 2.11.

$$A = \begin{bmatrix} G & 0 & 0 & 0 \\ F & H & 0 & 0 \\ 0 & F & H & 0 \\ 0 & J & 0 & J \end{bmatrix} \quad (2.10)$$

$$\theta^* = \arg\min_{\theta} \|A\boldsymbol{\theta} - \mathbf{b}\|^2 \quad (2.11)$$

2.3.4 Incremental Matrix Factorization

Solving the least-squares problem can be achieved by using Cholesky decomposition of $A^T A$; however, to achieve computationally efficient incremental updates, QR factorization proves more useful. The QR factorization of the Jacobian A matrix is defined and solved as follows:

$$\|A\boldsymbol{\theta} - \mathbf{b}\|^2 = \|Q \begin{bmatrix} R \\ 0 \end{bmatrix} \boldsymbol{\theta} - \mathbf{b}\|^2 \quad (2.12)$$

$$= \| \begin{bmatrix} R \\ 0 \end{bmatrix} \boldsymbol{\theta} - \begin{bmatrix} \mathbf{d} \\ \mathbf{e} \end{bmatrix} \|^2$$

$$= \|R\boldsymbol{\theta} - \mathbf{d}\|^2 + \|\mathbf{e}\|^2$$

$$R\boldsymbol{\theta}^* = \mathbf{d}$$

Note that ($[\mathbf{de}]^T = Q^T\mathbf{b}$) and when ($R\boldsymbol{\theta} = \mathbf{d}$) it removes the residual ($\mathbf{e}^2$). The $\boldsymbol{\theta}$ vector is solved by back substitution. R is upper right triangular. Therefore, most of the work has already been done, resulting in the complete trajectory and map that includes all measured landmarks. The true advantage of QR factorization is in its ability to quickly and efficiently perform updates. Rather than factoring the Jacobian matrix with every new measurement, Kaess performs QR-updating, which augments the upper right triangular sparse R matrix with a new measurement W^t and the corresponding right-hand side $\mathbf{d}$ vector as shown in eq. 2.13. This results in a system that can quickly be solved by givens rotations and efficiently perform smoothing and mapping.

$$\begin{bmatrix} Q^T & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} A \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} R \\ \mathbf{w}^T \end{bmatrix} \tag{2.13}$$

2.4 USBL Navigation Example

This section will further explain the inner workings of iSAM2 as a continuation of the example presented in 2.2. Beginning at the origin and traveling directly ahead at a speed of 1 m/s, we will observe the A and R matrices as the REMUS vehicle uses range and bearing measurements to a transponder to assist in localization. Each pose node in the factor graph summarizes 5 seconds of travel or 5 m in an ideal scenario. An in-depth look into the vehicle's travel with its two latest poses gives us the graph represented in figure 2.2.

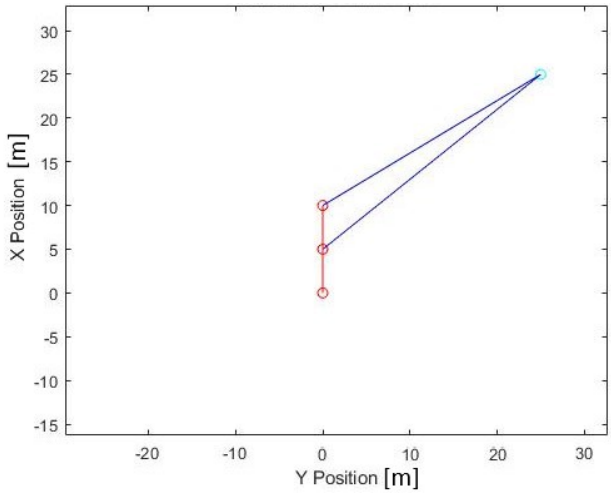


Figure 2.2. Single Iteration MATLAB Example of REMUS Traveling Along Heading 000°

Figure 2.3 displays the gradual graphical development of iSAM2. The red circles represent the pose nodes of the REMUS vehicle or X^* from eq. 2.5, and the light blue circle represents the transponder or L^* from 2.5. Lines between any two nodes are commonly referred to as constraints. Another advantage of iSAM2 is that the entire trajectory is maintained as well as all pose nodes and feature positions. Since only the additional entries to the A matrix are augmented to the lower left portion to include updated measurements to known features or any new detection, the A matrix remains sparse, lending itself to QR decomposition.

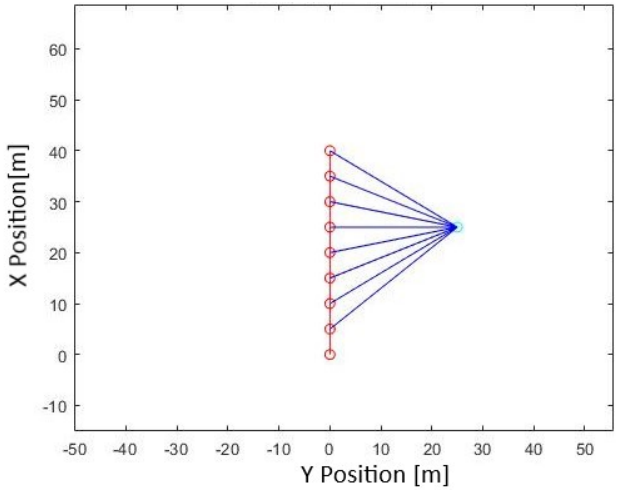


Figure 2.3. Matlab Example of REMUS Traveling Along Heading 000°

Recall equation 2.9 that details the components of each term in the A matrix. The first term $F_i^{i-1}\delta x_{i-1}$ is calculated by taking the Jacobian of our process model with respect to the X term. Subsequent terms $G_i^i\delta x_i$, $H_k^{ik}\delta x_{ik}$, and $J_k^{jk}\delta l_{jk}$ are similarly calculated to generate a sparse A matrix as shown in Figure 2.4.

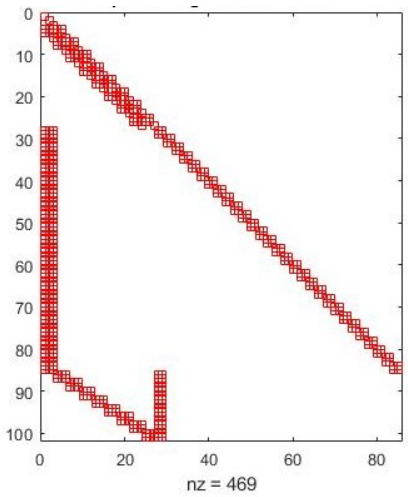


Figure 2.4. Visually Sparse A Matrix

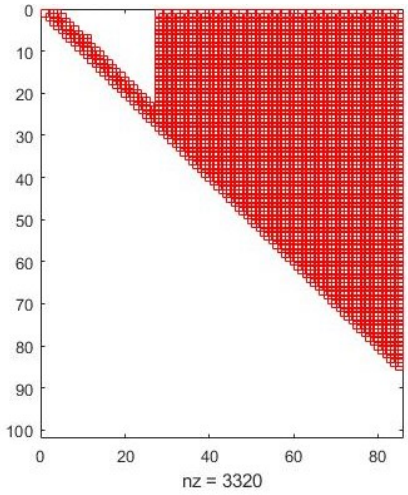


Figure 2.5. R Matrix After QR Decomposition of Sparse A Matrix

Figures 2.4 and 2.5 visually display the Augmented A matrix and non-linear least-squares solution derived by QR decomposition. The variables in the A matrix will have undergone variable reordering via Givens rotations in order to extract the R matrix. As described in

section 2.3.4, rather than restarting the A matrix decomposition process, a new measurement will be augmented to the R matrix to generate R^*, which will again be solved via Givens rotations to achieve an updated solution efficiently.

2.5 Summary

This chapter discussed uncertainty and the stochastic representation of positional uncertainty. A high-level development of iSAM2 was discussed and implemented in a simple example of AUV navigation with a USBL transponder as an external sensor. The implementation iSAM2 will continue to expand as we incorporate an additional visual sensor to assist with short-range localization.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Sensor Processing

This chapter discusses the primary sensor systems used onboard the REMUS vehicle to conduct image-based SLAM. First, I discuss the long-range sensor used as the primary aid in navigation to the area of interest (AOI). Then, a brief summary of underwater imaging and its associated challenges. I introduce the short range sensor used for underwater imaging and explain the system configuration for interpreting these images. As an intermediary step, I introduce and demonstrate a fiducial marker system called AprilTags as the short-range system for fine granular position estimation. Ultimately, the goal is to use an a priori image mosaic loaded onboard REMUS and computer-vision-feature-extraction techniques to localize the vehicle and locate the un-exploded ordnance (UXO).

3.1 Long Range Sensor

The digital ultra-short baseline (D-USBL) array is a modular acoustic navigation aid used by the REMUS vehicle as a sensor for range and bearing measurements to a transponder. The main components of the sensor system include a four-channel planar hydrophone array and an integral receiver/processor circuit board. During operation, the D-USBL emits interrogation pings at three-second intervals. The transponder emits an acoustic reply which is received by the four hydrophones. By measuring the difference in receive times of each of the hydrophones, a bearing and range can be resolved from the vehicle to the transponder [16]. Since the transponder position is known, the REMUS vehicle can localize itself from this information. Figure 3.1 displays the D-USBL and transponder interaction.

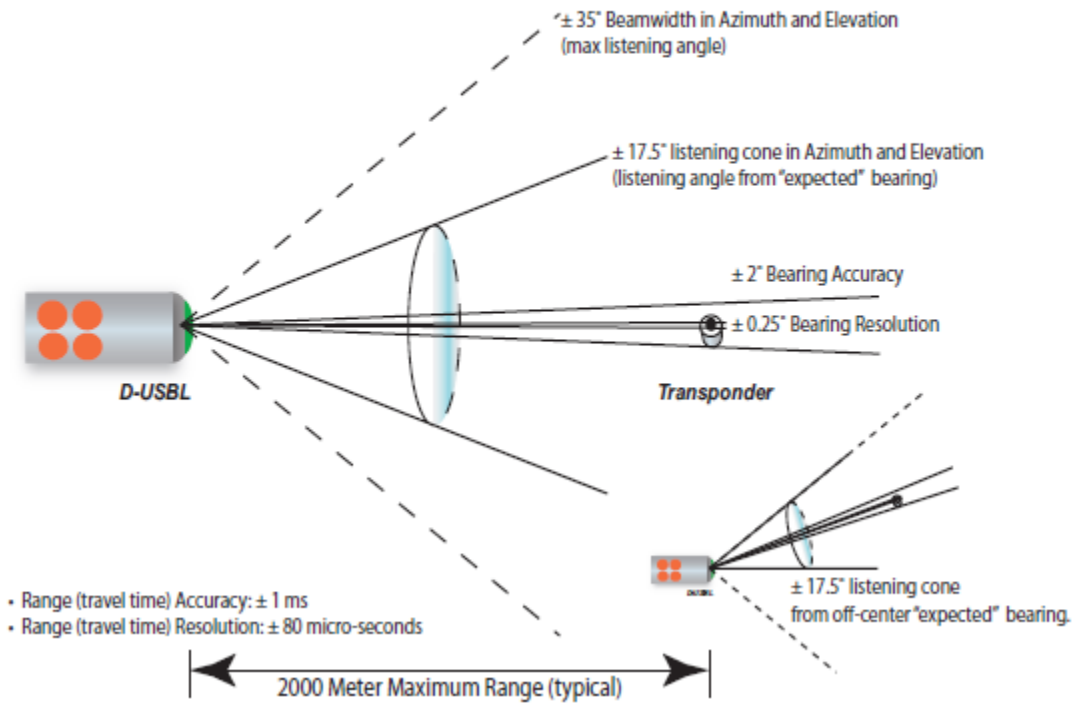


Figure 3.1. Diagram of D-USBL Operation. Source: [16].

The D-USBL sensor has a maximum listening angle of 60 degrees along with the vertical and horizontal directions within the local reference frame. A narrower "listening" angle is established at 35 degrees, which is used to calculate good fixes based on the estimated position of the vehicle and the expected bearing and range of the transponder. Table 3.1 details performance specifications of the D-USBL array.

Table 3.1. D-USBL Specifications. Adapted from [16].

Digital Ultra Short Baseline Array Specifications	
Specification	Capability
Beamwidth	±35°
Listening Angle	±17.5°
Maximum Range	2000 m
Frequency Range	20-30 kHz
Bearing Accuracy	±2.5°
Bearing Resolution	±0.25°
Range Accuracy	$\pm 1ms$
Range Resolution	$\pm 80micro - seconds$

Particular attention should be paid to the bearing and range accuracy specifications. As typical with most sonars, the range resolution is more accurate than the bearing measurement. This is important because greater confidence is given to the range measurement with the iSAM2 information matrix. The D-USBL sensor will be the initial sensor in assisting the vehicle navigating to the area of interest. It will then transition into low-speed control and use the camera for feature correlation with the ocean floor (represented by the AprilTag detection for this thesis) as an additional in-close navigation sensor.

3.2 Underwater Imaging

Underwater imaging is primarily categorized into two types, active optical imaging and passive optical imaging. During active optical imaging, the camera image is enhanced by an active light source. Conversely, passive optical imaging uses ambient light in the environment to produce images [17]. The underwater environment poses many challenges to optical imaging systems. Sea state, lighting, particulates, scattering, and attenuation are some of the challenges [18]. Particulates are biological phenomena that make the transmission of light through water highly variable. Light can be absorbed, refracted, reflected, scattered, and depolarized, further complicating the environment. An example of underwater characterization is Jerlov Water-type Classification that classifies attenuation in the sea of spectral-downward irradiance [19].

3.3 Short Range Sensor

In this section I introduce the camera used for underwater imaging and its specifications. Next the module that stores the camera for underwater operations is presented. A brief summary of Fiducial markers, an imaging tool used in robotics, is provided. AprilTags, the specific family of fiducial markers used for testing, is then discussed to include the optimal conditions for tag detection.

3.3.1 Camera

The underwater imaging sensor used for this thesis is a WATEC 910HX/RC camera with a Computar 2.6mm 1/3" Auto -Iris Lens. This ultra low light camera is operable at 3.0E-6 lumens per square meter (LUX) and has a 768(H) and 494(V) pixel definition. As a reference, a 100W incandescent bulb emits roughly 1600 lumens. [20]. The camera is mounted on board. The camera module is shown in Figure 3.3, along with other components described in the next section. This module is mounted at the forward-most position of the REMUS vehicle and directly down facing in the north east down (NED) positive Z direction. At the intended operating depth of 3 m, the camera maintains a horizontal and vertical field of 4 m and 5.5 m views, respectively.

Figure 3.2. Image of WATEC 910HX/RC Camera. Source: [20].

3.3.2 Camera Module

The camera module was developed with the technical expertise of Center for Autonomous Vehicle Research (CAVR) technician Lee Van Houtte. The watertight cylindrical capsule containing the necessary electronic equipment was developed in SOLIDWORKS. Within

the capsule, the WATEC camera is mounted facing downwards and connected to the "stack." The stack consists of two boards, one being a SENSORAY audio and video encoder/decoder capable of two simultaneous video streams. The second board in the stack is an Intel atom-based 4 core PCIe/104 processor that connects to REMUS's secondary controller. Complete specification information for the SENSORAY encoder/decoder and Intel processor can be found in [21] [22]. Figures 3.3 and 3.4 display the stand-alone camera module and the REMUS configuration with the camera module attached.

Figure 3.3. Image of Detached Camera Module

Figure 3.4. Image of REMUS With Camera Module Attached

3.3.3 Fiduicial Markers

Fiducial markers are pictorial objects used as visual reference points within an image. Originally developed for computer vision and virtual reality software development applications, these markers have expanded into several applications to include medicine, object identification, and robotics [18].

In 2015, Cesar et al. conducted a comparative analysis of three known fiducial marker systems in underwater environments, comparing their performance under variable conditions. Figure 3.5 shows examples of the three tag families evaluated. The top left image displays an AprilTag followed by an ARToolKit marker on the right. The bottom left marker is an example of an ArUco marker. The bottom right image contains smaller sizes of all three tag families. The goal of the series of experiments was to evaluate tag detection on the basis of six criteria: lighting, turbidity, marker size, camera distance, camera angle, and processing time. In summary, the study used a tank with black walls and fluorescent lamps to control lighting, powdered clay to regulate turbidity, and different sizes of each marker to evaluate marker size and camera distance. After several experiments, results showed that AprilTags outperformed other fiducial marker systems in every category except for time to detect. AprilTags recorded the highest time to detect with up to 150ms detection time [18].

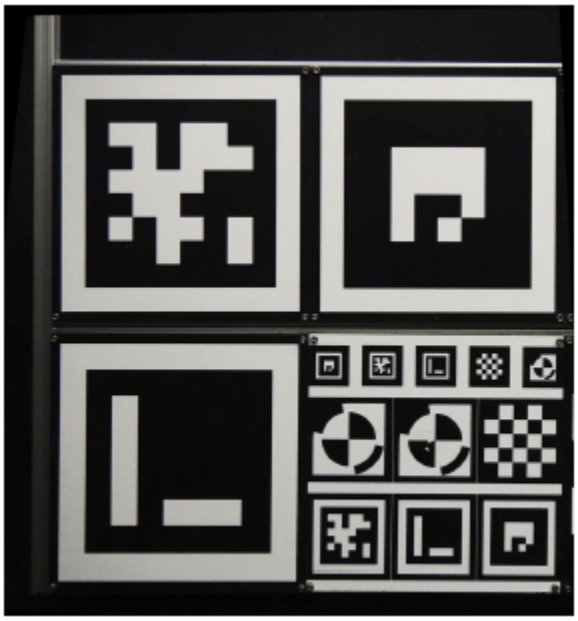


Figure 3.5. Example of Fiducial Markers. Source: [18].

AprilTags demonstrated high detection robustness in a multitude of environmental conditions, making it the obvious choice for this thesis. The primary goal in using a fiducial marker to represent a feature on the ocean floor is to maximize its probability of detection. The implementation of visual SLAM using iSAM2 will be conducted in relatively shallow waters (10-40m) with the REMUS vehicle operating 3m off the sea bottom. The littoral nature of this environment makes the turbidity, visibility, and lighting more significant. Another factor contributing to the use of AprilTags is the point where the REMUS vehicle enters the area of interest. The vehicle will enter slow speed control (further discussed in Chapter 4) to relocate the object of interest, meaning that the AprilTag will remain in the field of view for an ample amount of time to achieve a positive detection.

3.3.4 AprilTags

The AprilTag system was developed in 2011 by professor Edwin Olson of the University of Michigan [23]. His fiducial system attacked two key components in designing robust markers: marker detection and payload extraction. The tag detection sequence is separated into two jobs. First, a quad detection algorithm is used by first identifying all line segments present in the image. Next, a depth-first search is used to identify orthogonal adjacent line segments in the counterclockwise direction. Once four segments or a "quad" have been reached, a candidate for detection has been identified. Using known camera characteristics and the four quad corners, the AprilTag is represented as a single point. From this point, the payload can be extracted using the positional information of each corner by using a tag relative coordinate frame. The robustness of the AprilTag system is in its payload coding system. As a modified lexicode, the AprilTag system uses a minimum hamming distance between every payload in the tag family. This ensures that the fiducial markers are resistant to false detection due to rotation.

Since the release of his fiducial marker system, Olson and Wang have released newer models of the AprilTag systems: AprilTag2 and AprilTag3. For the AprilTag2 system, Olson improves his detection methodology by using adaptive thresholding and image segmentation [24]. The AprilTag3 system removes the boundary requirement present in most fiducial marker systems, further increasing the imaging application [25].

3.3.5 AprilTag Identification

In conducting detection of fiducial markers, it is important to ensure that the best possible detection occurs. In addition to the environmental challenges associated with underwater imaging, additional challenges must be considered and addressed to ensure the most accurate tag detection. Sources of degradation in marker detection are primarily from detection distance, viewing angle, and yaw angle. Figure 3.6 diagrams these sources of degradation.

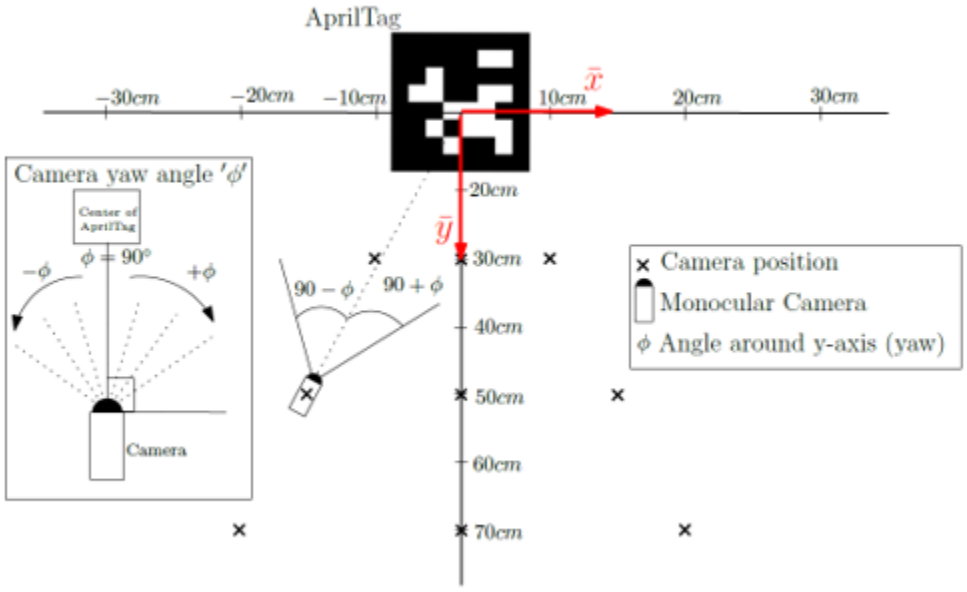


Figure 3.6. AprilTag Detection Degradation Sources. Source: [26].

The most intuitive source of degradation is the distance between the camera and the marker. As distance increases, the resolution of the AprilTag decreases, which, in turn, decreases the ability to identify the marker. The information or payload contained within the marker is based on its center. Sub-optimal marker detection results in inaccurate payload information. The next source of degradation is the viewing angle. Beyond distance, the optimal viewing angle is directly orthogonal to the plane of the marker with the origin at its center. The greater is the deviation in the viewing angle, the higher is the degradation in marker detection accuracy. Continuing with the topic of viewing angle, the yaw angle of the camera introduces further potential degradation. In any detection scenario, the alignment of the marker origin and camera center will produce the best marker detection. Misalignment

between the camera center and the tag origin introduce detection degradation within the payload [26]. These sources of degradation, especially the alignment of marker and camera centers, highlight the level of control required of the REMUS vehicle.

3.4 System Setup

Figure 3.7 displays a diagram of the system setup for AprilTag testing. The input of the camera is the raw image of the camera at the bottom of the tank and eventually the seafloor. The WATEC camera outputs the images to a digital encoder. A Robot Operating System (ROS) process running on board the Intel processor connected to the encoder takes the image and exports it as information that can be used by another ROS process. This process is conducting AprilTag identification and outputting the information contained within the payload of the marker. ROS is a robot development architecture that will be used in conjunction with other software for simulation described in detail in Chapter 4.

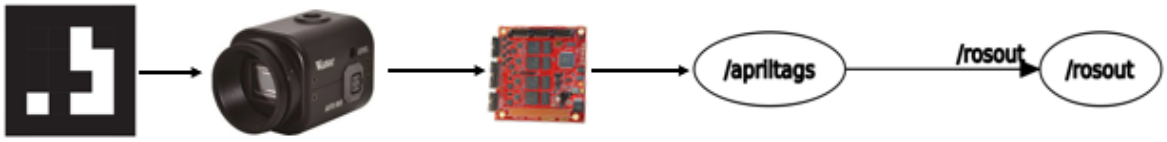


Figure 3.7. Flow of Information Diagram

3.5 AprilTag Testing

After successfully mounting the WATEC camera module on-board the REMUS vehicle and verifying its proper function, manual testing began in the CAVR lab verifying camera and AprilTag performance. After initiating the camera sensor, the AprilTags node and recording the data via a ROS .bag file the REMUS was manually maneuvered in a tank at the CAVR Lab shown in Figure 3.8.

Figure 3.8. Watec Camera Test Performed at the NPS CAVR Lab

The vehicle was placed at the far end of the tank, ensuring that the AprilTag was out of the camera's field of view. The REMUS vehicle was slowly pushed towards the AprilTag using a manual hooking tool as described in the Chapter 1 scenario. An example of the image from the downward-looking camera is shown in Figure 3.9.

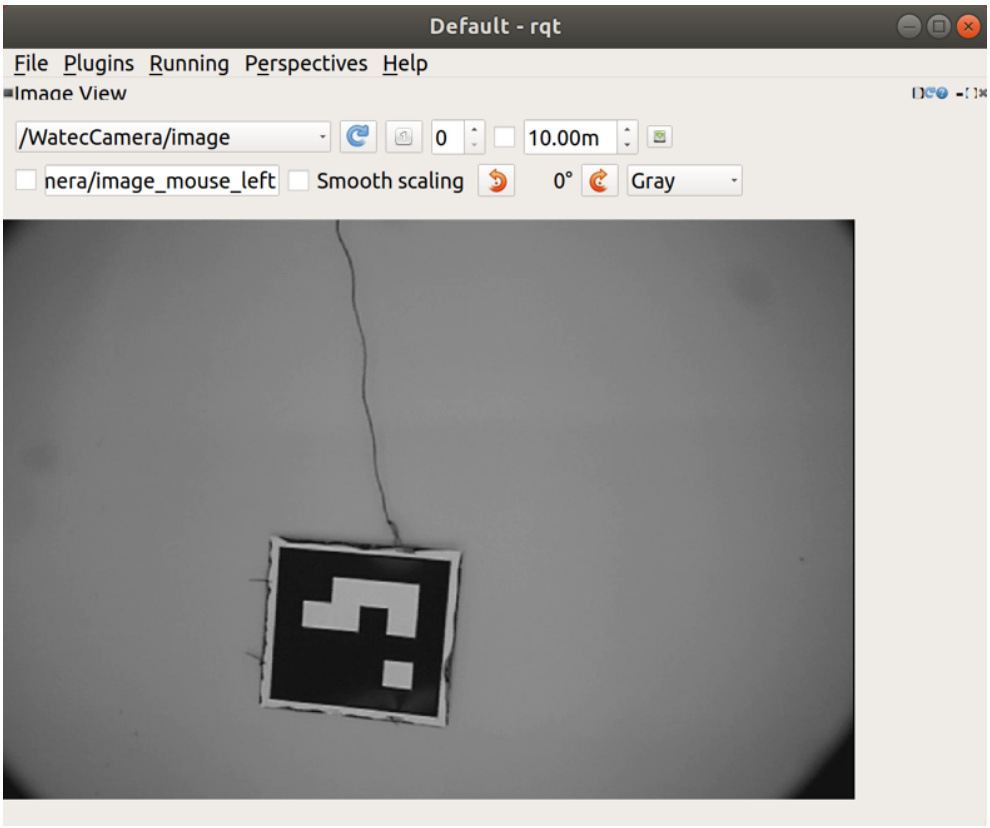


Figure 3.9. Detection of AprilTag Using Watec Camera

The vehicle was moved directly over the tag to verify proper detection. Next, the REMUS vehicle was manually maneuvered methodically in the XY plane to understand the limitations of the camera's field of view. A display of the position information is shown in Figure 3.10.

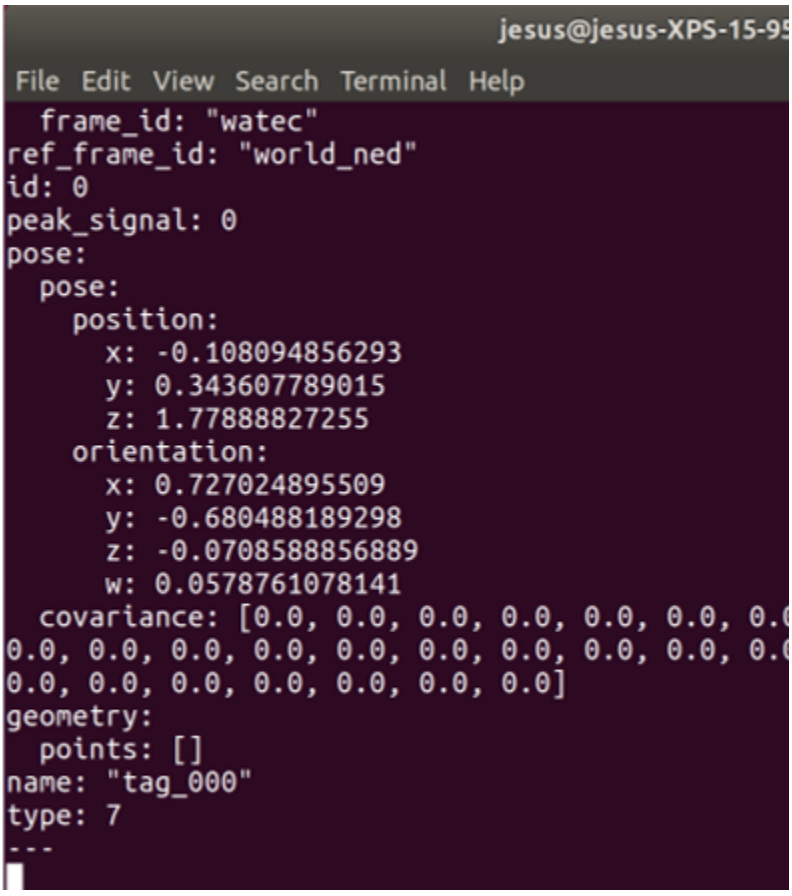


Figure 3.10. Position Information of Identified AprilTag

The information of the AprilTag payload contains the position information of the tag in Cartesian coordinates and the orientation of the tag displayed in quaternions. The simplest method of verifying the AprilTag information was by comparing the z or depth measurement to the depth of the tank. This measurement proved to be accurate, with the depth of the tank at 1.67m and the measurement reading 1.77m. In maneuvering the vehicle in the XY plane, the appropriate changes in orientation and Cartesian measurements were observed, verifying the proper function of the AprilTags code. A significant lag was discovered from the maneuvering of the vehicle to the updated position information. This was attributed to the high frame rate of the camera, which was adjusted to 2Hz to allow for near real-time position information. Further tank testing of the camera performance was not performed due to operational limitations observed during the spring of 2020 to combat the COVID-19 virus.

3.6 Visual Representation

The RIN proposed solution discussed in this thesis includes two fundamental problems that must be addressed. First, a position estimation problem occurs in which the REMUS position is degrading as a function of distance traveled. This problem can be addressed through the use of iSAM2 with the relatively inexpensive sensors discussed in this chapter. The real-world application of the proposed solution in this thesis requires the extraction of features on the seafloor through computer imaging. To reduce the complexity of the problem and focus primarily on position estimation, AprilTags were chosen as a visual representation of the images on the seafloor as a substitute for image correlation between the camera and an image mosaic built from camera data collected from the first SCM mission.

A real-world application of a camera sensor coupled with an a priori image mosaic would require near real-time image correlation using computer imaging techniques to extract features from the image and match them to the mosaic for fine granular position estimation. Video images received from the camera are turned into portable network graphics (PNG) files and must then be turned into or "stitched" into a mosaic in order to conduct image correlation. An example of the process is shown in Figures 3.11 and 3.12.

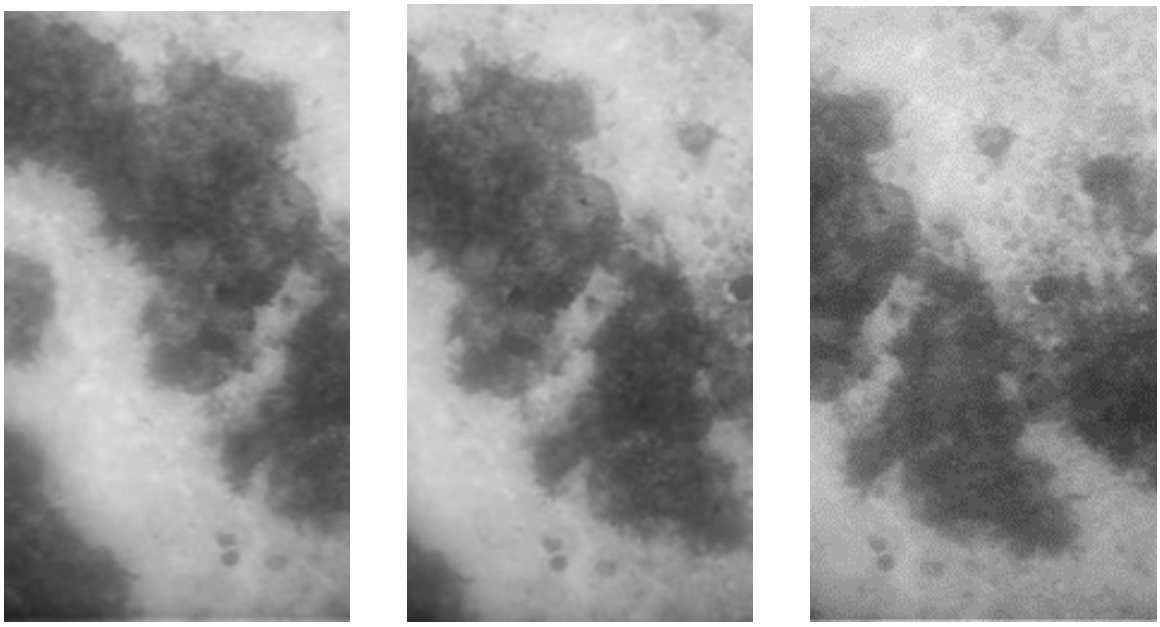


Figure 3.11. Three Sequential Seafloor Images

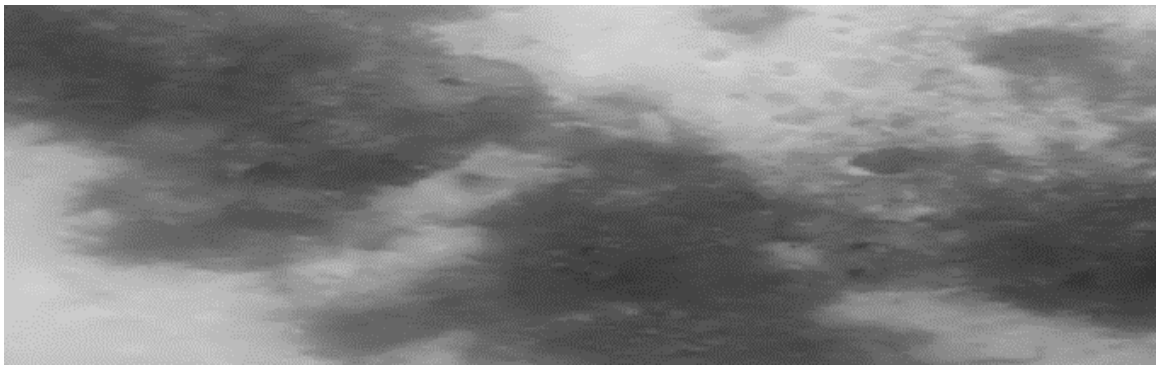


Figure 3.12. Stitched Seafloor Image of Three Sequential Images

This methodology presents various problems such as image distortion, environmental effects, and camera calibration, which will not be addressed in this thesis. The purpose of this section is to reiterate the use of fiducial markers as a simplification of the problem.

3.7 Summary

This chapter discussed the use of the D-USBL as a long-range sensor that provides acoustic range and bearing measurements. In contrast, the camera coupled with AprilTags gives necessary information for fine granular position estimation. Next, I discuss the implementation of software processes necessary to perform simulation of the RIN scenario.

CHAPTER 4:
Simulation and Implementation

This chapter discusses the tools, framework, and methodology for rapid AUV SLAM development. The approach for this framework is focused on hardware/software-in-the-loop (H/SIL) simulation. Industry often uses H/SIL to enhance testing quality, overcome tight development schedules, and reduce the economic impacts of laboratory testing [27]. A H/SIL architecture is advantageous. It replicates the real-world environment mathematically and allows for the software used in simulation to be directly applied to real-world testing. This reduces wear and tear on the AUV, reduces costs, and permits rehearsals for operations when a tactical surprise is critical.

In the case of REMUS operations and specifically for the image-based SLAM application, the goal is to develop a system that principally does two things:

- Develop a simulation environment that replicates the ocean environment to permit robust testing of software.
- Develop software that is the same code and communication architecture that is run onboard the vehicle.

In short, the same processes running onboard REMUS in real-world applications are also running in simulation. The H/SIL framework allow for rapid development and testing. In this chapter, I introduce the components of the architecture structure and combine the topics covered in this chapter as well as in chapters 2 and 3 to discuss the simulation in its entirety and the control required to conduct image-based SLAM in the RIN scenario.

4.1 Architecture

The software architecture implemented in this thesis has several components. They are the Linux Operating System, Robot Operating System, GAZEBO, Unmanned Undersea Vehicle Simulator (UUV-SIM) and NPS CAVR structure. The UUV-SIM and NPS CAVR components of the architecture are software-based and can reside on a laptop or the sec-

ondary controller CPU onboard the REMUS vehicle. They contain the components that permit augmenting military-grade vehicles with autonomy. Control commands are sent via the ethernet API to the main REMUS controller for actual operations. In simulation, similar control commands can be provided to the simulation environment. Figure 4.1 depicts the relationships between GAZEBO and ROS that encompass UUV-SIM. Figure 4.2 displays the CAVR structure and information flow. Next I briefly describe each software component beginning with the Linux Operating System.

4.1.1 Linux

The Linux operating system, also known as GNU/Linux, is traced primarily to two people: Linus Torvalds and Richard Stallman. Frustrated with the restrictions of the available operating systems at the time, Torvalds set out to build his operating system focusing on the component governing the communication between hardware and applications known as the kernel. At the same time, Stallman, who championed free, open-source software, began developing tools, utilities, and compilers necessary for an operating system. Naturally, the duo would combine their work into a fully working operating system. Since its development in 1991, several distributions of Linux have been released to include Ubuntu 18.04, released in April of 2018. Linux is a preferred operating system among developers because it is free and open-source software. Additionally, it is designed for compatibility with other operating systems and supports most programming languages, fostering collaboration in a variety of applications among software developers [28].

4.1.2 ROS

Seeing the need for a collaborative environment for robotics developers, developers at Stanford University initiated the Robot Operating System (ROS) environment. In 2007, funding from Willow Garage launched this project into the great collaborative environment used today. Since its initial public release, several updated ROS environments have emerged, including the version ROS Melodic Morenia, which was used in this thesis. The ROS framework follows a “publish” and “subscribe” framework to support a collaborative environment. This framework works by having processes run as “nodes” that publish information as “topics” and subscribing to that information to be used by another process [29].

4.1.3 GAZEBO

The Open Source Robotics Foundation (OSRF) funded simulation platform GAZEBO allows robotics developers to operate in various indoor and outdoor environments. It supports the widely used physics-based Open Dynamics Engine (ODE) and can be configured to use others as required by the user. This robust combination of dynamics, graphics, and sensor simulators makes it ideal for visualizing real- world scenarios. GAZEBO's pictorial display and simulation capabilities coupled with ROS's robot development tools provide a reliable open-source platform for robot development, simulation, and testing. GAZEBO is a very capable simulator; however, the physics engines are not properly configured to support the difference in dynamics and kinematics of the undersea environment. In order to accurately simulate AUV scenarios, an extension must be used, in this case, the Unmanned Underwater Vehicle Simulator (UUV-SIM) [30].

4.1.4 UUV-SIM

The Unmanned Underwater Vehicle Simulator was developed for the simulation of collaborative underwater vehicle operations. It was designed as an extension to GAZEBO to be implemented through ROS so that it can be used to test vehicles developed within the ROS framework. Through a series of plugins, UUV-SIM provides actuator models, underwater sensors, hydrodynamic and hydrostatic forces and moments, and underwater environment and world loads. Figure 4.1 displays the UUV-SIM software structure within Gazebo and ROS [31].

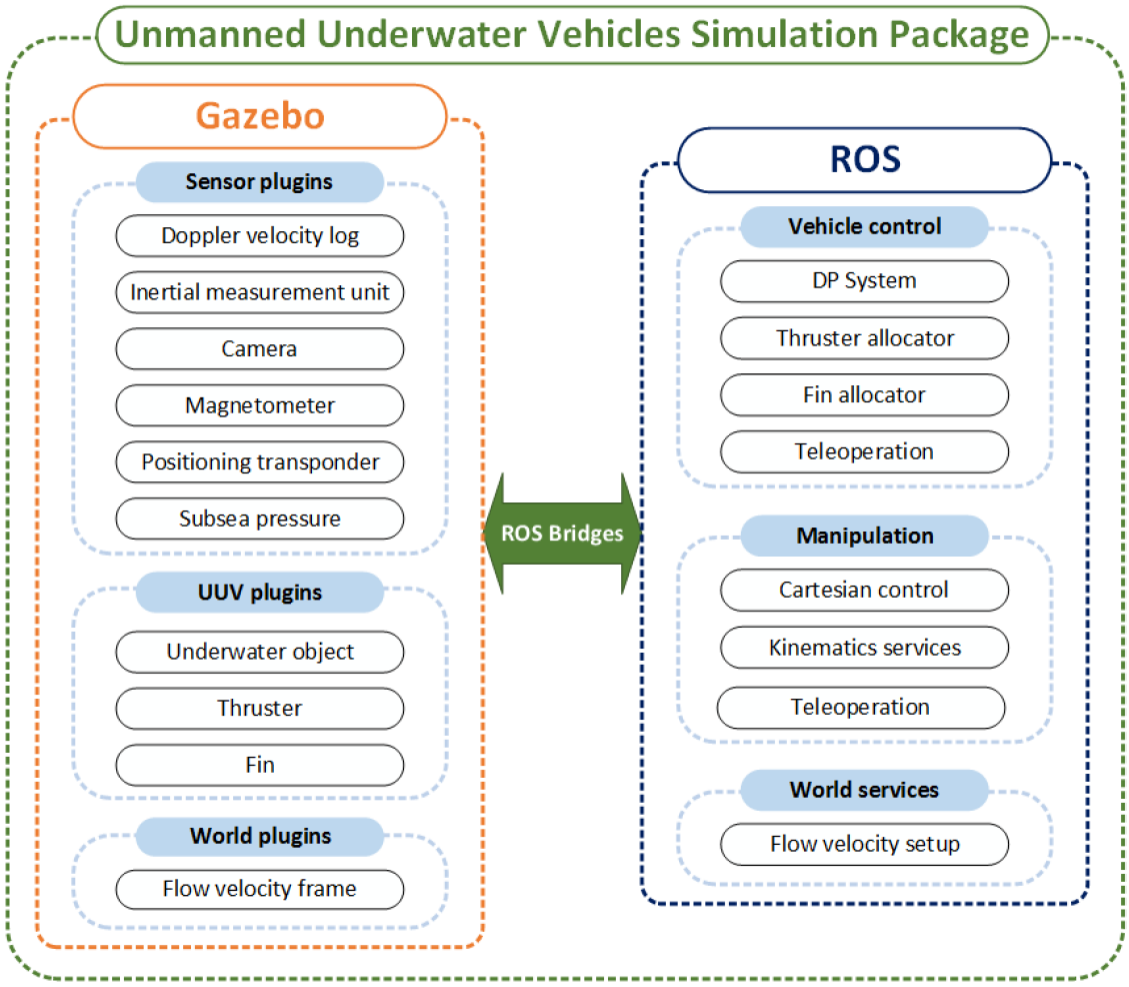
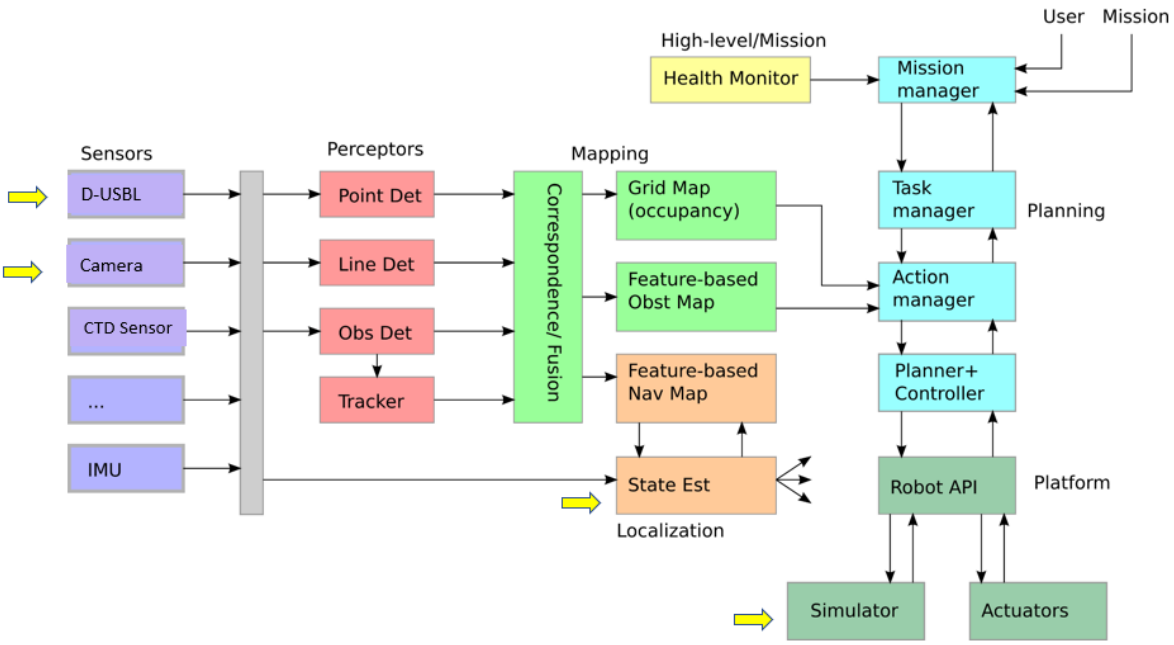


Figure 4.1. Diagram of UUV-SIM Package. Source: [31]

4.1.5 Information Flow

Up until this point, the architecture facilitates the ability to support robotic autonomy software. These are software processes for increasing the autonomy of the unmanned system to operate safely and more independently in its environment. The top of the pyramid is a software architecture developed by NPS researchers that permits autonomy on a wide class of unmanned systems. It has been employed as a secondary control architecture onboard aerial, surface, and underwater vehicles. It is shown in Figure 4.2. From an organizational perspective, information flow goes from left to right. Groups are organized into Sensors, Perceptors, Mapping, Localization, Planning, and Platform. Sensors are the tools mounted onboard the REMUS Vehicle to collect information about the environment, such as seafloor images from the WATEC camera or range and bearing measurements by the D-USBL. Preceptors interpret the information received from these sensors and use it to extract useful information. As an example, the Obstacle Detector ingests sonar images and detects imminent collisions. Mapping develops and maintains a feature-based obstacle or terrain map for feature-based navigation. Localization provides position estimation based on the sensor measurements and the localization algorithm. Planning determines the logic to execute the specified mission and selects the appropriate planners and controllers for mission execution. Finally, the Platform determines the proper actuators and processors for the different autonomous platforms used by NPS.

Next, I discuss integrating these components into a simulation using the Center for Autonomous Vehicle Research CAVR structure displayed in Figure 4.2 as a visual representation.



This diagram is a visual representation of the information flow within the CAVR structure. Highlighted by the yellow arrows are the key components discussed in this section. Sensors feed information to the iSAM2 process, processed information is then sent to the controller which then executes commands within the simulation.

Figure 4.2. CAVR Structure

The mission profile is executed via the mission manager, which contains the waypoint navigation information that REMUS will execute. Beginning from the left is the D-USBL and camera sensors sending information to the state estimator or iSAM2. The long-range D-USBL sensor is modeled by the `perc_usbl_sim` node, which takes the position information of the simulated transponder and outputs bearing and range measurements from REMUS to the transponder. Similarly, the `camera_sim` node simulates the down-looking camera and the images it captures of the seafloor. From here, a running sub-process summarized as a screenshot of the down-looking aspect of the camera is taken, converted into a jpg file, and run through a node named `perc_april_tags`. The `perc_april_tags` node is an adaptation of the AprilTag software, described in Chapter 3, for this scenario.

The output of the `perc_april_tags` node is the position information of the AprilTag. The information deduced from the long and short-range simulators is received by the iSAM2 process, which will take the sensor measurements and position information and provide the REMUS vehicle's trajectory as the location of the transponder and AprilTag.

4.2 Hydrodynamic Model

The hydrodynamic and hydro-static forces used for simulation by UUV-SIM stem from Thor I. Fossen's equations of motion described in [31]. Fossen's equations of motion are shown in equation 4.1.

$$(M_{RB} + M_A)\dot{v_r} + (C_{RB}(v_r) + C_A(v_r))v_r + D(v_r)v_r + g_0 + g(\eta) = \tau \tag{4.1}$$

Here, the M matrices represent a rigid body M_{RB} and added mass M_A; the C matrices are the coriolis and centripetal effects of these components; D is the damping matrix. g_0 and g_n are restoring forces for gravity and buoyancy, and τ are the control forces and torques. A detailed description of Fossen's equations is described in Chapter 6 of [3]. Adapting these equations for the NPS REMUS 100 vehicle is challenging in part because of the unique capability to control the vehicle both at slower and faster speeds due to the vertical and horizontal cross tunnel thrusters.

The NPS REMUS 100 vehicle possesses horizontal and vertical cross tunnel thrusters CTT that allow greater control over pitch and yaw than simple AUV configurations. UUV-SIM is effective in modeling the forces on AUVs; however, previous hydrodynamic models only address ideal scenarios for speed and acceleration and do not encompass the variable speeds encompassed in an AUV mission [4]. Efforts to achieve a greater understanding of CTTs and their effects on the hydrodynamics of AUVs have been conducted by [4] [9]. This research was largely to deduce how the cross tunnel thrusters affected the hydrodynamic coefficients of previous models. [32] researched the effects of cross tunnel thruster efficiency due to increasing speed.

Research efforts for CTTs have largely consisted of zero speed, constant speed, or constant

acceleration; however, they do not encompass all speed and acceleration ranges used in a full AUV mission profile. Ian Taylor conducted research to develop an accurate variable speed model of an AUV using CTTs. This research provided a 6 DOF model that simulates the REMUS vehicle at various speeds [4]. Taylor's research also investigates the effect of slow speeds on control surface efficiency, which will be further discussed in this section.

4.2.1 Control Problem

In determining the proper control of the REMUS vehicle, an important control transition must be addressed. In his development of a variable speed hydrodynamic model using CTTs, Taylor discovered that Coefficient of Drag (C_d) had the greatest impact on the remaining hydrodynamic coefficients. Though (C_d) varies depending on vehicle size and speed, Taylor's experimental field testing revealed (C_d) remained approximately constant over the majority of operating speeds. Investigation into the Reynolds number to determine when flow over REMUS became turbulent revealed that flow remains turbulent while speed is over 0.3 m/s [4]. Recall that in implementing the RIN portion of our scenario, the REMUS vehicle will have to transition to the camera sensor to identify the AprilTag. The complete navigation profile of RIN encompasses REMUS speeds where the *Re* is turbulent and laminar.

4.2.2 Control Transition

During the beginning phases of the RIN navigation, the REMUS vehicle will use aft thrust as its sole source of propulsion and the aft fins as its depth and heading control surfaces. As the vehicle approaches the object of interest, it will transition to close in control (CIC) where the vehicle will switch to its short-range sensor, the Watec camera, in order to identify a known feature, the AprilTag. This transition of flow over the vehicle from turbulent flow to laminar flow at around 0.3 m/s will encompass a time where control via aft thrust will no longer be effective due to the physical characteristics of hydrodynamic flow over the vehicle. The REMUS vehicle will have to shift its primary means of control from aft thrust to using the horizontal and vertical CTTs to control the pose of the vehicle.

To address this transition, hover control is initiated via a series of waypoints closely spaced together to dictate the desired pose of the vehicle rather than the desired speed. In order to

ensure the vehicle behaves as directed by the waypoints, the vertical and horizontal CTTs were tuned to maintain pitch and yaw at a hover position.

4.3 Summary

This chapter outlined the major components of the H/SIL architecture that support a simulation framework for rapid SLAM development to include the CAVR architecture used by NPS researchers. For the RIN scenario, a control problem and its solution were addressed and how they will be implemented in simulation.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Results

The focus of this thesis was to present an autonomous solution to the UXO problem space and a platform to evaluate this solution and other AUV scenarios. This chapter focuses on the performance of the REMUS model within the UUV-SIM ocean environment and determine what improvements can be made to the simulation in order to optimize its representation of real-world operations. A summary of a PID controller and the REMUS model in simulation is described prior to discussing results. I first present a comparative analysis of a depth and pitch controller in simulation to similar controllers tested in a freshwater tank. Next, I discuss a comparative analysis of depth, pitch, and heading controllers in simulation compared to a saltwater hover conducted by the NPS REMUS AUV with horizontal and vertical CTTs in the Monterey Bay, CA. This chapter concludes with a "proof of concept" demonstration of an AUV Image-based SLAM simulation that encompasses all of the concepts presented in this thesis: SLAM, undersea imaging, and a H/SIL simulation framework.

5.1 Background

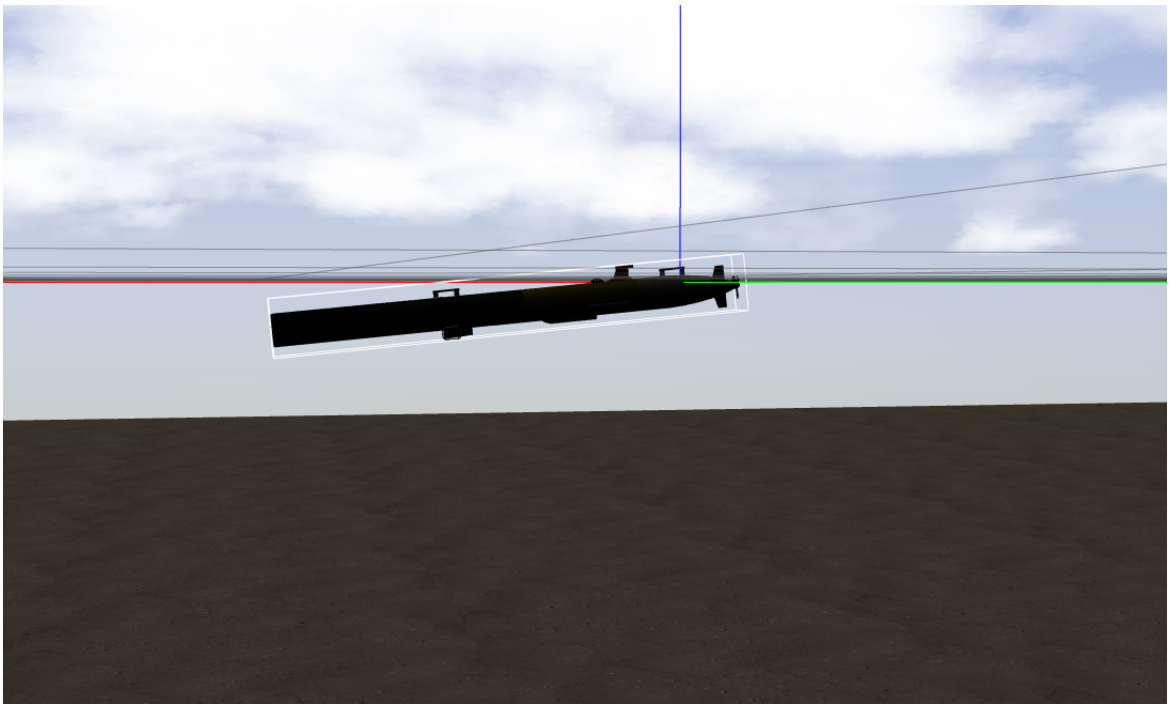
Proportional Integral Derivative (PID) control is a classical control technique that uses feedback to minimize the error between a desired and actual signal. Equation 5.1 summarizes a PID controller.

$$K_P e + K_I \int e dt + K_d \frac{de}{dt} \tag{5.1}$$

Consisting of three parts, a PID controller takes the error input and multiplies it with its gains, then sums the components resulting in the controller output. The proportional controller (K_P) is responsible for reducing the rise time and helping to eliminate steady-state error. The integral controller (K_I) has the effect of eliminating steady-state error but can, however, negatively affect the transient response. The derivative controller (K_D) will increase the stability of the system, reduce overshoot and improve transient response [33].

As is standard with PID control, the tuning of the controller gains was completed through field testing. First, the (K_P) was tuned to achieve an appropriate response time. Once complete, the (K_I) was tuned to minimize the error between the desired and actual signals. Finally, the (K_D) was adjusted to minimize steady-state error.

5.1.1 REMUS Model



REMUS model used to conduct experimentation within the Gazebo/UUV-Sim simulation.

Figure 5.1. REMUS Model

To properly compare the performance of REMUS in simulation and in-water testing, it is important to understand the REMUS model within UUV-SIM and how it was developed. Figure 5.1 displays REMUS within the simulation environment. The REMUS 3-d visualization is a six degree of freedom model consisting of a combination of smaller models, known as meshes, in the form of a digital asset exchange (DAE) model. The components of the REMUS model include the hull, pitch and rudder fins, horizontal and vertical CTT propellers, and the aft propeller. These components are developed with their associated

hydrodynamic coefficients, but there is a distinct difference between the CTT thrusters and other actuators.

The pitch and rudder fins and the aft propeller are developed as actuators, and these components move within the simulation. For example, a forward thrust sent to the aft propeller is executed as propeller rotations within simulation based on the design properties of the propeller. The CTTs are not modeled as moving actuators, rather forces that act on REMUS at the point of the thruster location. These forces were measured by the previous theses [4] [9].

5.2 UUV-SIM and Fresh Water Tank Comparison

Using the software architecture described in Chapter 4, together with the Matrix Laboratory (MATLAB) ROS Toolbox, PID controllers were developed and tuned to both maintain a 4-meter depth and a 5-degree pitch within the UUV-SIM simulation. Separately, the freshwater tank experiment was conducted by developing depth and pitch controllers for REMUS to hover at .65-meters with a 5-degree pitch.[2]

5.2.1 Results

Figures 5.2 and 5.3 demonstrate the REMUS model floating on the surface and descending to the goal depth. When tuning the PID controllers, special consideration was given to initialize the AUV on the surface and ensure that it floated flat on the surface. This was necessary because of inaccuracies in the AUV model dynamics within UUV-SIM while on surface and could result in the AUV having unrealistic overly positive or negative pitches while in a static position.

[2]Further testing in a larger tank allowing for similar depth requirements was not completed due to operational restrictions associated with the COVID-19 pandemic.

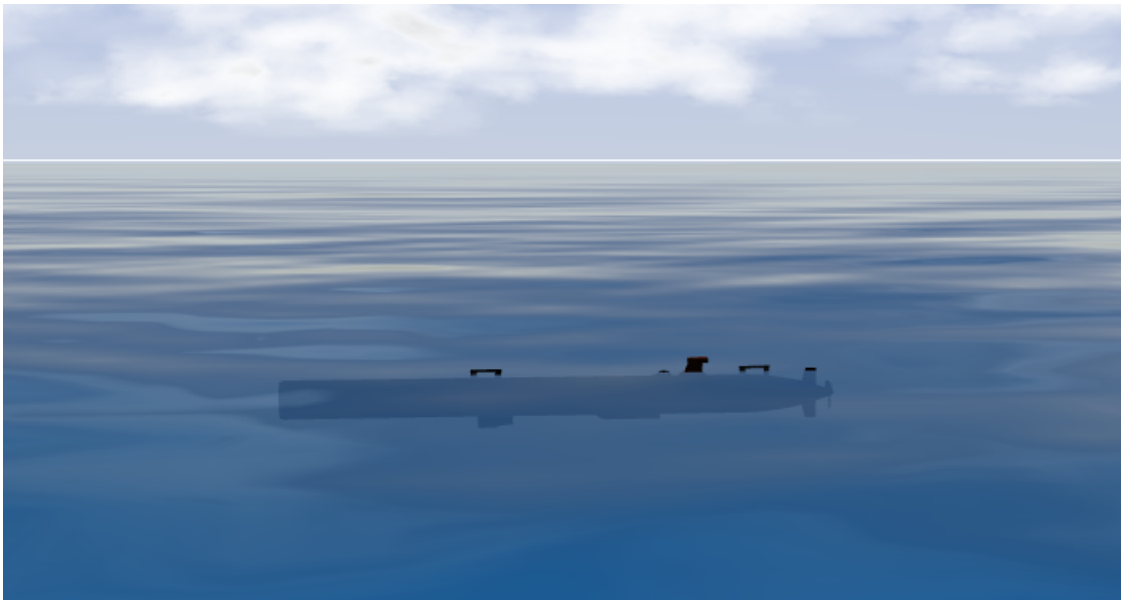


Figure 5.2. REMUS Vehicle on Surface in Simulation

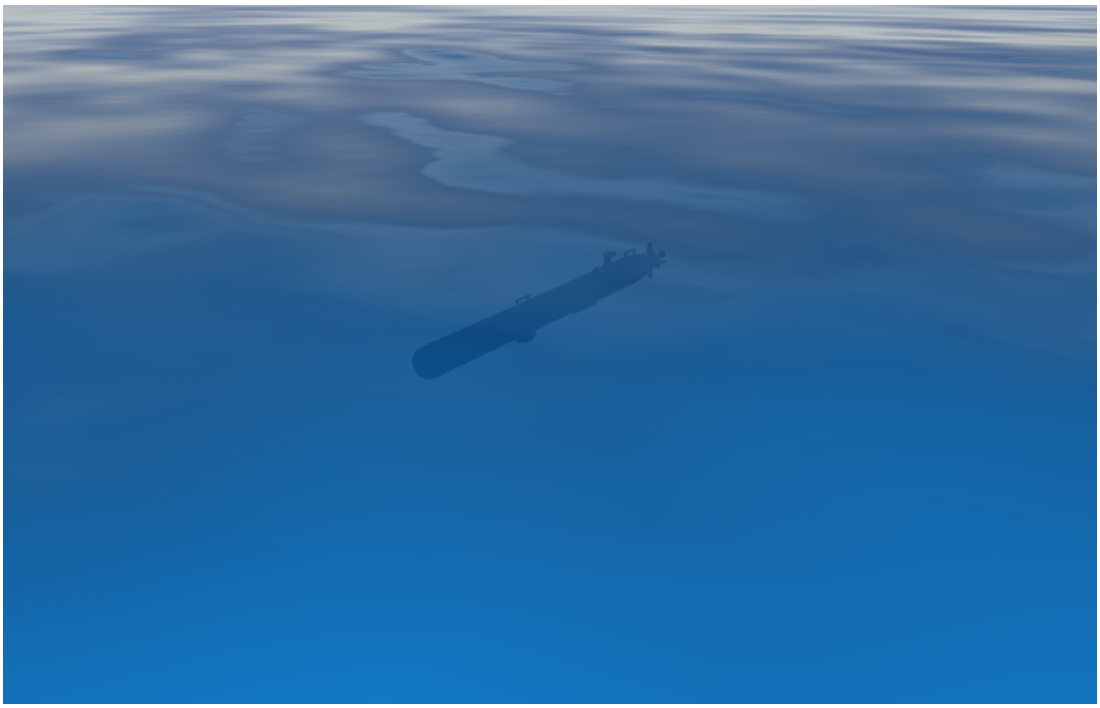
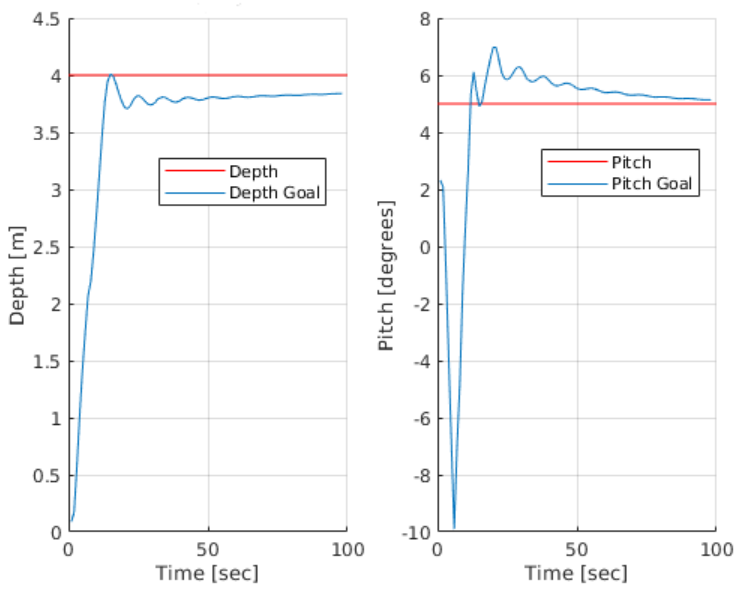


Figure 5.3. REMUS Vehicle Descending in Simulation

Table 5.1. Depth Tuning Comparison

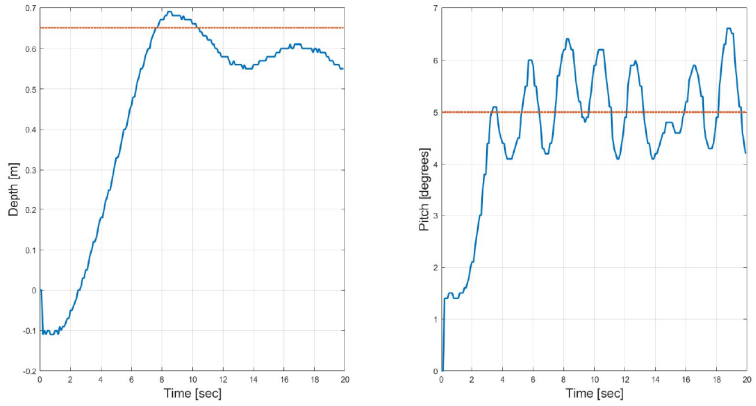
UUV-SIM and Freshwater Tank Controller Comparison			
Tune Goal	Kp	Ki	Kd
UUV-SIM Controller Values			
Depth	1000	5	30
Pitch	120	0	5
Freshwater Tank Controller Values			
Depth	1500	1000	500
Pitch	75	200	175

Table 5.1 displays the final gains established in UUV-SIM and the freshwater tank experiment. Note that the freshwater environment required stronger gain values in almost all components.



UUV-SIM performance with PID controller gains set to $K_d = 1000$, $K_i = 5$, and $K_d = 30$ for depth and $K_d = 120$, $K_i = 0$, and $K_d = 5$ for pitch.

Figure 5.4. UUV-SIM PID Performance



Freshwater tank performance with PIDcontroller gains set to $K_d = 1500$, $K_i = 1000$, and $K_d = 500$ for depth and $K_d = 75$, $K_i = 200$, and $K_d = 175$ for pitch.

Figure 5.5. Freshwater PID Performance

Figures 5.4 and 5.5 show the comparison between the depth and pitch controller performance for both the UUV-SIM simulation and the freshwater tank experiment. Due to limitations of the freshwater tank dimensions (approximately 6 m length x 6 m width x 2 m depth), REMUS was controlled to a limited depth of 0.65 meters but the same pitch as UUV-SIM.

5.2.2 Controller Gain Comparison

Table 5.1 shows the PID coefficients for the tuned depth and pitch controllers from both experiments. The most notable difference between coefficients used in simulation and those in the freshwater tank is the stronger tuning requirements necessary in freshwater. Prior to each mission, a checklist includes a buoyancy test to ensure neutral buoyancy in the water, so the REMUS experiences minimal effects on mobility due to positive or negative buoyancy. Factors that affect buoyancy include saltwater vs. freshwater, vehicle configuration, and the specific salinity of the water on that testing day. Normally the vehicle is ballasted at 1/4 pound positively buoyant.

Figure 5.6. REMUS Outfitted With Flotation

REMUS is designed to operate in saltwater and requires added weight to ensure its near-neutral buoyancy. Conversely, for freshwater operations such as tank testing, REMUS is negatively buoyant and requires added flotation to remain neutrally buoyant. Figure 5.6 displays positive flotation attached to REMUS and the location of the weight chute used to add weight as needed.

The UUV-SIM underwater world environment is configured to represent a saltwater ocean environment with the REMUS configured to be slightly positively buoyant. The added flotation to REMUS in freshwater was a rough approximation, and the changes in the buoyancy of REMUS are unknown. In this particular experiment, the added flotation created a significant buoyant force to be overcome by REMUS. When leaving the surface, REMUS manifests greater PID coefficients specifically in its depth controller because REMUS is working harder to maintain depth in freshwater.

5.2.3 Depth Control

Figure 5.7 shows a side-by-side comparison of the depth controllers developed in simulation and a freshwater tank, respectively. As shown, both controllers reach a steady-state depth slightly below their goal depths of 4 meters and .65 meters. The UUV-SIM depth controller exhibits gentle oscillations and a steady-state error. The controller in the tank experiment displays greater oscillations throughout the investigation once it reaches its goal depth. An

important factor when comparing these controllers is the limitations within the experimental tank used at the NPS CAVR facility. The tank has an approximate dimension of 6 meters in length, 6 meters in width, and 2 meters in depth. The limited depth makes it difficult for REMUS to correct itself before reaching the bottom of the tank or returning to the surface.

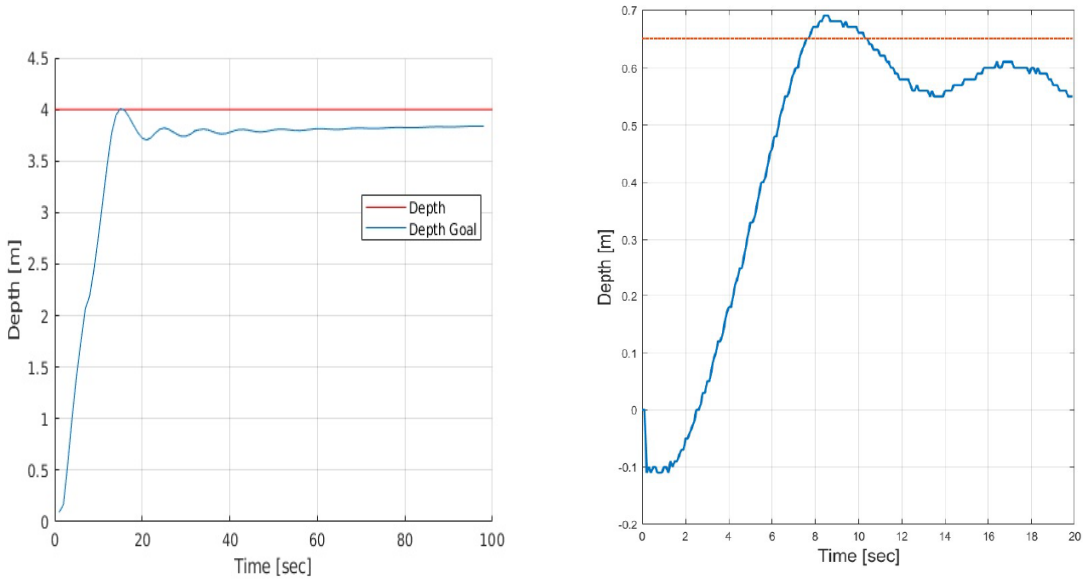


Figure 5.7. UUV-SIM vs. Tank Experiment Depth Performance

When testing in the fresh water tank, an essential component of obtaining the desired depth is overcoming the buoyant force of the vehicle with the added flotation. As shown in Table 5.1 with the largest value coefficient being the proportional gain and again in Figure 5.7 a significant amount of thrust from the vertical cross tunnel thrusters (VCTT) is required to overcome the buoyant force of REMUS and flotation on the surface. This large upward thrust emitted from the forward and aft VCTTs works against REMUS as it quickly approaches and passes its goal depth of .65 meters and must overcorrect to avoid reaching the bottom of the tank. This series of overshooting and undershooting continues throughout the experiment. Further testing in deeper fresh and saltwater tanks is appropriate for comparing the similarities and differences of the UUV-SIM and freshwater tank experiments.

Beyond the limitations of the freshwater tank, both depth controllers show a similar trend of a strong proportional gain required to leave the surface and minor adjustments at depth

to maintain a steady hover. Both controllers display similar gradual oscillations and quickly reach steady state slightly above the intended depth.

5.2.4 Pitch Control

Figure 5.8 compares the pitch controllers generated in UUV-Sim and the freshwater tank. Similar to the depth controllers, the pitch controllers exhibit identical trends in obtaining the desired command pitch. In particular, leaving the surface requires a significant thrust from the forward VCTT, resulting in a sizeable positive pitch rate and REMUS obtaining a "nose down" orientation greater than the pitch goal of 5 degrees. This is then corrected prior to reaching the goal depth where both thrusters must lower their RPMs. This pitch correction was particularly evident in the tuning of the UUV-SIM depth and pitch controllers, where the Gazebo display clearly showed REMUS correcting itself before reaching the goal depth.

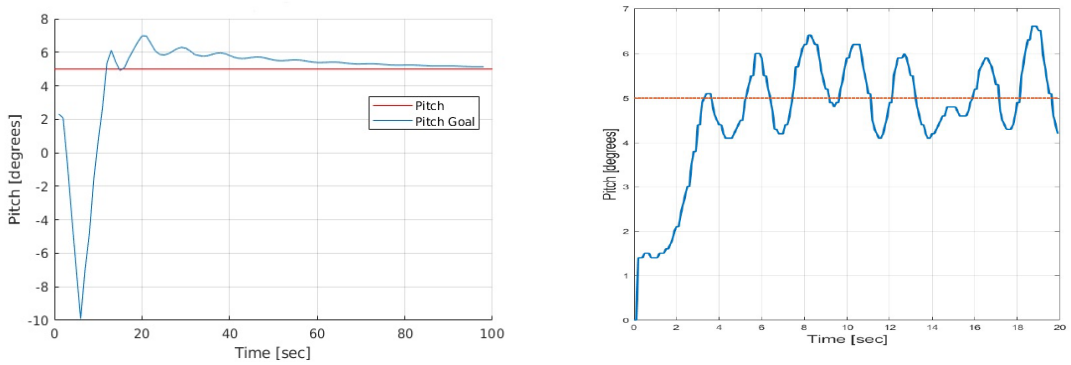
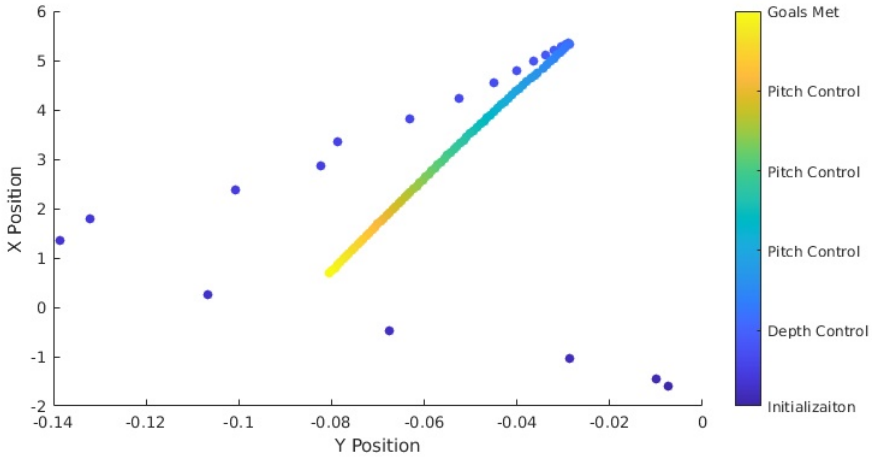


Figure 5.8. UUV-SIM vs. Tank Experiment Pitch Performance

When developing the depth and pitch controllers in simulation, strong proportional gain values would often result in the forward VCTT, overpowering the aft VCTT and causing the vehicle to lose pitch control. This will result in a vertical orientation, and REMUS will return to the surface. This example indicates the model's limitations, which is discussed further within the analysis section of this thesis. Additionally, tuning the depth controller within UUV-SIM revealed that a value of 500 RPMs from the forward thruster was required to leave the surface while only 300 RPMs were needed for the aft thruster. The center of buoyancy (COB) is closer to the forward thrusters, which results in a moment arm. We would expect the forward 500 RPMs to be greater than the aft. This is a confirmation of the

model. Conversely, we also see the vehicle pitching forward when going down in depth. This is an example of a limitation in the model.

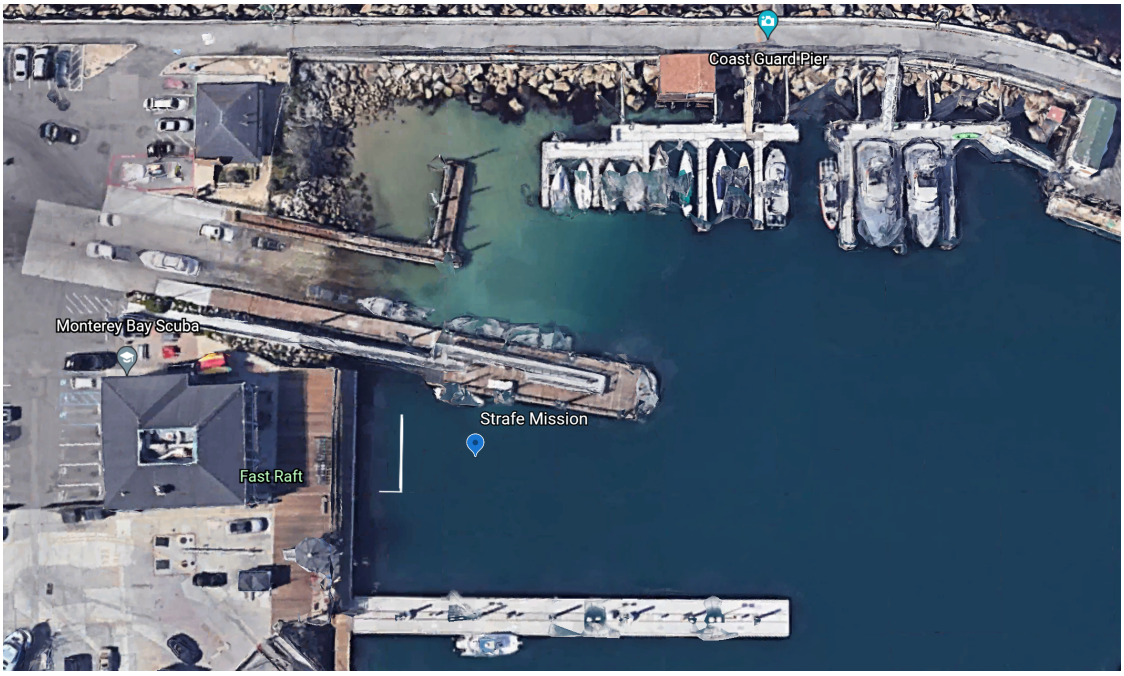
Figure 5.9 demonstrates the slip associated with the pitch control graph for UUV-SIM in Figure 5.8. As discussed, the RPMs required to leave the surface differs between the forward and aft VCTTs. As shown in Figure 5.3, the vehicle takes an initial positive pitch attitude to leave the surface, causing it to translate in the positive X direction. Next, an increased aft VCTT thrust is needed to reach the desired pitch requirement, which moves the vehicle back towards the Y-axis, ultimately settling at the desired pitch and causing slight "backward drift." This subtle translation occurring during pitch control illuminates the effectiveness of our model and simulation in reflecting minor translations expected of REMUS.



A position graph demonstrating slip associated with pitch control. System initialization occurs denoted in purple where REMUS initially drifts. Upon commencement of depth control shown in blue REMUS slips forward until reaching goal depth, where pitch control then initiates shown in light blue, causing a steady "backward drift" with small gradual position changes. Lighter colors demonstrate the translation associated with maintaining a slight pitch while at depth.

Figure 5.9. Slip Associated with Depth and Pitch Control

5.3 UUV-SIM and Open Water Hover



Overhead Google Earth image of Monterey Bay Coast Guard Pier where the strafe experimentation was conducted.

Figure 5.10. Strafe Experiment Location

Comparing the REMUS model in the Gazebo/UUV-SIM environment with data from the small, freshwater test tank has limited utility. A more useful comparison is with the REMUS operations in the ocean. Testing of the REMUS hover capability was conducted in the Monterey Bay inner harbor. This section compares the performance of the two. Similar to section 5.2, the controllers in UUV-SIM were tuned to maintain a hover at 4m; however, this time the pitch command was set for zero degrees. This section also compares the capability of REMUS to control to heading while at the same time conducting a strafe using horizontal cross tunnel thrusters (HCTT).

5.3.1 Results

The results of the depth and pitch controllers in this section are of the REMUS model in the UUV-SIM simulation controlling to four meters and zero-degrees pitch and a saltwater

operation of REMUS conducting a hover in the inner harbor of Monterey Bay at a depth of 1.5 meters and zero degrees pitch.

REMUS hover to 4 meters and zero degrees pitch in simulation.

Figure 5.11. REMUS Hover Position

Figure 5.11 shows the REMUS model at a depth of four meters and a zero-degree pitch. The associated VCTT values for reaching a four meter depth in the simulation are shown within the vertical thruster image of Figure 5.12. In attaining the depth goal, we notice that the forward thruster is working harder than the aft thruster shown in greater RPMs. This is a confirmation of the accuracy of the model, which is discussed in the thruster and analysis subsections of this section.

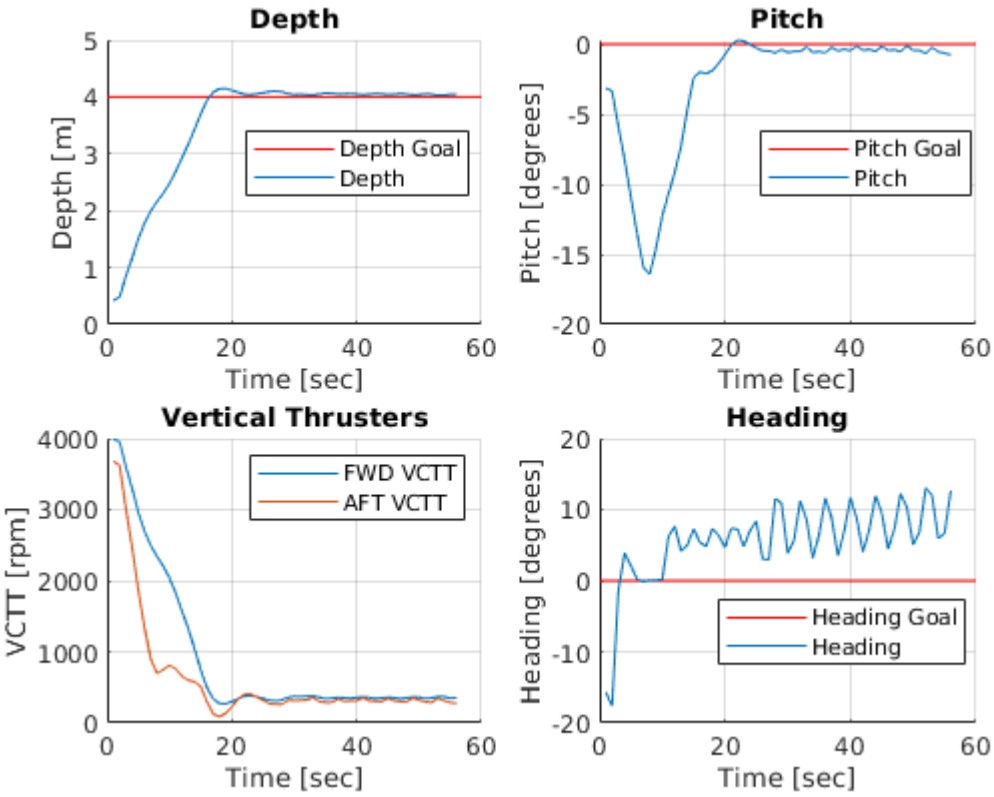


Figure 5.12. UUV-SIM Depth, Pitch, VCTTs and Heading

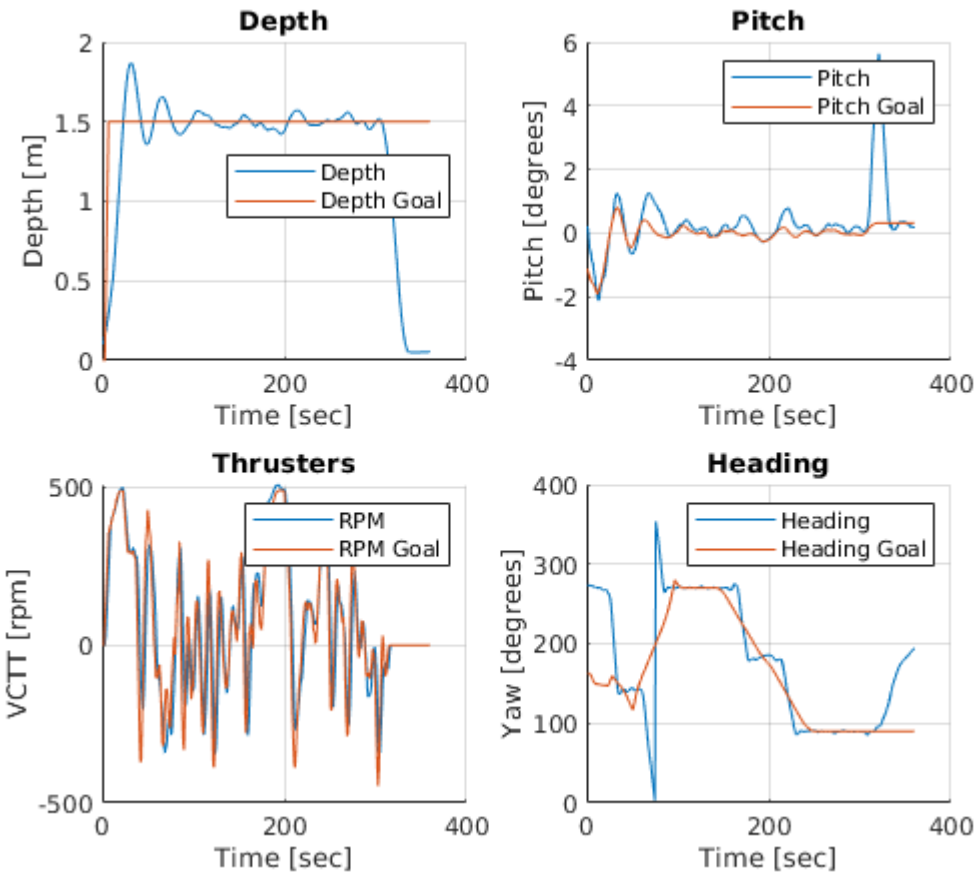


Figure 5.13. Open Water Depth, Pitch, VCTTs and Heading

Figures 5.12 and 5.13 show the results of both experiments in simulation and the saltwater environment. The open water hover encompasses six minutes of a strafe experiment where REMUS was given several strafe commands; however, for comparative analysis, we will focus only on a section that encompasses 50 seconds where the vehicle is strafing in one direction. This experiment was replicated within UUV-SIM to compare the performance of the model to that seen in real-life operations.

5.3.2 Depth

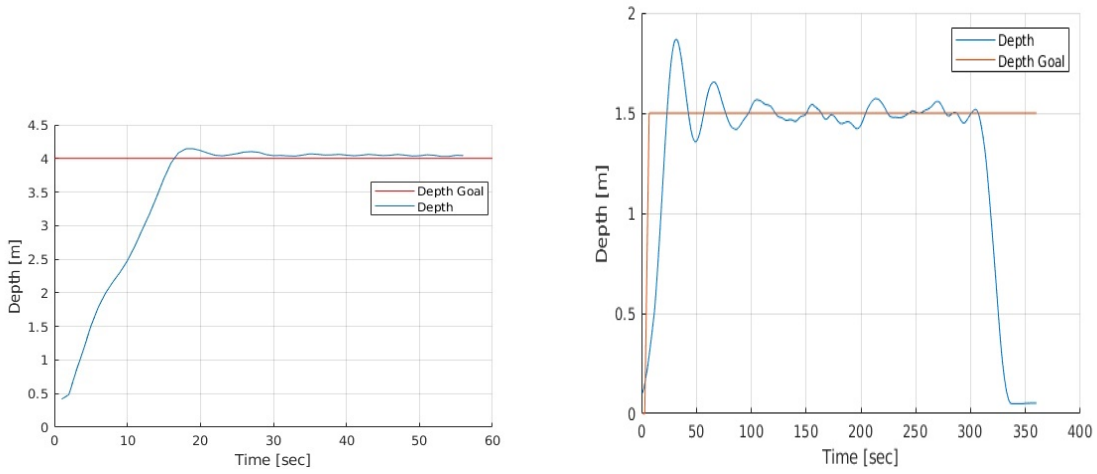


Figure 5.14. UUV-SIM vs. Open Water Depth Performance

Figure 5.14 shows a side-by-side comparison of the depth controllers developed in simulation and saltwater operations. Similar trends from the tank experiment show that the controllers have a slight steady-state error with respect to their goal depths of 4 meters and 1.5 meters. Removing the requirement of maintaining a 5-degree pitch provides for a smooth transition from surface to depth.

5.3.3 Pitch

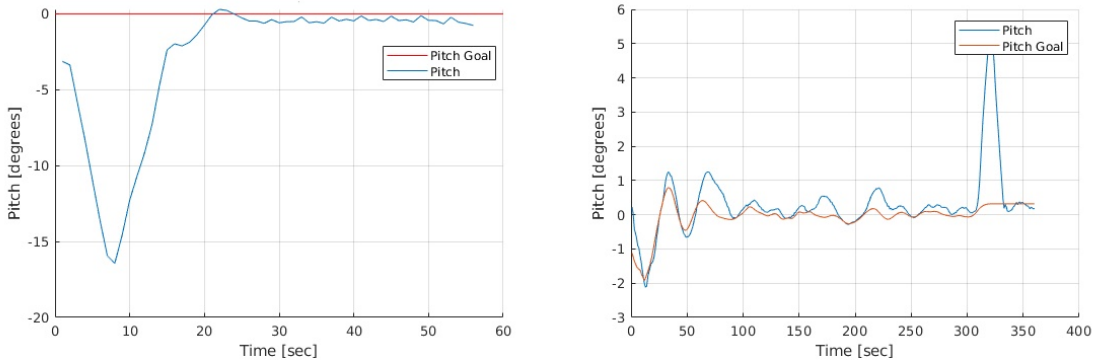


Figure 5.15. UUV-SIM vs. Open Water Pitch Performance

A pitch comparison between UUV-SIM and saltwater operations shows similar results. Hovering at a zero-degree pitch allows for REMUS both in simulation and saltwater to leave the surface smoothly and quickly control pitch.

5.3.4 Thrusters

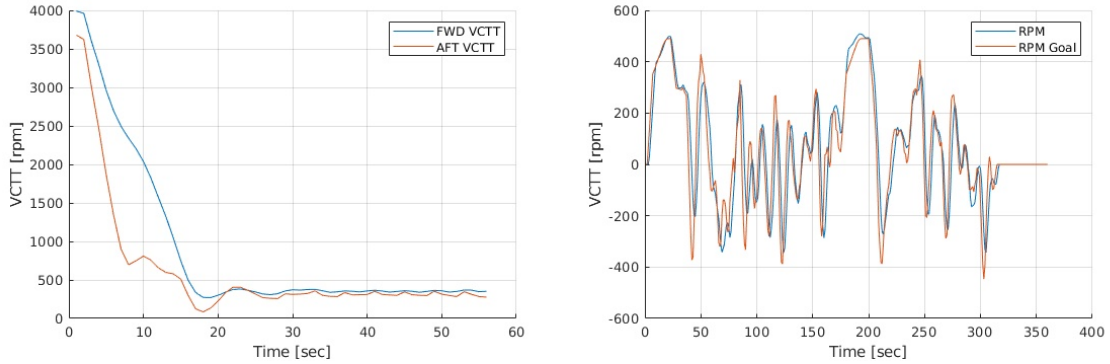
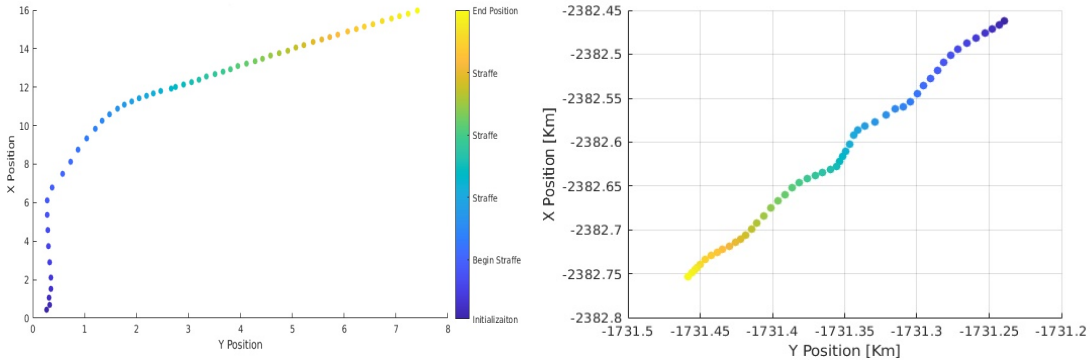


Figure 5.16. UUV-Sim vs. Tank Experiment Depth Performance

The greatest difference between the two experiments is the requirement on the thrusters in simulation and real-world operations. As shown in Figure 5.16, the model reaches a steady state at a RPM value of 400 RPM while the saltwater hover maintains a steady-state range of RPMs between -400 and 500. The hover is performed at a specific depth and pitch but does not require inputs from the aft thruster, resulting in a mission that operates in the control problem range. The flow of information within these systems previously discussed in Section 4.1.5 provides further insight into the differences between simulation and saltwater operation. The depth and pitch controllers calculate the necessary thrust required to reach their goals and provide a signal to the forward and aft thrusters. With a maximum thrust of 5000 RPMs, a high-level command such as depth is converted to a low-level command of RPM to the VCTTs. This command is sent as an electrical signal to the electric motors. The difference between simulation and real-world operations is, as previously discussed in Section 2.2, an RPM command of 3000 can manifest as a range of values from 2950 to 3050 RPM in the real world. In a simulation, an exact RPM of 3000 is achieved. The resulting differences between the forward and aft controllers can lead to oscillations within the pitch of ±5 degrees that are constantly corrected through the PID controller.

5.3.5 Heading

A strafe experiment was conducted within UUV-SIM to conduct a comparative analysis of heading control. Then it was compared to the open water results of a strafe mission performed in the Monterey Bay. The experiment consisted of REMUS maintaining a zero degree pitch and 4m of depth while strafing for 50 seconds.



System initialization is annotated in purple and the strafe mission is performed in colors ranging from dark blue to yellow.

Figure 5.17. UUV-SIM vs. Open Water Strafe

Figure 5.17 shows the UUV-SIM strafe mission compared alongside an open-water strafe. The image on the left is of UUV-SIM. During the first 10 seconds of the UUV-SIM mission, REMUS is only responsible for reaching its set depth, pitch, and heading commands with heading set to zero degrees. Once the 10 seconds elapse, REMUS is then commanded to strafe "east" along the positive Y-axis. In the corresponding open water experiment, the vehicle is given a command to strafe while maintaining 270 degrees for 50 seconds. The vehicle's initial position is in the top right corner of the image, and the vehicle conducts a 50-second strafe completing its mission in the bottom left corner.

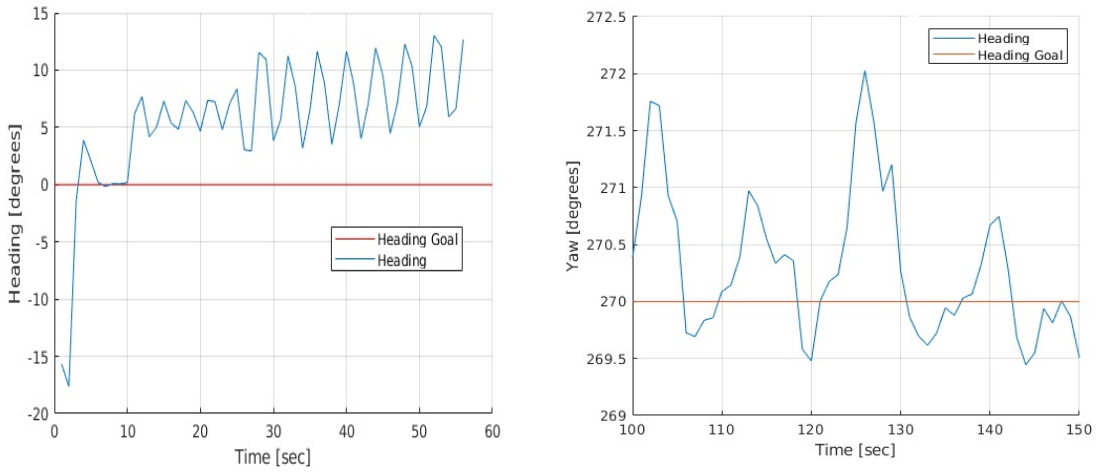


Figure 5.18. UUV-SIM vs. Open Water Heading Performance

Figure 5.18 shows the heading performance of REMUS within UUV-SIM and open water operations. The open water heading performance is extracted from the heading plot shown in 5.18 between 100 and 150 seconds where the vehicle is strafing in one direction. Both plots show similarities in how the vehicle performs its strafing action by creating alternating moments along with the vehicle's COB. The HCTTs are not placed equidistant to REMUS COB. For this reason, an equal thrust between the forward and aft HCTTs is insufficient for performing a strafe. The vehicle uses forced moments about the COB to "crab" horizontally. This is seen in both UUV-SIM and open-water operations in the oscillation of the vehicles heading. As shown, the vehicle oscillates between a range of values relative to the set heading. Within UUV-SIM, the heading control has difficulty maintaining the goal heading and performs increasing alternating moments to maintain the goal. As time progresses, the range of heading values also increases and thus increases the difference between the heading goal and heading. Open-water strafing operations demonstrated a better performance in maintaining heading during the strafe. As shown, oscillations and heading error do not increase over time as compared to UUV-SIM.

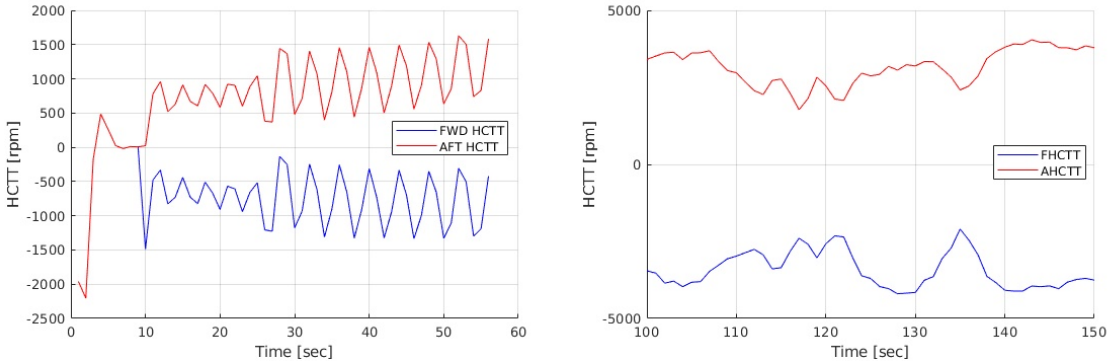


Figure 5.19. UUV-SIM vs. Open Water HCTT Performance

A plot of the HCTTs for both UUV-SIM and open water testing further demonstrate their respective performances. As shown in Figure 5.18, the UUV-SIM HCTT RPM values increase over the experiment causing for greater oscillations and increasing error over time. Though open-water operations outperformed UUV-SIM, some similarities demonstrate the accuracy of our model. In creating a moment, the forward HCTT is located closer to the COB than the aft HCTT resulting in more RPMs required to create the moment necessary to perform an even strafe than would be required by the aft HCTT. This is consistent with the previous VCTT comparison used in-depth control and further illustrated the model's accuracy. This difference in RPMs is further complicated by the gains of the PID controller that increases the difficulty of the vehicle to conduct a smooth strafe with large differences in RPM values between the forward and aft thrusters. This control is a problem that, when implemented within UUV-SIM requires a more sophisticated controller to ensure the forward and aft moment requirements of the vehicle are met as well as maintaining an even strafe. Both experiments demonstrate that the forward thrusters require greater thrust to produce a moment than the aft thrusters.

5.3.6 Analysis

Comparative analysis of REMUS within UUV-SIM with both fresh water and saltwater testing gives insight into the accuracy and limitations of the model. Beginning with accuracy, several behaviors of the REMUS model indicate that the model performs as expected in the real world. The parameters of the REMUS model itself are slightly positively buoyant in the water. This is accurate of real-world operations, as previously explained. Dependent on the environment and the salinity of the water, REMUS requires weight or flotation to remain near neutrally buoyant; however, it will not be perfectly neutrally buoyant in the water. The RPM requirements of the forward and aft VCTT and HCTTs accurately reflect REMUS COB. We notice that the forward thruster requires greater RPM values to reach the bottom than the aft thruster during depth and pitch control operations. We see this again during strafe control operations. The forward thruster requires greater RPM values than the aft thruster to maintain an even strafe. The consistency between forward and aft VCTT and HCTTs confirm that the model is performing properly. Similarly, vertical thruster values obtained within simulation correlate to those found in real-world testing. An observation of vertical thrusters at steady state, as shown in Section 5.3.4, demonstrate similar RPM ranges when conducting hover operations of ±500RPM.

Another confirmation of the model is the slip exhibited by the vehicle during control experimentation. During vehicle system initialization, especially the initial portion of depth, pitch, and strafe control, REMUS is subject to slip based on the UUV-SIM environment and any discrepancies in forces acting on REMUS while it commences its mission. As shown in Figure 5.9, REMUS is subject to slip in various directions as it initially conducts a depth-only maneuver and transitions to depth and pitch control where its slip decreases, and it becomes stationary.

In conducting experiments within UUV-SIM, there were limitations of the model that require further investigation. Some of the behaviors exhibited by the model were inconsistent with what we would see in the real world. An inaccuracy within the simulation occurred while REMUS was on the surface. The vehicle would often maintain a “nose down” attitude, as shown in Figure 5.1, that is inaccurate of what we would expect to see in salt water. This presented problems during testing, particularly in poor data, if the vehicle did not have a proper orientation at the beginning of an experiment. In building and testing our

PID controller, the model would often become unstable and quickly gain or lose depth. A particular inaccuracy was displayed when REMUS would surface unexpectedly, and half of the vehicle would be out of the water. This would never happen in real-world operations and requires further investigation.

A particular limitation of the model was discovered upon implementing strafe control. As shown in the testing in Monterey Bay, (Figure 5.17) REMUS is capable of conducting a side strafe using HCTTs. When attempting to replicate this in simulation, we could not achieve a true side strafe using HCTTs that did not include heavy slip perpendicular to the goal direction. This is shown on the simulation portion of the image in Figure 5.17. Extensive testing to include manual manipulation of the HCTTs to "force" a true side strafe using HCTTs was not fruitful, and slip along the opposing axis was always present.

Further research is required to understand the reason for this limitation. Initial hypotheses include limitations within the Fossen hydrodynamic model using HCTTs at low speeds and limits in implementations of the hydrodynamic model within UUV-SIM at low speeds. Another reason could be the limitations within the modeling of cross tunnel thruster forces in UUV-SIM as previously discussed when describing the REMUS model and finally, the limitations of the Fossen hydrodynamic model while on the surface or near-surface depths. This observed limitation is beyond the scope of this thesis, but it should be investigated further to improve the fidelity and utility of the simulation model.

5.4 Simulation

Figure 5.20 shows the GAZEBO environment providing a pictorial representation of our physics-based simulation of the REMUS mission. The REMUS model shown is a three-dimensional UUV-SIM DAE file allowing the specification of NPS REMUS hydrodynamic coefficients for realistic behavior. The image window displayed on the right side of the image is a simulated camera capturing images of the simulated ocean floor as a ROS image stream. Other processes not displayed in the simulation are CAVR processes, which navigate the vehicle. As discussed, the H/SIL architecture uses the same processes for in-water operation. The SLAM algorithm iSAM2 is using a USBL transponder for long-range position estimation and the mounted camera for low-speed position estimation. A heading error has been introduced to the vehicle, which is corrected. It permits the AUV to navigate over the tag accurately. Finally, a computer vision algorithm for AprilTags is working to identify the AprilTag.

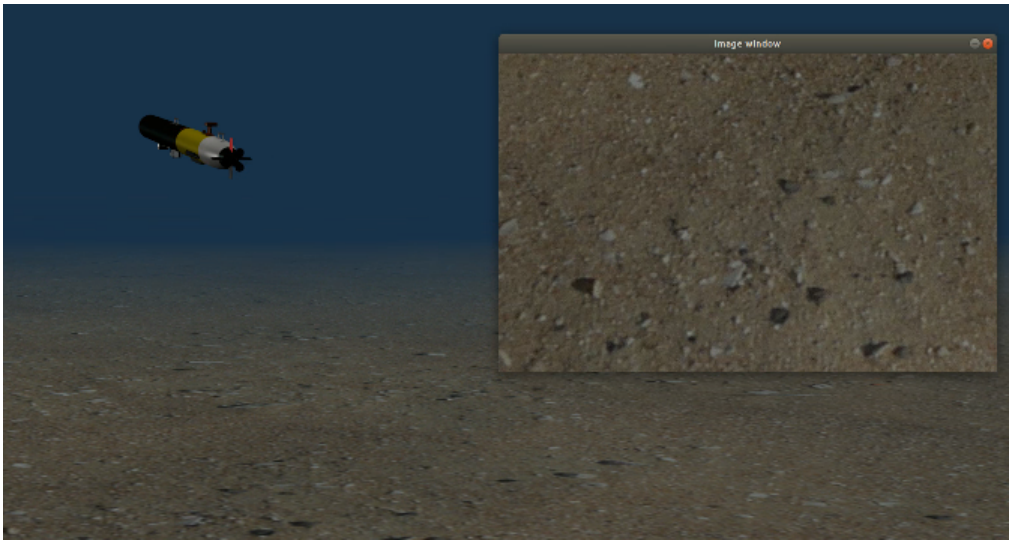


Figure 5.20. Image Capture of Working Simulation

5.4.1 AprilTags in Simulation

As the vehicle enters the area of interest, it transitions to slow speed control using pose commands and the vertical and horizontal CTTs to control its pose while it passes over the AprilTag. As previously discussed in Chapter 3, the primary sources of error when locating an AprilTag are camera distance, angle of the camera center, and yaw angle.

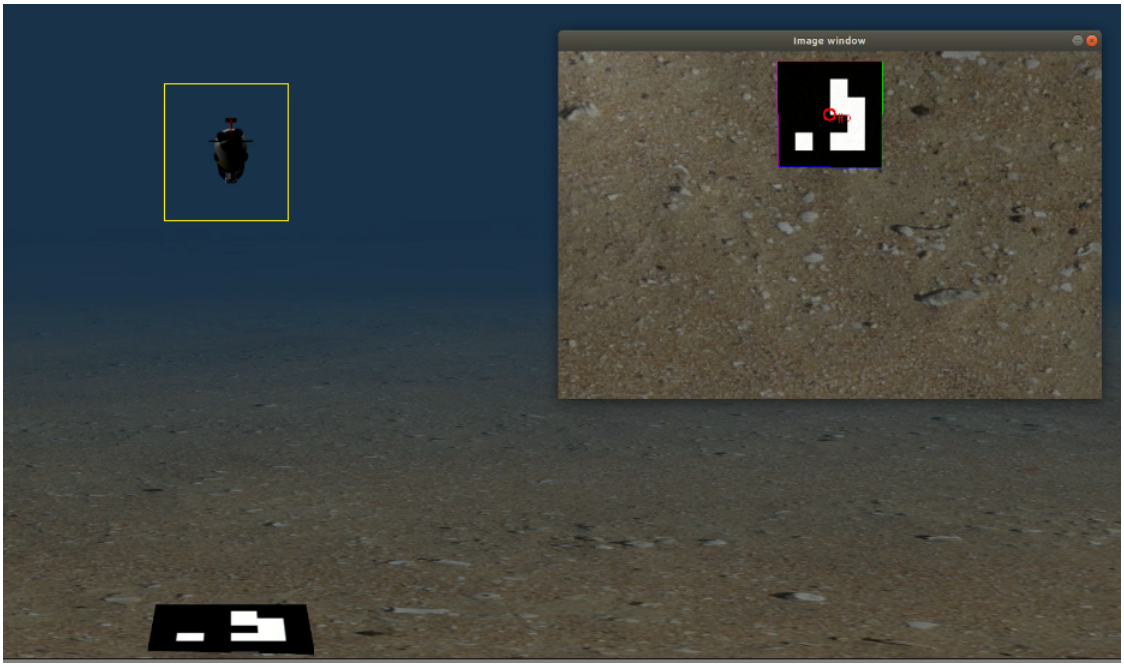
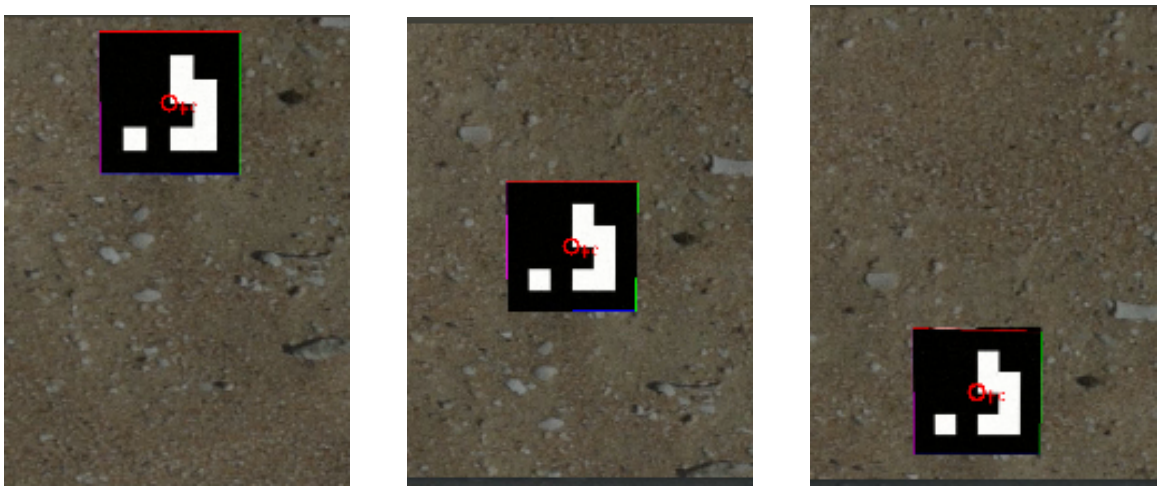


Image capture of AprilTag identification showing the spatial relationship between the vehicle and the AprilTag as well as the onboard camera image.

Figure 5.21. AprilTag Identification

Pose commands are used to minimize any changes in pitch, roll, or yaw of the camera and ensure steady detection while the tag is in the field of view. Figure 5.22 shows the marker as it enters and exits the field of view. At its optimal detection point, when the tag is in the center of the camera's field of view, the tag is generally equidistant from both the horizontal and vertical edges of the camera's field of view. This ensures the most accuracy of the decoded information received from the AprilTag, which is then used for further position estimation.



Three sequential seafloor AprilTag images demonstrating the AprilTag is entering and exiting the onboard camera's field of view. The AprilTag remains in the horizontal center of the field of view. The optimal image capture is when the AprilTag is in the center of the horizontal and vertical field of view.

Figure 5.22. AprilTag within Field of View

5.4.2 Discussion

The image-based SLAM simulation enables control of the vehicle, marker, and environmental conditions that can affect performance. The H/SIL framework allow for evaluating of the processes onboard REMUS and their capability to control to a specific underwater position while assessing the position estimating algorithm. This allows testing through the range of environments that may be present in the real world. For example, water visibility can be adjusted by changing the light or turbidity present in the environment. Similarly, the position estimation algorithm can be improved or degraded to understand its performance and limitations better. This provides the ability to run various simulations through a range of variables to better understand the image-based SLAM performance.

This demonstration lays the foundation for attacking the Navy's UXO problem and can be extended to many problems and mission sets encompassed by the AUV problem space. The next step in evaluating the performance of AUV image-based SLAM will be to continue experimenting in simulation, allowing the vehicle to navigate using the iSAM2 algorithm combined with USBL sensors and capturing images of AprilTags. The simulation frame-

work allows for the flexibility to evaluate each of these components individually and as a combined system. One example of these experiments is increasing the introduced error bias into the iSAM2 algorithm to evaluate its performance as accuracy degrades over time. Another example of the experiment is introducing turbidity and variable lighting into the simulation to observe the performance of marker detection and test the resilience of the undersea imaging process. With respect to the USBL, the transponder position can be manipulated through various simulations to understand the optimal positioning of the sensor and get the most accurate range and bearing measurements for optimal navigation. These future experiments are highlighted to stress the utility of the simulation framework. When the operator is intentionally degrading the sensors and the algorithm, testing is not practical in open-water operations. Many risks are involved in such testing, such as operational safety, risk of damaging or losing equipment, the costs associated with those risks, and logistical requirements of manning, equipment, funding, etc., to support the testing. The simulation framework allows for low-risk, high-reward testing of AUVs that provide insight into their limitations and optimize the time spent out in the field.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 6:
Conclusion

Simulations are crucial for increasing autonomy and enabling systems to improve their functions through iteration and improvement. This thesis provides a starting point and only touches the tip of the iceberg for what is possible within a simulation. The ability to replicate environments and explore these environments and scenarios through models of autonomous vehicles and their sensors removes the classic limitations developers endure, such as cost, resources, logistics, and funding. Furthermore, simulations allow developers to train models and use these models to train and evaluate controls. This tool will enable developers to explore the capabilities of autonomous vehicles, in this case, REMUS, in a safe environment where its undersea performance and limitations can be assessed prior to entry into the water.

Undersea ordnance is an ongoing problem that the US and many countries continue to face. Whether remnant of previous conflicts or actively staged by adversaries, the ability to neutralize undersea mines safely and cost-efficiently presents a capability gap currently in place. Within the Navy, AUVs' capability to effectively navigate with affordable sensors provides a gateway to neutralizing this ordnance. The private sector and education sector can also benefit from the AUV H/SIL architecture. Through UUV-SIM, entities can replicate the real-world environment in a physics-based simulation and use their software in simulation before live testing, which reduces wear and tear on the AUV, minimizes costs, and maximizes the effectiveness of real-world testing.

6.1 Contributions

Autonomy consists of four key components, and this thesis addresses three of the four major components as follows:

- **Position Estimation**: The simulation framework provides a dynamic environment capable of representing the wide range of undersea environments and its features to include images, bathymetry, external signals, and temperature gradients. Navigation by way of image-based SLAM coupled with the iSAM technique provides a method for addressing position estimation within the undersea ordnance scenario presented at the beginning of this thesis.
- **Sensor Processing**: The simulation architecture presented allows for the modeling of sensors to the vehicle's specifications, allowing for sensor processing of the information they provide. The CAVR architecture described, processes the information received from REMUS sensors to include depth, images, and environmental conditions. This thesis specifically addresses the use of underwater imaging through a camera and navigation through a D-USBL transponder.
- **Building an Internal Representation of the Environment**: The simulation framework presented in this thesis provided an architecture that allows the vehicle to sense its environment, record the necessary information to develop a map of its surroundings, and be able to act on it as necessary to complete its mission.
- **Dynamically Planning Routes**: This item is not addressed in this thesis; however, the techniques described within iSAM2 allow for an extension of map building to route planning within that map and using its sensors to navigate along this route.

6.1.1 Simulation Framework

This thesis developed a H/SIL framework for various AUV scenarios and laid the foundation for addressing complex undersea problems with AUVs. The physics-based UUV-SIM provides realistic underwater environments and the associated hydrostatic and hydrodynamic forces and moments AUVs and underwater objects will experience. This powerful tool, coupled with the work completed by Taylor in developing a hydrodynamic model representative of REMUS in variable speed environments, combines to make an accurate real-world simulation environment. The advantages of this framework are that REMUS can

be tested in simulation with reasonable accuracy of what is expected in real-world mission sets. Additionally, the H/SIL capability heavily simplify the transition from simulation to the real world. The same software is used in simulation that the secondary controller will execute on board REMUS. Except for tuning for configuration and environmental factors, REMUS testing in this simulation framework will reduce person-hours on station, equipment operation, and cost, allowing for more efficient real-world experimentation and data collection.

6.1.2 AUV Scenarios

This thesis lays the foundation for an autonomous solution to the recovery, identification, and neutralization of undersea ordnance. It demonstrated the use of the SLAM technique iSAM2 in conjunction with cheap sensors, a camera, and D-USBL array, for implementing image-based SLAM for AUVs. A computationally efficient position estimation algorithm in conjunction with sensors makes for an effective and expendable tool for MCM and EOD operations. Additionally, locating an undersea object and using it as a navigation tool is helpful in the scientific community and the private sector.

6.2 Future Work

As previously discussed, the framework described in this thesis lays the foundation for a variety of work in the AUV space. This section will discuss opportunities for further research to improve the simulation environment. Additionally, I provide suggestions for improvements in image-based SLAM.

6.2.1 Simulation Improvements

Experimentation with the REMUS vehicle discovered deficiencies observed in simulation about the hydrodynamic model. The buoyancy of REMUS proved to be inaccurate in specific scenarios, such as on the surface before an experiment, or if the vehicle were to surface during the mission execution phase, the vehicle dynamics would often be affected for the rest of the mission. Further research should be conducted to address the limitations of a Fossen hydrodynamic model while on the surface. Other limitations in the model

were exposed when REMUS would exhibit behaviors in simulation that we would not see in the real world, such as observing the vehicle oriented vertically with more than fifty percent of the vehicle exposed out of the water. This vehicle's positioning in the water is not possible based on the vehicle's characteristics and requires further investigation. Further investigation should be conducted into performing lateral translation maneuvers using a Fossen hydrodynamic model to quantify the limitations observed in strafe testing. Additionally, further research is required in implementing the proper control technique for lateral translation maneuvers that account for the limitations observed.

6.2.2 Image-Based SLAM

Operational limitations observed during the spring of 2020 to combat the COVID-19 virus limited the amount of testing performed on the Watec camera. Further testing is required on the camera's capabilities both in a controlled tank environment and in the ocean to test the real-world field of view of the camera and its performance with AprilTags. This thesis demonstrated the RIN mission; however, there are opportunities for software improvement to include integrating the latest AprilTags system, AprilTags3, which increases image detection capabilities by improving detection robustness and extending the fiducial marker system to irregularly shaped markers [25]. Additionally, implementation of GTSAM, the latest iteration of the iSAM algorithm family, which provides a user friendly toolbox and offers integration with MATLAB for post-processing data sets [14].

Ultimately, image-based SLAM aims to transition away from fiducial markers and into the stitching of seafloor images into an image mosaic to be used for image correlation and object detection. This simulation architecture is well-suited for testing image-based SLAM using the image mosiac.

List of References

[1] SUPSALV, "US Navy salvage report operations Desert Shield/Desert Storm Vol 1," Report No. T0800-AE-RPT-010/SUPSALV, June 1992 [Online]. Available: https://www.navsea.navy.mil/Home/SUPSALV/00C2-Salvage/Salvage-Publications/

[2] R. D. Christ and R. L. Wernli. *ROV Manual - A User Guide for Remotely Operated Vehicles*, 2nd ed. Elsever, Waltham, MA, USA, 2014 [Online]. Available: https://app.knovel.com/hotlink/khtml/id:kt00UBNEN1/rov-manual-user-guide/expendable-mcm-vehicles

[3] T. I. Fossen, *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Trondheim, Norway: Marine Cybernetics, 2002.

[4] I. Taylor, D. Homer, and E. Bermudez, "Variable Speed Hydrodynamic Model of an AUV Utilizing Cross-Tunnel Thrusters," in *OCEANS 2017 - Aberdeen*, 2017 [Online]. Available: https://calhoun.nps.edu/handle/10945/56183

[5] Sulzberger, G and Bono, J and Manley, R J and Clem, T and Vaizer, L and Holtzapple, R, "Hunting sea mines with UUV-based magnetic and electro-optic sensors," in *OCEANS 2009*, 2009 [Online]. Available: doi:10.23919/OCEANS.2009.5422086

[6] J. C. Hyland and C. M. Smith, "Optimal Reload Strategies for Identify-and-Destroy Missions," in *Detection and Remediation Technologies for Mines and Minelike Targets IX*, 2004. Available: https://doi.org/10.1117/12.562760

[7] J. Čejka, F. Bruno, D. Skarlatos, and F. Liarokapis, "Detecting square markers in underwater environments," *Remote Sensing*, vol. 11, no. 4, Feb 2019 [Online]. doi:http://search.proquest.com/docview/2333922945/.

[8] E. Westman and M. Kaess, "Underwater AprilTag SLAM and Calibration for High Precision Robot Localization," Carnegie Mellon University, Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-18-43, 2018 [Online]. Available: https://www.cs.cmu.edu/~kaess/pub/Westman18tr.pdf

[9] Bermudez, Eric B., "Terminal Homing for Autonomous Underwater Vehicle Docking," M.S. thesis, Dept. of Mechanical and Aerospace Engineering., NPS, Monterey, CA, USA, 2016 [Online]. Available: https://calhoun.nps.edu/handle/10945/49385

[10] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping: Part I," *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 99–110, June 2006. [Online]. doi: https://ieeexplore-ieee-org.libproxy.nps.edu/document/1638022/.

[11] T. Basar. *Control Theory: Twenty-Five Seminal Papers (1932-1981)*. John Wiley Sons, Hoboken, NJ, USA, 2000 [Online]. Available: https://ieeexplore-ieee-org.libproxy.nps.edu/book/5265919

[12] T. Bailey and H. Durrant-Whyte, "Simultaneous Localization and Mapping (SLAM): part II," *IEEE Robotics Automation Magazine*, vol. 13, no. 3, pp. 108–117, September 2006. [Online]. doi: https://ieeexplore-ieee-org.libproxy.nps.edu/document/1678144".

[13] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, 1987 [Online]. Available: doi:10.1109/ROBOT.1987.1087846

[14] F. Dellaert, "Factor Graphs and GTSAM: A Hands-on Introduction." Accessed June 18, 2019 [Online]. Available: https://gtsam.org/get_started/

[15] M. Kaess, "Incremental smoothing and mapping," Ph.D. dissertation, College of Computing, Georgia Institute of Technology, Atlanta, GA, 2008 [Online]. Available: https://ieeexplore-ieee-org.libproxy.nps.edu/document/4682731

[16] *REMUS Operations and Maintenance Manual*, Hydroid, Pocasset, MA, USA.

[17] J. S. Jaffe, "Underwater optical imaging: The past, the present, and the prospects," *IEEE Journal of Oceanic Engineering*, 2015 [Online]. doi: https://doi.org/10.1109/JOE.2014.2350751.

[18] D. B. dos Santos Cesar, C. Gaudig, M. Fritsche, M. A. dos Reis, and F. Kirchner, "An evaluation of artificial fiducial markers in underwater environments," in *OCEANS 2015 - Genova*, 2015 [Online]. Available: doi:10.1109/OCEANS-Genova.2015.7271491

[19] N. Højerslev, J. Høkedal, and K. Sørensen, "Optical water types of the Nordic Seas and adjacent areas," *Oceanologia*, vol. 16, May 2013 [Online]. doi: https://doi.org/110.5697/oc.55-2.471.

[20] CamTronics, "1/2" WAT-910HX/RC." Accessed Jan. 6, 2019 [Online]. Available: http://www.wateccamerashop.com/watec-products/watec-color-cameras/wat-910hx-rc

[21] SENSORAY, "4-channel A/V H.264 Ecoder/Decoder | Model 953-ET." Accessed Jan. 12, 2019 [Online]. Available: http://www.sensoray.com/products/953.htm

[22] Embedded Solutions, "Intel Atom Dual/Quadcore 4th Generation SOC/BAY Trial PCIE/104 PC/104 Form Factor." Accessed Jan. 12, 2019 [Online]. Available: https://www.adl-usa.com/product/adle3800pc/

[23] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, 2011 [Online]. Available: doi:10.1109/ICRA.2011.5979561

[24] J. Wang and E. Olson, "Apriltag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016 [Online]. Available: doi:10.1109/IROS.2016.7759617

[25] M. Krogius, A. Haggenmiller, and E. Olson, "Flexible layouts for fiducial tags," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019 [Online]. Available: doi:10.1109/IROS40897.2019.8967787

[26] M. Abbas, S. Aslam, K. Berns, and A. Muhammad, "Analysis and improvements in apriltag based state estimation," *Sensors (Basel, Switzerland)*, vol. 19, no. 24, Dec 2019 [Online]. doi: https://doi.org/10.3390/s19245480.

[27] A. Soltani and F. Assadian, "A hardware-in-the-loop facility for integrated vehicle dynamics control system design and validation," *IFAC PapersOnLine*, vol. 49, no. 21, Oct 2016 [Online]. doi: https://doi.org/10.1016/j.ifacol.2016.10.507.

[28] C. Negus. *Linux Bible*. John Wiley Sons, Incorporated, 2015 [Online]. Available: https://ebookcentral-proquest-com.libproxy.nps.edu/lib/ebook-nps/detail.action?pq-origsite=primo&docID=1895205#

[29] Open Robotics, "History." Accessed Feb. 18, 2019 [Online]. Available: https://www.ros.org/history/

[30] Open Source Robotics Foundation, "Why Gazebo?" Accessed Mar. 1, 2019 [Online]. Available: http://gazebosim.org/

[31] M. M. M. Manhães, S. A. Scherer, M. Voss, L. R. Douat, and T. Rauschenbach, "UUV Simulator: A gazebo-based package for underwater intervention and multi-robot simulation," in *OCEANS 2016 MTS/IEEE Monterey*, 2016 [Online]. Available: https://ieeexplore-ieee-org.libproxy.nps.edu/document/7761080

[32] A. Palmer, G. E. Hearn, and P. Stevenson, "Modelling tunnel thrusters for autonomous underwater vehicles," *IFAC Proceedings Volumes*, vol. 41, no. 1, 2008 [Online]. doi: https://doi.org/10.3182/20080408-3-IE-4914.00017.

[33] T. Prestero, "Development of a six-degree of freedom simulation model for the remus autonomous underwater vehicle," in *MTS/IEEE Oceans 2001. An Ocean Odyssey. Conference Proceedings (IEEE Cat. No.01CH37295)*, 2001. Available: doi:10.1109/OCEANS.2001.968766

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California