MODELING AND GUIDANCE OF AN UNDERACTUATED AUTONOMOUS UNDERWATER VEHICLE

by

Ali H M Wadi

A Thesis presented to the Faculty of the
American University of Sharjah
College of Engineering
In Partial Fulfillment
of the Requirements
for the Degree of

Master of Science in
Mechanical Engineering

Sharjah, United Arab Emirates

December 2017

Approval Signatures

We, the undersigned, approve the Master's Thesis of Ali H M Wadi

Thesis Title: Modeling and Guidance of an Underactuated Autonomous Underwater Vehicle.

Signature **Date of Signature**
(dd/mm/yyyy)

___________________________ _______________
Dr. Jin-Hyuk Lee
Assistant Professor, Department of Mechanical Engineering
Thesis Advisor

___________________________ _______________
Dr. Shayok Mukhopadhyay
Assistant Professor, Department of Electrical Engineering
Thesis Co-Advisor

___________________________ _______________
Dr. Lotfi Romdhane
Professor, Department of Mechanical Engineering
Thesis Committee Member

___________________________ _______________
Dr. Mohammad Jaradat
Associate Professor, Department of Mechanical Engineering
Thesis Committee Member

_________________________ _______________
Dr. Mamoun Abdel-Hafez
Head, Department of Mechanical Engineering

___________________________ _______________
Dr. Ghaleb Husseini
Associate Dean for Graduate Affairs and Research,
College of Engineering

___________________________ _______________
Dr. Richard Schoephoerster
Dean, College of Engineering

___________________________ _______________
Dr. Mohamed El-Tarhuni
Vice Provost for Graduate Studies

Acknowledgement

I would like to take this opportunity to express my gratitude to the people that made this work possible. I would like to thank my advisors Dr. Jin-Hyuk Lee and Dr. Shayok Mukhopadhyay for providing knowledge, guidance, support, and motivation throughout my research stages. I’m deeply grateful for their great assistance, worthy discussion, and suggestions.

I would like to thank Dr. Lotfi Romdhane and Dr. Mohammed Jaradat for serving on my examining committee and providing me with great feedback.

I would like to thank the professors of the Mechanical Engineering department who taught me the master level courses with mighty teaching methods and skills. I appreciate their dignified advices and motivation.

I would also like to thank the Graduate Program and the College of Engineering at the American University of Sharjah for offering me the Graduate Teaching Assistantship which allowed me to continue my higher education in Mechanical Engineering.

Dedication

Dedicated to whom I owe it all,

my ever-loving and ever-supporting mother…

Abstract

Autonomous Underwater Vehicles (AUVs) have become an indispensable tool that is employed by an array of fields. From the inspection of underwater cables and pipelines, to the monitoring of fish pens and coral reefs, to the detection and disposal of mines, and to the executing search and rescue operations, AUV research and development has received a lot of attention. This thesis is concerned with the mathematical modeling of an underactuated AUV to execute its missions. The modeling task entails identification of the numerous parameters of the vehicle. A finite element analysis software was used to estimate the parameters describing drag and hydrodynamic mass phenomena. While the proposed underactuated configuration promotes the deployment of more energy-efficient vehicles, this configuration imposes complications on the guidance and motion control tasks as the vehicle becomes constrained in the way it can reach certain positions or perform certain motions (anholonomy). To tackle this trajectory tracking guidance problem, a model-based controller that overcomes the underactuated nature of the vehicle was designed. This controller was further enhanced by the novel development and application of a Universal Adaptive Stabilizer-based adaptation law that aims to minimize controller effort, reject noise, and provide robust trajectory tracking. The adaptation is governed by a statistical management system to ensure proper operation in a noisy underwater environment. Moreover, the navigation problem is touched upon by implementing a sensor fusion algorithm to estimate the vehicle state in its noisy environment. The algorithm investigates an Extended Kalman Filter as well as an Unscented Kalman Filter to fuse the available information from sensors with the modeled dynamics of the vehicle and provide better estimates of the vehicle state. Additionally, the hardware and software was integrated in a Robot Operating System setting, and a Gazebo-based simulation environment that enables the visual depiction and testing of algorithms on the considered AUV was developed. The parameter identification methodology compared well to published analytical and empirical forms, the proposed adaptation law outperformed traditional techniques like Adaptive Proportional Controllers, and the gain management system demonstrated excellent potential at maintaining stable operation of the vehicle in very noisy environments.

Keywords: *Autonomous Underwater Vehicle; Modeling; Underactuated Systems; Adaptive Control; State Estimation; System Identification; Robot Operating System*

Contents

List of Figures

List of Tables

List of Abbreviations

ANC	Adaptive Nussbaum Controller
APC	Adaptive Proportional Controller
AUV	Autonomous Underwater Vehicle
AWGN	Additive White Gaussian Noise
CB	Center of Buoyancy
CG	Center of Gravity
DOF	Degrees of Freedom
EMF	Electromotive Force
KF	Kalman Filter
EKF	Extended Kalman Filter
ESC	Electronic Speed Controller
GPS	Global Positioning System
IMU	Inertial Measurements Unit
INS	Inertial Navigation System
PI	Proportional Integral
PID	Proportional Integral Derivative
UKF	Unscented Kalman Filter
UV	Underwater Vehicles
ROS	Robot Operating System
ROV	Remotely Operated Vehicle
SoC	State of Charge
ToF	Time of flight
VS	Visual Servoing

Chapter 1. Introduction

Underwater vehicles have progressed a lot over the course of the past 70 years. Research and development of Underwater Vehicles (UVs), however, has lagged behind when compared to aerial vehicles, land vehicles and watercrafts. Numerous problems hindering the deployment of underwater vehicles are still considered unsolved when compared to the other vehicles. Primarily is the problem of accurate localization underwater. Vehicles operating above sea level make use of the Global Positioning System (GPS) in localization and velocity estimation tasks; on the other hand, the high frequency nature of the electromagnetic waves of GPS causes conductive seawater to attenuate it heavily and hinder its utility. Other more costly techniques are used to localize UVs, but those techniques run into limitations when it comes to accuracy and/or the effective span of the space they can cover. Further, many classes of autonomous underwater vehicles are underactuated, which gives rise to nonholonomic constraints on the kinematically admissible motion of the vehicle. Consequently, the control process becomes nontrivial in the sense that nonlinear approaches are sometimes required, for stabilization through time-invariant static state-feedback is not generally possible. Other issues affecting the development of underwater vehicles include the high cost of equipment, the high-pressure nature of the environment, and the high cost maintenance required to keep the vehicles operational.

This chapter introduces the different types of Underwater Vehicles (UV), their use in the industry, and the difficulties associated with their deployment. It then presents the problems investigated in this work as well as the contribution. Finally, the organization of the thesis is presented.

1.1. Overview

UVs can be split into two categories; manned vehicles and unmanned vehicles. The former includes submarines as well as atmospheric diving suits, and the latter includes Remotely Operated Vehicles (ROVs), Autonomous Underwater Vehicles (AUVs), and Hybrid Vehicles. The last category can either be man-operated or autonomous. The several types of UVs are deployed in many industries today for various purposes. Those uses are summarized adequately by the Department of Navy for Research, Development and Acquisition of the United States. Categorized under defense purposes, UVs can perform intelligence, surveillance and reconnaissance missions, mine countermeasures operations, offence missions, inspection and identification tasks, oceanography surveys, payload delivery assignments, and information acquisition/jamming operations. Categorized under commercial uses, UVs are involved in offshore Oil and Gas works like rig inspection as well as underwater construction/welding, pipeline and undersea-cable deployment and inspection works, salvage operations, and monitoring and protection of aquaculture fish farms. Categorized under science missions, UVs are involved in oceanographic observation tasks and marine archaeology studies [1].

The aforementioned missions depend heavily on the ability of the vehicle to locate itself with respect to the environment as well as the awareness of the surrounding environment. Each mission requires some types of sensor packages; to exemplify, aquaculture monitoring enforces that the vehicle be able to map underwater current movement, take environmental readings like temperatures, oxygen levels, and salinity of the water, and detect and localize fish swarms [2]. Some of the missions necessitate the vehicle to have a manipulator arm to carry the different works. Others require autonomy, the removal of a tether, and high endurance to be deployed for extended periods of time. Consequently, it is argued that all of them can make use of an accurate mathematical model that can describe adequately the dynamics of the vehicle.

One of the major constituent subsystems of an Underwater Vehicle is the pressurized hull, which houses most of the electronics. It also protects them from the pressure outside, and it prevents water from seeping inside. Next is the hydrodynamic hull, which encases the vehicle in a shroud that enhances drag characteristics by allowing for a smoother and more aerodynamic water flow-path around the vehicle. This hull is usually made from syntactic foam, which also serves to permanently control the vehicle buoyancy. Buoyancy Ballast system is the subsystem that provide buoyancy control at any time. An example is submarine ballast tanks that can store water or replace it with air to adjust stability or buoyancy. The power delivery system supplies the vehicle with power through means of internal high capacity and power density batteries, or it can be delivered through an external umbilical/tether. Thrusters or thruster-control surface combinations are used to actuate the vehicle and steer it through water. Thrusters consist of motors with carefully selected propellers installed. Brushed motor thrusters need to be pressure and water proofed through the design of special housings that are usually filled with oil. In addition, periodic maintenance in the form of changing the brushes of the motor is necessary. More recently, brushless motor thrusters have been adopted for cost-efficient underwater vehicles. They need not be water or pressure proofed, for they have an open design and water can seep into the inside of the motor. Adverse effects include a shorter lifespan of the thruster, though. Also, brushless motors require specialized control circuits to function properly. Navigation and positioning subsystems take care of localization of the vehicle as well as other dependent tasks like path-planning, obstacle-avoidance, and navigating the environment. Finally, communication links transmit data to ground station or allow remote operation of the vehicle. It can take a wireless acoustic form or a wired form through the tether interface.

1.2. Thesis Objectives

This research work develops a mathematical model for an underwater vehicle, and it utilizes the developed model in the creation of an adaptive guidance controller as well as a navigation filter. The derived highly-coupled nonlinear model accounts for various phenomena a marine craft experiences in an underwater setting. The aforementioned applications in section 1.1 of the vehicle model can improve the performance and endurance of various marine crafts working in the field. A better model-based control law can help minimize the control effort required to realize certain functionality. The proposed adaptive control low exhibits robustness to external environmental disturbance while requiring lower control effort when compared to a more traditional approach. The model-based state estimator serves as a possible substitute for an Inertial Navigation System (INS) for solving the navigation problem.

1.3. Research Contribution

This work accomplishes the following:

- Propose a mathematical model for an underwater marine craft that is advantageous in automating the vehicle in its missions.
- Perform parameter identification through means of software packages to describe various underwater environmental phenomena.

- Utilize the devised model in:
    - Devise a robust adaptive controller capable of overcoming the underactuated nature of the vehicle, the coupled nonlinear dynamics of the vehicle, and the unpredictable disturbances an underwater environment imposes on the vehicle.
    - Propose and implement a model-based navigation solution capable of providing the position and attitude of the vehicle in a reliable manner.
- Integrate the hardware and implement the devised controller and the in a Robot Operating System setting.

1.4. Thesis Organization

After the introduction chapter, this thesis is organized as follows:

- Chapter 2 details the use of UVs in the industry and presents a survey of the practices involving UVs. It also reviews the literature available on modeling, parameter identification, navigation, guidance, and control of UVs.
- Chapter 3 derives the mathematical model describing the equations of motion of the vehicle.
- Chapter 4 involves in the identification process of rigid body, drag, and added inertia parameters of the proposed vehicle.
- Chapter 5 proposes the guidance control strategy developed to overcome the underactuated nature of the underwater vehicle. It also presents the adaptation laws and the devised gain update algorithm.
- Chapter 6 implements sensor-fusion algorithms that make use of the devised mathematical model and the information available from sensor packages to estimate the state of the vehicle.
- Chapter 7 details the proposed experimental setup and sensor packages to be used. It also presents the architecture for hardware-software integration, and it showcases a Gazebo underwater environment for testing of underwater vehicle algorithms.
- Chapter 8 concludes on the outcomes of this work and proposes improvements on future efforts in the field.

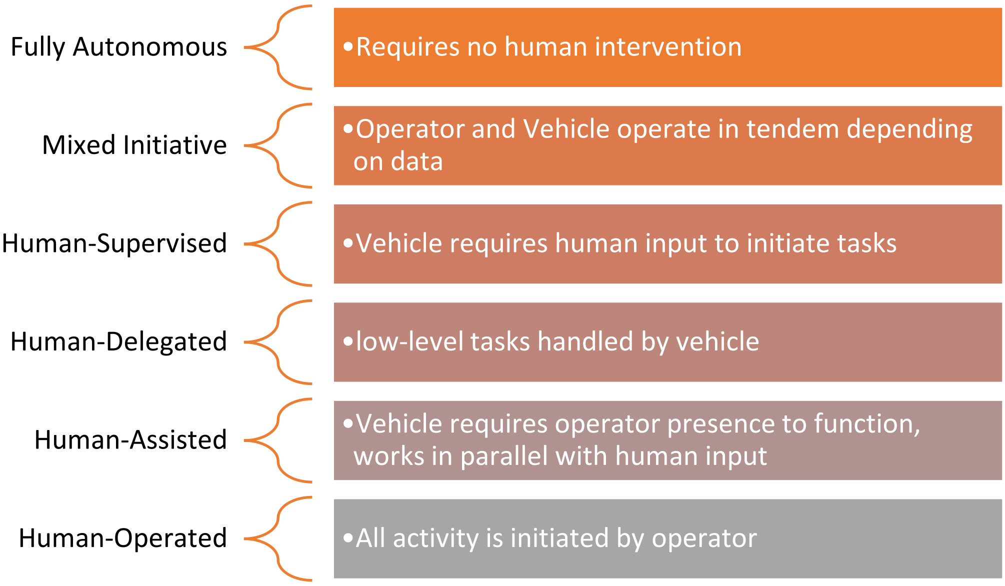
Chapter 2. Background and Literature Review

This chapter engages in market research that aims to identify what product AUVs in the industry utilize to carry the tasks required of them. Areas of improvement are identified and are taken as a basis for contribution of this thesis. Literature review is then presented to describe the modelling process for UVs and to showcase the latest model-aided techniques employed in navigation and control of UVs.

2.1. Survey of AUVs in the Field

As part of literature review, a survey of AUVs operating in the field was conducted in order to classify underwater vehicles, find out how they perform their tasks, and identify possible areas of improvement. This section categorizes the vehicles based on the levels of autonomy possible and the levels of sizes that the vehicle can take.

The levels of autonomy differ from one field to another; nonetheless, be it an automotive vehicle, a robot, or an underwater vehicle; all fields identify a range of classes that span the gap between full human control to full autonomy of the vehicle. Figure 1 presents 6 different levels of autonomy as classified by the National Defense Research Institute [1].



Fully Autonomous – •Requires no human intervention

Mixed Initiative – •Operator and Vehicle operate in tendem depending on data

Human-Supervised – •Vehicle requires human input to initiate tasks

Human-Delegated – •low-level tasks handled by vehicle

Human-Assisted – •Vehicle requires operator presence to function, works in parallel with human input

Human-Operated – •All activity is initiated by operator

Figure 1: Vehicle Autonomy Levels

One more possible classification of AUVs is one based on the size of the vehicle. The size of an AUV can allow or limit what missions it can perform. A survey done by United States Department of the Navy identifies 4 main classes of AUVs available in the market [1]. Figure 2 presents the identified classes.

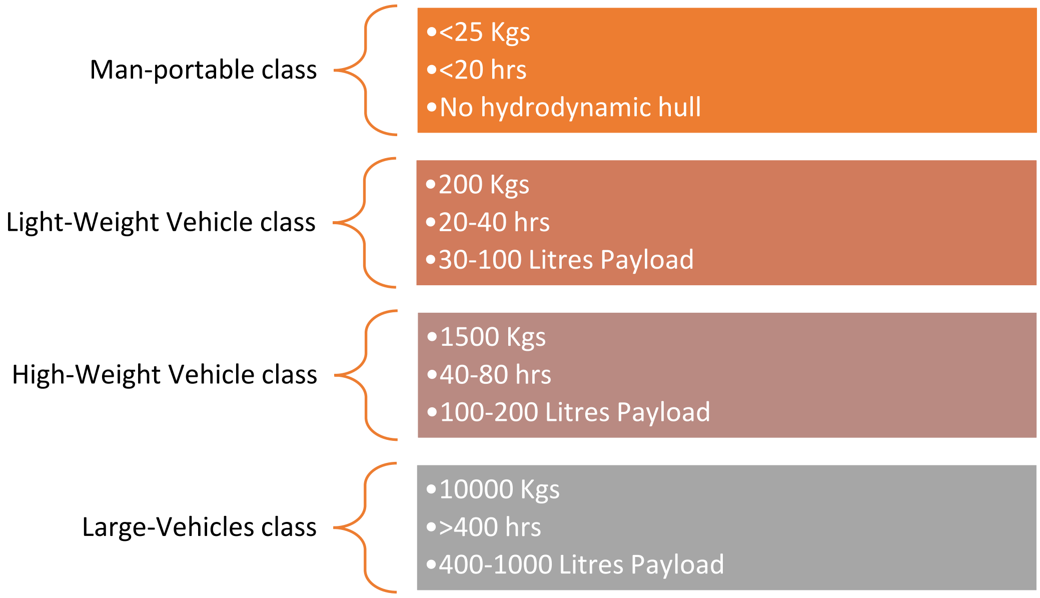


Figure 2: Autonomous Underwater Vehicle Classes

The surveyed AUVs follow the classification detailed in Figure 2. Categorized under the first class, Man-portable class, the Bluefin-9 in Figure 3 is chosen. It is a product of Bluefin Robotics, and it is one of the smallest AUVs in deployment today. It was designed to carry mapping missions in shallow waters.

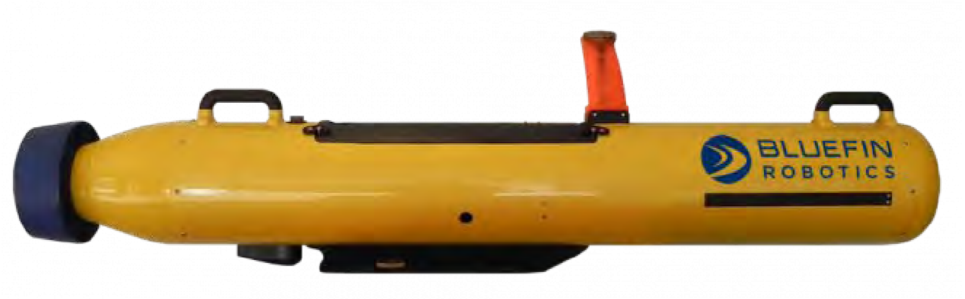


Figure 3: Bluefin-9 [1]

The REMUS-6000 shown in Figure 4 is an example of a Light-Weight Vehicle class product. It is developed by Naval Oceanographic Office as a substitute for other vehicles of this class with low endurance and payload ratings. This vehicle is aimed at deep sea operations; mainly mapping.

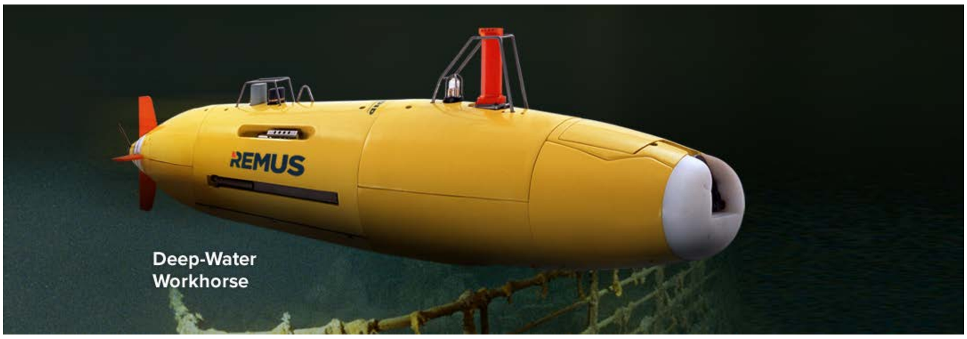


Figure 4: REMUS-6000 [1]

To demonstrate defense/military use of AUVs, the High-Weight class HUGIN-1000 Military Version was chosen, which is shown in Figure 5. This vehicle was developed by the Royal Navy of Norway based on another product, the HUGIN-1000. The HUGIN product line is heavily engaged in Oil and Gas underwater works and inspection. The military version of the HUGIN is feature rich, and it is one of the few vehicles employing sensor fusion and map-based navigation techniques.

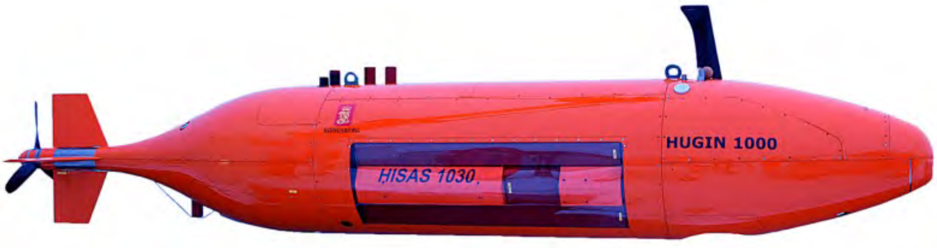


Figure 5: HUGIN-1000 [1]

An example of the Large-Vehicles class is the Theseus in Figure 6, which is an AUV designed to carry installation and inspection works for undersea fibre-optic cables. Its large size and payload capacity make it suitable for its tasks. They also enable it to carry payload delivery missions as well.

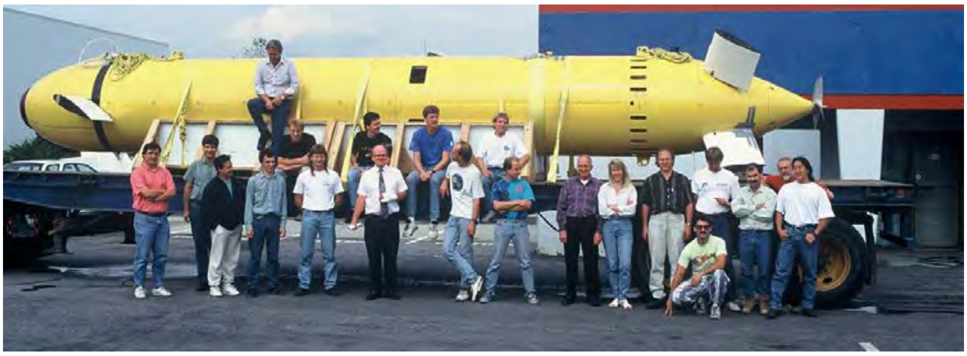


Figure 6: Theseus [1]

One last example of a very small vehicle that man-operated is the VideoRay, which is shown in Figure 7. It is an inspection class remotely operated vehicle that is considered to be one of the most portables vehicles in the industry. Like other small vehicles, it lacks proper navigation sensor packages, and it is linked to a ground control station through means of a tether. This tether also supplies it with power.

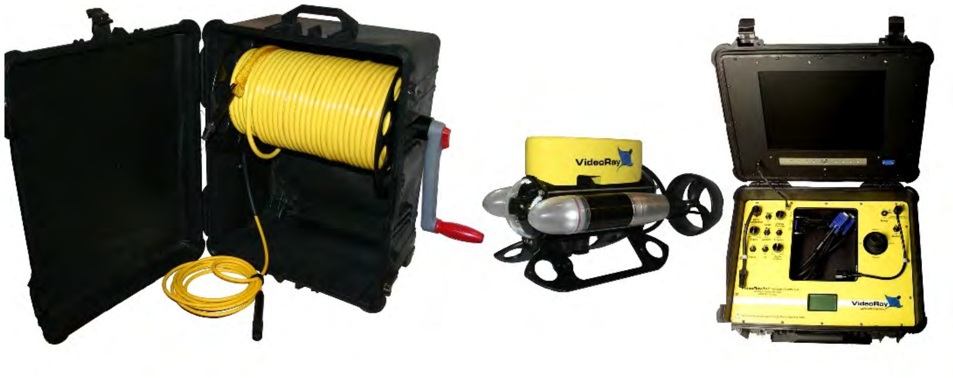


Figure 7: VideoRay [1]

To summarize the results of the survey, Table 1 compares the surveyed vehicles. The conducted research provided some valuable insight into what types of vehicles the different industries utilize and how those vehicles perform their given tasks. The combination of different sensor packages enables each vehicle to carry its given tasks. Further, most vehicles that carry localization elected to use acoustic time of flight techniques, which provide the location of the vehicle relative to a known truth landmark. Drawbacks here include the high cost in addition to the need for accompanying GPS-enabled floats to acquire the accurately known true position. Navigation through unknown territory is accomplished with the help of multi-beam sonar systems that map the environment and localize the vehicle in the generated map. Few high-end vehicles designed for military used especially make use of sensor fusion techniques to combine inertial information from IMUs with other sources like pressure sensors and Echo Sounders [3]. All the surveyed vehicles are underactuated. The exact nature of the controllers implemented on the surveyed vehicles is not disclosed; however, no indication of model-based control or application of advanced techniques exist [1, 4]. This is explored further in the next subsection.

Table 1: Market Survey Specifications List

Feature	**Bluefin-9** **(man-portable class)**	**REMUS-6000** **(light-weight class)**	**HUGIN-1000** **(high-weight class)**	**Theseus** **(large-vehicle class)**	**VideoRay** **(remotely operated)**
Hull	1.65 m long 0.24 m Diameter 50 Kg Dry weight	3.84 m long 0.71 m Diameter 862 Kg dry weight	3.85 – 5.0 m long 0.75 m Diameter 600 - 850 Kg dry weight	10.8 m long 1.3 m Diameter 8845 Kg Dry weight	0.3 m long 0.3 m Diameter 4.5 Kg Dry weight
Speed	1.0 – 2.6 m/s	0 – 2.6 m/s	0 – 3.0 m/s	0 – 2.1 m/s	0 – 2 m/s
Operating Depth	100 m	4000 m	1000 m	1000 m	70 m
Endurance	12 Hours (speed dependant)	22 Hours (speed dependant)	50 - 60 Hours (speed/payload dependant)	>1 week (speed dependant)	Power supplied through tether interface
Communications	Acoustic and Radio links	Acoustic link WiFi 2.4 GHz Iridium Satellite	Ethernet (when tethered) Acoustic and Radio links WiFi Iridium Satellite	Acoustic link to a Radio-modem float	Ethernet through tether interface
Sensor Packages	• Scanning Sonar • Optical quadrant detector system	• Acoustic Doppler Current meter • Multiple Scanning Sonar solutions • Video Camera • Conductivity Sensor • Inertial, LBL, ULBL Navigation solutions	• Multiple Scanning Sonar solutions • Conductivity Sensor • Doppler Velocity log • GPS, Inertial, ULBL Navigation solutions • Sensor fusion	• Magnetometers • Video Camera • Still Camera	• Pressure sensor • Compass • Video Camera
Payload	N/A	N/A	N/A	2.5 m^3 Max Volume 2041 Kg Max Dry Weight	N/A

2.2. Literature Review

This section of literature review will be split into subsections, each dealing with one area of research.

2.2.1. Modeling of underwater vehicles.

A lot of work has been put into characterization of the dynamics of underwater vehicles. These efforts aim to capture the various physical phenomena that an UV experiences in a fluid environment like that of water. The vehicle is usually taken as a rigid body that can be described by the dynamics of rigid bodies undergoing general motion in the 6 degrees of freedom that are kinematically admissible [5, 6]. These equations of motion turn out to be a set of 6 second order nonlinear differential equations. Those set are accompanied by a kinematics model that can describe the motion of the vehicle in a number of coordinate frames [7, 8].

Since water is more viscous and more dense that of air, effects of the likes of damping have to be taken into account in a water environment [9]. Drag is the main energy dissipating component in the vehicle model, and it is a function of water properties of the likes of density and salinity. It is a resistive hydrodynamic force resulting from the movement of the vehicle relative to the fluid as well is the movement of the fluid relative to other fluid layers; those constituents of drag are called potential drag and skin friction, respectively [10].

In a six degrees-of-freedom (DOF) setting such as that of the vehicle, drag is a result of the coupled motion of all the motions of the vehicle. [11] presents a novel procedure that aims to identify the hydrodynamic damping coefficients that describe the two types of drag. The procedure includes constructing a pendulum-like setup with a scaled model of the vehicle, whose parameters are to be identified, pinned at the end of the pendulum. As a result of the exact knowledge of the dynamics of pendula, a model for this new system is derived with the drag coefficients governing the damping behavior. Decay pendulum motion experiments were conducted and validated against the real vehicle. The results proved to be descriptive of the phenomenon. More traditional approaches are taken in [12-16], where finite element software packages are utilized to simulate the flow around the vehicle, and the drag coefficients are extracted from simulation and validated against experimental data. In [15], the equations of motion are formulated in such a way for the equations to be linear in the unknown drag coefficients. Accordingly, a number of Least squares techniques were employed to identify the parameters of a set of ROV free roaming data collected. Furthermore, 6 different plant models were used, each with a different level of complexity in terms of the coupling between the different degrees of freedom. The algorithms were successful at finding the parameters, with the difference being converging time and the amount of data needed to reach a solution.

Other noticeable crucial effects when a solid body moves in a fluid is the added mass or inertia of the water surrounding the solid [17]. When acceleration is imposed on a fluid either by acceleration of a solid body in the fluid or by acceleration of the fluid itself, additional forces will act on the surfaces in contact with the fluid. These fluid inertial forces are of importance in many practical situations.

Added mass is a fluid-structure interaction phenomenon that affects the motion of solid structures in a fluid medium. This interaction appears in the form of an additional force required to accelerate the body in the medium. The interaction is governed by numerous criteria like the geometry of the moving body, the orientation of the body, the type and amount of fluid surrounding the body, and the frequency of excitation of waves [18]. The effect appears as a strongly coupled phenomenon or as a weakly coupled one. Strong coupling is denoted as two-way interaction between the solid and the fluid domains, which means that movement of one medium affects the other in a significant manner such that the resulting interaction on the second medium affects the first and so on. This type of fluid-structure interaction is observable in flexible solids undergoing vibration. Weak coupling, however, is a one-way interaction that manifests from one medium to the other. Being a rigid body and not a flexible one, an underwater vehicle experiences only one-way weak coupling. Moreover, when it comes to Underwater Vehicles, the assumption is made to neglect the varying nature of the phenomenon under the premise that underwater currents roughly do not vary at typical working depths for these vehicles ($Depth > 20m$) [10]. Consequently, the effect can be modeled as a constant term that is added to the mass matrix of the vehicle and not a function of frequency [12]. A lot of effort is expensed in finding cost-effective ways to identify added inertia parameters. [19] presents a novel adaptive algorithm is implemented online to fine tune all the different parameters of the vehicle in a real-time setting. The algorithm is tested in a real-world test and it is shown to give comparable results to the more conventional least squares parameter identification, which was run offline. other approaches taken to identify added inertia include modal analysis-based approaches and kinetic energy-based approaches. The former relies on comparing the natural frequencies of the rigid body of the vehicle in vacuum to the natural frequencies of the vehicle in water, and then it uses this information to identify the parameters [20]. The other approach is based on potential fluid theory, and it matches the kinetic energy of the added inertia moving at the speed of the vehicle to the kinetic energy of the fluid domain in which the vehicle moves [21].

2.2.2. Underactuated systems. Underactuated systems are ones in which the number of actuators is less than the number of degrees of freedom. They are defined as systems, usually mechanical, that cannot follow arbitrary paths and trajectories in the configuration space of the system. While motion can happen in those unactuated degrees of freedom due to external disturbances for example, the actuator configuration does not have control over them, and the system cannot be accelerated in the unactuated degrees of freedom. Sometimes the reason behind the under-actuation is the existence of nonholonomic constraints that prevent the motion in certain directions [22]. Nonholonomic constraints are kinematic constraints that are expressed in terms of the generalized coordinates and velocities. They limit the motion through restricting the set of allowable generalized velocities.

Underactuated mechanical systems often arise in applications involving space/underwater robots, aerial vehicles, mobile robots, robotic manipulators, and legged robots [23]. The study of underactuated systems is essential to the deployment of such types of robots in their respective fields. It also is important when a fully actuated system loses an actuator on a mission and is still required to continue its task [24]. All of the surveyed AUVs were underactuated.

2.2.3. Guidance. Guidance is the action of determining the desired position, velocity, and acceleration of a marine craft to be used by the control system [25]. Many solutions to this problem exist in literature. The trajectory generation task is preferred to take a simple form and be computationally inexpensive. To that end, techniques of the likes of polynomials, splines, and linear paths with blends have been developed [26]. Dubins, clothoids (Euler spirals), and Fermat spirals are reported to be used in connecting linear path segments [27, 28]. Monotone cubic Hermite splines are an approach used in generating spline-based trajectories [29]. The guidance system utilizes one of the aforementioned techniques to generate paths that connect reference waypoints for the vehicle to follow.

2.2.4. Control of underactuated systems. Control is the action of determining the effort in terms of forces and moments required of a marine craft to achieve an objective, which lies in accordance with the guidance task [25]. The AUV control problem is a rather challenging one [30, 31]. One reason behind the claim includes the underactuated nature of the vehicle. Although the nature of the motion is kinematically admissible, there are less actuators than degrees of freedom. In other words, the vehicle is capable of reaching a certain target position, but it has to move in such a way that is permissible by the thruster arrangement installed. This arrangement is referred to as nonholonomic constraints, which translates to the vehicle not being able to move in one or more directions. Advanced control approaches are often necessary to get around those constraints. These approaches have to exploit the coupling between the degrees of freedom to design control laws that are able to carry the task required of the vehicle. In fact, it has been proven that nonholonomic control systems with restricted mobility cannot be stabilized by smooth time-invariant state feedback laws [32]. The motion control task can be a point-to-point motion problem where the system has to reach a final configuration from an initial one, a path following problem where the system has to both reach a final configuration from an initial one and follow a geometric path, or a trajectory following problem where the system has to perform a path following problem that is associated with a timing law, which dictates when the system should reach a certain configuration [33].

Literature presents a multitude of approaches to tackle the control problem. One of the simpler approaches to solve the control problem is line-of-sight (LOS) path following, where the marine craft is required to move at a constant forward speed and a heading controller leads the marine craft to converge to the parameterized path [29, 34]. The heading controller drives two angles between the current position of the vehicle and a desired waypoint to accomplish the control objective. Sliding mode control is used in literature to design controllers that are exponentially stable and capable of following trajectories. In [33], the authors tackle the problem of designing control laws for a 4-thruster underactuated marine craft. First, the dynamics are transformed into chained form coordinates. A kinematic control law is then devised for point-to-point as well as trajectory following motion tasks. Finally, a dynamic control law, which is also designed around the transformed coordinates, is coupled with the kinematic control law to realize force- and moment-based control rather than velocity control. The stability of the laws is investigated and proven. An AUV with 4 actuators is treated in [35], where a second order chained form transformation is coupled with a discontinuous control method to attain a globally stable control of the system. The stabilizing problem was split into three, each concerned with a part of the dynamics. The controller is verified through simulation. A more traditional methodology to deal with underactuated systems is presented in [36], and it was tested on an underactuated prismatic-prismatic-rotary (PPR) robotic manipulator. The procedure includes expressing the dynamic equations into a 2nd order chained form, and it devises a backstepping controller for this new subsystem. Backstepping is a nonlinear control methodology that allows for more flexibility than state-feedback linearization. It permits keeping some nonlinear terms in the dynamics as well as adding nonlinear damping, which makes for "more robust control laws that are less susceptible to errors arising from imprecise models" [37]. In [38], a quadcopter underactuated system is stabilized in attitude using a time-delayed controller. The results of which outperform that of a similar PID controller, especially for disturbance rejection. The control method is concerned with estimating the dynamic state of the system and providing compensation through control action.

2.2.5. Navigation. Navigation is science of directing a marine craft through determining pose, course, and distance travelled [10]. The missions that Underwater Vehicles execute require the vehicle to know its position relative to the environment in which it is deployed. This requires the use of sensor packages that enable the vehicle to carry the different tasks and missions it is assigned. In contrast to vehicles with direct line of sight to the sky, where localization is a solved problem through means of the Global Positioning System (GPS), an underwater setting renders GPS ineffective since it operates on high-frequency electromagnetic waves (~1.5 GHz). Those waves get highly attenuated by water, and they are unable to penetrate well in water.

The latest developments in this field come in the form of integration of computer vision and mapping techniques. [39] provides a comprehensive survey that details most of the technologies being deployed recently. A lot of those techniques aim to carry simultaneous localization and mapping (SLAM) of the underwater environment, which makes later visits to a site that has been mapped easier since the vehicle only has to localize itself relative to the known map. A number of sensor packages are used to implement the techniques; sonar sensors in the single and multi-beam form and one or more camera sensors comprise the most widely used techniques. The algorithms are often coupled with other sensor packages and sensor fusion is usually carried to have better measurements of the quantities of interest [40].

High-frequency Long Baseline (LBL) as well as Ultra-Short Baseline (USBL) acoustic time of flight methods are used to acquire the position of the vehicle relative to a transponder/network of transponders. Both methods utilize triangulation of the acoustic waves. The former utilizes a network of transponders with known locations, while the latter utilizes sonar technology to determine the range to the vehicle [3, 41]. The errors that occur in these methods are attributed to inaccurate knowledge of the precise locations of the transponders, inaccurate knowledge of the sound velocity and the variations that happen to the velocity of sound as a result of environmental factors such as water temperature and density [4]. It is possible to measure some of those quantities and update the estimate of the speed of sound underwater frequently onboard the vehicle.

The relative distance of the vehicle to the surface of the water, depth, is a quantity that can be measured indirectly. Pressure sensors can be used to estimate the depth of the vehicle since pressure is a linear function of depth. However, those sensors operate with some uncertainty, for movements of the vehicle affect the pressure field surrounding it. As a result, the quality of measurements when the vehicle is not stationary degrades.

Other quantities that are essential for navigation and tracking are the attitude, heading, and velocity of the vehicle. Inertial measurement units (IMU) comprising of accelerometers, magnetometers and gyroscopes are used to provide estimates of the aforementioned quantities. Those sensors excel at providing an accurate attitude of the vehicle. Moreover, IMUs enable inertial navigation through dead reckoning, which is the process of estimating the trajectory through knowledge of an initial position and availability of IMU measurements. The result is highly driven by the drift resulting from the integration process of IMU measurements to achieve position estimates. Errors accumulate over time, and the drift grows to cause unbounded growth in the quantities integrated. Inherent errors with these systems include misalignment of the sensor axis and those of the vehicle, errors due to geographic magnetic interferences, and errors due to magnetic disturbance from the vehicle components magnetic signature [42]. Velocity can also be acquired through a single integration of the acceleration, but the drift problem will persist. Another way of acquiring velocity estimates is through Doppler sonars, which utilize the Doppler effect to calculate the translational velocity of the vehicle relative to the sea-floor. Nonetheless, they work provided the sea-floor is in range of the sensor used [43].

State estimation enables the use of stochastic techniques to improve on the quality of measurements produced by the sensors used. It involves the use of stochastic filters that utilize any and all available measurements of the states to produce a measurement of superior quality. The filters fuse the measured quantities along with the dynamic or kinematic models governing the motion of the vehicle to provide a better estimate than the measurement [4]. Those filters are also optimal and unbiased, given that the correct measurement and dynamic model statistics are fed into the filters. Consequently, this property corrects for the noises corrupting the states as well as the uncertainty of the sensors. It is of interest to point that the singularity-free solution to represent rotation, quaternions, imposes difficulties that are acknowledged in the literature. For instance, the work of [44] derives the inertial dynamics of a body utilizing quaternions to represent attitude. Several approaches are followed, and the performance of them is compared. An Extended Kalman Filter (EKF) approach is first presented. Then, an optimization based quaternion rate Kalman is devised. The implemented optimization algorithms are based on newton and gauss-newton numerical methods, and they provide fast convergence as well as performance improvement over the EKF in a real-time setting. Instead of an EKF, [45] deals with the formulation of an Unscented Kalman Filter (UKF) to deal with the nonlinearity associated with the process and/or measurement equations used to construct the filter. Certain modifications were done on the formulation of the filter to accommodate for the problematic properties of the quaternions. Other works acknowledge the bias error terms present in the sensors and include it in the state vector to be estimated. To illustrate, [46] avoids the use of an EKF and uses a nonlinear observer instead, one in which global exponential convergence in terms of the estimation error can be established. The observer utilizes the kinematic equations of motion for an aerial vehicle and integrates information from both GPS and IMU sources. The results are verified using real test data. In [47], an algebraic nonlinear transform of generalized nonlinear measurement equations of a time of flight sensor solution is implemented such that the new model is a linear time-varying one. Integrating the new model with two observers for attitude and translational motion estimation enabled the presentation of a navigation solution that is proven to be globally convergent. Other works on nonlinear observers aim to couple the information provided by the active sensor packages and fuse the measurements to aid in navigating the vehicle [40, 48, 49]. The premise with nonlinear observers is that, unlike the EKF, it is often possible to prove local or even sometimes global stability of the observer.

Chapter 3. Mathematical Model

In this chapter, we study the dynamics that govern the motion of the vehicle. Standard notation in the marine vessels field is utilized here. Then, the kinematics model is formulated considering appropriate coordinate frames. Afterwards, the dynamic equations of motion are derived from both basic principles and empirical observations. Finally, the power source model of Chen and Mora is explained and detailed.

3.1. The kinematics of Motion

For an unconstrained six degrees of freedom marine vessel, six independent coordinates are required to fully describe the dynamics. The Society of Naval Architects and Marine Engineers have established standard terminology to describe position, attitude, velocities, forces, and moments. Table 2 presents the nomenclature used throughout this thesis.

Table 2: SNAME Nomenclature for Marine Vessels [7]

DOF	Motion	Forces and Moments	Velocities	poses
1	Motion in the x direction (Surge)	X	u	x
2	Motion in the y direction (Sway)	Y	v	y
3	Motion in the z direction (Heave)	Z	w	z
4	Motion around the x direction (Roll)	K	p	ϕ
5	Motion around the y direction (Pitch)	M	q	θ
6	Motion around the z direction (Yaw)	N	r	ψ

3.1.1. Coordinate frames. Dynamic motion of the AUV requires a reference frame relative to which motion can be described. This inertial frame has to be one in which the Newton-Euler equations of motion are valid; that is, the frame is nonaccelerating. A number of standard coordinate frames are in use to describe the terrestrial motion of Earth through space as well as motion on the surface or Earth.

The Earth-Centered Inertial reference frame (ECI) is a nonaccelerating frame that is fixed at the center of the Earth. It is often used to describe terrestrial navigation. Another frame of interest is the Earth-Centered Earth-Fixed (ECEF) reference frame.

Similar to ECI, ECEF is fixed at the center of the Earth. However, it rotates relative to ECI at a rate of a revolution per 24 hours or $7.2921x10^{-5}\ rad/s$.

The North-East-Down (NED) is a Cartesian coordinate frame that is defined relative to elliptical Earth. It is a tangent plane to the surface of the earth that moves with the vehicle, but its axes point in different directions than those of the vehicle. The three principal directions in this frame point towards true north, the east, and downwards. To map a location in this frame relative to ECEF, longitude, l, and latitude, μ, angles are used. Figure 8 shows the NED frames with the longitude and latitude angles.

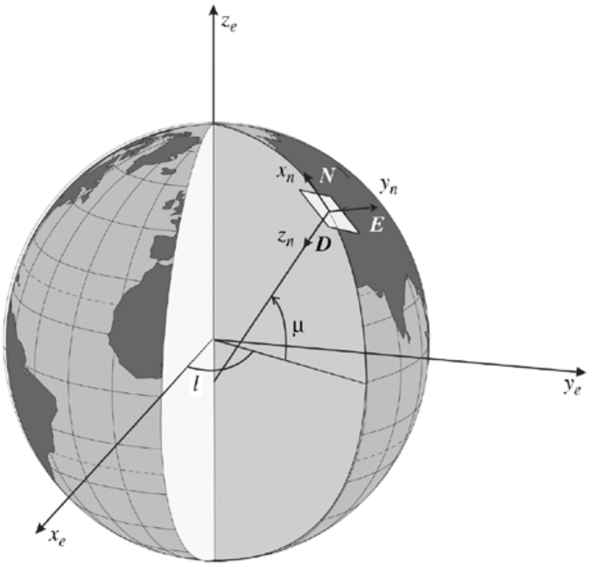


Figure 8: North-East-Down Coordinate Frame [10]

An approximation of the NED coordinate frame is denoted as the n-frame. It is defined as a tangent plane to the surface of earth with Cartesian axes pointing towards the directions of NED and constant longitude and latitude angles. This approximation is called flat Earth navigation, and the gravitational acceleration is constant in both direction and magnitude in this model [25]. The effects of having a rotating earth are neglected under the premise that AUVs are not made to transverse large distances. In addition, the effects that the rate of rotation of Earth introduces are negligible when compared to the dominant hydrodynamic effects on the vehicle. The rate of rotation of Earth is $\omega_e = 7.2921 \times 10^{-5}\ rad/s$.

Another reference frame of interest happens to be the body-fixed frame (b-frame), which is located at the center of gravity of the vehicle for convenience $(x_b y_b z_b)$. In this frame, x_b points forwards, y_b points rightwards and z_b points downwards. The rotations the vehicle can take are roll about the x_b axis, pitch about the y_b axis and yaw about z_b axis. Figure 9 depicts the axes of the b-frame on a marine vessel.

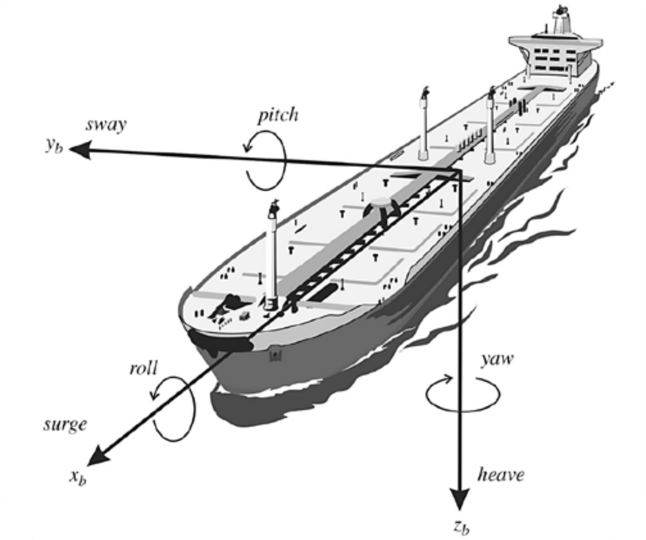


Figure 9: Body-fixed Frame [10]

3.1.2. Coordinate frame transformation. The orientation of the AUV can have be described using numerous approaches. The mapping between orientation in inertial and body frames can be done through three consecutive rotations about the axes of interest with an order that is important. There are numerous ways to represent this rotation. One can use axis-angle rotation, Euler angles, Gibbs vector, Pauli spin matrices, and Hamilton's quaternions to name but a few [10]. The physically meaningful Euler angles of roll, pitch, and yaw, can be used; however, the inverse kinematic problem of extracting the angles from the rotation matrix run into singularities when the second rotation sequence matches either of the other two. A singularity free solution comes in the form of unit quaternions, which is a four-parameter solution for the three rotations problem [50]. The velocity in the n-frame is realized through rotating the velocity of the body-fixed frame as follows

$$\dot{\mathbf{P}}_n = \mathbf{R}_b^n(\boldsymbol{q})\,\boldsymbol{\upsilon}_{\boldsymbol{b}} \quad (1)$$

where $\mathbf{R}_b^n$ is the rotation matrix mapping a vector quantity in the b-frame to the n-frame. Appendix A introduces quaternions.

The time rate of change of quaternions is related to the angular velocity vector of the body-fixed frame, and the is given by the following matrix

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{\eta} \\ \dot{\epsilon} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -\boldsymbol{\epsilon}^{\mathrm{T}} \\ \eta\ \mathrm{I}_{3x3} + \mathbf{S}(\boldsymbol{\epsilon}) \end{bmatrix} \boldsymbol{\omega}_b = \mathbf{T}(\boldsymbol{q})\ \boldsymbol{\omega}_b$$
$$\mathbf{T}(\boldsymbol{q}) = \frac{1}{2}\begin{bmatrix} -\epsilon_1 & -\epsilon_2 & -\epsilon_3 \\ \eta & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & \eta & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & \eta \end{bmatrix} \tag{2}$$

where $\mathbf{S}$ is the skew-symmetric matrix defined by

$$\mathbf{S}(\boldsymbol{\lambda}) = \begin{bmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{bmatrix} \tag{3}$$

and the vector cross product is defined by

$$\boldsymbol{\lambda} \times \mathbf{a} := \mathbf{S}(\boldsymbol{\lambda})\mathbf{a} \tag{4}$$

Standard terminology in the marine vessels field denote body-fixed displacements and velocities, whether they use quaternions or Euler angles, as

$$\boldsymbol{\eta} = \begin{bmatrix} \mathbf{P}_n \\ \boldsymbol{q} \end{bmatrix}, \qquad \boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{\upsilon}_b \\ \boldsymbol{\omega}_b \end{bmatrix} \tag{5}$$

where $\boldsymbol{\eta}$ represents position and attitude in the n-frame, $\boldsymbol{\nu}$ represents the linear and angular velocities in the b-frame. The components of the position and orientation vectors are

$$\mathbf{P}_n = \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix}, \qquad \boldsymbol{\upsilon}_b = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \qquad \boldsymbol{q} = \begin{bmatrix} \eta \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix}, \qquad \boldsymbol{\omega}_b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{6}$$

The kinematic equations of motion can then be shown to take the following form

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_q(\boldsymbol{q})\ \boldsymbol{\nu} \tag{7}$$

$$\begin{bmatrix} \dot{\mathbf{P}}_n \\ \dot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b^n(\boldsymbol{q}) & \mathbf{O}_{3x3} \\ \mathbf{O}_{4x3} & \mathbf{T}(\boldsymbol{q}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\upsilon}_b \\ \boldsymbol{\omega}_b \end{bmatrix}$$

where $\mathbf{O}_{\mathrm{i}x\mathrm{j}}$ is a $i \times j$ matrix of zeros.

The kinematic model presented above does not take the physical constraints that are typically present with AUVs, which is the existence of nonholonomic constraints. Nonholonomic constraints are kinematic constraints that involve generalized coordinates and velocities of the system of interest. For example, wheeled vehicles like cars and differential mobile robots cannot move the in lateral direction in typical operating conditions, for the friction between the tires and the ground prevents slipping from occurring [51]. Also, those vehicles cannot translate normal to the ground.

Autonomous Underwater Vehicles, which are usually neutrally buoyant, exhibit the notion of anholonomy. Thrusters propel the vehicle in the forward direction, while fins or thrusters exert moments on the vehicle to give it the ability to turn. Like wheeled vehicles, the motion of AUVs is only possible in surge and not in sway or heave in addition to the possible angular motions of pitch, roll, and yaw that depend on the thruster/fin arrangement installed. The kinematic model can then be rewritten as in equation (8).

$$\begin{bmatrix} \dot{\mathbf{P}}_n \\ \dot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b^n(\boldsymbol{q})_{1x3} & \mathbf{O}_{1x3} \\ \mathbf{O}_{4x3} & \mathbf{T}(\boldsymbol{q}) \end{bmatrix} \begin{bmatrix} u \\ \boldsymbol{\omega}_b \end{bmatrix} \tag{8}$$

where $\mathbf{R}_b^n(\boldsymbol{q})_{1x3}$ is the first row of $\mathbf{R}_b^n(\boldsymbol{q})$, which describes the relation between the surge motion and the change of position coordinate in the n-frame. In the context of navigation, these constraints can be exploited to provide pseudo zero-velocity measurements in the constrained direction for a navigation filter to make use of.

3.2. The Dynamic Equations of Motion

3.2.1. Rigid-body kinetics. The vehicle is taken to be a rigid body for which the dynamics are to be described; that is, the distance between any two points in the object is taken to be invariant. The general motion of a rigid body can be described through the translation and rotation of a point in that body. The position of vector describing such point can take the following form:

$$\boldsymbol{r}_G = \boldsymbol{r}_O + \boldsymbol{r}_{G/O} \tag{9}$$

where $\boldsymbol{r}_O$ is the position vector to a point of observation, $\boldsymbol{r}_G$ is the position vector to the center of gravity (CG) of the vehicle, and $\boldsymbol{r}_{G/O}$ is the relative position of CG with respect to point O. Figure 10 depicts the aforementioned quantities.

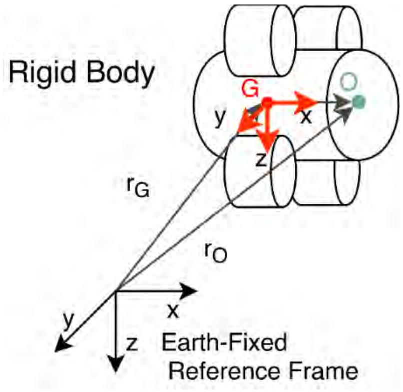


Figure 10: Coordinate Frames of the AUV

The velocity is then the derivative of the position vector, which comprises of a translation and a rotation representing a change in magnitude and a change in direction shown as:

$$\boldsymbol{v}_G = \frac{\boldsymbol{d}}{\boldsymbol{dt}}(\boldsymbol{r}_G) = \frac{\boldsymbol{d}}{\boldsymbol{dt}}(\boldsymbol{r}_O + \boldsymbol{r}_{G/O}) = \boldsymbol{v}_O + \dot{\boldsymbol{r}}_{G/O} + \boldsymbol{\omega} \times \boldsymbol{r}_{G/O} = \boldsymbol{v}_O + \boldsymbol{\omega} \times \boldsymbol{r}_{G/O} \quad (10)$$

where $\dot{\boldsymbol{r}}_{G/O} = 0$ for a rigid body when expressed in the b-frame.

Similarly, the acceleration of the rigid body through another differentiation is shown to be:

$$\begin{aligned} \boldsymbol{a}_G &= \frac{\boldsymbol{d}}{\boldsymbol{dt}}(\boldsymbol{v}_G) = \dot{\boldsymbol{v}}_G + \boldsymbol{\omega} \times \boldsymbol{v}_G \\ &= \dot{\boldsymbol{v}}_O + \boldsymbol{\omega} \times \boldsymbol{v}_O + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_{G/O}) \end{aligned} \quad (11)$$

It is common practice to place the body-fixed coordinate frame at the center of gravity of the vehicle. That will simplify equation (11) since $\boldsymbol{r}_{G/O} = [0\ 0\ 0]^T$ to the following form, written in light of the Newton-Euler equations:

$$\begin{gathered} \frac{\boldsymbol{d}}{\boldsymbol{dt}}(m\,\boldsymbol{v}_G) = \boldsymbol{f}_G \\ \mathrm{m}\,\boldsymbol{a}_G = m\,(\dot{\boldsymbol{v}}_O + \boldsymbol{\omega} \times \boldsymbol{v}_O) = \boldsymbol{f}_G \\ \frac{\boldsymbol{dH}}{\boldsymbol{dt}} + \boldsymbol{r}_G \times m\,\boldsymbol{a}_G = \boldsymbol{M}_G \end{gathered} \quad (12)$$

$$\frac{\boldsymbol{d}}{\boldsymbol{dt}}(I_O\,\boldsymbol{\omega}) + \boldsymbol{r}_G \times m(\dot{\boldsymbol{v}}_O + \boldsymbol{\omega} \times \boldsymbol{v}_O) = I_c\,\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (I_c\,\boldsymbol{\omega}) = \boldsymbol{M}_G$$

The devised equations of motion describing the 6-DOF vehicle can be rearranged and written in the standard form used by the society of naval architects and marine engineers in equation (13). The following utilizes the previously defined standard notations of $\boldsymbol{\eta}$ and $\boldsymbol{\nu}$.

$$\mathbf{M}_{RB}\,\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau} \tag{13}$$

where the rigid-body mass matrix, $\mathbf{M}_{RB}$, consists of the total mass of the vehicle, m, as well as the second moment of area, $\mathbf{I}_b$, taken with respect to the b-frame principle axes (moment of inertia), $\mathbf{I}_{3\text{x}3}$ is the identity matrix of size $\mathbb{R}^{3x3}$, and $\boldsymbol{S}(\boldsymbol{r}_\text{g})$ is the previously defined skew-symmetric matrix. Equation (14) presents the form.

$$\mathbf{M}_\text{RB} = \begin{bmatrix} \text{m}\,\mathbf{I}_{3\text{x}3} & -m\boldsymbol{S}(\boldsymbol{r}_\text{g}) \\ \text{m}\boldsymbol{S}(\boldsymbol{r}_\text{g}) & \mathbf{I}_b \end{bmatrix}$$
$$\mathbf{I}_b = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \tag{14}$$

When the observation point, $\boldsymbol{r}_\text{g}$, is taken to be the center of the body-fixed coordinate frame, the matrix reduces to:

$$\mathbf{M}_\text{RB} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & I_{xy} & I_{xz} \\ 0 & 0 & 0 & I_{xx} & I_{yy} & I_{yz} \\ 0 & 0 & 0 & I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \tag{15}$$

The Coriolis and Centripetal forces matrix is a compact form of the resulting equations of motion from the derived rigid body dynamics of equation. This matrix factorization can also be utilized to account for the added mass contribution to Coriolis and Centripetal forces. The matrices corresponding to body and added mass Coriolis and Centripetal forces are presented below:

$$\mathbf{C}(\boldsymbol{\nu}) = \begin{bmatrix} \mathbf{0}_{3\text{x}3} & -\mathbf{S}(\mathbf{M}_{11}\boldsymbol{\nu}_1 + \mathbf{M}_{12}\boldsymbol{\nu}_2) \\ -\mathbf{S}(\mathbf{M}_{11}\boldsymbol{\nu}_1 + \mathbf{M}_{12}\boldsymbol{\nu}_2) & -\mathbf{S}(\mathbf{M}_{12}^\text{T}\boldsymbol{\nu}_1 + \mathbf{M}_{22}\boldsymbol{\nu}_2) \end{bmatrix} \tag{16}$$

$$\mathbf{C}_A(\boldsymbol{\nu}) = \begin{bmatrix} \mathbf{0}_{3x3} & -\mathbf{S}\left(\mathbf{M}_{A_{11}}\boldsymbol{\nu}_1 + \mathbf{M}_{A_{12}}\boldsymbol{\nu}_2\right) \\ -\mathbf{S}\left(\mathbf{M}_{A_{11}}\boldsymbol{\nu}_1 + \mathbf{M}_{A_{12}}\boldsymbol{\nu}_2\right) & -\mathbf{S}\left(\mathbf{M}_{A_{21}}\boldsymbol{\nu}_1 + \mathbf{M}_{A_{22}}\boldsymbol{\nu}_2\right) \end{bmatrix}$$

3.2.2. Hydrostatics. Hydrostatic effects on the vehicle comprise the restoring forces matrix, which describes two forces. The first one is the effect of gravitational force acting through the center of gravity and the resultant moment about it, while the second one is the effect of buoyancy in the form of force acting through the center of buoyancy and corresponding moment about it. Those two effects have a direct effect on the natural stability of the vehicle since the resulting moments are dependent on the attitude of the vehicle as well as the locations of the centers of buoyancy and gravity. $\mathbf{f}_{\mathrm{b}}^{n}$ and $\mathbf{f}_{\mathrm{g}}^{n}$ in equation set (17) represent the buoyancy and weight forces, respectively.

$$\begin{aligned} \mathbf{f}_{\mathrm{b}}^{n} &= -\begin{bmatrix} 0 \\ 0 \\ \mathrm{B} \end{bmatrix}, \qquad \mathrm{B} = \rho \mathrm{g} \nabla \\ \mathbf{f}_{\mathrm{g}}^{n} &= \begin{bmatrix} 0 \\ 0 \\ \mathrm{W} \end{bmatrix}, \qquad \mathrm{W} = \mathrm{mg} \end{aligned} \qquad (17)$$

where ρ is the density of the water, g is the gravitational constant, ∇ is the submerged volume of the vehicle, and m is the mass of the vehicle.

Representing the effects in the body-fixed frame through proper rotations yields equations set (18), where $\mathbf{R}_b^n(\boldsymbol{q})^{\mathrm{T}}$ represents the rotation matrix mapping from the n-frame to the b-frame, $\boldsymbol{r}_{\mathrm{b}}^{\mathrm{b}}$ is the location of the center of buoyancy in b-frame, and $\boldsymbol{r}_{\mathrm{g}}^{\mathrm{b}}$ is the location of the center of gravity in b-frame.

$$\begin{aligned} \mathbf{f}_{\mathrm{b}}^{\mathrm{b}} &= \mathbf{R}_b^n(\boldsymbol{q})^{\mathrm{T}}\, \mathbf{f}_{\mathrm{b}}^{n} \\ \mathbf{f}_{\mathrm{g}}^{\mathrm{b}} &= \mathbf{R}_b^n(\boldsymbol{q})^{\mathrm{T}}\, \mathbf{f}_{\mathrm{g}}^{n} \\ \mathcal{G}(\boldsymbol{\eta}) &= -\begin{bmatrix} \mathbf{f}_{\mathrm{b}}^{\mathrm{b}} + \mathbf{f}_{\mathrm{g}}^{\mathrm{b}} \\ \boldsymbol{r}_{\mathrm{b}}^{\mathrm{b}} \times \mathbf{f}_{\mathrm{b}}^{\mathrm{b}} + \boldsymbol{r}_{\mathrm{g}}^{\mathrm{b}} \times \mathbf{f}_{\mathrm{g}}^{\mathrm{b}} \end{bmatrix} = -\begin{bmatrix} \mathbf{R}_b^n(\boldsymbol{q})^{\mathrm{T}}\left(\mathbf{f}_{\mathrm{b}}^{\mathrm{n}} + \mathbf{f}_{\mathrm{g}}^{\mathrm{n}}\right) \\ \boldsymbol{r}_{\mathrm{b}}^{\mathrm{b}} \times \mathbf{R}_b^n(\boldsymbol{q})^{\mathrm{T}}\, \mathbf{f}_{\mathrm{b}}^{n} + \boldsymbol{r}_{\mathrm{g}}^{\mathrm{b}} \times \mathbf{R}_b^n(\boldsymbol{q})^{\mathrm{T}}\, \mathbf{f}_{\mathrm{g}}^{n} \end{bmatrix} \\ &= \begin{bmatrix} 2(\mathrm{B}-\mathrm{W})(\epsilon_1\,\epsilon_3 - \eta\,\epsilon_2) \\ 2(\mathrm{B}-\mathrm{W})(\epsilon_2\,\epsilon_3 + \eta\,\epsilon_1) \\ (\mathrm{W}-\mathrm{B})(1 - 2(\epsilon_1^2 + \epsilon_2^2)\,) \\ r_{b_x}(2\mathrm{B})(\epsilon_1\,\epsilon_2 + \eta\,\epsilon_3) \\ r_{b_x}(\mathrm{B})\left(1 - 2(\epsilon_1^2 + \epsilon_2^2)\right) \\ r_{b_x}(2\mathrm{B})(\epsilon_2\,\epsilon_3 - \eta\,\epsilon_1) \end{bmatrix} \end{aligned} \qquad (18)$$

3.2.3. Hydrodynamics. Hydrodynamic effects on the vehicle are twofold. They are added mass in addition to damping. Hydrodynamic added mass is a component that is a consequence of the fluid-structure interaction between the vehicle and the surrounding environment. This phenomenon results in a change to the apparent mass of the vehicle that impedes the acceleration of the vehicle. In the devised model, the effect of added mass is accounted for as a change in the apparent mass matrix of the vehicle as in equation (19).

$$\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A$$
$$\mathbf{M}_A = \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} \tag{19}$$

where $\mathbf{M}_{RB}$ represents the previously defined rigid body mass matrix. Each variable in $\mathbf{M}_A$ denotes a force or a moment in the direction of the variable due to an acceleration denoted in the subscript. Table 2 explains the force and moment notation. To illustrate, $Z_{\dot{r}}$ is a force the vehicle experiences in the z-axis (down in b-frame terminology) due to an angular acceleration $\dot{r}$ about the z-axis.

The empirical Drag model shown in equation (20) depicts the highly coupled nature of drag. The model accounts for the quadratic and linear components of drag phenomenon [10].

$$\mathrm{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \begin{bmatrix} |\boldsymbol{\nu}|^{\mathrm{T}}\mathbf{D}_{n1}\boldsymbol{v} \\ |\boldsymbol{\nu}|^{\mathrm{T}}\mathbf{D}_{n2}\boldsymbol{v} \\ |\boldsymbol{\nu}|^{\mathrm{T}}\mathbf{D}_{n3}\boldsymbol{v} \\ |\boldsymbol{\nu}|^{\mathrm{T}}\mathbf{D}_{n4}\boldsymbol{v} \\ |\boldsymbol{\nu}|^{\mathrm{T}}\mathbf{D}_{n5}\boldsymbol{v} \\ |\boldsymbol{\nu}|^{\mathrm{T}}\mathbf{D}_{n6}\boldsymbol{v} \end{bmatrix} + \begin{bmatrix} X_u \\ Y_v \\ Z_w \\ K_p \\ M_q \\ N_r \end{bmatrix} \circ \boldsymbol{\nu} \tag{20}$$

where $\boldsymbol{\nu}$ is the vector of linear and angular velocities, each $\mathbf{D}_{ni}$ is a $\mathbb{R}^{6x6}$ matrix that captures the coupling between the degrees of freedom and affects the i^{th} degree of freedom, and the ∘ operator represents the Hadamard product (element-wise multiplication). The notation used here for the forces is similar to that of the added mass forces. For example, X_u is the coefficient of drag force in the x-direction due to a velocity u in the x-direction. This model has 222 parameters to be identified.

While the aforementioned model is expected to be representative of drag phenomenon, it would be difficult to use because of the extensive number of parameters. The identification process in this case might not converge, or it might require a lot of independent and representative experimental data points for any identification algorithm to converge. More information on the identification process is presented in the next section. As a consequence of the difficulty associated with the use of the fully coupled drag model, a simpler approach needs to be followed. This approach is one that is representative of the phenomenon, yet it is simple enough with a fewer number of parameters to identify. The vehicle does not move in a manner in which all degrees of freedom are changing at the same time, so uncoupling the drag components will be done. The uncoupled degrees of freedom model exhibits the basic nature of drag; Potential drag is shown to be proportional to the square of vehicle velocity, while skin friction is linearly proportional to the vehicle velocity. This is shown in equation (21).

$$\mathbf{D}(\boldsymbol{\nu}) = \begin{bmatrix} X_{|u|u}|u| \\ Y_{|v|v}|v| \\ Z_{|w|w}|w| \\ K_{|p|p}|p| \\ M_{|q|q}|q| \\ N_{|r|r}|r| \end{bmatrix} + \begin{bmatrix} X_u \\ Y_v \\ Z_w \\ K_p \\ M_q \\ N_r \end{bmatrix} \tag{21}$$

where the notation for linear drag is the same as in the coupled degrees of freedom model, and $X_{|u|u}$ is the quadratic drag force in the x-direction due to a velocity u in the x-direction. The $\mathbf{D}(\boldsymbol{\nu})$ is multiplied by $\boldsymbol{\nu}$ in the equations of motion.

3.2.4. External forces. The external forces exerted on the vehicle are those arising from actuation through thrusters and those arising from the environmental currents. Actuation of the vehicle is achieved through thrusters, which are modeled by the thruster allocation matrix. Actuation of each thruster results in a thrust force applied on the location of the thruster, a moment of that force about the center of gravity of the vehicle, and a reactional torque on the vehicle that is a result of the rotary motion of the propeller. Equation (22) presents the external forces vector where f_i is the force exerted by thruster i and R is the perpendicular distance from the center of gravity to any of the thrusters.

$$\boldsymbol{\tau}^b = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{r} \times \boldsymbol{f} \end{bmatrix} = \begin{bmatrix} f_1 + f_2 + f_3 + f_4 \\ 0 \\ 0 \\ 0 \\ R\,(-f_1 - f_2 + f_3 + f_4) \\ R\,(f_1 - f_2 + f_3 - f_4) \end{bmatrix} \tag{22}$$

The thruster allocation matrix maps the force exerted by each of the actuators to the resulting forces, moments, and torques on the vehicle. This matrix is shown below:

$$\boldsymbol{\tau}^b = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -R & -R & R & R \\ R & -R & R & -R \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \tag{23}$$

Often, thruster performance curves relate the force, f_i, that any of the thrusters needs to exert to the input electrical signal sent to the thrusters. Lookup tables or fitting tools are then used to send the signal required to generate the desired force.

Finally, considering all the above, the model can be written as:

$$\mathbf{M}\,\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{\mathcal{G}}(\boldsymbol{\eta}) = \boldsymbol{\tau}_{\boldsymbol{actuat}\mathrm{or}} \tag{24}$$

The form in equation (24) can be expanded to:

$$\begin{bmatrix} m\,\dot{u} + m(w\,q - v\,r) + X_u u + X_{|u|u}|u|u + 2(\mathrm{B} - \mathrm{W})(\epsilon_1\,\epsilon_3 - \eta\,\epsilon_2) \\ m\,\dot{v} + m(u\,r - w\,q) + Y_v v + Y_{|v|v}|v|v + 2(\mathrm{B} - \mathrm{W})(\epsilon_2\,\epsilon_3 + \eta\,\epsilon_1) \\ m\,\dot{w} + m(v\,p - u\,q) + Z_w w + Z_{|w|w}|w|w + (\mathrm{W} - \mathrm{B})(1 - 2(\epsilon_1^2 + \epsilon_2^2)\,) \\ I_{xx}\dot{p} + I_{xy}\dot{q} + I_{xz}\dot{r} + \left(I_{zz} - I_{yy}\right) q\,r + K_p p + K_{|p|p}|p|p + r_{b_x}(2\mathrm{B})(\epsilon_1\,\epsilon_2 + \eta\,\epsilon_3) \\ I_{yx}\dot{p} + I_{yy}\dot{q} + I_{yz}\dot{r} + (I_{xx} - I_{zz})\,p\,r + M_q q + M_{|q|q} + r_{b_x}(\mathrm{B})\left(1 - 2(\epsilon_1^2 + \epsilon_2^2)\right) \\ I_{zx}\dot{p} + I_{zy}\dot{q} + I_{zz}\dot{r} + \left(I_{yy} - I_{xx}\right) p\,q + N_r r + N_{|r|r} + r_{b_x}(2\mathrm{B})(\epsilon_2\,\epsilon_3 - \eta\,\epsilon_1) \end{bmatrix} \tag{25}$$

$$= \begin{bmatrix} f_1 + f_2 + f_3 + f_4 \\ 0 \\ 0 \\ 0 \\ R\,(-f_1 - f_2 + f_3 + f_4) \\ R\,(f_1 - f_2 + f_3 - f_4) \end{bmatrix}$$

Chapter 4. Parameter Identification

The problem of parameter identification appears because some parameters of the vehicle model are not known with great certainty, while other parameters are not known at all. To exemplify, the mass, the inertia and locations of the centers of gravity/buoyancy are parameters that can be estimated thorough software packages with some certainty; however, the quality of materials, the manufacturing process, and the assembly process might impose unforeseen changes in the parameters mentioned. Moreover, parameters like the added mass coefficients and damping coefficients are financially very costly to estimate since they require specialized experimental setups. Nevertheless, finite element computational methods allow the identification of these parameters, and experimental validation often shows that the finite element analysis gives realistic results when applied properly.

4.1. Proposed Vehicle Classification

The proposed vehicle, which is shown in Figure 11, is a man-portable class Autonomous Underwater Vehicle (AUV) with a quadcopter-like thruster arrangement. The vehicle in its stable configuration has the axis of the cylinder horizontal, unlike a quadcopter. This stable configuration is induced by controlling the locations of the centers of gravity and buoyancy of the vehicle.

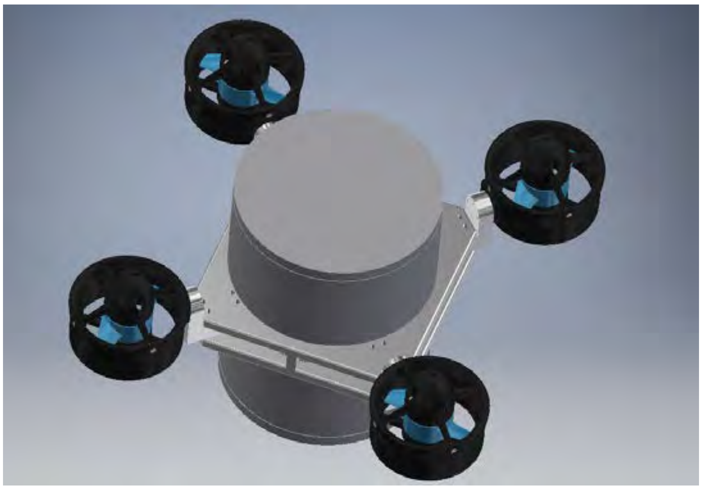


Figure 11: Vehicle CAD model

4.2. Mass Matrix Identification

The components relevant to construct this matrix are total mass of the vehicle, the inertia tensor, and the center of gravity. Knowledge of the materials involved as well as their properties allow for Autodesk Inventor to compute the desired quantities. It is of significance to mention that the center of buoyancy is also calculable through the software package. That is doable by replacing all the material properties with those of water and re-evaluating the center of gravity, which will describe the center of buoyancy. The quantities are tabulated below.

Table 3: AUV properties

Quantity	value
Mass	$4.82\ kg$
Inertia	$\begin{bmatrix} 0.06346 & -8.680x10^{-6} & -5.175x10^{-5} \\ -8.680x10^{-6} & 0.05195 & 5.200x10^{-7} \\ -5.175x10^{-5} & 5.200x10^{-7} & 0.05281 \end{bmatrix} kg.m^2$
Center of Gravity	$[0 \quad 0 \quad 0]^T m$
Center of Buoyancy	$[1 \times 10^{-4} \quad 0 \quad -0.030]^T m$

It is evident that the nearly symmetrical shape of the AUV caused the Inertia tensor to almost be purely diagonal. In addition, the centers of gravity and buoyancy result in natural stability in the roll and pitch motions.

4.3. Damping Parameters Identification

Commercial software package ANSYS was used to obtain an estimate of the different drag components. The general methodology followed is that of [52]. A 3D model for the vehicle was created in Autodesk Inventor for simulation purposes. Two scenarios were considered in the simulation environment; translational motion in the directions of the three principal axes as well as rotational motion about them. A multitude of flow velocity conditions for the two scenarios were simulated. Those scenarios are referred to as scenario 1 and scenario 2.

Figure 12 and Figure 13 show the meshed model in the ANSYS environment in the two scenarios mentioned. An enclosure that is 4-5 times as large as the vehicle was created to model each scenario, since general fluid simulation guidelines suggest a minimum of 2-3. Mesh statistics are presented in Table 4. The required level of refinement of the discretized model was determined through a mesh convergence study. An additional level of refinement was applied to the model to run the scenarios for good measure.

Table 4: Mesh statistics

	Number of elements	Number of nodes
Scenario 1	405354	82067
Scenario 2	1902365	365667

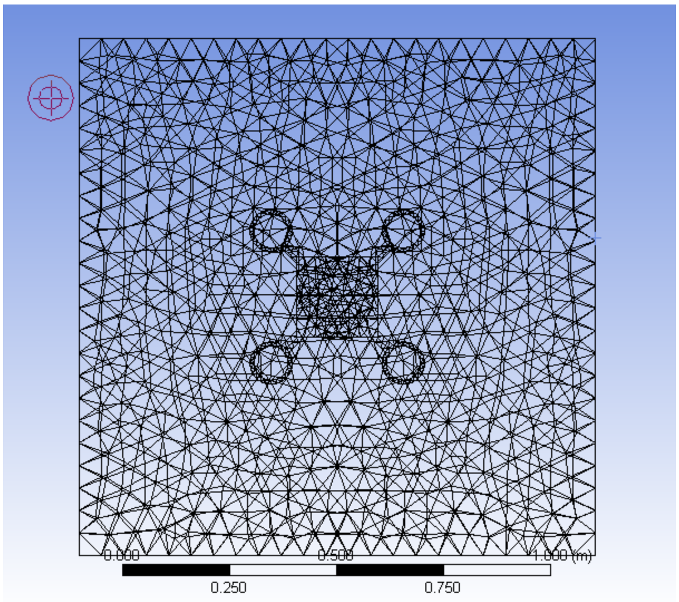


Figure 12: Mesh Front View of AUV

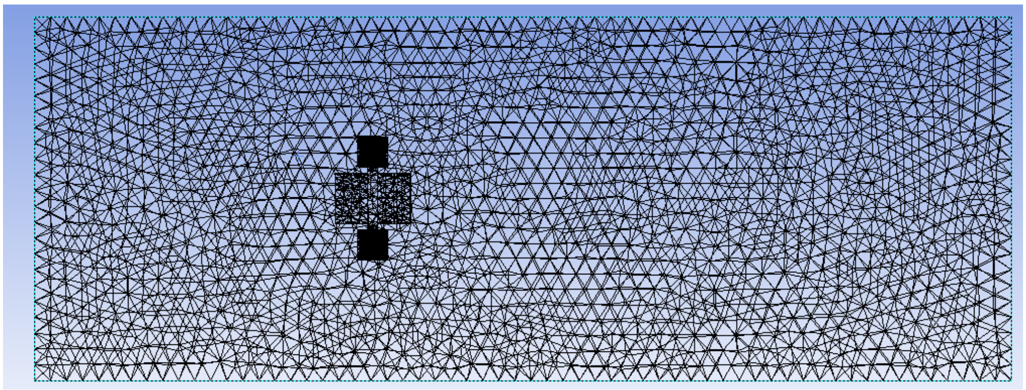


Figure 13: Mesh Side View of AUV

As for the boundary conditions, the first scenario was simulated in the Fluent ANSYS package. It consisted of an enclosure with the AUV model inside. The former domain was chosen to be a water fluid domain with auto initialization, and the latter domain was a no-slip boundary condition. The enclosure had an inlet with a specified velocity on one side, and it had a zero relative-pressure outlet on the other side. All the other sides of the enclosure were free boundaries with zero shear interaction with the fluid domain. A parametric study was implemented for several inlet velocities. The resulting total pressure fields were integrated over the entire boundary of the AUV for every inlet velocity to compute the total drag force. Figure 14 and Figure 16 present the pressure contour field on the AUV boundary as well as the velocity stream lines throughout the fluid boundary. The figures correspond to a case of translational flow with 1 m/s inlet velocity. For brevity, cases of the flow in the other directions are not presented.

The second scenario was simulated in the CFX ANSYS package. It also consisted of an enclosure and an AUV model inside. Here, rotary motion of the AUV was realized through mesh motion of the AUV domain. The former domain was chosen to be a water fluid domain with auto initialization, and the latter domain was a no-slip boundary condition with mesh motion handling. All the sides of the enclosure were free boundaries with zero shear interaction with the fluid domain. A parametric study was implemented for several rotational velocities of the AUV domain. The resulting total pressure fields were integrated over the entire boundary of the AUV for every inlet velocity to compute the total drag moment. It is of interest to indicate that CFX, unlike FLUENT, is a time domain solver and not a steady state one when mesh motion handling is enabled. Figure 15 and Figure 17 report the pressure contour field on the AUV boundary as well as the velocity stream lines throughout the fluid boundary for two flows. For brevity, other cases are not presented.

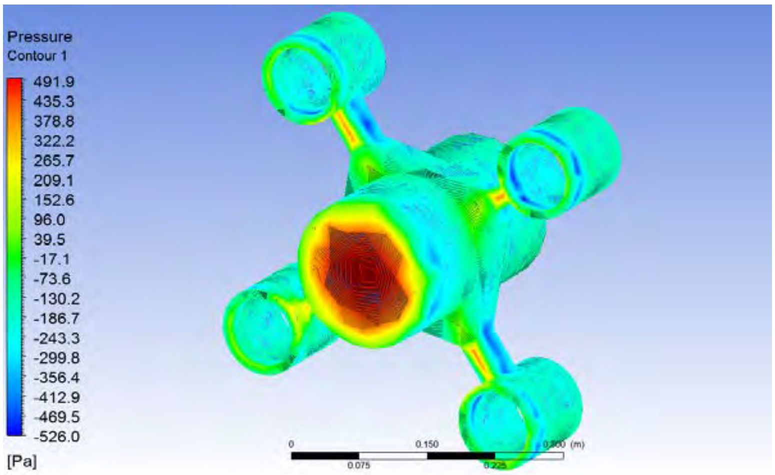


Figure 14: Pressure Contour Plot for Translational Motion

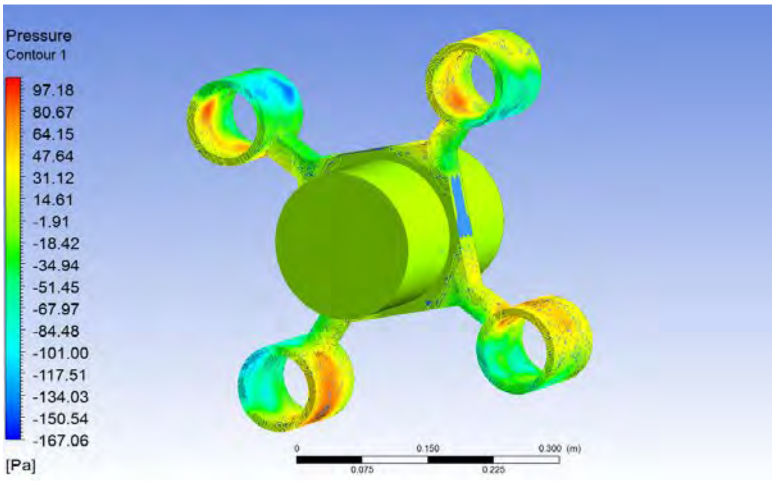


Figure 15: Pressure Contour Plot for Rotational Motion

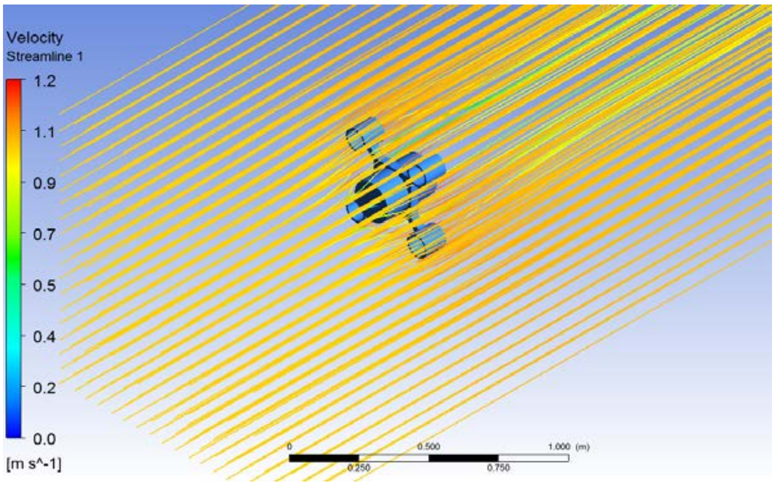


Figure 16: Flow Velocity Streamlines for Translational Motion

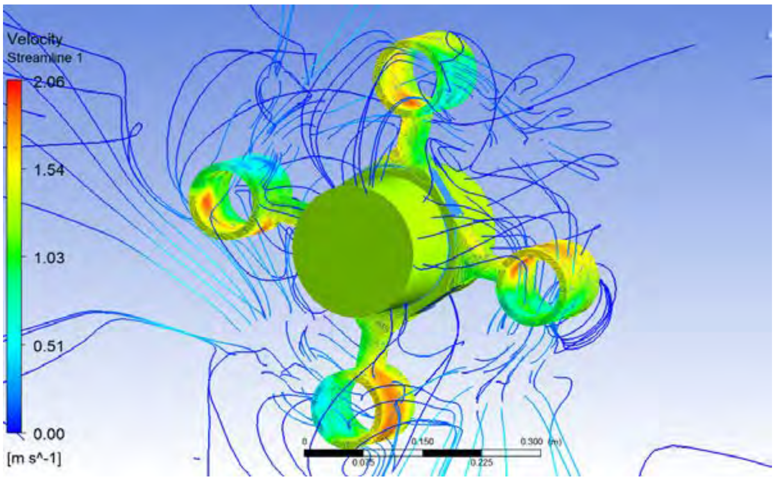


Figure 17: Flow Velocity Streamlines for Rotational Motion

Figure 18 shows the drag forces and drag torques on the vehicle as a function of the linear or angular velocity of the vehicle. To get the quadratic and linear drag coefficients, the ANSYS data was fitted to quadratic functions to find the drag profiles. The drags coefficients documented in Table 5.

Table 5: Identified Drag Model Parameters

	1	**2**	**3**	**4**	**5**	**6**
Linear drag coefficients	1.073	1.168	1.168	−0.0219	0.0504	0.0504
Quadratic drag coefficients	23.382	28.770	28.770	0.0471	0.1533	0.1533

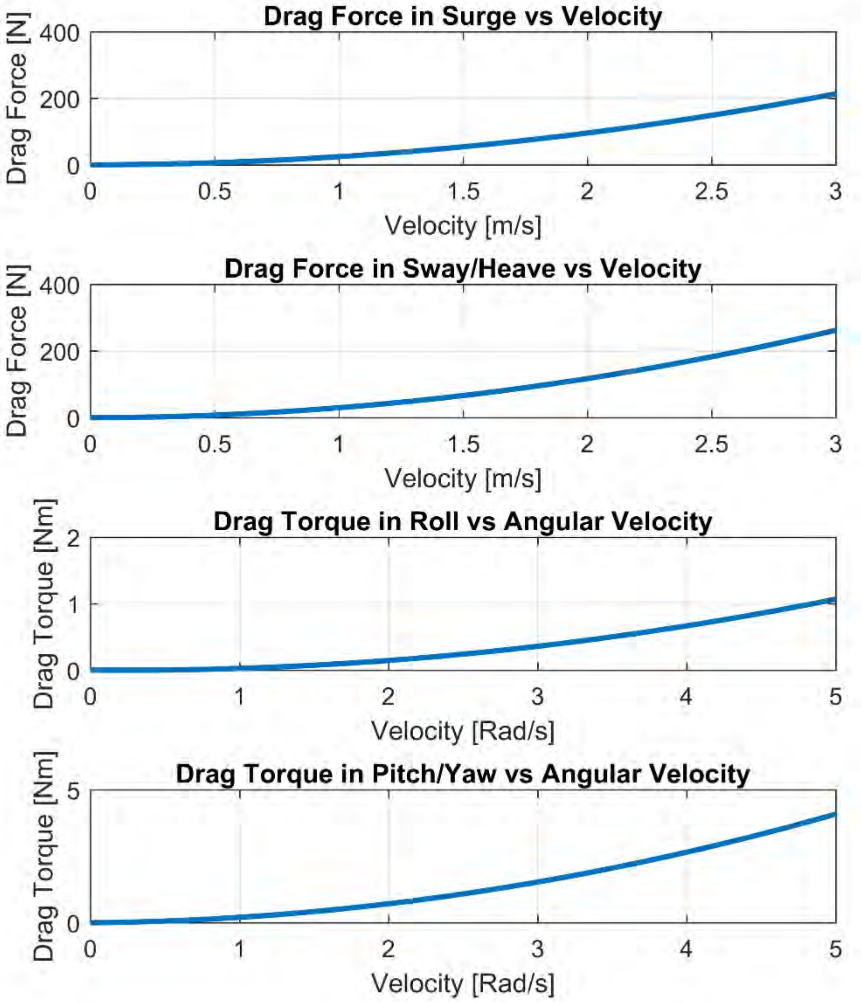


Figure 18: Drag Identification Results

4.4. Hydrodynamic Mass Identification

To estimate the hydrodynamic parameters of the vehicle, experimental testing and/or simulation through finite element packages is necessary. Here, the relevant parameters are identified through the ANSYS software suite.

The added mass coefficients for a slender body can be derived through strip theory methodology like in [53]. Many vehicles, however, are not slender in shape, and the proposed vehicle is no exception. Therefore, a more fundamental view of the phenomenon is taken to identify the added mass parameters. Since the added mass determines the necessary work required to change the kinetic energy of the fluid surrounding a body in motion, one can relate the kinetic energy of the fluid surrounding the body and the kinetic energy of the body [21]. The kinetic energy of a fluid volume can be calculated through the volume integral:

$$\begin{aligned} \mathrm{KE} &= \frac{1}{2}\int \left(u_x^2 + u_y^2 + u_z^2\right) dm \\ &= \frac{\rho}{2}\int_V \left(u_x^2 + u_y^2 + u_z^2\right) dV \\ &= \frac{\rho}{2}\iiint \left(u_x^2 + u_y^2 + u_z^2\right) dxdydz \end{aligned} \tag{26}$$

where u_i, $i = x, y, z$ are the Cartesian components of the fluid velocity, and V is the entire fluid domain. The integral above assumes stationary fluid far from the body. For a flowing fluid with a constant uniform velocity, Wi, the integral becomes:

$$\mathrm{KE} = \frac{\rho}{2}\int_V (u_i - W_i)(u_i - W_i)\, dV \tag{27}$$

According to potential theory, the added mass can be computed by equating the kinetic energy of the fluid relative to the body as in equations (26) and (27) to an equivalent body kinetic energy.

$$\begin{gathered} \mathrm{KE}_{\mathrm{fluid}} = \mathrm{KE}_{\mathrm{body}} \\ \frac{\rho}{2}\int_V (u_i - W_i)(u_i - W_i)\, dV = \frac{1}{2} m_a U^2 \end{gathered} \tag{28}$$

where U is the instantaneous velocity of the body moving in the fluid.

According to [54], in context of potential fluid flow, the added mass matrix for a symmetric body must be symmetric. The aforementioned integrals were implemented in ANSYS for the simulations described earlier in the chapter. First, the methodology is tested through simulating the flow over a primitive shape for which the added mass is known. Then, the added mass of the vehicle can be estimated using the validated methodology.

The added mass for a cylinder is shown through analytical and empirical approaches to take the form in equation set (29) [18, 37], where $X_{\dot{u}}$ is the only empirical form. The cylinder had a diameter of $0.4\ m$ and a length of $0.3\ m$.

$$\boldsymbol{M}_{\boldsymbol{Cylinder}_{\boldsymbol{Added}}} \begin{cases} X_{\dot{u}} = 1.08\ \rho\pi r^{2.93} l^{0.07} \\ Y_{\dot{v}} = \pi\rho r^2 l \\ Z_{\dot{w}} = \pi\rho r^2 l \\ K_{\dot{p}} = 0 \\ M_{\dot{q}} = \dfrac{1}{12}\pi\rho r^2 l^3 \\ N_{\dot{r}} = \dfrac{1}{12}\pi\rho r^2 l^3 \end{cases} \tag{29}$$

where r is the radius of the cylinder, l is the length of the cylinder, and ρ is the density of the fluid in which the body is submerged.

The results of the added mass identification results for a cylinder are documented in Table 6.

Table 6: Added Mass Results for a Cylinder

	$\boldsymbol{X_{\dot{u}}}\ [\boldsymbol{kg}]$	$\boldsymbol{Y_{\dot{v}}, Z_{\dot{w}}}\ [\boldsymbol{kg}]$	$\boldsymbol{K_{\dot{p}}}\ [\boldsymbol{kg.m^2}]$	$\boldsymbol{M_{\dot{q}}, N_{\dot{r}}}[\boldsymbol{kg.m^2}]$
Real Added Mass	32.200	37.699	0	0.283
ANSYS Added Mass	34.996	38.646	0.085	0.328

The methodology gives accurate results for the translational degrees of freedom. However, the rotational degrees of freedom did not exactly closely match the analytical solution. This can be explained by the mesh not being fine enough in the CFX module that utilizes moving mesh to solve the problem. The module used does not support smaller mesh sizes.

The added mass matrix for the vehicle was then simulated. Convergence results for the simulated runs are shown in Figure 19. As the mesh became finer, the solution converged to the values presented in Table 7. The added mass in the roll motion is observed to be somewhat large, and the convergence analysis suggests that improvement is possible.

Table 7: Added Mass Results for the Vehicle

	$X_{\dot{u}}\ [kg]$	$Y_{\dot{v}}, Z_{\dot{w}}\ [kg]$	$K_{\dot{p}}\ [kg.m^2]$	$M_{\dot{q}}, N_{\dot{r}} [kg.m^2]$
Real Added Mass	38.583	31.258	0.9285	0.116

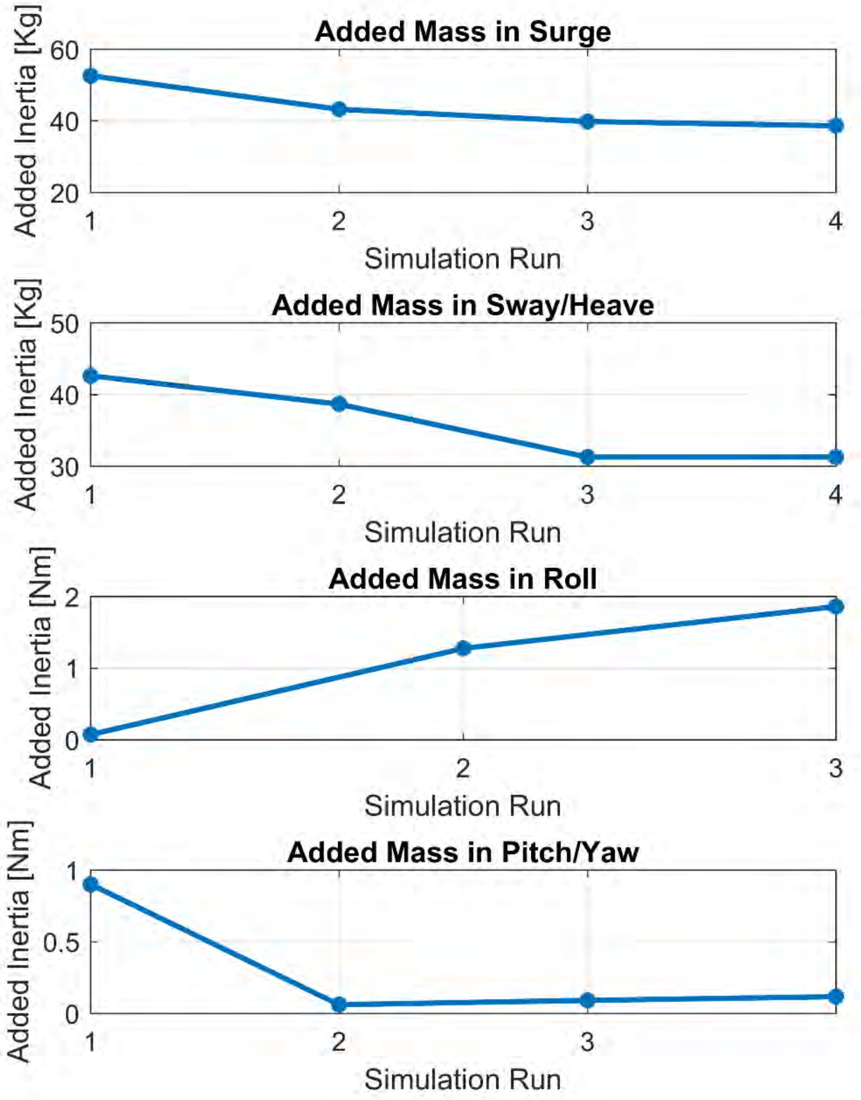


Figure 19: Added Mass Convergence Results

4.5. Fine-tuning of the Parameters

It is often of interest to improve upon the estimated parameters when experimental data is present. Parameter identification can be used to fine-tune the parameters of the vehicle [14, 15, 19]. The identification vector accommodates all the variables to be identified, which are attributed to mass, inertia, center of buoyancy, damping, and added mass. Nonlinear least square optimization techniques can be used to identify the parameters that best fit the simulation results to the experimental ones. The Levenberg-Marquardt algorithm is a numerical routine that operates in a hybrid fashion between Gradient-Decent and Gauss-Newton numerical routines.

Let $\boldsymbol{f}$ be a 6-row vector containing the difference between the experimental response, $\mathbf{y}$, and the simulation results, as follows:

$$\boldsymbol{f} = \mathbf{y} - (\mathbf{M}\,\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{\mathcal{G}}(\boldsymbol{\eta}) - \boldsymbol{\tau}_{\boldsymbol{actuat}\mathrm{or}}) \tag{30}$$

Let $\mathbf{J}$ be the Jacobian matrix, where the partial derivatives of $\boldsymbol{f}$ with respect to the elements of $\boldsymbol{\lambda}$ are evaluated at the values of the vector at every iteration i:

$$\mathbf{J}_{\boldsymbol{i}} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial \lambda_1} & \frac{\partial f_1}{\partial \lambda_2} & \cdots & \frac{\partial f_1}{\partial \lambda_n} \\ \frac{\partial f_2}{\partial \lambda_1} & \frac{\partial f_2}{\partial \lambda_2} & \cdots & \frac{\partial f_2}{\partial \lambda_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial \lambda_1} & \frac{\partial f_n}{\partial \lambda_2} & \cdots & \frac{\partial f_n}{\partial x_\lambda} \end{bmatrix}\right|_{\boldsymbol{\lambda} = \boldsymbol{\lambda_i}} \tag{31}$$

Approximated by the linearization, $\boldsymbol{f}$ can be rewritten with $\boldsymbol{\delta}$ representing a change in the parameter vector $\boldsymbol{\lambda}$, as:

$$\boldsymbol{f}(\boldsymbol{\lambda_i} + \boldsymbol{\delta}) \approx \boldsymbol{f}(\boldsymbol{\lambda_i}) + \boldsymbol{J_i}\boldsymbol{\delta} \tag{32}$$

The Levenberg-Marquardt numerical routine yields a minimum $\boldsymbol{f}$, given by:

$$(\boldsymbol{J^T J} + \boldsymbol{\psi I})\boldsymbol{\delta} = \mathbf{j}^{\mathrm{T}}\boldsymbol{f} \tag{33}$$

where $\boldsymbol{\psi}$ is a parameter that is varied by the algorithm between iterations such that convergence is reached, and $\boldsymbol{\delta}$ is the parameter that linear system can be solved for to be added to $\boldsymbol{\lambda}$. Many software packages like MATLAB have an implementation of the described routine.

Chapter 5: Guidance and Control

This chapter deals with the motion planning task and the associated control to force the vehicle to follow some trajectory. The devised control method is twofold; a high-level controller and a low-level controller. Given an input trajectory, the high-level controller, derived from the vehicle kinematic model, computes the requires vehicle velocities required to force the vehicle to track the input trajectory, and the low-level controller, derived from the vehicle dynamic model, takes the input from the high-level controller and computes the necessary thrust and torque required of the vehicle to realize the input velocities.

5.1. Trajectory Planning

To perform the tasks required of the vehicle, motion planning is necessary. Trajectory panning is the process of describing the desired motion in terms of positions, velocities and accelerations for the different degrees of freedom involved in the motion. The trajectory generation task is preferred to take a simple form and be computationally simple. To that end, techniques of the likes of polynomials, splines, and linear paths with blends have been developed [26]. Here, a quintic polynomial scheme is described to generate the history of required position, velocity, and acceleration for the vehicle. This higher order scheme allows for specifying position waypoints that the vehicle will go through to reach an end goal. It also allows for continuous velocity and acceleration trajectories, unlike cubic polynomials that only have continuous velocity trajectories.

The quintic polynomial describing the motion between two waypoints $x(t_0)$ and $x(t_f)$ is described by:

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{34}$$

The velocity and acceleration trajectories are then realized by differentiation equation (34) to get:

$$\begin{aligned} \dot{x}(t) &= a_1 + a_2 t + a_3 t^2 + a_4 t^3 + a_5 t^4 \\ \ddot{x}(t) &= a_2 + a_3 t + a_4 t^2 + a_5 t^3 \end{aligned} \tag{35}$$

To solve for the polynomial describing the motion, six constraints must be specified to solve for $a_0, a_1, \dots, a_5$. They are the two position waypoints, initial and final velocities, and initial and final accelerations. The constraints form a linear system of equations in the needed parameters, which is shown in matrix form in equation (36).

$$\begin{bmatrix} x_0 \\ \dot{x}_0 \\ \ddot{x}_0 \\ x_f \\ \dot{x}_f \\ \ddot{x}_f \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} \tag{36}$$

5.2. Control

As a consequence of the underactuated nature of the underwater vehicle, nonholonomic constraints interfere with the ability of the vehicle to move freely in 3D space. That is, while the vehicle can reach any position in its configuration space, it is constrained in the way that motion can happen. Submarines are examples of underactuated underwater vehicles with nonholonomic constraints, for they are usually equipped with a thruster that propels the vehicle forward and two stern planes that introduce pitch and yaw moments. Similarly, the AUV can only generate thrust in the forward direction and moments about all axis. This makes the control nontrivial as the simpler techniques like state feedback become inapplicable. Consequently, a subset of the degrees of freedom that can be actuated are exploited in a manner that allows the vehicle to accomplish the tasks required of it. Here, trajectory tracking is the task we focus on, for it is a common task that underwater vehicles are required to perform. The suggested control architecture is presented in Figure 20.

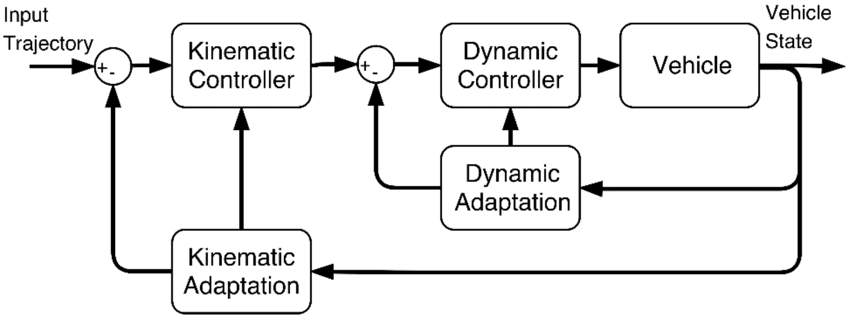


Figure 20: Control Diagram

5.2.1. Kinematic controller. Based on the kinematic model derived in equation set (7), it is of interest to device a control law that exclusively makes use of the actuatable degrees of freedom in tracking a given trajectory. First, the position and velocity of point ***E*** in Figure 21 is to be described, where point ***E*** is a lookahead distance usually employed in line-of-sight path following algorithms. In the inertial frame, point ***E*** is described as:

$$\begin{gathered}\boldsymbol{\eta}_E = [x_E \quad y_E \quad z_E]^T \\ \boldsymbol{\upsilon}_E = [u_E \quad v_E \quad w_E]^T\end{gathered} \tag{37}$$

with the velocities written in terms of the translational velocity, $\boldsymbol{\upsilon}_G$, and rotational velocity, $\boldsymbol{\omega}$, of the vehicle as:

$$\boldsymbol{\upsilon}_E = \boldsymbol{\upsilon}_G + \boldsymbol{\omega} \times \boldsymbol{r}_{G/E} \tag{38}$$

where the distance between points G and E is:

$$\boldsymbol{r}_{G/E} = [\epsilon \quad 0 \quad 0]^T \tag{39}$$

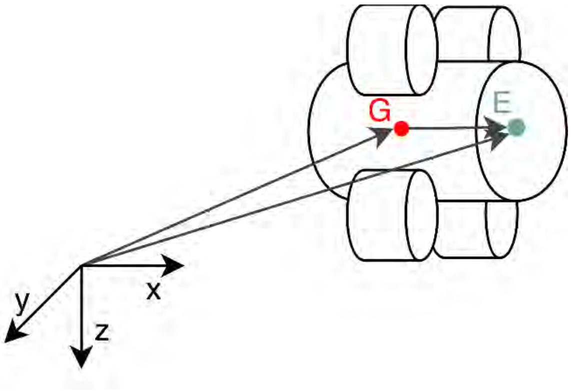


Figure 21: Kinematic control scheme

To formulate the kinematic control law, the kinematic model is first used to describe the velocities in equation (38) in the n-frame as:

$$\begin{gathered}\dot{\boldsymbol{\eta}}_E = \mathbf{R}_b^n(\boldsymbol{q})\left(\boldsymbol{\upsilon}_G + [0 \quad v \quad w]^T + \boldsymbol{\omega} \times \boldsymbol{r}_{G/E}\right) \\ \dot{\boldsymbol{q}} = \mathbf{T}(\boldsymbol{q})\, \boldsymbol{\omega}\end{gathered} \tag{40}$$

where $\boldsymbol{\upsilon}_G = [u \quad 0 \quad 0]^T$ and $\boldsymbol{\omega} = [0 \quad q \quad r]^T$ are inputs, and the rest of the velocities are taken as disturbances.

The trajectory following problem is to design a control law that can force the vehicle whose position is described by (40) to follow the desired continuous trajectory described by some $\boldsymbol{\eta}_{E_{des}}, \dot{\boldsymbol{\eta}}_{E_{des}}$.

The control law can be derived by noticing that the following relationship holds when describing the motion in terms of point E:

$$\begin{aligned}\boldsymbol{\upsilon}_G + \boldsymbol{\omega} \times \boldsymbol{r}_{G/E} &= \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \epsilon \begin{bmatrix} 0 \\ q \\ -r \end{bmatrix} \\ &= \mathbf{S}\,\boldsymbol{\upsilon}_a + [0 \quad v \quad w]^T \end{aligned} \tag{41}$$

where $\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & \epsilon \\ 0 & -\epsilon & 0 \end{bmatrix}$, $\boldsymbol{\upsilon}_a = [u \quad q \quad r]^T$, and $[0 \quad v \quad w]^T$ is the sway and heave velocities of the vehicle due to environmental disturbance.

Consequently, the kinematic relationship between points E and G can be exploited to devise the following control law:

$$\boldsymbol{\upsilon}_a = \mathbf{S}^{-1}\left\{\mathbf{R}_b^n(\boldsymbol{q})\left[\dot{\boldsymbol{\eta}}_E + \boldsymbol{\Lambda}\left(\boldsymbol{\eta}_{E_{des}} - \boldsymbol{\eta}_E\right) - [0 \quad v \quad w]^T\right]\right\} \tag{42}$$

where $\boldsymbol{\Lambda}$ is a diagonal design matrix that is positive definite and Hurwitz. $\boldsymbol{\Lambda}$ serves as the proportional gain in this control law.

The closed loop dynamics are noticed to take the form of

$$\begin{aligned}\dot{\boldsymbol{\eta}}_E &= \mathbf{R}_b^n(\boldsymbol{q})\mathbf{R}_b^n(\boldsymbol{q})^T\left(\dot{\boldsymbol{\eta}}_{E_{des}} + \boldsymbol{\Lambda}\left(\boldsymbol{\eta}_{E_{des}} - \boldsymbol{\eta}_E\right)\right) \\ &= \dot{\boldsymbol{\eta}}_{E_{des}} + \boldsymbol{\Lambda}\left(\boldsymbol{\eta}_{E_{des}} - \boldsymbol{\eta}_E\right)\end{aligned} \tag{43}$$

where the simplification is valid as a result of the orthogonal property of the rotation matrix. Denoting the error as $\boldsymbol{e} = \boldsymbol{\eta}_{E_{des}} - \boldsymbol{\eta}_E$, the error can be described by the dynamics that exponentially converge to zero:

$$\dot{\boldsymbol{e}} + \boldsymbol{\Lambda}\boldsymbol{e} = \boldsymbol{0} \tag{44}$$

It is of interest to note that under drift due to ocean currents and/or nonzero lateral vehicle motion, the control law of (42) does not guarantee tracking of any reference orientation trajectories. [55] proves that under zero disturbances, asymptotic orientation tracking is also guaranteed.

The kinematic control law in (43) was supplemented with integral control action, for steady state error was present despite the gain adaptation proposed later in this chapter. The kinematic control law is then updated to be:

$$\dot{\boldsymbol{\eta}}_E = \dot{\boldsymbol{\eta}}_{E_{des}} + \boldsymbol{\Lambda}\left(\boldsymbol{\eta}_{E_{des}} - \boldsymbol{\eta}_E\right) + \boldsymbol{\Lambda}_{\mathbf{i}}\int_0^t \left(\boldsymbol{\eta}_{E_{des}} - \boldsymbol{\eta}_E\right) dt \tag{45}$$

where $\mathbf{\Lambda_i}$ is a diagonal matrix with small integral gain, and $\mathbf{\Lambda}$ is the proportional gain matrix in this control law.

5.2.2. Dynamic controller. The second part of solving the trajectory tracking problem lies in the vehicle being able to realize the velocity commands that the kinematic controller produces. This command is a function of the state of the vehicle. The dynamic controller receives this velocity command, which is now denoted as $\boldsymbol{\upsilon}_{a_{des}}$, and tries to match the real velocity $\boldsymbol{\upsilon}$ with the kinematic controller output. Denoting the error between the two velocities as $\boldsymbol{e_v} = \boldsymbol{\upsilon} - \boldsymbol{\upsilon}_{\mathrm{a}_{des}}$, the dynamics of $\boldsymbol{\eta}_E$ can be described as:

$$\dot{\boldsymbol{\eta}}_E = \mathbf{R}_b^n(\boldsymbol{q})\big(\mathbf{S}\big(\boldsymbol{e}_v + \boldsymbol{\upsilon}_{\mathrm{a}_{des}}\big) + [0 \quad v \quad w]^T\big) \tag{46}$$

which leads to the following error dynamics:

$$\dot{\boldsymbol{e}} + \mathbf{\Lambda}\boldsymbol{e} = \mathbf{R}_b^n(\boldsymbol{q})\mathbf{S}(\boldsymbol{e}_v) \tag{47}$$

The trajectory tracking error here does not converge to zero as equation (47) suggests. To force the position error dynamics, $\boldsymbol{e}$, to zero, the mismatch between the command velocity and the vehicle velocity, $\boldsymbol{e}_v$, must be driven to zero.

Recalling that the dynamics of the vehicle can be described by:

$$\mathbf{M}\,\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{\mathcal{G}}(\boldsymbol{\eta}) = \boldsymbol{\tau_a} \tag{48}$$

A reduced order model that only takes into consideration the controlled degrees of freedom is written as

$$\mathbf{M}^{3\times3}\,\dot{\boldsymbol{\upsilon}}_a + \mathbf{C}(\boldsymbol{\nu}_a)^{3\times3}\boldsymbol{\upsilon}_a + \mathbf{D}(\boldsymbol{\nu}_a)^{3\times3}\boldsymbol{\upsilon}_a + \boldsymbol{\mathcal{G}}(\boldsymbol{\eta}) + \boldsymbol{d}(\boldsymbol{\nu}) = \boldsymbol{\tau_a} \tag{49}$$

in which $\boldsymbol{\upsilon}_a = [u \quad q \quad r]^T$, $\boldsymbol{\nu} = [u \quad v \quad w \quad p \quad q \quad r\,]^T$, $\boldsymbol{\tau_a} = [X \quad L \quad M]^T$, and $\boldsymbol{d}(\boldsymbol{\nu})$ is a disturbance vector arising from the coupling between the dynamics in the nonlinear model that are not fully accounted for in the reduced model. The computation of such vector is possible since knowledge of the entire state vector is assumed to be known. Also, $\mathbf{M}^{3\times3}$, $\mathbf{C}(\boldsymbol{\nu})^{3\times3}$, and $\mathbf{D}(\boldsymbol{\nu})^{3\times3}$ are only compromised of the components directly induced by $\boldsymbol{\upsilon}_a$.

The dynamic controller now is required to produce a $\boldsymbol{\tau_a}$ that can track the velocity vector $\boldsymbol{\upsilon}_a$ despite of the disturbance vector $\boldsymbol{d}(\boldsymbol{\nu})$. Given that $\boldsymbol{\upsilon}_{a_{des}}$ and $\dot{\boldsymbol{\upsilon}}_{a_{des}}$ are available, a computed-torque-like control law is applicable as:

$$\boldsymbol{\tau_a} = \mathbf{M}^{3\times 3}\left[\dot{\boldsymbol{\upsilon}}_{\mathrm{a}_{des}} + \mathbf{K}_d \mathbf{e}_v\right] + \mathbf{C}(\boldsymbol{\nu})^{3\times 3}\boldsymbol{\upsilon}_{a_{\mathrm{des}}} + \mathbf{D}(\boldsymbol{\nu})^{3\times 3}\boldsymbol{\upsilon}_{a_{\mathrm{des}}} + \boldsymbol{\mathcal{G}}(\boldsymbol{\eta}) + \boldsymbol{d}(\boldsymbol{\nu}) \qquad (50)$$

in which $\mathbf{K}_d$ is a positive diagonal design matrix.

The dynamics of the error when substituting the control law in the model are now given by:

$$\dot{\boldsymbol{e}}_v + \mathbf{K}_{\mathrm{d}}\, \boldsymbol{e}_v = \mathbf{0} \qquad (51)$$

which implies exponential convergence to zero of $\boldsymbol{e}_v$ and, consequently, $\boldsymbol{e}$.

5.3. Adaptive Online Parameter Tuning

5.3.1. Adaptive Proportional Controller (APC). The performance of the previously devised controllers depends on the values of the positive design parameters $\boldsymbol{\Lambda}$ and $\mathbf{K_d}$. Further, the performance of the dynamic inner control loop must keep up with the trajectory fed by the kinematic outer control loop in presence of noise and possible model uncertainty for robust trajectory tracking to occur. Consequently, an adaptive strategy to tune the design parameter gains is proposed. Two approaches are presented; an Adaptive Proportional Control adaptation law that depends on the tracking error and an adaptation law based on Universal Adaptive Stabilization (UAS).

The APC tunes the design parameters based on the propagation of variable k, which depends on the error. The dynamic gain adaptation law is given as:

$$\begin{aligned} \dot{k}_x &= \gamma_x e_x^2 \\ \dot{k}_y &= \gamma_y e_y^2 \\ \dot{k}_z &= \gamma_z e_z^2 \end{aligned} \qquad (52)$$

where k_x, k_y, and k_z are the adapted gain parameters, γ_x, γ_y, and γ_z are positive constants that drive the adaptation rate with respect to the tracking error, and e_x, e_y, and e_z are the errors in the three principal directions. The control law for the kinematic loop then becomes

$$\dot{\boldsymbol{\eta}}_E = \dot{\boldsymbol{\eta}}_{E_{des}} + \boldsymbol{\Lambda}\left(\boldsymbol{\eta}_{E_{des}} - \boldsymbol{\eta}_E\right) \qquad (53)$$

where $\mathbf{\Lambda} = diag\left(\begin{bmatrix} k_x^2 & k_y^2 & k_z^2 \end{bmatrix}\right)$.

To prove the stability of the devised kinematic control law, consider the candidate Lyapunov function

$$V_x(e_x, k_x) = \frac{1}{2}(e_x^2 + k_x^2) \tag{54}$$

Taking the time derivative of candidate Lyapunov function yields

$$\begin{aligned}
\dot{V}_x(e_x, k_x) &= e_x\dot{e}_x + k_x\dot{k}_x \\
&= e_x\left(\dot{x}_{E_{des}} - \dot{x}_E\right) + k_x(\gamma_x e_x^2) \\
&= e_x\left(\dot{x}_{E_{des}} - \left(\dot{x}_{E_{des}} + k_x^2\left(x_{E_{des}} - x_E\right)\right)\right) + k_x(\gamma_x e_x^2) \\
&= e_x\left(\dot{x}_{E_{des}} - \dot{x}_{E_{des}} - k_x^2 e_x\right) + k_x(\gamma_x e_x^2) \\
&= -k_x^2 e_x^2 + k_x\gamma_x e_x^2 \\
&= -(k_x^2 - k_x\gamma_x)e_x^2 < 0
\end{aligned} \tag{55}$$

where equation (53) was used to write $\dot{x}_E = \dot{x}_{E_{des}} + k_x^2\left(x_{E_{des}} - x_E\right)$.

Given that $k_x(0) > 0$, then

$$\begin{aligned}
&V_x(0,0) = 0 \\
&V_x(e_x, k_x) > 0, \forall e_x, k_x \neq 0 \\
&\dot{V}_x(e_x, k_x) < 0, \forall e_x, k_x \neq 0 \; if \; k_x > \gamma_x
\end{aligned} \tag{56}$$

where the condition $k_x > \gamma_x$ can be enforced by initializing k_x to be larger in magnitude than the selected γ_x. The same stability proof can be realized for the y as well as the z dynamics. Following Lyapunov stability theory, the system is asymptotically stable, and the tracking error will converge to zero.

Let the dynamic gain adaptation law for the dynamic controller be given as:

$$\begin{aligned}
\dot{k}_{d_x} &= \gamma_{d_x} e_{v_x}^2 \\
\dot{k}_{d_y} &= \gamma_{d_y} e_{v_y}^2 \\
\dot{k}_{d_z} &= \gamma_{d_z} e_{v_z}^2
\end{aligned} \tag{57}$$

where k_{d_x}, k_{d_y}, and k_{d_z} are the adapted gain parameters, $\gamma_{d_x}, \gamma_{d_y}$, and γ_{d_z} are positive constants that drive the adaptation rate with respect to the tracking error, and

$e_{v_x}, e_{v_y},$ and e_{v_z} are the errors in the three principal directions. By substituting (50) into (49), the control law for the dynamic loop can be written as

$$\dot{\boldsymbol{\upsilon}}_{\mathrm{a}} = \dot{\boldsymbol{\upsilon}}_{\mathrm{a}_{des}} + \mathbf{K}_d \mathbf{e}_v \tag{58}$$

where $\mathbf{K}_d = diag\left(\begin{bmatrix} k_{d_x}^{2} & k_{d_y}^{2} & k_{d_z}^{2} \end{bmatrix}\right)$ and $\mathbf{e}_v = \boldsymbol{\upsilon}_{\mathrm{a}} - \boldsymbol{\upsilon}_{\mathrm{a}_{des}}$.

To prove the stability of the devised dynamic control law, consider the candidate Lyapunov function

$$V_{d_x}\left(e_{v_x}, k_{d_x}\right) = \frac{1}{2}\left(e_{v_x}^2 + k_{d_x}^{2}\right) \tag{59}$$

Taking the time derivative of candidate Lyapunov function yields

$$\begin{aligned} \dot{V}_{d_x}\left(e_{v_x}, k_{d_x}\right) &= e_{v_x}\dot{e}_{v_x} + k_{d_x}\dot{k}_{d_x} \\ &= e_{v_x}(\dot{u} - \dot{u}_{des}) + k_{d_x}\left(\gamma_{d_x} e_{v_x}^2\right) \\ &= e_{v_x}\left(\left(\dot{u}_{des} + k_{d_x}^{2}(u - u_{des})\right) - \dot{u}_{des}\right) + k_{d_x}\left(\gamma_{d_x} e_{v_x}^2\right) \\ &= e_{v_x}\left(\dot{u}_{des} - \dot{u}_{des} - k_{d_x}^{2} e_{v_x}\right) + k_{d_x}\left(\gamma_{d_x} e_{v_x}^2\right) \\ &= -k_{d_x}^{2} e_{v_x}^2 + k_{d_x}\gamma_{d_x} e_{v_x}^2 \\ &= -\left(k_{d_x}^{2} - k_{d_x}\gamma_{d_x}\right) e_{v_x}^2 < 0 \end{aligned} \tag{60}$$

Given that $k_{d_x}(0) > 0$, then

$$\begin{aligned} &V_{d_x}(0,0) = 0 \\ &V_{d_x}\left(e_{v_x}, k_{d_x}\right) > 0, \forall e_{v_x}, k_{d_x} \neq 0 \\ &\dot{V}_{d_x}\left(e_{v_x}, k_{d_x}\right) < 0, \forall e_{v_x}, k_{d_x} \neq 0 \; if \; k_{d_x} > \gamma_x \end{aligned} \tag{61}$$

where the condition $k_{d_x} > \gamma_{d_x}$ can be enforced by initializing k_{d_x} to be larger in magnitude than the selected γ_{d_x}. The same stability proof can be realized for the y as well as the z dynamics. Following Lyapunov stability theory, the system is asymptotically stable, and the tracking error will converge to zero.

The APC is noticed to be slow to provide the necessary adaptation required to track a given trajectory. Its performance is also severely affected by the amount of noise injected in the simulation environment. Therefore, another approach needs to be investigated, and it needs to exhibit robustness to noise as well as show quick adaptation as a response to disturbances.

5.3.2. Adaptive Nussbaum Controller (ANC). The Universal Adaptive Stabilizer (UAS) adaptation law devised here relies on a Nussbaum function to tune the design parameters. Nussbaum functions are usually exploited in the design of UAS's. Numerous Nussbaum functions have been tested, and a Mittag-Leffler function was chosen to act as a Nussbaum function for the purpose of tuning the design parameters. It is of interest to note that a Mittag-Leffler function can act as a Nussbaum function under certain conditions that are documented in [56]. The new control law takes the following form.

$$\dot{\boldsymbol{\eta}}_E = \dot{\boldsymbol{\eta}}_{E_{des}} + \boldsymbol{N}(\boldsymbol{\Lambda})\left(\boldsymbol{\eta}_{E_{des}} - \boldsymbol{\eta}_E\right) \tag{62}$$

where $\boldsymbol{N}(\boldsymbol{\Lambda}) = diag\left(\begin{bmatrix} N_x(k_x) & N_y(k_y) & N_z(k_z) \end{bmatrix}\right)$ and each $N_i(k_i)$ is the Mittag-Leffler function that depends on two positive real parameters, α and β. The conditions under which the Mittag-Leffler function acts as a Nussbaum function are investigated in [56], and they correspond to $\alpha \in (2,3]$ and $\beta = 1$. The Mittag-Leffler function is given as:

$$N_i(z) = \sum_{\gamma=0}^{\infty} \frac{z^{\gamma}}{\Gamma(\alpha\gamma + \beta)}\ , \quad i = x, y, z \tag{63}$$

where $\alpha = 2.5$ and $\beta = 1$ are the constant parameters and $\Gamma(z+1) = z\Gamma(z), z > 0$, is the standard Gamma function. The time propagation of the tunable gains is defined just like in equation (52) as:

$$\begin{aligned} \dot{k}_x &= \gamma_x e_x^2 \\ \dot{k}_y &= \gamma_y e_y^2 \\ \dot{k}_z &= \gamma_z e_z^2 \end{aligned} \tag{64}$$

The stability of the UAS approach is proved as follows. Rewriting equation (62) in terms of the tracking error for the x direction gives

$$\begin{aligned} \dot{e}_x + N_x(k_x)e_x &= 0 \\ \dot{e}_x &= -N_x(k_x)e_x \end{aligned} \tag{65}$$

Choose a candidate Lyapunov function as

$$V_x(e_x, k_x) = \frac{1}{2}e_x^2 + \int_{k_x(t_0)}^{k_x(t)} N_x\left(k_x(t)\right) dt \tag{66}$$

Taking the time derivative of candidate Lyapunov function yields

$$\begin{aligned}\dot{V}_x(e_x,k_x) &= e_x\dot{e}_x + N_x\big(k_x(t)\big)\dot{k}_x(t) \\ &= -N_x\big(k_x(t)\big)e_x^2 + N_x\big(k_x(t)\big)\gamma_x e_x^2\end{aligned} \tag{67}$$

Letting $\gamma_x = 1$ gives

$$\dot{V}_x(e_x,k_x) = 0 \tag{68}$$

Integrating $\dot{V}_x(e_x,k_x)$ gives

$$V_x(t) - V_x(t_0) = C \tag{69}$$

where C is a constant of integration.

Substituting (66) in (69) gives

$$\frac{1}{2}e_x^2 + \int_{k_x(t_0)}^{k_x(t)} N_x\big(k_x(t)\big)\,dt = C + V_x(t_0) \tag{70}$$

Dividing by $k_x(t) - k_x(t_0)$ gives

$$\frac{1}{2}\frac{e_x^2}{k_x(t)-k_x(t_0)} = \frac{C+V_x(t_0)}{k_x(t)-k_x(t_0)} - \frac{1}{k_x(t)-k_x(t_0)}\int_{k_x(t_0)}^{k_x(t)} N_x\big(k_x(t)\big)\,dt \tag{71}$$

Now if $k_x(t) \to \infty$ as $t \to \infty$, then the R.H.S. of equation (71) can take values approaching $-\infty$ following the nature of Nussbaum functions, and this will violate (71) as $k_x(t)$ is defined in (64) to be positive. Since the positive nature of the L.H.S. of (71) is violated by letting $k_x(t) \to \infty$, then the assumption that $k_x(t) \to \infty$ is false and $k_x(t)$ is, therefore, bounded. However, $\dot{k}_x(t)$ being a non-decreasing function by definition and $k_x(t)$ being bounded implies that $k_x(t) \to k_{x_\infty}$ as $t \to \infty$. This further implies that $\dot{k}_x(t) \to 0$ as $t \to \infty$, i.e. $e^2(t) \to 0$ as $t \to \infty$ or $e(t) \to 0$ as $t \to \infty$. The stability of the approach is concluded.

Like with the APC stability proof, the above ANC stability proof is also applicable to the dynamics control loop adaptation law.

It is of interest to describe the conditions under which the adaptive algorithms tune the controller parameters. This is important because the devised adaptation is unidirectional, so the gains only grow as a result of the adaptation. If the gains are too large in magnitude, oscillations due to large control effort might hinder the system from staying as close to the trajectory as possible. Moreover, large gains translate to large control effort, which has detrimental consequences on the endurance of the onboard power source.

5.4. Adaptive Gain Update Conditions

Since it is important to have control over the gain adaptation in a realistic setting, a gain update management algorithm is devised here. Let the Euclidean norm of the error between the path and trajectory is a zero-mean randomly distributed Gaussian variable $||\boldsymbol{e}||_2 \sim \mathcal{N}\left(0, \sigma_{e_n}\right)$. This assumption is valid since it is expected of the controller to ideally reduce the error to zero, and it is expected that the error will vary depending on the varying trajectory. The standard deviation of the error norm is updated at each time step to describe the variance of the history of the error. The Euclidian or second norm of the error is shown in equation (72).

$$||\boldsymbol{e}||_2 = \sqrt{e_x^2 + e_y^2 + e_z^2} = e_n \tag{72}$$

Further, let the expected value of the vector of error norms $\boldsymbol{e}_n$ be described by the average value of the error norm computed over a moving window of the variable, and let the standard deviation of the vector of error norms $\boldsymbol{e}_n$ be that of the entire time history of $\boldsymbol{e}_n$.

$$E\left[\boldsymbol{e}_{n_k}\right] = \sum_{i=k-L}^{k} \frac{e_{n_i}}{L} \tag{73}$$

When the controller is functioning properly, the expected value of the error norm ideally reduces to a small quantity. Further, the value of the error norm under typical operating conditions should not be far away from the expected value of the error norm. Typical trajectories that AUVs are required to track are parametrized to be slowly varying and smooth. Therefore, unless environmental disturbances affect the vehicle, the assumptions on the mean and standard deviation of the error norm should hold. To alleviate the effect of significant disturbances on the vehicle, the gain values are reset to their initial levels when such a disturbance is detected.

The proposed adaptation is set to happen when the current value of the error norm is more than two standard deviations away from the expected value of the error norm as shown in equation (74).

$$\frac{\left|e_{n_k} - E\left[\boldsymbol{e}_{n_k}\right]\right|}{\sigma_{\boldsymbol{e}_n}} > 2 \tag{74}$$

where $\sigma_{\boldsymbol{e}_n}$ is the standard deviation of $\boldsymbol{e}_n$. This condition serves to indicate with confidence that the value of error norm is

For the sake of computational performance, it is of interest to implement a recursive mean and standard deviation algorithm to avoid processing the entire time history of the error vector at each time step. The Welford algorithm was proposed by [57], and it has since been analyzed and observed to be stable and much less prone to loss of precision over alternative algorithms. The algorithm is presented in Figure 22.

Initialization	$\boldsymbol{n = 0}$ $\boldsymbol{\bar{x} = \sigma_x^2 = 0}$
Update when new value is available	$n = n + 1$ $\bar{x}_{new} = \bar{x}_{old} + \dfrac{x - \bar{x}_{old}}{n}$ $\sigma_{x\,new}^2 = \sigma_{x\,old}^2 + \dfrac{x - \bar{x}_{old}}{x - \bar{x}_{new}}$
Finalize update	$\bar{x}_{new} = \bar{x}_{new}$ $\sigma_{x\,old}^2 = \sigma_{x\,new}^2$

Figure 22: The Welford Recursive Algorithm

The expression in equation (74) represents the normal score or the z-score, which is defined to be the number of standard deviation by which an observed variable exceeds the mean. It is argued that if the computed normal score suddenly grows to exceed some threshold that is determined from the history of the computed normal score, it must be the case that a disturbance of some sort happened. The threshold is set to take the value of the last maximum computed normal score since it is expected of the controller to drive that score to low values as the error grows smaller. The suggested approach to effectively deal with these situations comes in the form of resetting the adapted controller gains as to avoid having gains that are larger than required in normal operating conditions. This condition serves to circumvent unnecessary large gains for the controllers. The algorithm flowchart is presented in Figure 23. The gains are left to adapt in the first couple of seconds since they are initialized with very small positive values as discussed in the stability analysis.

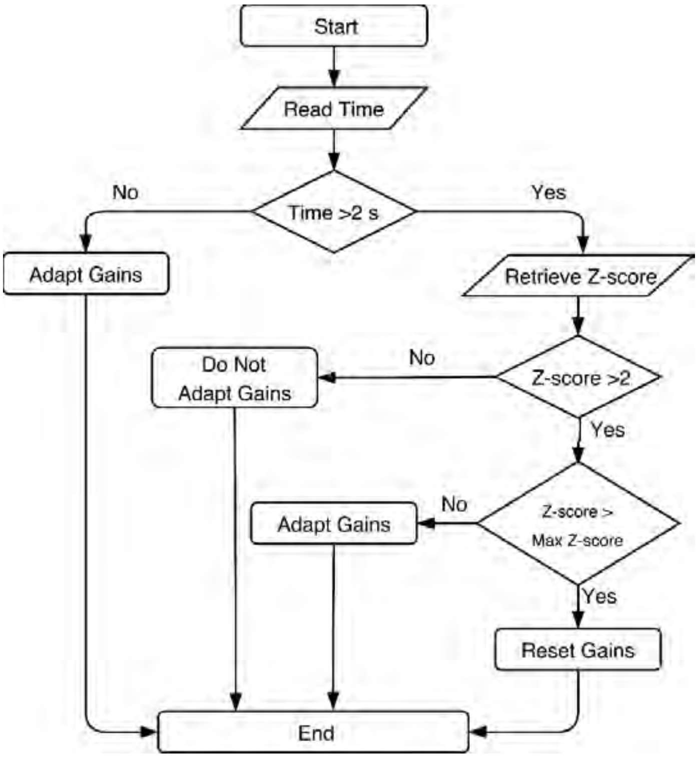


Figure 23: Gain Adaptation Conditions

To elaborate on the proposed algorithm, the vehicle first starts with gains of small magnitude, and it is commanded to track some trajectory. The gain adaptation happens for a brief window, which was chosen to be 2 seconds based on trial and error, to drive the gains to values that can successfully accomplish the trajectory tracking task. As that happens, the statistics of the error norm in terms of expected value and standard deviation should be low in magnitude relative to their initial values. When a disturbance happens, the error norm will suddenly increase in magnitude, the expected value of the error norm will also increase, and the standard deviation will grow. The Z-score was chosen to be the performance metric to detect such incidents, and the designed threshold was two standard deviations from the expected value of the error norm. This serves to establish a confidence interval that the error norm has increased beyond desired levels, and the corrective action here to prevent controller saturation is to reset the gains as they might have been driven too large.

5.5. Simulation Results

In order to evaluate the performance of the devised control algorithm, the controller was tested in simulation environment under MATLAB. The vehicle was commanded to cover two loops around each of the four parametrized trajectories, and the tracking results validate the performance of the controller. The simulation conditions are documented in Table 8. 20 Monte Carlo simulations for each of the parametrized trajectories allowed the generation of aggregate performance statistics for the controllers and adaptation routines.

Table 8: Simulation Conditions

	Value
Initial Pose **(Straight/ Polynomial Paths)**	$\boldsymbol{\eta} = [1 \quad 1 \quad 8 \quad 0.7071 \quad 0 \quad 0 \quad 0.7071\ ]^{\boldsymbol{T}}$
Initial Pose **(Helical/8-shaped Paths)**	$\boldsymbol{\eta} = [1 \quad 1 \quad 1 \quad 0.7071 \quad 0 \quad 0 \quad 0.7071\ ]^{\boldsymbol{T}}$
Dynamics noise	$\boldsymbol{w} \sim \mathcal{N}(0, \mathbf{10^{-2}}), \qquad \mathbf{w} \in \mathbb{R}^{13}$
Measurements noise	$\boldsymbol{v} \sim \mathcal{N}(0, \mathbf{10^{-2}}), \qquad \boldsymbol{v} \in \mathbb{R}^{13}$

The first test trajectory employs interpolated straight lines parametrized by constant velocity and zero acceleration constraints to connect between waypoints, and it connects those lines at waypoints in a non-smooth manner. The waypoints are presented in Table 9.

Table 9: Waypoints Used to Parametrize Trajectories

	From Waypoint $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ $[\boldsymbol{m}]$	**To Waypoint** $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})[\boldsymbol{m}]$
Path 1	$[\mathbf{7}, \mathbf{7}, \mathbf{7}]$	$[\mathbf{14}, \mathbf{7}, \mathbf{14}]$
Path 2	$[\mathbf{14}, \mathbf{7}, \mathbf{14}]$	$[\mathbf{14}, \mathbf{14}, \mathbf{14}]$
Path 3	$[\mathbf{14}, \mathbf{14}, \mathbf{14}]$	$[\mathbf{7}, \mathbf{14}, \mathbf{0}]$
Path 4	$[\mathbf{7}, \mathbf{14}, \mathbf{1}]$	$[\mathbf{7}, \mathbf{7}, \mathbf{7}]$

For the straight paths parameterized trajectory, Figure 24 shows the trajectory tracking results, Figure 25 shows the trajectory tracking errors, Figure 26 shows the vehicle tracking velocities, Figure 27 shows the control efforts required to track the trajectory, and Figure 28 shows the adaptation of the kinematic and the dynamic controller gains.

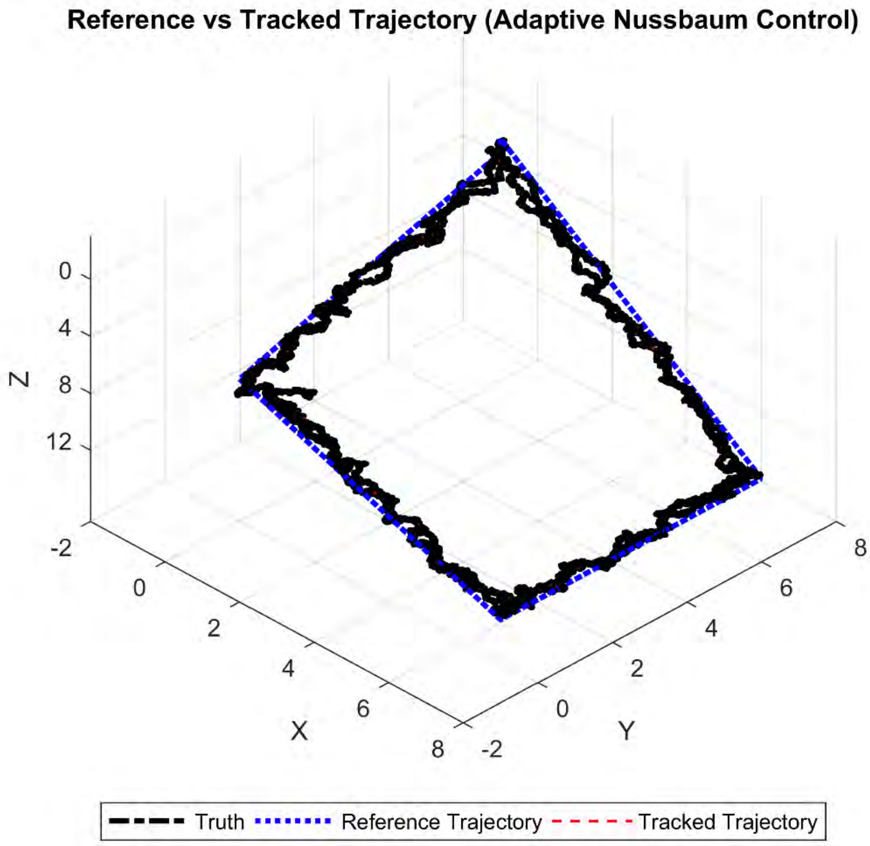


Figure 24: Trajectory Tracking (Parametrized Straight Paths)

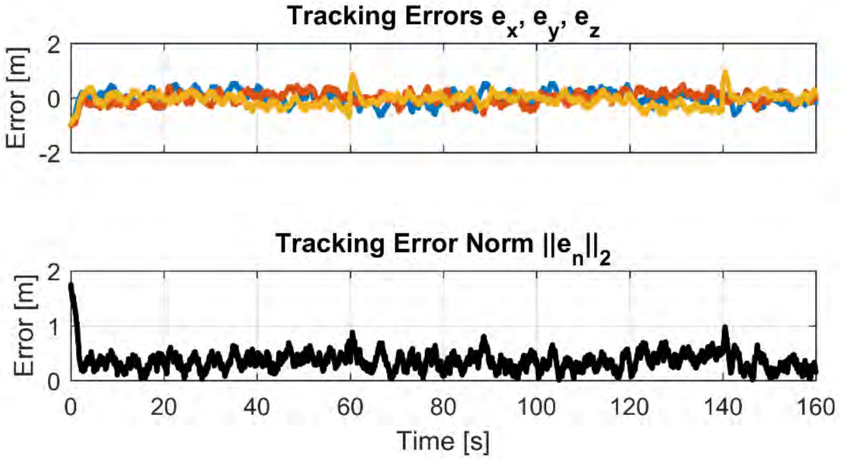


Figure 25: Trajectory Tracking Errors (Parametrized Straight Paths)

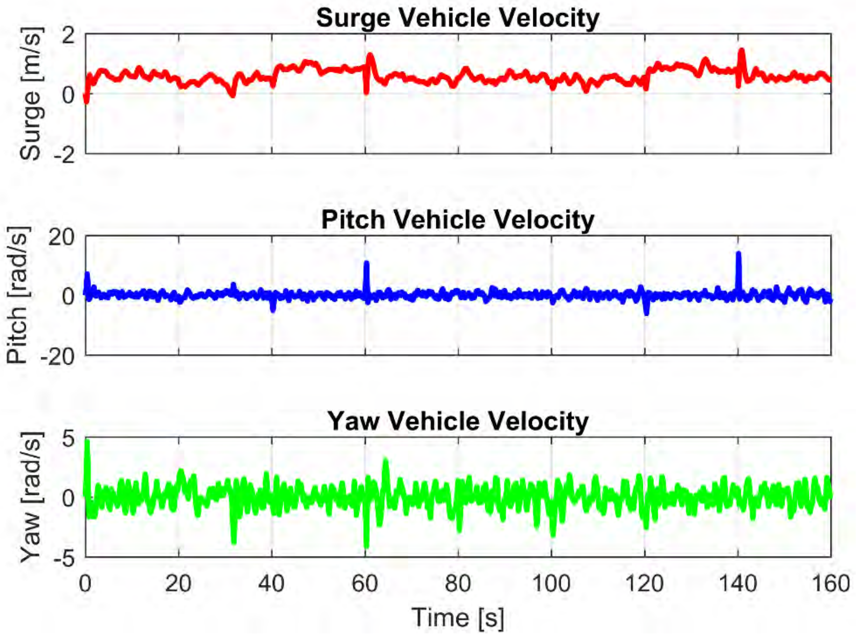


Figure 26: Vehicle Tracking Velocities (Parametrized Straight Paths)

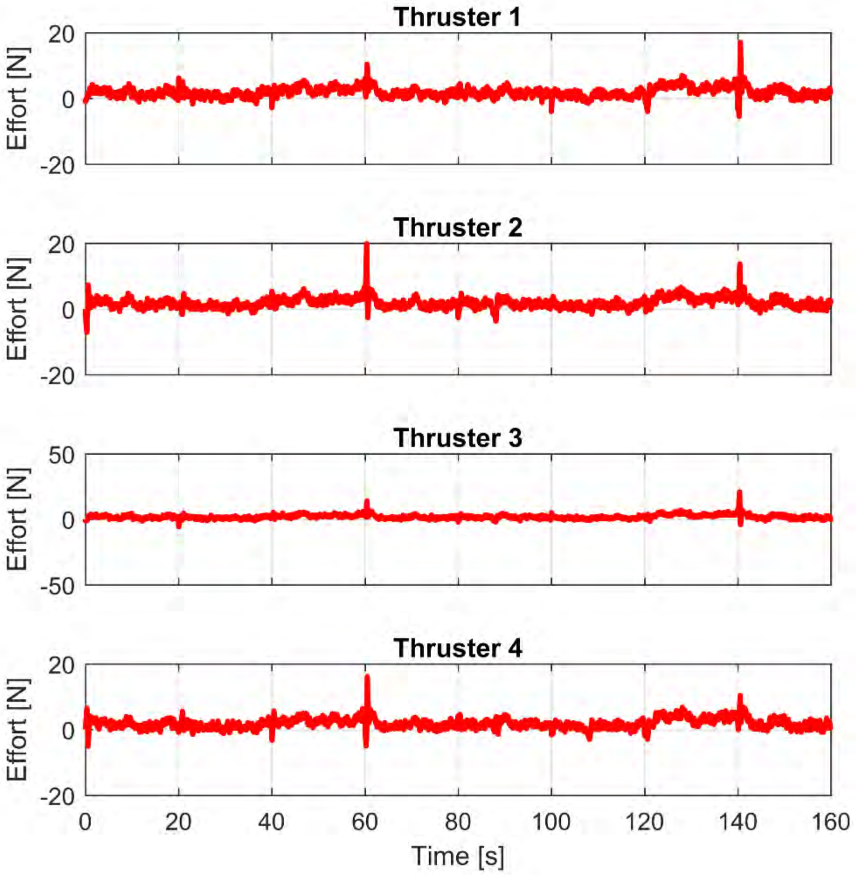


Figure 27: Control Efforts (Parametrized Straight Paths)

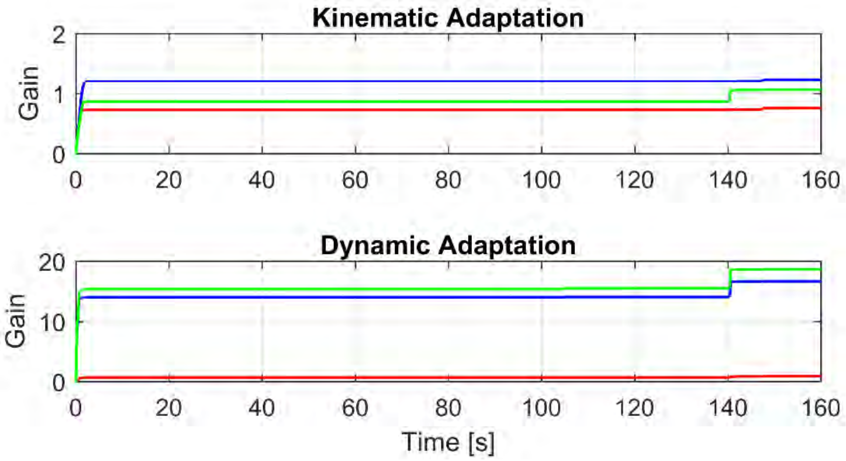


Figure 28: Gain Adaptation (Parametrized Straight Paths)

The second test trajectory employs quintic polynomials going through the same waypoints in the first test trajectory. The polynomial has continuous first and second derivatives, so motion is expected to be smooth and jerks in the acceleration should be avoided by feeding the polynomial with proper conditions in between waypoints.

For the quintic polynomials parameterized trajectory, Figure 29 shows the trajectory tracking results, Figure 30 shows the trajectory tracking errors, Figure 31 shows the vehicle tracking velocities, Figure 32 shows the control efforts required to track the trajectory, and Figure 33 shows the adaptation of the kinematic and the dynamic controller gains.

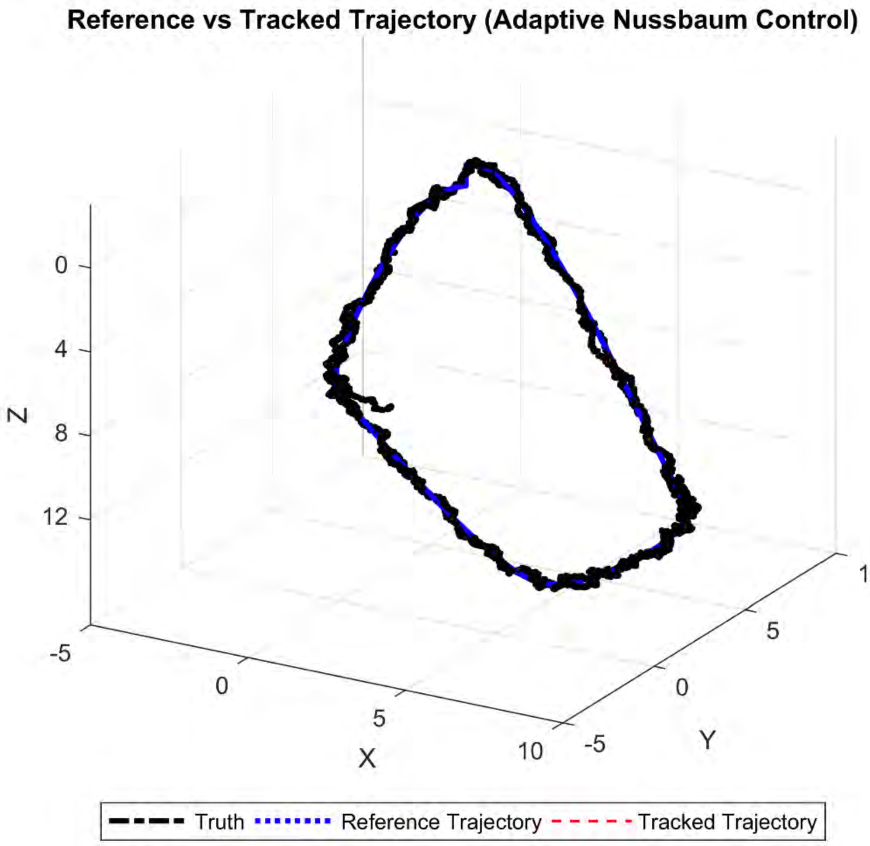


Figure 29: Trajectory Tracking (Parametrized Polynomial Paths)

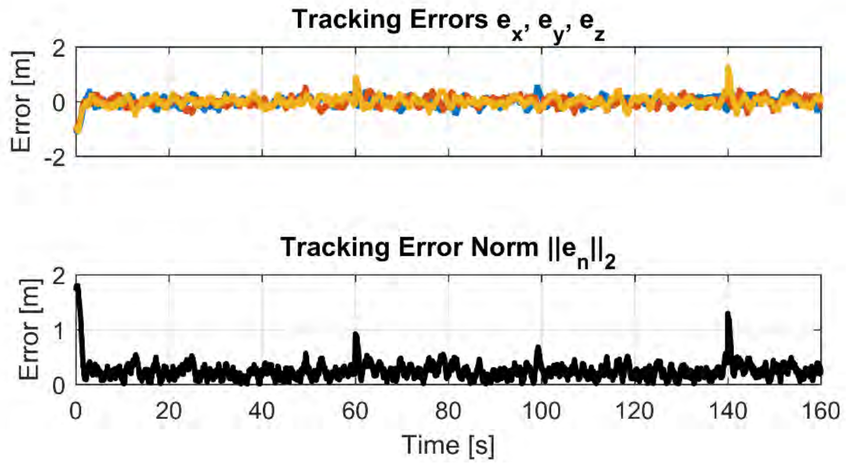


Figure 30: Trajectory Tracking Errors (Parametrized Polynomial Paths)

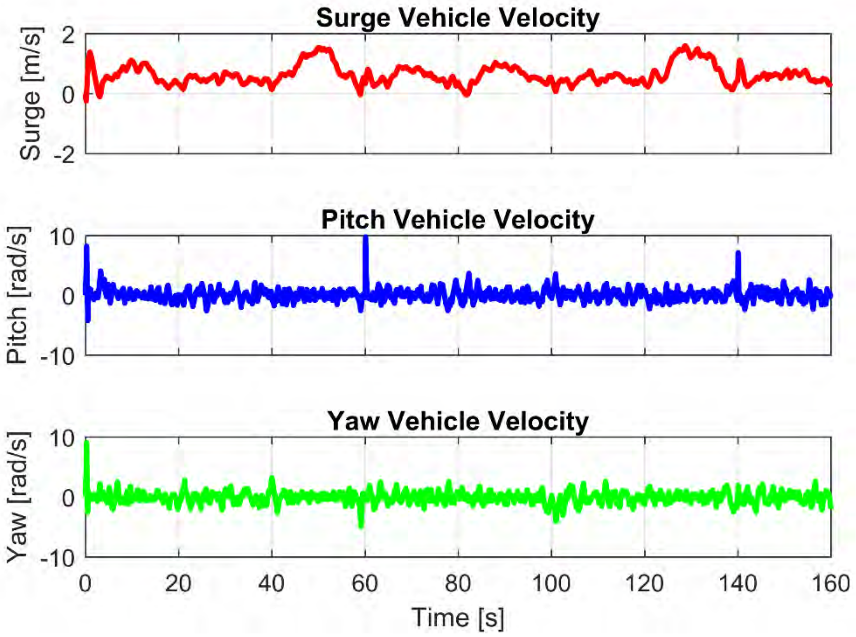


Figure 31: Vehicle Tracking Velocities (Parametrized Polynomial Paths)

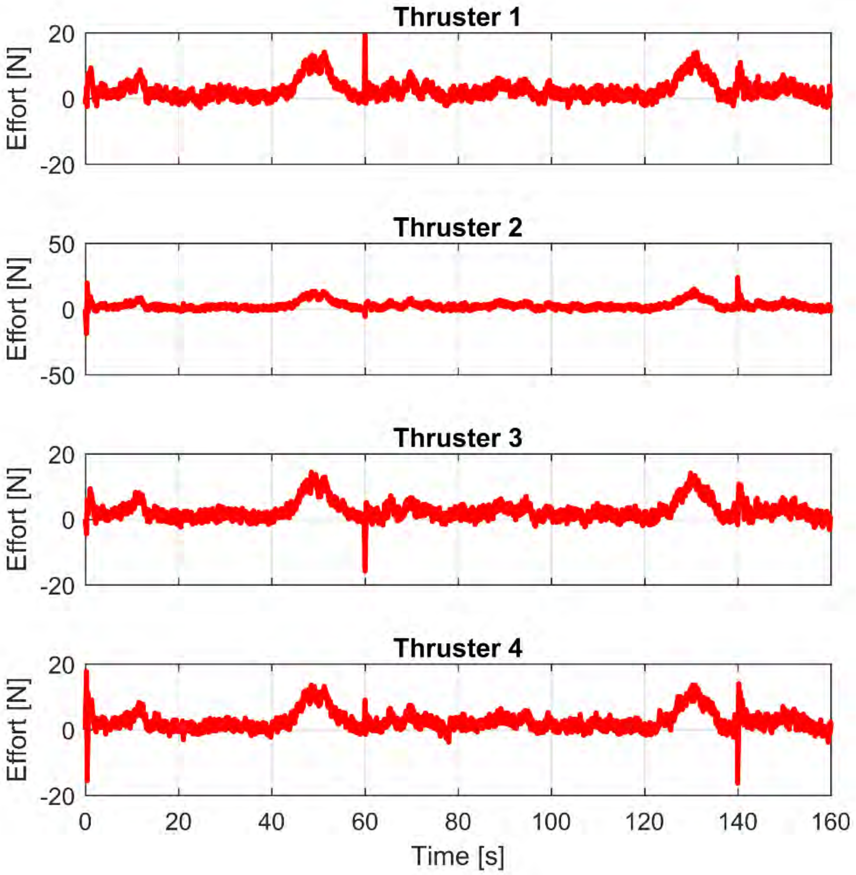


Figure 32: Control Efforts (Parametrized Polynomial Paths)

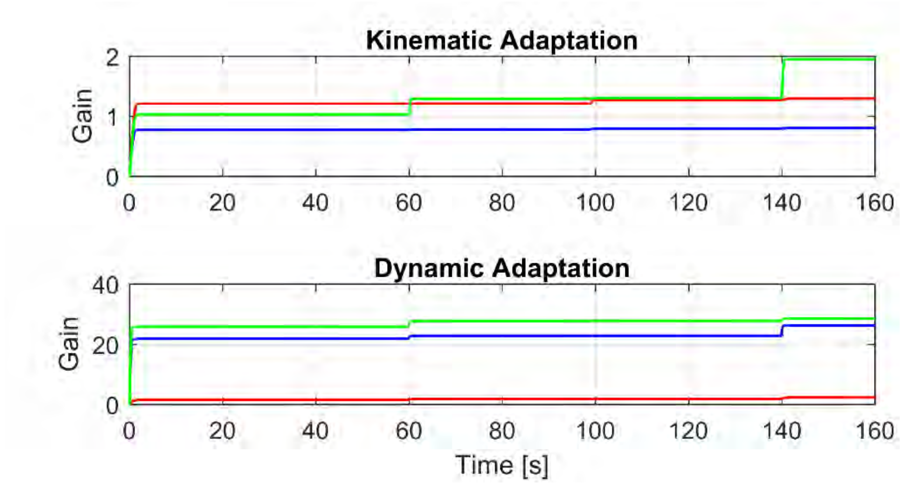


Figure 33: Gain Adaptation (Parametrized Polynomial Paths)

The third parametrized trajectory is a helix. The results for the trajectories as well as the required controller effort in terms of thrust and moments that the vehicle exerts are documented below.

For the helical parameterized trajectory, Figure 34 shows the trajectory tracking results, Figure 35 shows the trajectory tracking errors, Figure 36 shows the vehicle tracking velocities, Figure 37 shows the control efforts required to track the trajectory, and Figure 38 shows the adaptation of the kinematic and the dynamic controller gains.

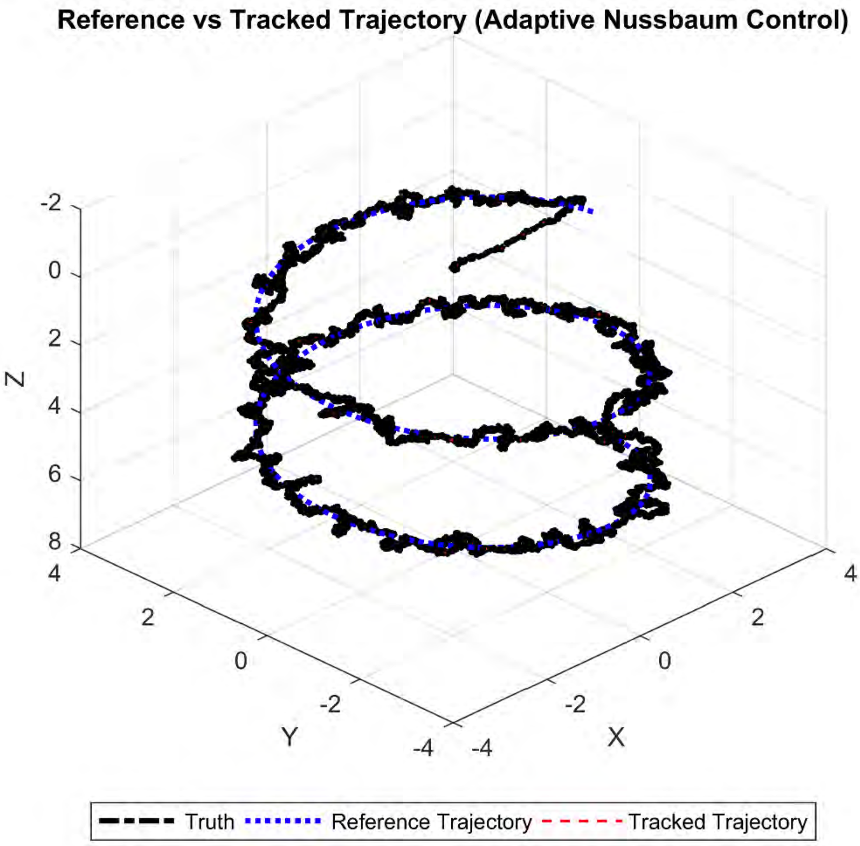


Figure 34: Trajectory Tracking (Parametrized Helical Paths)

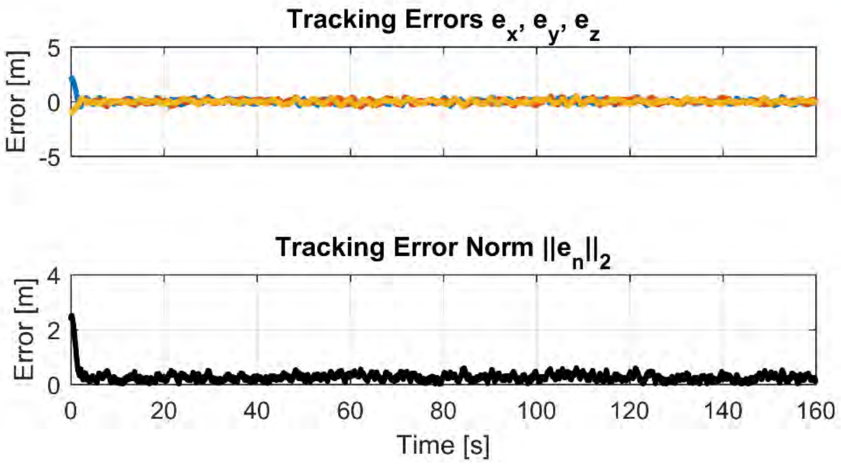


Figure 35: Trajectory Tracking Errors (Parametrized Helical Paths)

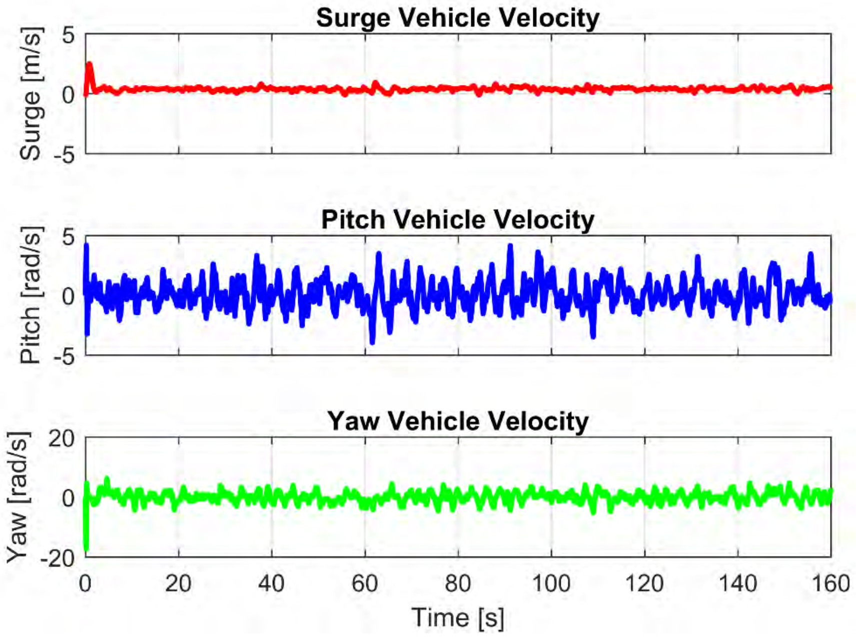


Figure 36: Vehicle Tracking Velocities (Parametrized Helical Paths)

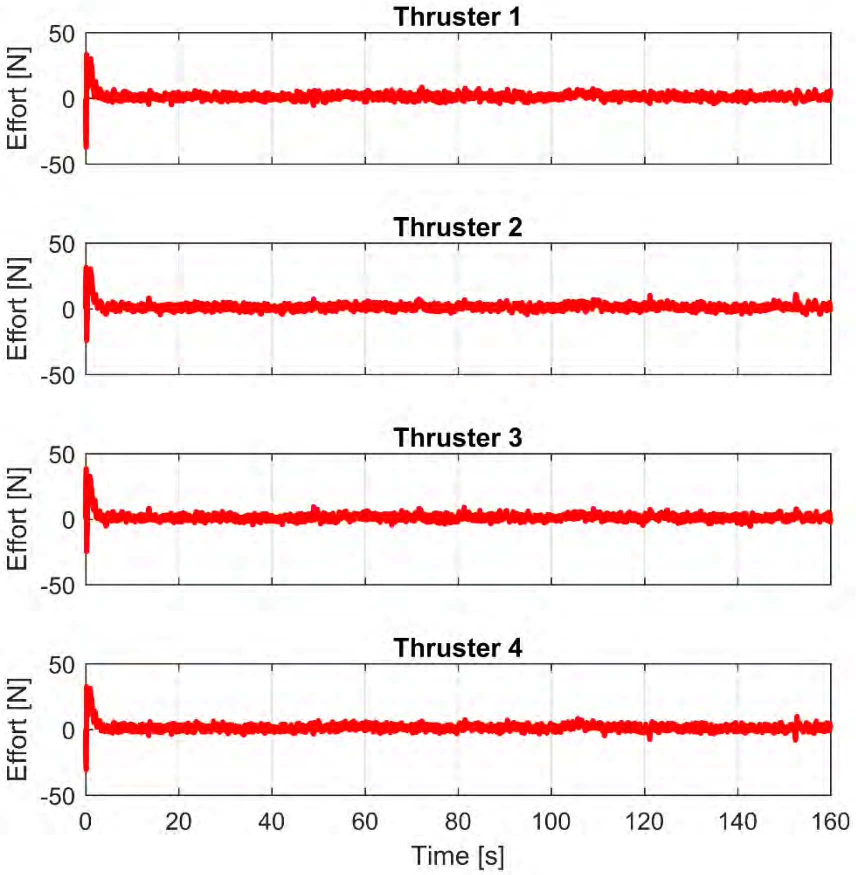


Figure 37: Control Efforts (Parametrized Helical Paths)

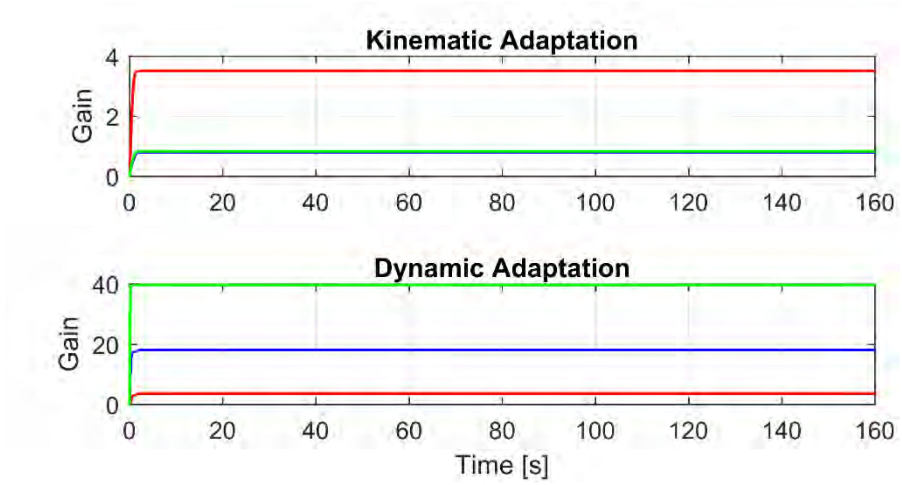


Figure 38: Gain Adaptation (Parametrized Helical Paths)

The fourth parametrized trajectory is a figure-8 with a varying height. The results for the trajectories as well as the required controller effort in terms of thrust and moments that the vehicle exerts are documented below.

For the 8-shaped parameterized paths, Figure 39 shows the trajectory tracking results, Figure 40 shows the trajectory tracking errors, Figure 41 shows the vehicle tracking velocities, Figure 42 shows the control efforts required to track the trajectory, and Figure 43 shows the adaptation of the kinematic and the dynamic controller gains.

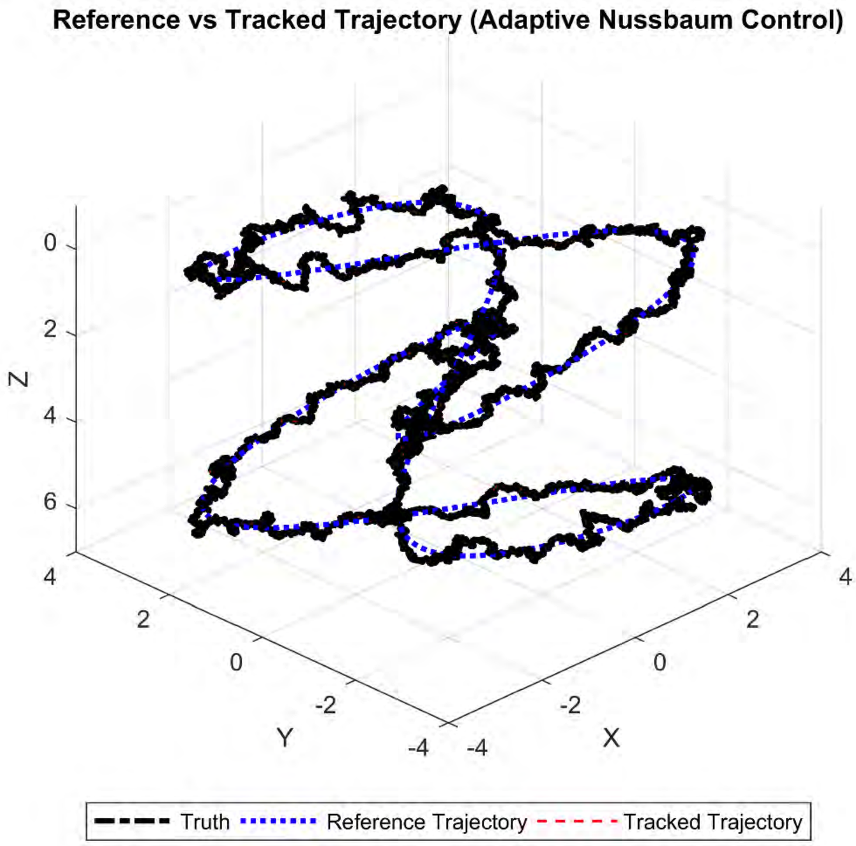


Figure 39: Trajectory Tracking (Parametrized 8-shaped Paths)

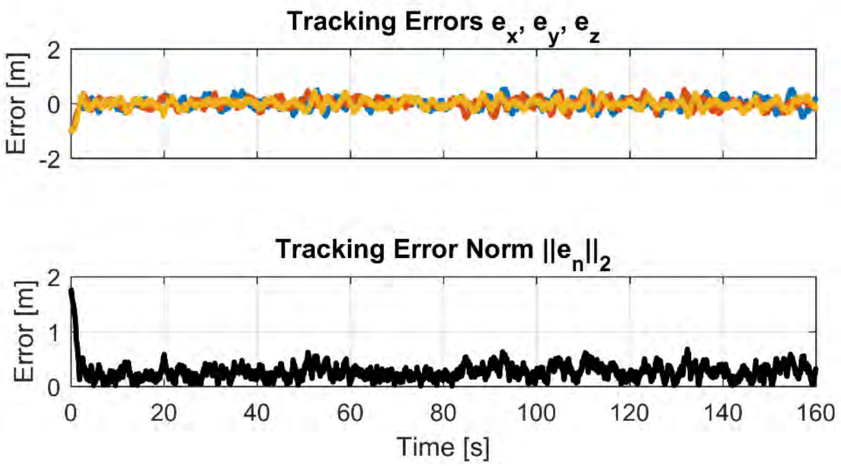


Figure 40: Trajectory Tracking Errors (Parametrized 8-shaped Paths)

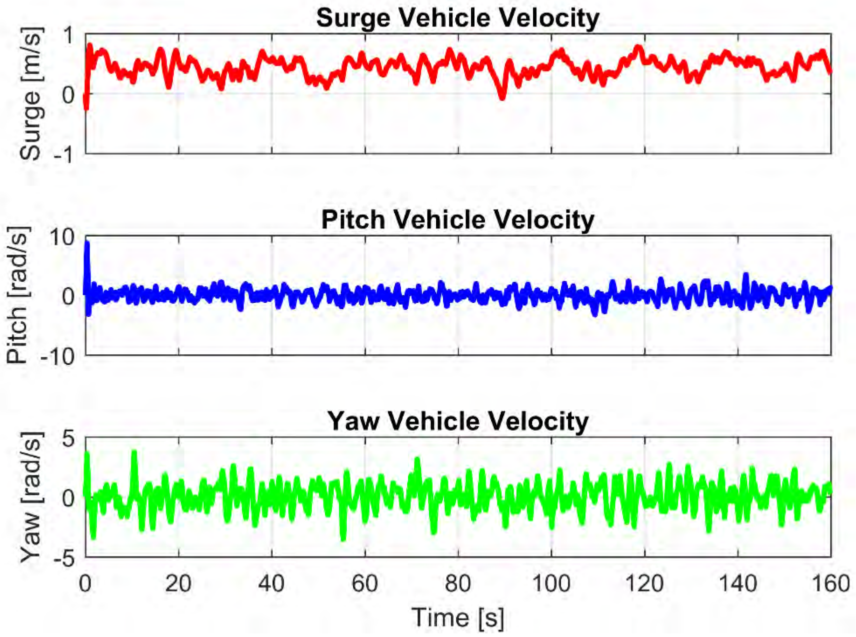


Figure 41: Vehicle Tracking Velocities (Parametrized 8-shaped Paths)

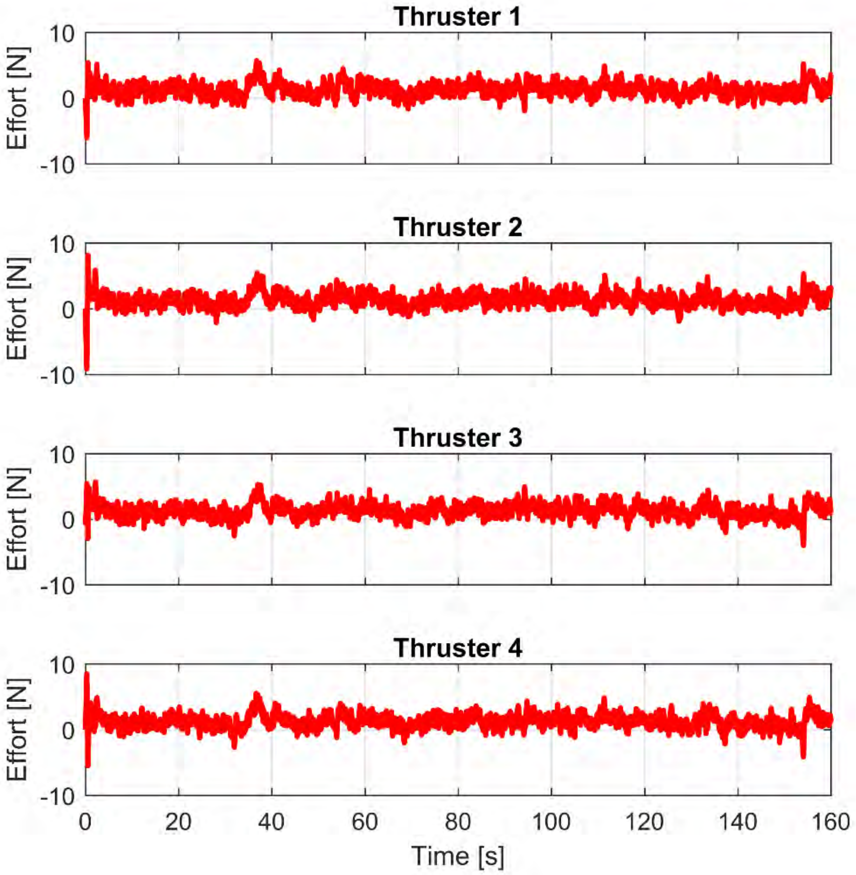


Figure 42: Control Efforts (Parametrized 8-shaped Paths)

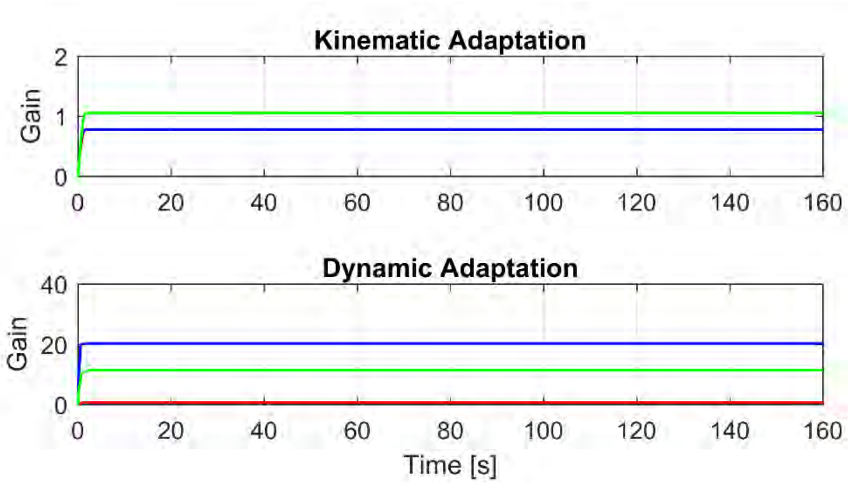


Figure 43: Gain Adaptation (Parametrized 8-shaped Paths)

Table 10 - Table 13 summarize the tracking performance of the two proposed controllers and the two adaptation routines in the presence of noise as well as the lack thereof. The mean error norm as well as the mean control effort for each thruster are documented for the tested trajectories.

Table 10: Trajectory Tracking Error (Conditional Adaptation Law)

	Noisy Environment		Noise-Free Environment	
	APC error $[m]$	ANC error $[m]$	APC error $[m]$	ANC error $[m]$
Straight Paths	0.4509	0.3524	0.3301	0.2850
Polynomial Paths	*Fails*	0.2515	0.0800	0.1029
Helical Paths	1.7676	0.2704	0.6477	0.0660
8-shaped Paths	0.2992	0.2477	0.0715	0.0755

Table 11: Trajectory Tracking Effort (Conditional Adaptation Law)

	Noisy Environment		Noise-Free Environment	
	APC effort $[N]$	ANC effort $[N]$	APC effort $[N]$	ANC effort $[N]$
Straight Paths	2.6492	2.0228	1.6845	1.7531
Polynomial Paths	*Fails*	2.7916	2.6433	2.4293
Helical Paths	2.1138	0.8369	0.3785	0.3529
8-shaped Paths	5.8636	1.7709	1.6688	1.1086

Table 12: Trajectory Tracking Error (Continuous Adaptation Law)

	Noisy Environment		Noise-Free Environment	
	APC error $[m]$	ANC error $[m]$	APC error $[m]$	ANC error $[m]$
Straight Paths	*Fails*	0.2857	*Fails*	0.2206
Polynomial Paths	*Fails*	0.2401	0.0853	0.0984
Helical Paths	*Fails*	0.2769	0.0808	0.0655
8-shaped Paths	*Fails*	0.2715	0.0599	0.0735

Table 13: Trajectory Tracking Effort (Continuous Adaptation Law)

	Noisy Environment		Noise-Free Environment	
	APC effort $[N]$	**ANC effort $[N]$**	**APC effort $[N]$**	**ANC effort $[N]$**
Straight Paths	*Fails*	3.2884	*Fails*	1.8329
Polynomial Paths	*Fails*	3.2886	3.0515	2.4310
Helical Paths	*Fails*	2.7089	0.3638	0.3523
8-shaped Paths	*Fails*	4.6655	1.6673	1.1082

Figure 44 - Figure 46 present a visual depiction of the mean tracking error and the mean control effort for the proposed controllers and adaptation routines. It is of significance to indicate that the APC with the continuous adaptation law was not included in the visual depiction, for it fails consistently in a noisy setting.

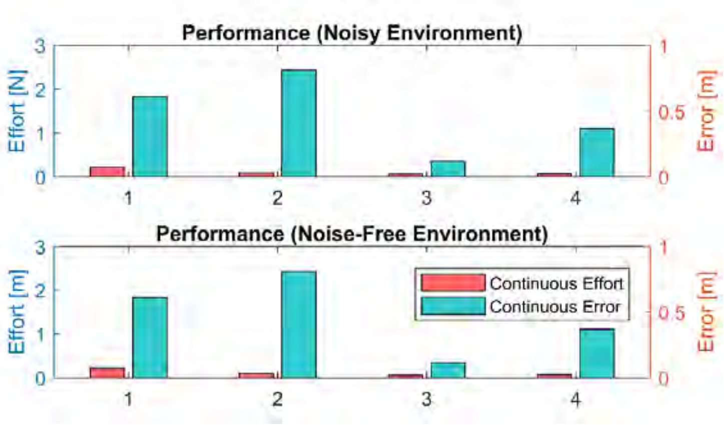


Figure 44: Adaptive Nussbaum Controller Performance with the Continuous Adaptation Law

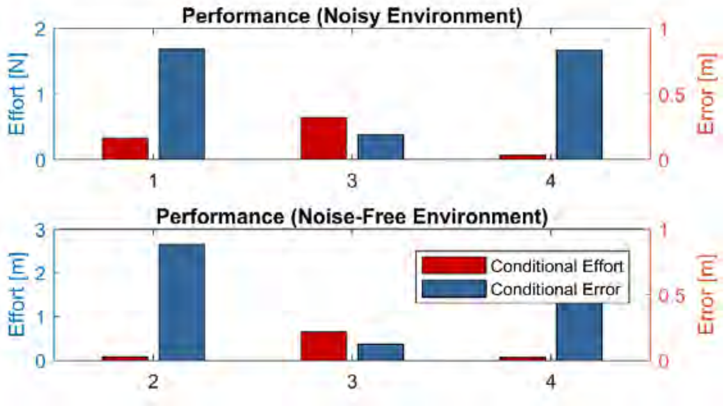


Figure 45: Adaptive Proportional Controller Performance with the Conditional Adaptation Law

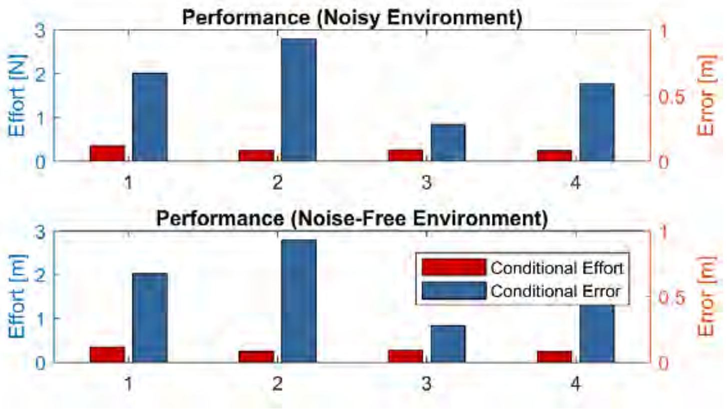


Figure 46: Adaptive Nussbaum Controller Performance with the Conditional Adaptation Law

With the continuous adaptation law, the APC consistently fails in the simulated noisy environment, and it sometimes fails in the noise free environment. On the other hand, the ANC exhibits robustness to the level of noise present. Moreover, when it does work, the APC results in approximately the same level of mean tracking error as the ANC, but it consistently requires larger control efforts than the ANC. With the conditional adaptation law, the trend is similar. Here, the APC sometimes fails in the noisy environment, which gives merit to the proposed conditional adaptation law. Similar to the other adaptation routine, the APC requires more control effort to achieve a similar or worse trajectory tracking error to the ANC. All in all, the results point to the ANC being superior to the APC in all test cases, when the control effort required to achieve a certain trajectory tracking error is taken into consideration. The results also give merit to the use of the conditional adaptation routine over the continuous adaptation routine such that the adapting gains reset to avoid overshoot and/or instability. It is important to note that the injected noise magnitudes were large, which was done to test the robustness of the controllers and the adaptation laws to the existence of noise and disturbances in the underwater environment. Further, the measurement noise choice is justified when considering that popular acoustic Time-of-Flight localization techniques can involve relatively large uncertainties that arise of the inaccurate knowledge of the density of the medium, systematic bias in the measurement apparatus, and other issues that hinder accurate localization of the vehicle [4].

The proposed gain management system is to be verified through simulating a case where the vehicle is required to track the parameterized polynomials trajectory through the waypoints of Table 9. To force the gains to adapt, the vehicle is disturbed by dynamics noise, measurements noise, and forces due to water currents. The water current effects are modeled to be first-order Gauss-Markov processes that exert forces in the three principal directions [10]. The exact form of the disturbance is shown below.

$$\dot{\boldsymbol{\tau}}_{current} + \delta \boldsymbol{\tau}_{current} = \boldsymbol{\epsilon} \tag{75}$$

where $\boldsymbol{\tau}_{current} = [X \quad Y \quad Z]^T$ are the current-induced forces in the three principal directions, δ is a time constant to determine how fast the current changes, and $\boldsymbol{\epsilon}$ is a vector of Gaussian white noises that drive the system.

To disturb the vehicle in some of the simulated scenarios, we inject water current disturbances in the three principal directions to induce gain adaptation. The simulated disturbance representing the current disturbance forces is shown in Figure 47.

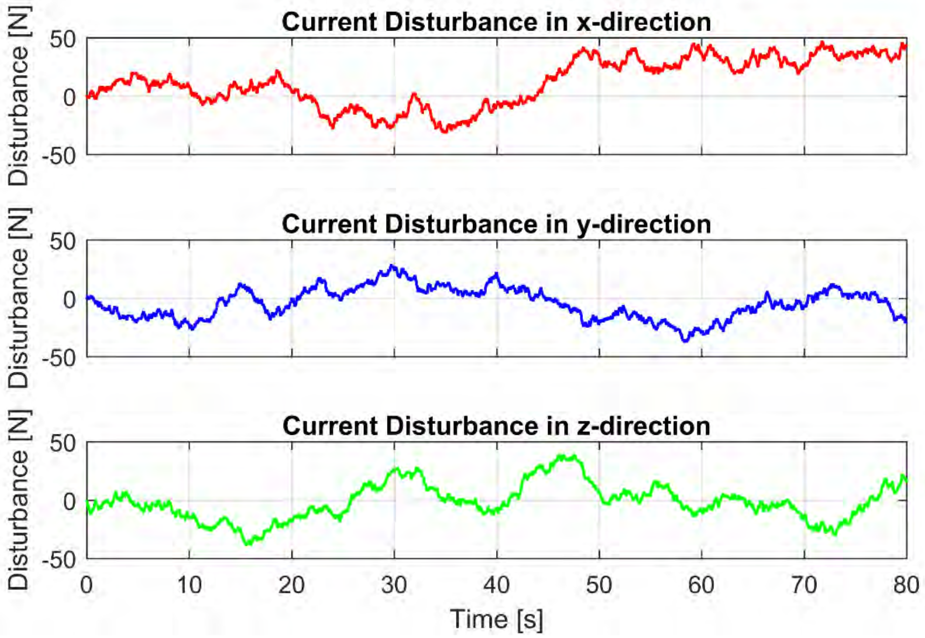


Figure 47: Injected Current-induced Disturbance Forces

Table 14: Simulation Conditions

	Value
Initial Pose	$\boldsymbol{\eta} = [1 \quad 1 \quad 8 \quad 0.7071 \quad 0 \quad 0 \quad 0.7071\,]^{\boldsymbol{T}}$
Dynamics noise	$\boldsymbol{w} \sim \mathcal{N}(0, \mathbf{10^{-2}}), \qquad \mathbf{w} \in \mathbb{R}^{13}$
Measurements noise	$\boldsymbol{v} \sim \mathcal{N}(0, \mathbf{10^{-2}}), \qquad \boldsymbol{v} \in \mathbb{R}^{13}$
Current noise	$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{10^{2}}), \qquad \boldsymbol{\epsilon} \in \mathbb{R}^{3}$

In Figure 48 and Figure 49, the performance of devised gain management algorithm is tested against the case where adaptation is continuous and is not managed by the algorithm. The ANC is run on the parametrized polynomial paths trajectory, and the algorithm results are documented. The trajectory has bends in which we expect aggressive maneuvering to induce large amounts of tracking error. This motions before and after this bend are clearly steeper when the trajectory plot is inspected. It is apparent from the error norm as well as the standard deviation plots that the vehicle starts off path, and the error in this case is large with some value for the standard deviation. As the controller drives the vehicle to the command trajectory, the error and standard deviation decrease in magnitude. The Z-score plot shows a similar trend. When a turn is reached, which happens every 20 seconds, the Z-Score performance metric experiences a sharp increase in value. If the controller manages to drive the vehicle back to the trajectory, the Z-score decreases and so does the error norm and the standard deviation. When the bend at 60 seconds time is reached, the Z-score grows to become larger than the previously recorded peak and larger than the threshold, and that resets the gains. During the same loop in which the gains reset, the adaptation law updates the gains and pushes them to the controller. This is different from the case in Figure 51 in which continuous adaptation is simulated with the same controller and trajectory type. Here, the gains are shown to be under adaptation throughout the tracking task, the gain values are higher when compared to the gain management algorithm, and the tracking performance is worse as previously detailed in the tracking performance results.

In Figure 50 and Figure 51, the performance of devised gain management algorithm is tested against the case where adaptation is continuous and is not managed by the algorithm. The ANC is run on the parametrized polynomial paths trajectory, and the algorithm results are documented. The simulation results clearly show the advantage of the gain management algorithm as the ANC in continuous adaptation mode eventually diverges as a consequence of the large gains. The adaptation governed by the proposed algorithm, however, still manages to keep the tracking error from increasing without bound.

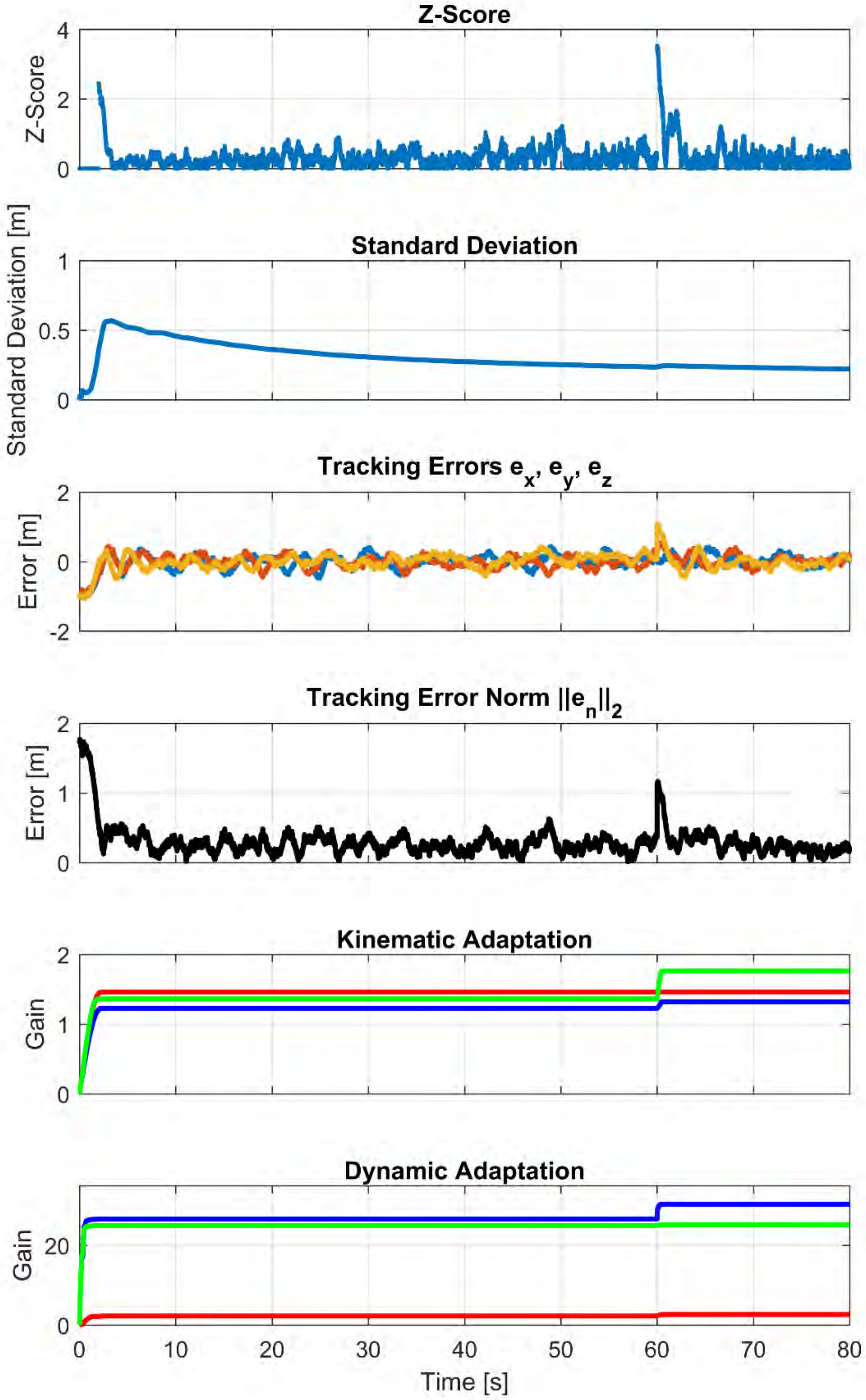


Figure 48: Gain Update Algorithm without Water Current Disturbance (Adaptive Nussbaum Controller on the Parametrized Polynomial Paths)

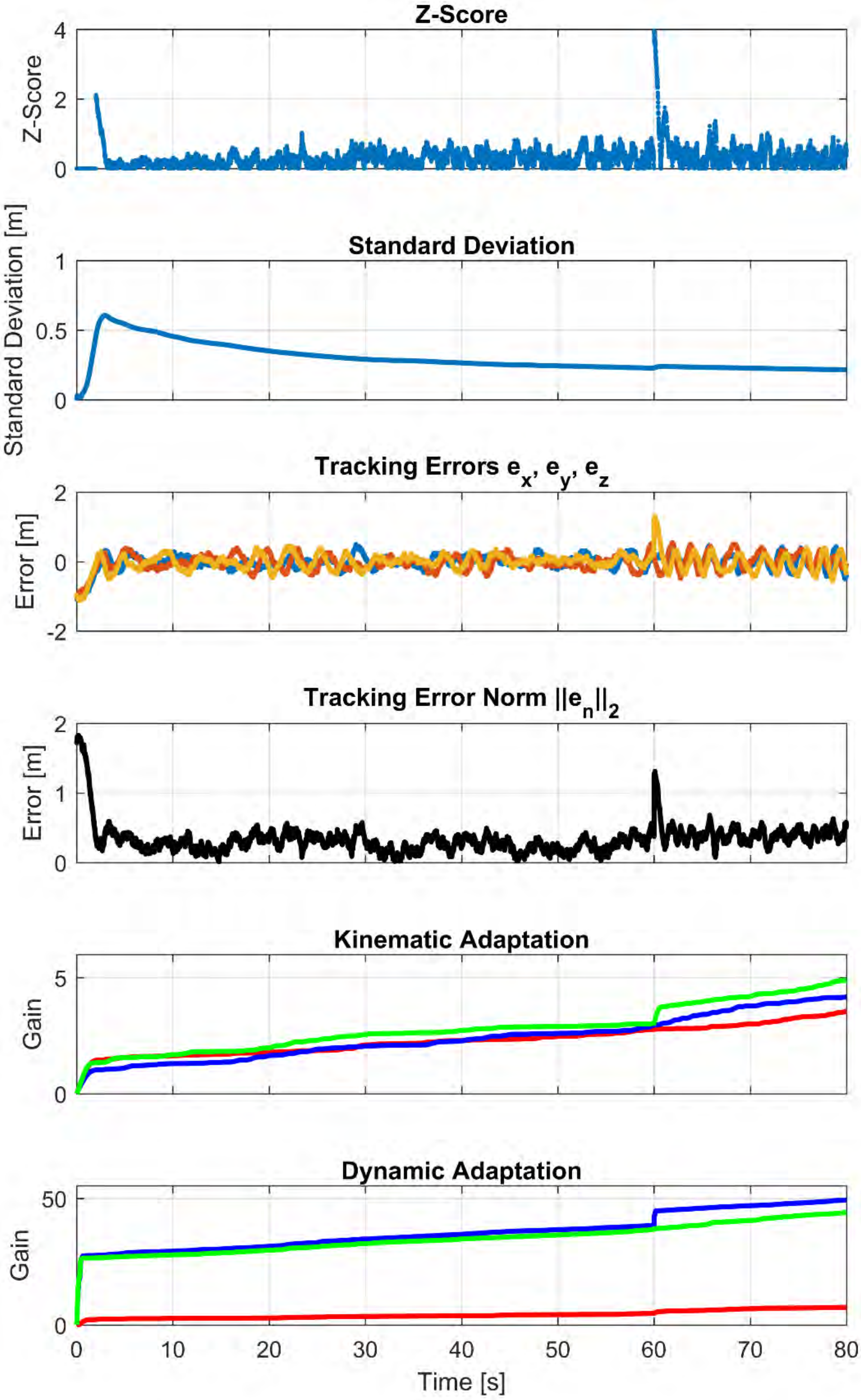


Figure 49 Continuous Gain Update without Water Current Disturbance (Adaptive Nussbaum Controller on the Parametrized Polynomial Paths)

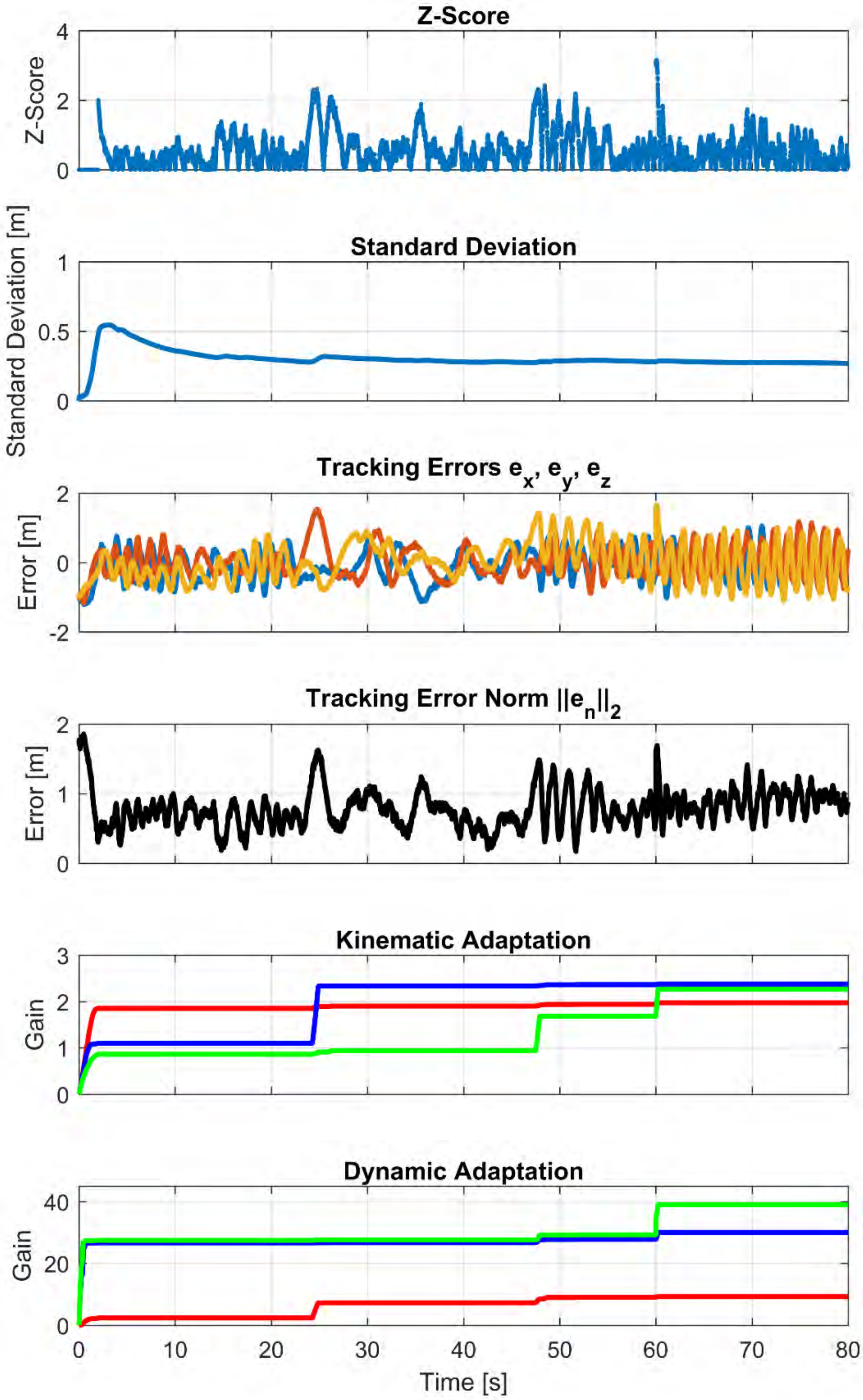


Figure 50: Gain Update Algorithm with Water Current Disturbance (Adaptive Nussbaum Controller on the Parametrized Polynomial Paths)

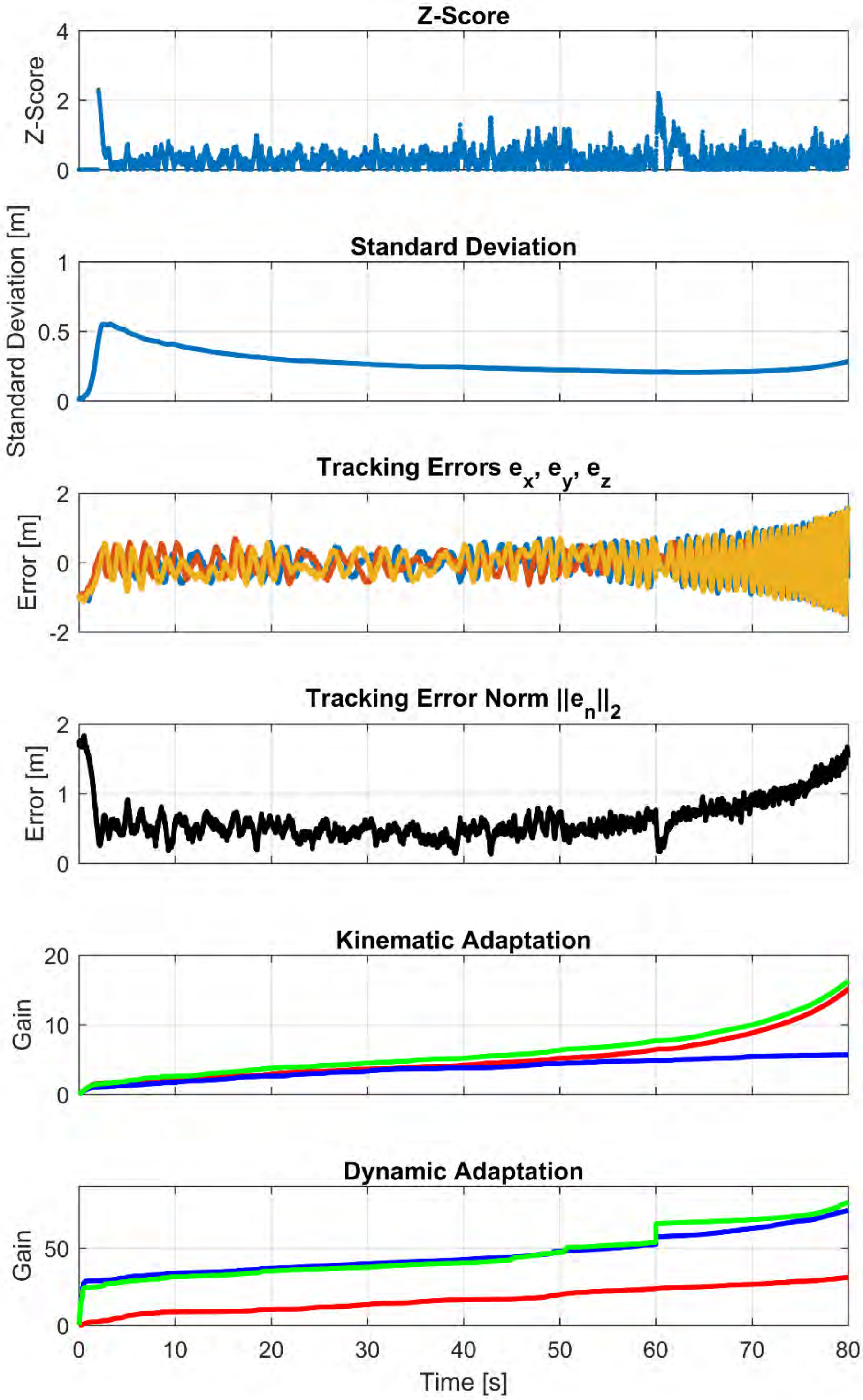


Figure 51: Continuous Gain Update with Water Current Disturbance (Adaptive Nussbaum Controller on the Parametrized Polynomial Paths)

Chapter 6: Navigation and Localization

Navigation and localization are crucial to the functionality of an AUV, for accurate position information is not always available. Measurements of the vehicle states are often corrupted with noise arising from both the sensors and the environment. The underwater environment is one in which slowly varying or constant disturbances in the form of ocean currents and waves are likely to happen [37]. A filter which makes use of the structural dynamic equations of motion in estimating the vehicle states is necessary for successful deployment of UVs. In this chapter, the problems that problems that Inertial Measurement Units (IMUs) suffer from are introduced. Then, an Unscented Kalman Filter (UKF) and an Extended Kalman Filter (EKF) are used to fuse the sensor measurements with the dynamic model to provide a vehicle state estimate.

6.1. Problems with Inertial Measurement Units

A strap-down IMU consists of a triad of orthogonally mounted accelerometers, gyroscopes, and, often, magnetometers. The sensor package is mounted along the b-frame or the body frame of the vehicle. Various errors affect the performance of an IMU; the accelerometer and gyroscope measurements typically suffer from biases, scale factors, misalignments, and noise. The biases that affect measurements are not necessarily constant. They are susceptible to drift. The scale factor of the IMU can be a linear or a nonlinear function of the measurement. The triad of the IMU sensors can also suffer from misalignment between the three axes of each sensor. The noise affecting IMU measurement can be a result of the sensor package internal circuitry or the body they are mounted on. The noise is usually modeled as Gaussian and white [44, 45, 58, 59]. These devices are used in inertial navigation to estimate motion, and they require correction from sources like GPS to function properly.

The following describes the quantities measured by the IMU, where $\boldsymbol{\varepsilon}_i$ is a vector of zero-mean white Gaussian noises. The acceleration measured by the IMU is a function of the real acceleration vector, $\boldsymbol{a}^b$, first order scale factors and misalignment error, $\boldsymbol{\Delta}(s_a, \varphi_a)$, an accelerometer bias, $\boldsymbol{b}_a$, the acceleration of gravity, $\mathbf{g}^b$, and some zero-mean white Gaussian noise, $\boldsymbol{\varepsilon}_i$. The rate of change of the bias is modeled as a random walk (Wiener process). The b superscript refers to the quantities that are describes in the b-frame.

$$\dot{\boldsymbol{v}}_{IMU}{}^{b} = [\mathbf{I} + \boldsymbol{\Delta}(s_a, \varphi_a)]\boldsymbol{a}^b + \boldsymbol{b}_a + \mathbf{g}^b + \boldsymbol{\varepsilon}_1$$
$$\dot{\boldsymbol{b}}_a = \boldsymbol{\varepsilon}_2 \tag{76}$$

Similar to the acceleration measurement model, the angular velocity measured by the IMU as a function of the real angular velocity vector, first order scale factors and misalignment error, bias, and noise. The rate of change of the bias is also modeled as a random walk (Wiener process). The b superscript refers to the quantities that are describes in the b-frame.

$$\boldsymbol{\omega}_{IMU}^{b} = \left[\mathbf{I} + \boldsymbol{\Delta}\left(s_g, \varphi_g\right)\right]\boldsymbol{\omega}^b + \boldsymbol{b}_G + \boldsymbol{\varepsilon}_3$$
$$\dot{\boldsymbol{b}}_G = \boldsymbol{\varepsilon}_4 \tag{77}$$

The magnetometer measurement model, similar to the acceleration and angular velocity measurement models, is a function of the real local magnetic field strength, first order scale factors, and misalignment error, bias, and noise. The rate of change of the bias is also modeled as a random walk (Wiener process). The b superscript refers to the quantities that are describes in the b-frame.

$$\boldsymbol{H}_{IMU}^{b} = \left[\mathbf{I} + \boldsymbol{\Delta}\left(s_g, \varphi_g\right)\right]\boldsymbol{H}^b + \boldsymbol{b}_m + \boldsymbol{\varepsilon}_5$$
$$\dot{\boldsymbol{b}}_G = \boldsymbol{\varepsilon}_6 \tag{78}$$

Unlike the other two sensors, the magnetometer reading is very easy to calibrate. [60, 61] present methods that can be used to calibrate the magnetometer and identify the scale and bias terms. Calibration is accomplished through knowing the local magnetic field strength, which is constant for a given location. Often, that information alongside a set of measurements are fed into an optimization routine to approximate the measurement model. Further, having a calibrated magnetometer allows for the use of it in the attitude estimation problem.

The $[\mathbf{I} + \boldsymbol{\Delta}(s, \varphi)]$ term in both measurement models represents the scale factors and misalignment errors of the IMU. The values that this matrix can take constant values, or they can be slowly changing like the biases of the sensors. Consequently, they are either determined through calibration of the IMU sensors before deployment, or they are included in the state estimation problem as random walk processes [46, 58]. **I** is the identity matrix, and the details of the comprising components are shown below.

$$\boldsymbol{\Delta}(s,\varphi) = \begin{bmatrix} s_x & \varphi_{xy} & \varphi_{xz} \\ \varphi_{yx} & s_y & \varphi_{yz} \\ \varphi_{zx} & \varphi_{zy} & s_z \end{bmatrix} \tag{79}$$

The sensors are often calibrated before deployment, and online identification of the errors is only possible if enough information about the motion of the vehicle is attainable. In other words, the observability of the dynamic system is a requirement [62].

6.2. Sensor Fusion Formulation

The need to use sensor fusion techniques on the vehicle arises because of the noisy nature of the underwater environment. The inertial solution is not perfect as discussed in the literature review, so successful deployment of the AUV requires the sensor packages to provide more information about the vehicle state than what an IMU offers. A doppler velocity log as well as an underwater localization sensor package are assumed to be available in the development of this chapter.

To formulate the filter, the dynamic model given by

$$\mathbf{M}\,\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{\mathcal{G}}(\boldsymbol{\eta}) = \boldsymbol{\tau}_{\boldsymbol{actuat}\mathrm{o}\boldsymbol{r}} \tag{80}$$

is rewritten to isolate $\dot{\boldsymbol{\nu}}$ as

$$\begin{aligned} \dot{\boldsymbol{\nu}} &= -\boldsymbol{M}^{-\mathbf{1}}\big(\mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{\mathcal{G}}(\boldsymbol{\eta})\big) + \boldsymbol{M}^{-\mathbf{1}}\boldsymbol{\tau}_{\boldsymbol{actuat}\mathrm{o}\boldsymbol{r}} \\ \dot{\boldsymbol{\nu}} &= \boldsymbol{M}^{-\mathbf{1}}\boldsymbol{f}(\boldsymbol{\eta},\boldsymbol{\nu}) + \boldsymbol{M}^{-\mathbf{1}}\boldsymbol{u} \end{aligned} \tag{81}$$

and the kinematic model is described by

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} \dot{\mathbf{P}}_n \\ \dot{\boldsymbol{q}} \end{bmatrix} = \mathbf{J}_{\boldsymbol{q}}(\boldsymbol{q})\,\boldsymbol{\nu} = \begin{bmatrix} \mathbf{R}_b^n(\boldsymbol{q}) & \mathbf{0}_{3x3} \\ \mathbf{0}_{4x3} & \mathbf{T}(\boldsymbol{q}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\upsilon}_b \\ \boldsymbol{\omega}_b \end{bmatrix} \tag{82}$$

The complete model can then be described by

$$\begin{bmatrix} \dot{\boldsymbol{\nu}} \\ \dot{\boldsymbol{\eta}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}^{-\mathbf{1}}\boldsymbol{f}(\boldsymbol{x},t) + \boldsymbol{M}^{-\mathbf{1}}\boldsymbol{u}(t) \\ \mathbf{J}_{\boldsymbol{q}}(\boldsymbol{q})\,\boldsymbol{\nu} \end{bmatrix} = \boldsymbol{F}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{t}) \tag{83}$$

and the state vector of the dynamic system is:

$$\boldsymbol{x} = [\boldsymbol{\upsilon}_b \quad \boldsymbol{\omega}_b \quad \mathbf{P}_n \quad \boldsymbol{q}\,]^T, \qquad \boldsymbol{x} \in \mathbb{R}^{13} \tag{84}$$

The discrete form of the above model is realized through Euler integration. Also, additive dynamics zero-mean white Gaussian noise, $\boldsymbol{w}_k \sim \mathcal{N}(0, \boldsymbol{Q}_{\boldsymbol{w}})$ is included to characterize any disturbances or uncertainties in the model. Based in forward Euler integration, the discrete form of the nonlinear dynamic plant model can be written as:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{t}_k)\boldsymbol{\Delta t} + \boldsymbol{w}_k \tag{85}$$

A measurement equation is necessary to depict the information the sensors provide and relate them to the state vector quantities. This measurement model is disturbed by a vector of zero-mean white Gaussian noise denoted $\boldsymbol{v}_k \sim \mathcal{N}(0, \boldsymbol{R}_v)$. The measurement model is presented below, where $\boldsymbol{h}(\boldsymbol{x}_k)$ is the vector-valued measurement function.

$$\boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{86}$$

The discrete form of the measurement model is:

$$\boldsymbol{h}(\boldsymbol{x}_{\mathrm{k}}, t_k) = \begin{bmatrix} \boldsymbol{\upsilon}_k^b & \boldsymbol{\omega}_k^b & \mathbf{P}_{n_k} & \boldsymbol{q}_k \end{bmatrix}^T \tag{87}$$

where $\boldsymbol{\upsilon}_k^b$ is the linear vehicle velocity measured in the b-frame by a doppler velocity log, $\boldsymbol{\omega}_k^b$ is the angular vehicle velocity measured in the b-frame by a calibrated inertial measurement unit, $\mathbf{P}_{n_k}$ is the position of the vehicle measured in the n-frame by an acoustic time-of-flight localization setup, and $\boldsymbol{q}_k$ is the quaternion describing the attitude of the vehicle as reported by the calibrated inertial measurements unit. The measurements model considered here is linear, and the measurement model can be written as:

$$\boldsymbol{z}_k = \mathbf{H}\, \boldsymbol{x}_k + \boldsymbol{v}_k \tag{88}$$

where $\mathbf{H} = \mathbf{I}_{13\times13}$ is the identity matrix.

6.3. Extended Kalman Filter (EKF)

The Kalman Filter (KF) is a linear Minimum Mean Square Error estimation technique, and the dynamic and measurements models dealt with are both nonlinear in the states. Therefore, it is necessary to linearize the models around the best estimate available at that instant. Unlike the KF, for which stability can be proven, the stability of the EKF filter is nontrivial and is heavily reliant on the linearization to not diverge [63]. The algorithm usually employs a first order Taylor series expansion for the dynamics and measurement equations around the estimate, $\hat{\boldsymbol{x}}$, to linearize them [64]. Alternate formulations of the EKF, like the one utilized here, make use of Euler discretization. Also, $\boldsymbol{f_x}(\hat{\boldsymbol{x}}, \boldsymbol{t})$ represents the Jacobian of the dynamics equation that is utilized in the EKF algorithm. Notice that the measurement equation is linear in the state vector. The Jacobian of the measurement equation is the identity matrix. The Jacobian for the dynamics equation is shown in equation (89).

$$\boldsymbol{f_x}(\hat{\boldsymbol{x}}, \boldsymbol{t}) = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}\right|_{\hat{\boldsymbol{x}}_k} \tag{89}$$

The Jacobian of the dynamics equation is presented in Appendix C. The EKF algorithm is presented in Figure 52.

Initialization	$\boldsymbol{E}[\boldsymbol{x_0} \mid \boldsymbol{z_0}] = \hat{\boldsymbol{x}}_{\boldsymbol{0}} \qquad \mid \qquad \boldsymbol{cov}[\boldsymbol{x_0} \mid \boldsymbol{z_0}] = \boldsymbol{P_0}$
State Propagation	$\bar{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k \mid k-1} = \boldsymbol{F}(\boldsymbol{x}_{k-1}, \hat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_{k-1})$
Covariance Propagation	$\boldsymbol{P}_{k \mid k-1} = \bar{\boldsymbol{P}}_k = \boldsymbol{\Phi}_{k-1} P_{k-1} \boldsymbol{\Phi}_{k-1}^{\mathrm{T}} + \boldsymbol{\Gamma}\mathbf{Q}_{k-1}\boldsymbol{\Gamma}^{\mathrm{T}}$
Kalman Gain	$\mathbf{K}_k = \bar{\boldsymbol{P}}_K \boldsymbol{h}_{\boldsymbol{x}_K}^{T} [\boldsymbol{h}_{\boldsymbol{x}_k} \bar{\boldsymbol{P}}_K \boldsymbol{h}_{\boldsymbol{x}_k}^{T} + \boldsymbol{R}_k]^{-\mathbf{1}}$
State Update	$\hat{\boldsymbol{x}}_K = \bar{\boldsymbol{x}}_K + \mathbf{K}_k[z_k - \boldsymbol{h}_k(\boldsymbol{x}_k)]$
Covariance Update	$\hat{\boldsymbol{P}}_K = [\mathbf{I} - \mathbf{K}_k \boldsymbol{h}_{\boldsymbol{x}_k}(\boldsymbol{x}_k)] \bar{\boldsymbol{P}}_K [\mathbf{I} - \mathbf{K}_k \boldsymbol{h}_{\boldsymbol{x}_k}(\boldsymbol{x}_k)]^T + \mathbf{K}_k \boldsymbol{R}_k \mathbf{K}_k^T$

Figure 52: Extended Kalman Filter Algorithm

where forward Euler integration yields:

$$\begin{gathered} \boldsymbol{F}(\boldsymbol{x}_k, \hat{\boldsymbol{x}}_k, \boldsymbol{u}_k) = \hat{\boldsymbol{x}}_k + \Delta t[\boldsymbol{M}^{-\mathbf{1}}\boldsymbol{f}(\boldsymbol{x}, t) + \boldsymbol{M}^{-\mathbf{1}}\boldsymbol{u}(t)] \\ \boldsymbol{\Phi}_{k-1} = \mathbf{I} + \Delta t[\boldsymbol{f_x}(\hat{\boldsymbol{x}}, \boldsymbol{t})] \\ \boldsymbol{\Gamma} = \Delta t\mathbf{M}^{-\mathbf{1}} \end{gathered} \tag{90}$$

6.4. Simulation Results

Here, the Adaptive Nussbaum Controller is integrated with the EKF as a sensor fusion algorithm to provide the controller feedback. The details of the simulation environment developed to test the algorithm are presented in Table 15. The waypoints used to generate the reference trajectory are shown in Table 16. Figure 53 and Figure 54 present the trajectory tracking results that employ an EKF to provide controller feedback and the EKF state estimation error, respectively.

.

Table 15: Simulation Conditions

	Value
Initial Pose	$\boldsymbol{\eta} = [0.1 \quad 0.1 \quad 0.1 \quad 0.7071 \quad 0 \quad 0 \quad 0.7071\ ]^{\boldsymbol{T}}$
Dynamics noise	$\boldsymbol{w} \sim \mathcal{N}\left(0, \mathbf{10^{-2}}\right), \qquad \mathbf{w} \in \mathbb{R}^{13}$
Measurements noise	$\boldsymbol{v} \sim \mathcal{N}\left(0, \mathbf{10^{-2}}\right), \qquad \boldsymbol{v} \in \mathbb{R}^{13}$

Table 16: Waypoints used to Parametrize Straight Trajectories

	From Waypoint $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ $[\boldsymbol{m}]$	**To Waypoint $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})[\boldsymbol{m}]$**
Path 1	$[\mathbf{0}, \mathbf{0}, \mathbf{0}]$	$[\mathbf{3}, \mathbf{0}, \mathbf{3}]$
Path 2	$[\mathbf{3}, \mathbf{0}, \mathbf{3}]$	$[\mathbf{3}, \mathbf{3}, \mathbf{3}]$
Path 3	$[\mathbf{3}, \mathbf{3}, \mathbf{3}]$	$[\mathbf{0}, \mathbf{3}, -\mathbf{3}]$
Path 4	$[\mathbf{0}, \mathbf{3}, -\mathbf{3}]$	$[\mathbf{0}, \mathbf{0}, \mathbf{0}]$

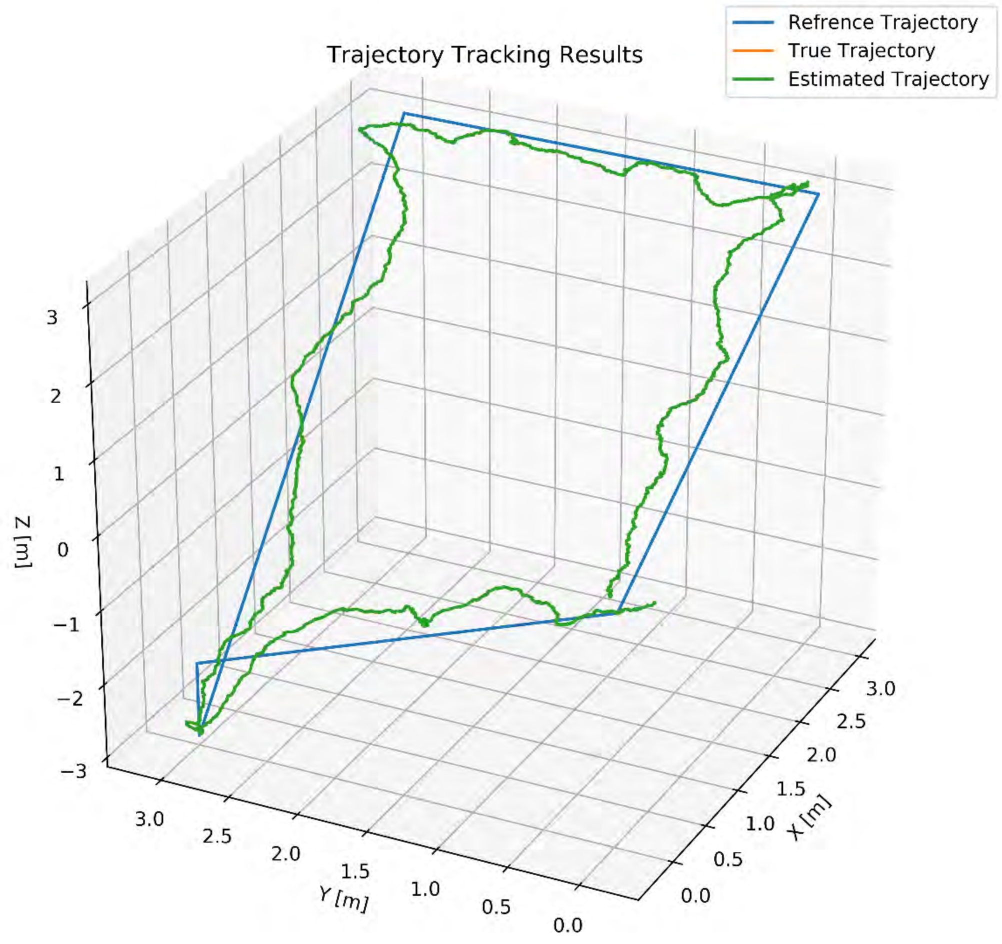


Figure 53: Trajectory Tracking with Sensor Fusion Providing Feedback

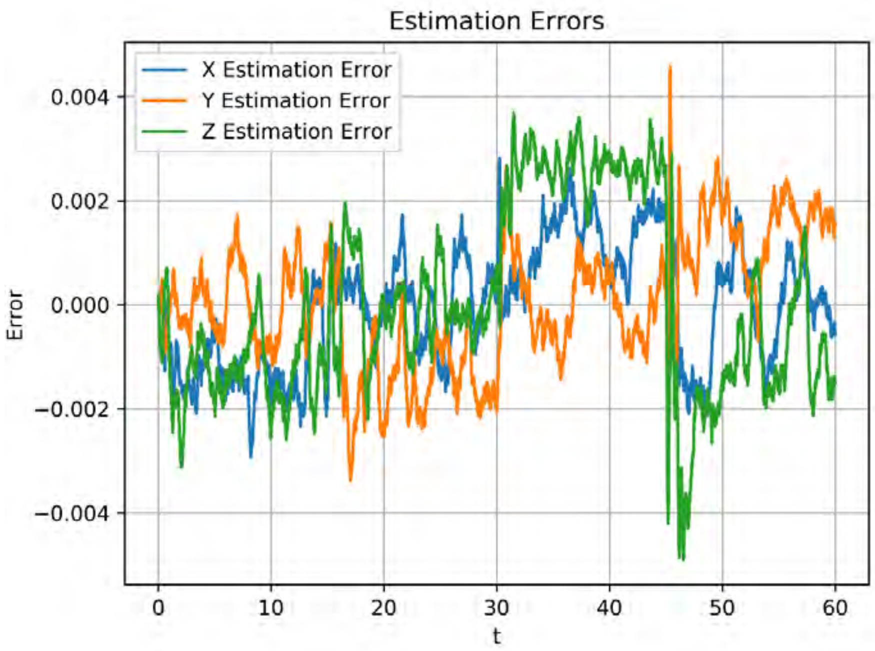


Figure 54: EKF Estimation Results

The estimate of the EKF is very close to the noisy true state, as shown in the reported estimation errors. The Mean Square Error for the run shown above is reported to be $6.1451 \times 10^{-6}\, m$. The estimation results are very good since the true dynamics and measurements statistics are known. The filter then works in an optimal manner.

6.5. Unscented Kalman Filter (UKF)

While the Extended Kalman Filter is a widely applied algorithm for state estimation, it has many shortcomings when the system nonlinearities are significant. The linearization performed to apply the Kalman Filter algorithm to nonlinear systems might not be of sufficient order to capture the behavior accurately. As a result, the algorithm might diverge, or it might give unreliable estimates. A substitute to propagating the mean and covariance of the state of interest comes in the form of the Unscented Kalman Filter. The idea behind the UKF is that it is easier to approximate a probability density function than to approximate a nonlinear system. This greatly reduces the approximation errors of the UKF when compared to the EKF since the former only approximates the probability distribution functions [64].

The tool used to approximate the probability distribution functions used in the estimation process is the unscented transformation. The unscented transformation is based on the fact that it is possible to perform a nonlinear transformation on a single point. Also, it possible to find a set of points in the state space that can be sampled to approximate the true probability density function of the state vector. These points whose ensemble mean and covariance represent the true mean and covariance for the state vector are called sigma points. The unscented transformation is summarized in Figure 55.

For the vector $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{1}}$ that gets transformed by the nonlinear function $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$,	
Choose $\boldsymbol{2n}$ sigma points, $\boldsymbol{x}^{(\boldsymbol{i})}$	$\boldsymbol{x}^{(i)} = \overline{\boldsymbol{x}} + \widetilde{\boldsymbol{x}}^{(i)}, \quad i = 1,\ldots,2n$ $\widetilde{\boldsymbol{x}}^{(i)} = \sqrt{n\boldsymbol{P}}_i, \qquad i = 1,\ldots,n$ $\widetilde{\boldsymbol{x}}^{(n+i)} = -\sqrt{n\boldsymbol{P}}_i, \quad i = 1,\ldots,n$
Transform the sigma points	$\boldsymbol{y}^{(\boldsymbol{i})} = f\left(\boldsymbol{x}^{(i)}\right), \qquad i = 1,\ldots,2n$
Mean approximation	$\boldsymbol{y}_{\boldsymbol{u}} = \frac{1}{2n}\sum_{i=1}^{2n} \boldsymbol{y}^{(\boldsymbol{i})}$
Covariance approximation	$\boldsymbol{P}_{\boldsymbol{u}} = \frac{1}{2n}\sum_{i=1}^{2n} \left(\boldsymbol{y}^{(i)} - \boldsymbol{y}_{\boldsymbol{u}}\right)\left(\boldsymbol{y}^{(i)} - \boldsymbol{y}_{\boldsymbol{u}}\right)^{\boldsymbol{T}}$

Figure 55: Unscented Transformation Algorithm

where $\sqrt{n\boldsymbol{P}}$ is the matrix square root of $n\boldsymbol{P}$ such that $\sqrt{n\boldsymbol{P}}\sqrt{n\boldsymbol{P}}^T = n\boldsymbol{P}$ and $\sqrt{n\boldsymbol{P}}_i$ is the ith column of $\sqrt{n\boldsymbol{P}}$. Cholesky Decomposition or a Singular Value Decomposition are approaches through which the matrix square root can be carried out.

The Unscented Kalman Filter algorithm is similar in form to the Extended Kalman Filter one except for the usage of unscented transformations to propagate the means and covariances. Figure 56 summarizes the algorithm.

The need to compute the matrix square root eliminates the UKF as an option to provide real time filtering. Testing in the MATLAB environment showed that the Cholskey decomposition as well as Singular Value Decomposition fail to provide a result as the covariance matrix loses full rank at times during the simulations.

6.6. Uncertainty structure Identification

The underlying uncertainty structure arising from the process and the measurement method/device are two noise sequences modeled to be zero-mean normal Gaussian white noise processes. The magnitudes of the process and measurement noise covariance processes are needed for optimal estimation of the state vector. Literature presents a multitude of methods that exploit a set of available data and the structure of the plant and measurement equations in formulating optimization problems that can be solved for the unknown uncertainty structure. Bayesian, Maximum likelihood, Correlation, and Autocovariance least-squares are examples of such approaches [65, 66].

Initialization	$\boldsymbol{E}[\boldsymbol{x_0}\|\boldsymbol{z_0}] = \boldsymbol{\hat{x}_0}$ | $\boldsymbol{cov}[\boldsymbol{x_0}\|\boldsymbol{z_0}] = \boldsymbol{P_0}$
Unscented Transformation	$\hat{x}_{k-1} \xrightarrow[\substack{Unscented \\ Transformation}]{} \boldsymbol{\hat{x}}_k{}^{(i)}$
State *a priori* estimate	$\boldsymbol{\hat{x}}_K^- = \frac{1}{2n}\sum_{i=1}^{2n} \boldsymbol{\hat{x}}_k{}^{(i)}$
Covariance *a priori* estimate	$\boldsymbol{P}_k^- = \frac{1}{2n}\sum_{i=1}^{2n} \left(\boldsymbol{\hat{x}}_k{}^{(i)} - \boldsymbol{\hat{x}}_K^-\right)\left(\boldsymbol{\hat{x}}_k{}^{(i)} - \boldsymbol{\hat{x}}_K^-\right)^{\boldsymbol{T}} + \boldsymbol{Q}_k$
Unscented Transformation	$\boldsymbol{\hat{x}}_K^- \xrightarrow[\substack{Unscented \\ Transformation}]{} \boldsymbol{\hat{x}}_k{}^{(i)}$
Measurement Transformation	$\boldsymbol{\hat{z}}_k{}^{(i)} = \boldsymbol{h}\left(\boldsymbol{\hat{x}}_k{}^{(i)}\right)$
Measurement prediction	$\boldsymbol{\hat{z}}_k = \frac{1}{2n}\sum_{i=1}^{2n} \boldsymbol{\hat{z}}_k{}^{(i)}$
Covariance prediction	$\boldsymbol{P}_z = \frac{1}{2n}\sum_{i=1}^{2n} \left(\boldsymbol{\hat{z}}_k{}^{(i)} - \boldsymbol{\hat{z}}_k\right)\left(\boldsymbol{\hat{z}}_k{}^{(i)} - \boldsymbol{\hat{z}}_k\right)^{\boldsymbol{T}} + \boldsymbol{R}_k$ $\boldsymbol{P}_{xz} = \frac{1}{2n}\sum_{i=1}^{2n} \left(\boldsymbol{\hat{x}}_k{}^{(i)} - \boldsymbol{\hat{x}}_K^-\right)\left(\boldsymbol{\hat{z}}_k{}^{(i)} - \boldsymbol{\bar{z}}_k\right)^{\boldsymbol{T}}$
Kalman Gain	$\mathbf{K}_k = \boldsymbol{P}_{xz}\boldsymbol{P}_z^{-1}$
State Update	$\boldsymbol{\hat{x}}_K^+ = \boldsymbol{\hat{x}}_K^- + \mathbf{K}_k[z_k - \boldsymbol{\hat{z}}_k]$
Covariance Update	$\boldsymbol{P}_K^+ = \boldsymbol{P}_k^- - \mathbf{K}_k\boldsymbol{P}_z\mathbf{K}_k^{\boldsymbol{T}}$

Figure 56: Unscented Kalman Filter Algorithm

Chapter 7. System Integration

This chapter details the integration of the different components and subsystems of the vehicle. First, the hardware specifications are presented. Second, a suggested localization approach is detailed. Finally, the software architecture with which the hardware is integrated is explained, and a developed simulation environment that enables prototyping and testing of algorithms for underwater vehicles is discussed.

7.1. Hardware

The experimental hardware used is detailed here. First is the Blue Robotics T100 thrusters are used to provide actuation. The three-phase brushless motors running the thruster are controlled using electronic speed controllers (ESC), and a pulse-width modulated (PWM) signal is used as input to the thruster. The manufacturer provides steady state thrust, speed, and power draw charts for the thrusters. Table 17 presents the thruster specifications. For the IMU, a high-end consumer grade MIDG II solution is used. It provides acceleration, angular velocity, and magnetic field strength measurements at a rate of 50 Hz. Table 18 and Table 19 summarize the relevant specifications. For depth estimation, a pressure sensor is used. It provides a temperature-compensated high-resolution pressure output, and it is water proof up to 300 m. Table 20 summarizes the relevant specifications. Figure 57 outlines the architecture of the system. It shows how the components are integrated and how they communicate with each other.

Table 17: T100 Thruster Specification

Quantity	**Value**
Maximum Thrust (Forward | Reverse)	$2.36\ kgf$ | $1.82\ kgf$
Minimum Thrust	$0.01\ kgf$
Rotational Speed	$300 - 4200\ RPM$
Maximum Power	$130\ W$
Recommended Voltage	$12\ V$

Table 18: Rate Gyro Specifications

Quantity	**Value**
Range of Operation	$\pm\ 300^o/Sec$
Non-linearity	$0.1\ \%\ of\ Full\ Scale$
Noise Density	$0.1^o/Sec\ /\sqrt{Hz}$
3dB Bandwidth	$20\ Hz$

Table 19: Accelerometer Specifications

Quantity	**Value**
Range of Operation	$\pm 6\ g$
Non-linearity	$0.3\ \%\ of\ Full\ Scale$
Noise Density	$150\ \mu g/\sqrt{Hz}$
3dB Bandwidth	$20\ Hz$

Table 20: Pressure Sensor Specification

Quantity	**Value**
Maximum Mechanical Pressure	$50\ bar$
Operating Pressure	$0 \rightarrow 30\ bar$
Resolution	$0.2\ mbar$

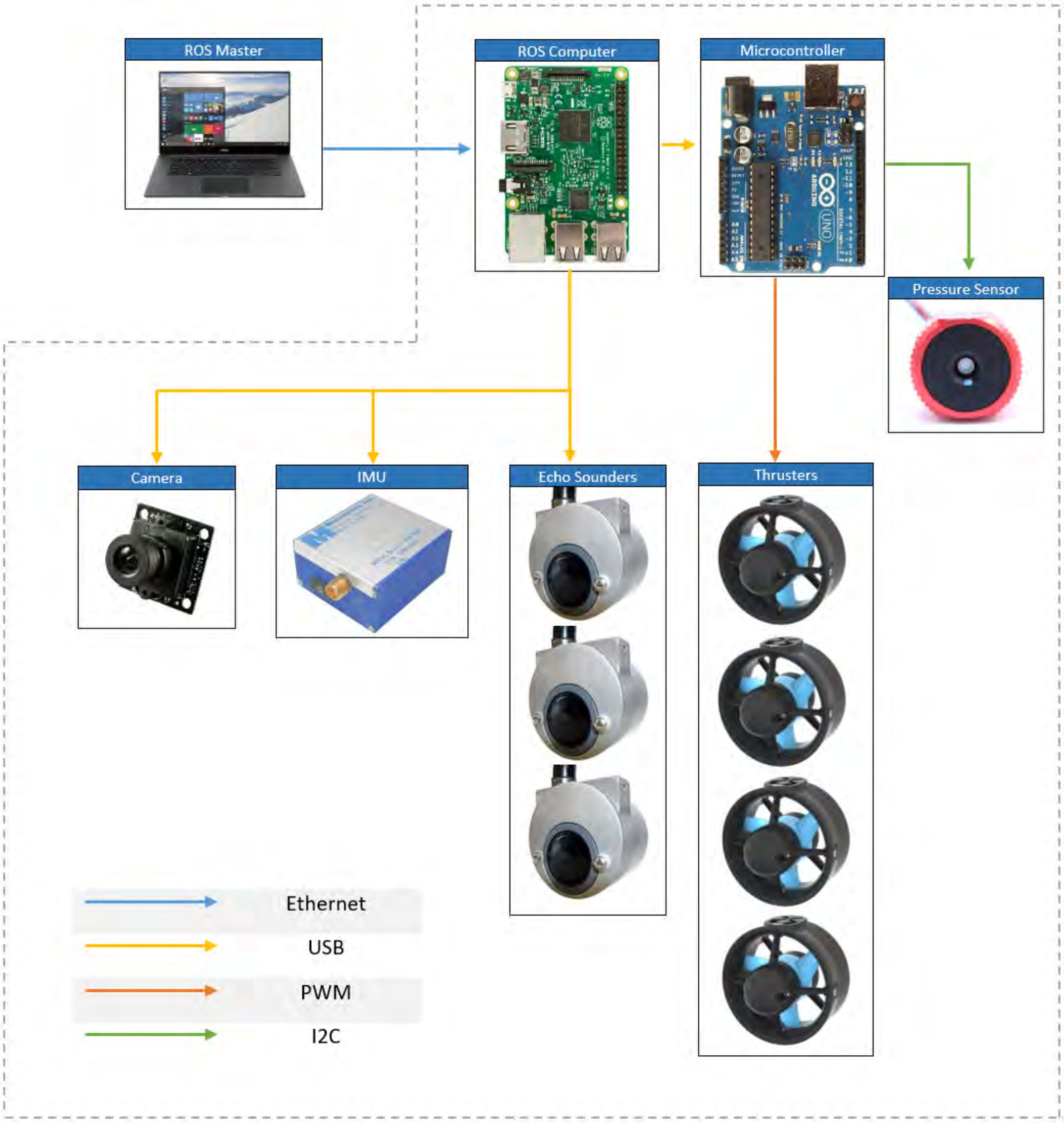


Figure 57: System Integration Diagram

7.2. Localization

Acoustic localization equipment is the most widely used option when it comes to underwater localization; however, they are very expensive and were not available to validate this work. Instead, the proposed localization approach was through Visual Servoing (VS) and pressure measurements.

VS is the technique by which visual information is utilized in the control of robots. Here, VS provided measurements of the location of a float that represents the position of the vehicle inside the pool. The float was tied to the vehicle, which were given a constant depth test trajectory such that the float stays within a reasonable range from the true location of the vehicle. Pressure measurements provided a high-resolution measurement of the depth of the vehicle.

Visual Servoing was implemented on a monocular camera. The camera tracked a target of know size. Having the size of the object and a camera model allows the estimation of the position of the object within the visible image frame. The location here is measured with respect to a characteristic point called the principal point, which represents the origin of the coordinate frame in the images captured by the camera.

To localize an object in a camera image frame, the camera must be modeled. The pinhole camera model is a popular model used to represent images captured by monocular cameras. In this model, the scene is formed by projecting 3D points into the image plane using a perspective transformation [67]. The model is given by

$$s\,m' = ATM' = A[R|t]M'$$
$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{91}$$

where $x, y,$ and z are the coordinates of a 3D point in the world coordinate space; u and v are the coordinates of the projection point in pixels; A is the intrinsic parameters matrix of the camera; c_x and c_y are the principal points located at the image center; and f_x and f_y are the focal lengths expressed in pixels units. Figure 58 presents an overview of the camera image frame and how a point is depicted in its view.

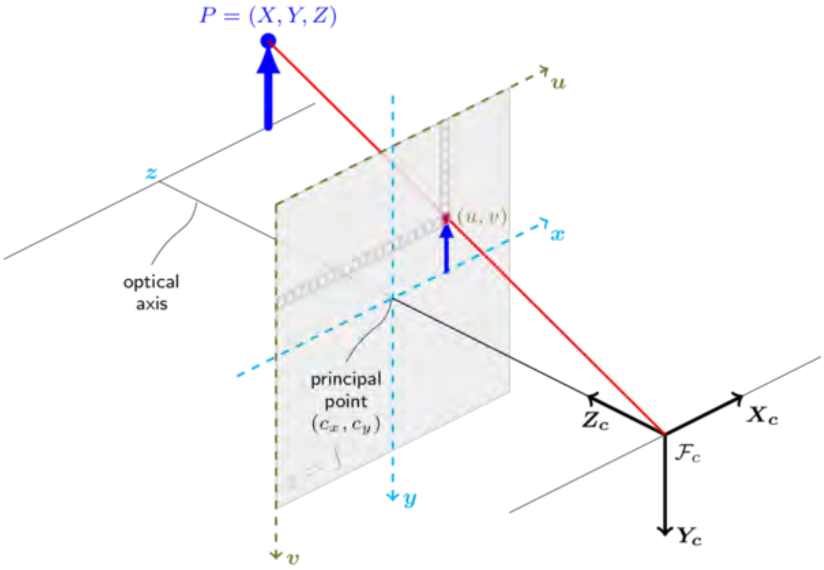


Figure 58: Pinhole Camera Model [67]

It is of interest to note that if an image from the camera is scaled by a factor due to a change in the camera resolution, all the aforementioned parameters should be scaled by the same factor. The characteristic matrix of camera intrinsic parameters does not depend on the scene viewed. Rather, it is fixed for a fixed focal length. The transformation matrix, $T = [R|t]$, combines rotation and\or translation of the camera. Consequently, it describes camera motion around a static scene, or rigid motion of an object in front of a still camera. The transformation matrix is called the matrix of camera extrinsic parameters, and it relates the coordinates of a point to the camera-fixed coordinate system.

Real lenses typically suffer from distortion. Figure 59 presents the different types of distortion. Distortion can be corrected for after calibration.

$$\begin{aligned} x_{undistorted} &= x_d \frac{1 + K_1 r^2 + K_2 r^4 + K_3 r^6}{1 + K_4 r^2 + K_5 r^4 + K_6 r^6} + 2P_1 x_d y_d + P_2(r^2 + 2x_d^2) \\ y_{undistorted} &= y_d \frac{1 + K_1 r^2 + K_2 r^4 + K_3 r^6}{1 + K_4 r^2 + K_5 r^4 + K_6 r^6} + 2P_2 x_d y_d + P_1(r^2 + 2y_d^2) \end{aligned} \tag{92}$$

where x_d and y_d are the distorted image coordinates, K_i and P_i are the radial and tangential distortion coefficients, respectively, and $r = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}$ is the distance between the distorted point coordinate and the camera principal point. Unlike the other coefficients, the distortion parameters are insensitive to changes in the resolution of the camera and do not need to be scaled accordingly.

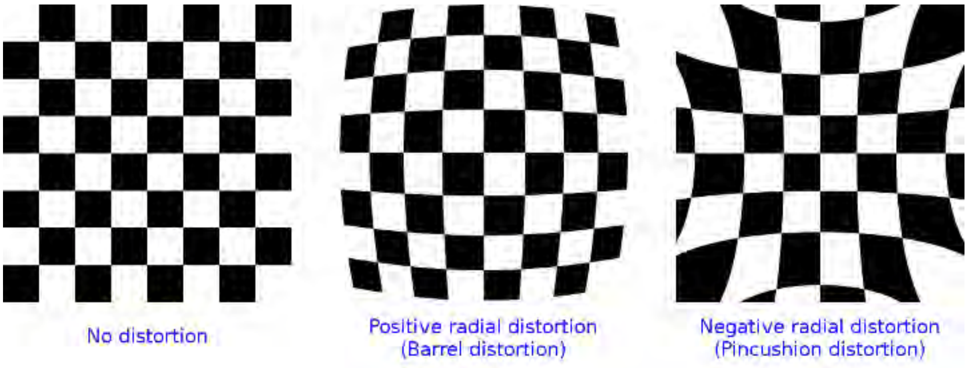


Figure 59: Types of Distortion

To test the proposed localization approach through Visual Servoing. A tag, whose dimensions are known, is moved along a known trajectory, and the measurements are processed using the open source Visual Servoing Platform (ViSP), which provides functions that are exploited to work with the available equipment. Figure 60 shows the reported trajectory of the moving tags based on ViSP measurements against the known ground truth that the tag is moved along.

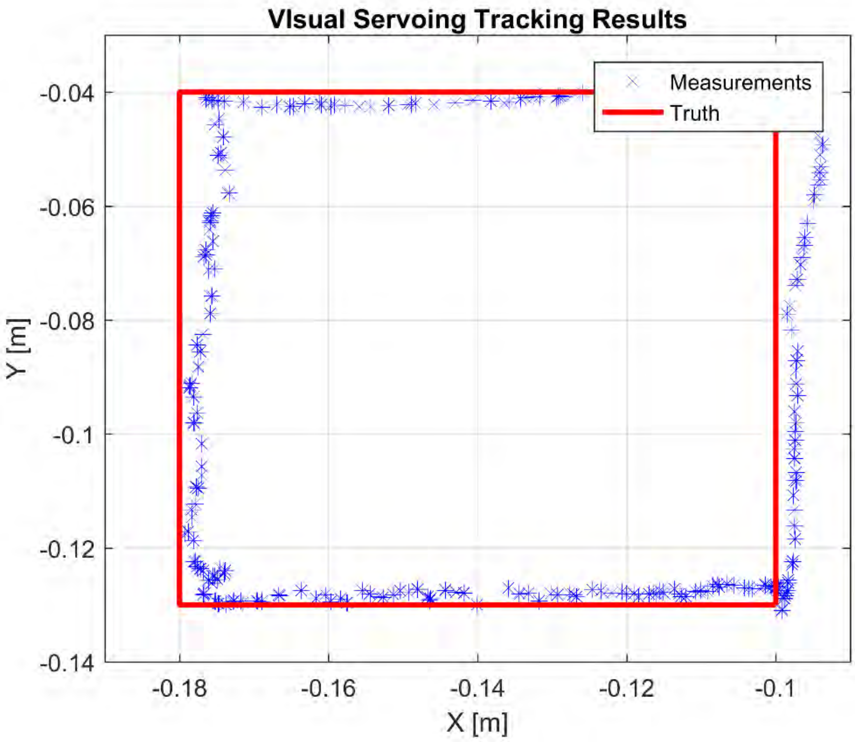


Figure 60: Visual Servoing Motion Tracking

To evaluate the performance of ViSP, the Mean Square Error of the distance between the measurement and truth was computed for the recorded motion according to:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x - x_m)^2 + (y - y_m)^2 \tag{93}$$

The MSE for the trajectory shown was $0.0055\ m$. For proper deployment of the suggested solution, a high-resolution camera needs to be used. Resolutions that are significantly larger than 1280×720, which was the only one available, are recommended to cover a large area. With the resolution used, a usable area of less than $0.5 \times 0.5\ m$ was achievable; an area too small for trajectory tracking and testing of the hardware.

7.3. The Robot Operating System

The Robot Operating System (ROS) is a middleware for robots; that is a collection of software frameworks that aid robot software development. ROS provides the services an operating system usually offers such as “hardware abstraction, low-level device control, implementation of commonly used functionality, message-passing between processes, and package management” [68]. The pieces of code in ROS where all the processing takes place are called nodes. Nodes post and receive messages that can contain anything from sensor data to be processed to control signals to be supplied to actuators. The communication or the messaging pattern between the comprising components of a ROS network is a publish-subscribe pattern, where the nodes that publish a message to a topic trigger the nodes that subscribe to that topic to receive the newly published information. A group of nodes is called a package, and a group of packages are called a stack.

ROS was chosen to exploit the ease with which code can run in a network of connected computers. The computers can be placed onboard the vehicle or on a surface station. The different algorithms were implemented in five main nodes. The first node was a driver node for the pressure sensor, and it publishes the depth of the vehicle from the pressure reading to a topic called */depth*. The second node was a driver node for MIDG IMU, and it publishes the inertial measurements of the sensor to a topic called */imu*. The third node is the estimation node, which implements the sensor fusion algorithm based on the Extended Kalman Filter. This node subscribes to the sensor messages and publishes the state estimate to a topic called */state_estimate*. The fourth node is the adaptive control node. It subscribes to the */state_estimate* topic, it processes the data to compute the control signal as well as update the gains of the controllers, and it published the control signal to a topic called */Thrust*. The final implemented node is an Arduino node. This node subscribes to the */Thrust* topic, and it computes the motor control signal required to realize the controller computed thrust.

7.4. ROS-based Simulation Environment

Here, we describe a developed simulation environment based around the gazebo simulator for which ROS integration and support is already present. The simulation environment should serve as a visual means through which algorithm prototyping and testing is possible. The gazebo simulator by default can handle rigid body motion, and the vehicle is assumed to be a rigid body undergoing general motion. The gazebo simulator can also handle simulating buoyancy. The other hydrodynamic effects are, however, not included by default in the simulator. Therefore, the effects that gazebo does not simulate by default are computed according to the developed equations of motion, and the resulting forces and moments are applied on the vehicle. [69] developed C++ gazebo plugins that give direct access to the quantities of interest. The identified parameters of the vehicle are then fed into the UUV plugin, and the vehicle equations of motion can then be simulated in the gazebo environment. Figure 61 shows the vehicle visualization when spawned in the Gazebo environment.

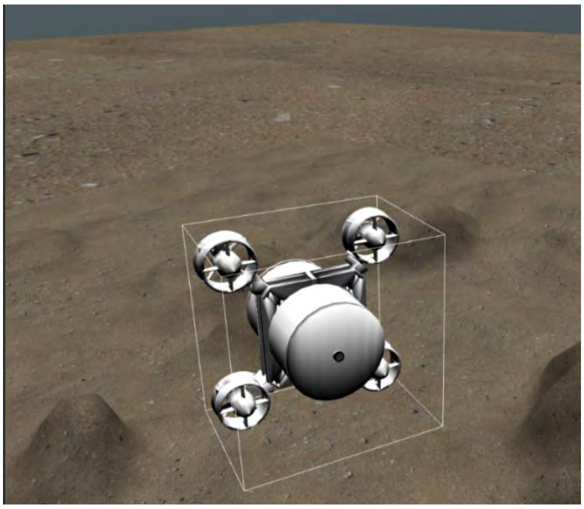


Figure 61: AUV in the Gazebo environment

The used package and the code written give access to a ROS topic called *thruster_manager/input*, which accepts wrench messages that contain the forces and torques to be applied on the vehicle at each instance in time. This enables the development and visual testing of effort-based controllers through Python scripts, C++ scripts, or MATLAB/Simulink links. Figure 62 and Figure 63 show the running nodes and topics necessary for the Gazebo simulation and the ROS integration to run.

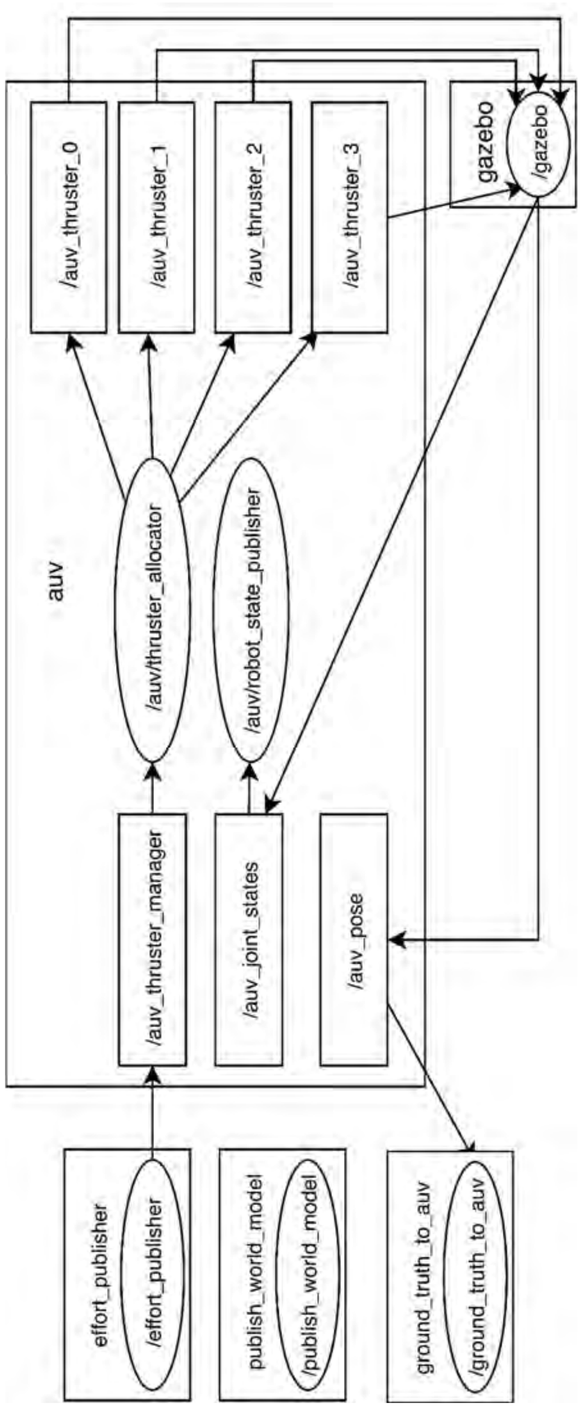


Figure 62: ROS Simulation Environment Running Nodes

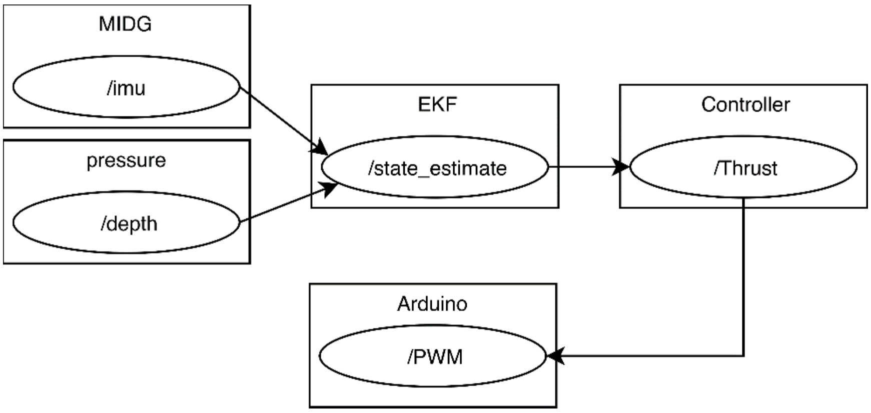


Figure 63: ROS Node Architecture

Chapter 8. Conclusions, Limitations, and Future Work

8.1. Conclusions

This work set out to mathematically model an Autonomous Underwater Vehicle and to estimate the parameters of the vehicle. The devised model accounted for a multitude of effects a marine craft experiences when operating in an underwater environment, and the parameter identification was primarily based on finite element analysis software packages. That allowed the approximation of quantities of the likes of the drag profile that the vehicle experiences as well as the added inertial effects resulting from the coupling between the rigid body of the vehicle and the fluid environment.

The vehicle mathematical model is also exploited in the design of a novel adaptive guidance controller. The designed control law relies on the structural model to account for and then compensate for the nonlinear effects experienced in an underwater setting as well as the nonlinear dynamics of a six degrees of freedom rigid body undergoing general motion. The control law was compared against a classic control law from literature, and the proposed adaptive routine was also compared against a classic approach. The designed controller and adaptation routine proved superior to classical alternatives.

Finally, the navigation problem was touched upon through the design and evaluation of two classical approaches to the navigation problem. Namely, an extended Kalman filter estimation problem in addition to an Unscented Kalman Filter estimation problem were formulated to make use of the structural mathematical model of the vehicle for navigation purposes. The classical approach to solving the navigation problem in literature is to deploy an Inertial Navigation System to localize the vehicle in its working environment.

8.2. Limitations

The primary limitation faced in this work was the unavailability of a localization system that provides position measurements of the vehicle. That prevented the experimental validation of the proposed work.

8.3. Future Work

Here, we mention some possible improvements that can be applied to the proposed work. First, the mathematical model can be further enhanced by considering the effects of water currents, which are of interest in some environments. Assuming that the velocity of the vehicle as well as the flowrate of the ocean current are measurable, the effects of water currents on the dynamics of motion can then be incorporated into the model as

$$\mathbf{M}\,\dot{\boldsymbol{\nu}}_r + \mathbf{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{\mathcal{G}}(\boldsymbol{\eta}) = \boldsymbol{\tau}_{\boldsymbol{actuat}\mathrm{o}r} \tag{94}$$

where $\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu_c}$ and $\boldsymbol{\nu}_c$ is the ocean current velocity.

The added inertia for a body can be estimated using other techniques other than the simplistic approach used here. An approach based on modal analysis is presented in [20]. The vehicle is modeled and imported in a finite element analysis software package, and the modal analysis is run in vacuum in addition to in water. The natural frequencies reported by both approaches can then be exploited to estimate all the elements of the added mass matrix.

Another possible improvement on this work lies in changing the adaptation law. This can be further investigated by substituting the input driven systems by a first order dynamic system in the gain that is driven the error. So, the new dynamic gain adaptation law can be described by the stable dynamics given by

$$\dot{k} + \gamma_1 k = \gamma_2 e^2 \tag{95}$$

where γ_1 is a design parameter that has to be properly selected for proper function of the adaptation law and γ_2 is a positive constant.

The advantage this adaptation law offers is that the gain is allowed to increase and to decrease depending on the error. So, as the error decreases, the gain values will decrease as well. This will eliminate the need for resetting the gain based on the running statistics of the error norm, but it will also need the effect of the γ_1 parameter to be thoroughly investigated.

Extensive experimental testing of the filters and the control algorithms with proper localization equipment is also necessary to validate the proposed work. Also, the benefit that the structural information of the mathematical model of the vehicle provides when it comes to estimating the state can be investigated with the aforementioned equipment.

Finally, it is possible to exploit the structural information that the model provides in fault detection. For example, [70] develops a T^2-statistic-based fault detection and classification algorithm that can determine online any faults that could happen to the control surfaces of the marine craft.

References

[1] R. Button, J. Kamp, T. Curtin, and J. Dryden, "A survey of missions for unmanned undersea vehicles," RAND Corporation, Tech. Report, 2009.

[2] D. Chang, W. Wu, C. R. Edwards, and F. Zhang, "Motion tomography: Mapping flow fields using autonomous underwater vehicles," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 320-336, 2017.

[3] B. Jalving, K. Gade, O. K. Hagen, and K. Vestgard, "A toolbox of aiding techniques for the HUGIN AUV integrated inertial navigation system," in *Oceans 2003. Celebrating the Past ... Teaming Toward the Future (IEEE Cat. No.03CH37492)*, 2003, pp. 1146-1153.

[4] J. Kinsey, R. Eustice, and L. Whitcomb, "A survey of underwater vehicle navigation: Recent advances and new challenges," in *Proceedings of IFAC Conference of Manoeuvering and Control of Marine Craft*, pp. 1-12, 2006.

[5] K. Alam, T. Ray, and S. G. Anavatti, "Design and construction of an autonomous underwater vehicle," *Neurocomputing*, vol. 142, pp. 16-29, 2014.

[6] C. Wang, F. Zhang, and D. Schaefer, "Dynamic modeling of an autonomous underwater vehicle," *Journal of Marine Science and Technology,* vol. 20, no. 2, pp. 199-212, 2015.

[7] T. I. Fossen and O.-E. Fjellstad, "Nonlinear modelling of marine vehicles in 6 degrees of freedom," *Mathematical Modelling of Systems*, vol. 1, no. 1, pp. 17-27, 1995.

[8] P. Ridao, J. Batlle, and M. Carreras, "Dynamics model of an underwater robotic vehicle," University of Girona, Spain, Tech. Report, 2001.

[9] J. N. Newman, *Marine hydrodynamics*. Cambridge, MA: The MIT Press, 2017.

[10] T. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 2011.

[11] Y. H. Eng, M. W. S. Lau, E. Low, G. Seet, and C. S. Chin, "Estimation of the Hydrodynamics Coefficients of an ROV using Free Decay Pendulum Motion," *Engineering Letters*, vol. 16, pp. 326-331, 2008.

[12] G. Conte, S. M. Zanoli, D. Scaradozzi, and A. Conti, "Evaluation of hydrodynamics parameters of a UUV. A preliminary study," in *First International Symposium on Control, Communications and Signal Processing*, 2004, pp. 545-548.

[13] J. Geisbert, "Hydrodynamic Modeling for Autonomous Underwater Vehicles Using Computational and Semi-Empirical Methods," Masters of science, Ocean Engineering, Virginia Tech, Virginia Tech Electronic Theses and Dissertations, 2007.

[14] S. C. Martin and L. L. Whitcomb, "Preliminary results in experimental identification of 3-DOF coupled dynamical plant for underwater vehicles," *OCEANS 2008*, 2008, pp. 1-9.

[15] S. C. Martin and L. L. Whitcomb, "Experimental Identification of Six-Degree-of-Freedom Coupled Dynamic Plant Models for Underwater Robot Vehicles," *IEEE Journal of Oceanic Engineering,* vol. 39, no. 4, pp. 662-671, 2014.

[16] K. Tan, T. Lu, and A. Anvar, "Drag coefficient estimation model to simulate dynamic control of autonomous underwater vehicle (AUV) motion," presented at *20th International Congress on Modelling and Simulation*, Adelaide, South Australia, 2013, pp. 963-969.

[17] S. Fackrell, "Study of the Added Mass of Cylinders and Spheres," Ph.D., Mechanical, Automotive, and Materials Engineering, University of Windsor, 2011.

[18] R. P. S. Han and H. xu, "A simple and accurate added mass model for hydrodynamic fluid—Structure interaction analysis," *Journal of the Franklin Institute,* vol. 333, no. 6, pp. 929-945, 1996.

[19] C. J. McFarland and L. L. Whitcomb, "Comparative experimental evaluation of a new adaptive identifier for underwater vehicles," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 4614-4620.

[20] L. Jezequel L, "Hydrodynamic added-mass identification from resonance tests," *American Institute of Aeronautics and Astronautics Journal,* vol. 21, no. 4, p. 608-613, 1983.

[21] F. White, *Fluid Mechanics*. New Delhi, India: McGraw-Hill Education, 2017.

[22] J. H. Ginsberg, *Advanced Engineering Dynamics*. Harper & Row Limited, 1988.

[23] M. W. Spong, "Underactuated mechanical systems," in *Control Problems in Robotics and Automation*, 1998, pp. 135-150.

[24] W. E. Dixon, *Nonlinear Control of Engineering Systems a Lyapunov-based Approach.* Boston: Birkhäuser, 2003, pp. 269-335.

[25] L. McCue, "Handbook of Marine Craft Hydrodynamics and Motion Control," *IEEE Control Systems,* vol. 36, no. 1, pp. 78-79, 2016.

[26] J. J. Craig, *Introduction to Robotics: Pearson New International Edition: Mechanics and Control*. Pearson Education Limited, 2013.

[27] M. Candeloro, A. M. Lekkas, A. J. Sørensen, and T. I. Fossen, "Continuous Curvature Path Planning using Voronoi diagrams and Fermat's spirals," *IFAC Proceedings Volumes,* vol. 46, no. 33, pp. 132-137, 2013.

[28] T. I. Fossen, K. Y. Pettersen, and R. Galeazzi, "Line-of-Sight Path Following for Dubins Paths With Adaptive Sideslip Compensation of Drift Forces," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820-827, 2015.

[29] A. M. Lekkas and T. I. Fossen, "Integral LOS Path Following for Curved Paths Based on a Monotone Cubic Hermite Spline Parametrization," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2287-2301, 2014.

[30] S. C. Martin and L. L. Whitcomb, "Preliminary experiments in underactuated nonlinear model-based tracking control of underwater vehicles with three degree-of-freedom fully-coupled dynamical plant models: Theory and experimental evaluation," in *2012 Oceans*, 2012, pp. 1-7.

[31] N. P. I. Aneke, "Control of underactuated mechanical systems." Ph.D Thesis, Technische Universiteit Eindhoven, 2003.

[32] R. W. Brockett and H. U. C. MA., *Asymptotic Stability and Feedback Stabilization*. Defense Technical Information Center, 1983.

[33] S. Wadoo and P. Kachroo, *Autonomous Underwater Vehicles: Modeling, Control Design and Simulation*. CRC Press, 2016.

[34] T. I. Fossen and A. M. Lekkas, "Direct and indirect adaptive integral line-of-sight path-following controllers for marine craft exposed to ocean currents," *International Journal of Adaptive Control and Signal Processing*, vol. 31, no. 4, pp. 445-463, 2017.

[35] Z. M. Zain, K. Watanabe, K. Izumi, and I. Nagai, "Exponential Stabilization of Second-Order Nonholonomic Chained Systems," in *Intelligent Robotics and Applications: 6th International Conference, ICIRA 2013, Busan, South Korea, September 25-28, 2013, Proceedings, Part II*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 96-107.

[36] N. P. I. N. Aneke, H.de Jager, A. G., "Tracking control of second-order chained form systems by cascaded backstepping," *International Journal of Robust and Nonlinear Control*, vol. 13, no. 2, pp. 95-115, 2003.

[37] T. I. Fossen, *Marine control systems : guidance, navigation and control of ships, rigs and underwater vehicles*. Trondheim, Norway: Marine Cybernetics, 2002.

[38] J. G. Lim and S. Jung, "Design of a Time-Delayed Controller for Attitude Control of a Quadrotor System," in *Intelligent Robotics and Applications: 6th International Conference, ICIRA 2013*, 2013, pp. 274-280.

[39] M. Massot-Campos and G. Oliver-Codina, "Optical Sensors and Methods for Underwater 3D Reconstruction," *Sensors*, vol. 15, no. 12, pp. 31525-31557, 2015.

[40] J. M. Hansen, J. Roháč, M. Šipoš, T. A. Johansen, T. I. Fossen, and G. Wang, "Validation and Experimental Testing of Observers for Robust GNSS-Aided Inertial Navigation," in *Recent Advances in Robotic Systems.* Rijeka: InTech, 2016, Ch. 06.

[41] J. C. Kinsey, D. A. Smallwood, and L. L. Whitcomb, "A new hydrodynamics test facility for UUV dynamics and control research," in *Oceans 2003. Celebrating the Past ... Teaming Toward the Future*, 2003, pp. 356-361.

[42] J. C. Kinsey and L. L. Whitcomb, "Preliminary field experience with the DVLNAV integrated navigation system for oceanographic submersibles," *Control Engineering Practice,* vol. 12, no. 12, pp. 1541-1549, 2004.

[43] R. McEwen, H. Thomas, D. Weber, and F. Psota, "Performance of an AUV navigation system at Arctic latitudes," *IEEE Journal of Oceanic Engineering,* vol. 30, no. 2, pp. 443-454, 2005.

[44] J. L. Marins, Y. Xiaoping, E. R. Bachmann, R. B. McGhee, and M. J. Zyda, "An extended Kalman filter for quaternion-based orientation estimation using MARG sensors," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium*, 2001, pp. 2003-2011.

[45] E. Kraft, "A quaternion-based unscented Kalman filter for orientation tracking," in *Proceedings of the Sixth International Conference of Information Fusion, 2003*, 2003, pp. 47-54.

[46] B. Vik and T. I. Fossen, "A nonlinear observer for GPS and INS integration," in *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228)*, 2001, pp. 2956-2961.

[47] H. F. Grip, T. I. Fossero, T. A. Johansent, and A. Saberi, "A nonlinear observer for integration of GNSS and IMU measurements with gyro bias estimation," in *2012 American Control Conference (ACC)*, 2012, pp. 4607-4612.

[48] J. M. Hansen, T. I. Fossen, and T. Arne Johansen, "Nonlinear observer design for GNSS-aided inertial navigation systems with time-delayed GNSS measurements," *Control Engineering Practice,* vol. 60, no. Supplement C, pp. 39-50, 2017.

[49] T. A. Johansen, J. M. Hansen, and T. I. Fossen, "Nonlinear Observer for Tightly Integrated Inertial Navigation Aided by Pseudo-Range Measurements," *Journal of Dynamic Systems, Measurement, and Control,* vol. 139, no. 1, pp. 11007-11017, 2016.

[50] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," Tech. Report, 2006.

[51] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[52] C. Chin and M. Lau, "Modeling and testing of hydrodynamic damping model for a complex-shaped remotely-operated vehicle for control," *Journal of Marine Science and Application,* vol. 11, no. 2, pp. 150-163, 2012.

[53] T. I. Fossen, *Guidance and control of ocean vehicles*. Wiley, 1994.

[54] C. Yih, *Fluid Mechanics: A Concise Introduction to the Theory*. West River Press, 1977.

[55] F. Alonge, F. D. Ippolito, and F. M. Raimondi, "Trajectory tracking of underactuated underwater vehicles," in *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228)*, 2001, vol. 5, pp. 4421-4426.

[56] Y. Li and Y. Chen, "When is a Mittag–Leffler function a Nussbaum function?," *Automatica,* vol. 45, no. 8, pp. 1957-1959, 2009.

[57] B. P. C. F. p. d. A. Welford, "Note on a Method for Calculating Corrected Sums of Squares and Products," *Technometrics,* vol. 4, no. 3, pp. 419-420, 1962.

[58] W. Al Masri, "Inertial Navigation System of In-pipe Robot," M.Sc. Thesis, Mechatronics Engineering, American University of Sharjah, 2016.

[59] M. F. Abdel-Hafez, "On the development of an inertial navigation error-budget system," *Journal of the Franklin Institute,* vol. 348, no. 1, pp. 24-44, 2011.

[60] R. Valenti, I. Dryanovski, and J. Xiao, "Keeping a Good Attitude: A Quaternion-Based Orientation Filter for IMUs and MARGs," *Sensors,* vol. 15, no. 8, pp. 19302-19330, 2015.

[61] V. Renaudin, M. H. Afzal, and G. Lachapelle, "New method for magnetometers based orientation estimation," in *IEEE/ION Position, Location and Navigation Symposium*, 2010, pp. 348-356.

[62] M. Abdel-Hafez, I. Rhee, and J. Speyer, "Observability of an integrated GPS/INS during maneuvers - IEEE Journals & Magazine," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 40, no. 2, pp. 526-535, 12 July 2004.

[63] Y. Bar-Shalom and X.-R. Li, *Estimation with Applications to Tracking and Navigation*. Wiley, 2001, p. 584.

[64] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley, 2006.

[65] M. A. Zagrobelny and J. B. Rawlings, "Identifying the uncertainty structure using maximum likelihood estimation," in *2015 American Control Conference (ACC)*, 2015, pp. 422-427.

[66] A. Wadi, M. Abdel-Hafez, and G. A. Husseini, "Identification of the Uncertainty Structure to Estimate the Acoustic Release of Chemotherapeutics

from Polymeric Micelles," *IEEE Transactions on NanoBioscience*, vol. 16, no. 7, pp. 609-617, 2017.

[67] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, vol. 25, no. 11, pp. 120-123, 2000.

[68] O. S. R. Foundation. (2017, June). *The Robot Operation System*. Available: http://www.ros.org

[69] 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, pp. 1-8.

[70] J. H. Kim and G. O. Beale, "Fault Detection and Classification in Underwater Vehicles Using the T2 Statistic," *Automatika: Journal for Control, Measurement, Electronics, Computing and Communications*, vol. 43, no. 1-2, pp. 29-37, 2017.

[71] J. Solà, "Quaternion kinematics for the error-state kalman filter," Cornell University Library, Tech. Report, 2015.

Appendix A: Introduction to Quaternions

This appendix is a brief introduction about quaternions, details the used properties, and presents the rotation matrix formed through quaternions.

If we have two complex numbers $A = a + bi$ and $C = c + di$, then $Q = A + Cj$ with $k \triangleq ij$ yields a number in the space of quaternions denoted $\mathbb{H}$.

$$Q = a + bi + cj + dk \in \mathbb{H} \tag{96}$$

where $\{a, b, c, d\} \in \mathbb{R}$ and $\{i, j, k\}$ are three imaginary unit numbers defined such that

$$i^2 = j^2 = k^2 = ijk = -1, \tag{97}$$

which leads to

$$ij = -ji = k, \qquad jk = -kj = i, \qquad ki = -ik = j. \tag{98}$$

We are interested in unit quaternions for which the real part is η and three imaginary parts form the vector $\boldsymbol{\epsilon} = [\epsilon_1 \quad \epsilon_2 \quad \epsilon_3]^{\boldsymbol{T}}$. The unit quaternion is defined to have a second norm that is equal to 1. Therefore, the set, Q, of quaternions can be defined as:

$$\text{Q} \coloneqq \left\{\boldsymbol{q} \mid \boldsymbol{q}^{\mathrm{T}}\boldsymbol{q} = \mathbf{1}, \qquad \boldsymbol{q} = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix}, \qquad \eta \in \mathbb{R}, \qquad \boldsymbol{\epsilon} \in \mathbb{R}^3, \qquad \boldsymbol{q} \in \mathbb{H}\right\} \tag{99}$$

The norm requirement constrains the degrees of freedom of the quaternion to three. The unit quaternion, like regular complex numbers, can encode rotations which are three-dimensional in the case of quaternions. The quaternion rotation is performed using the quaternion double product defined as

$$x' = \boldsymbol{q} \otimes x \otimes \boldsymbol{q}^* \tag{100}$$

where $\boldsymbol{q}^*$ is the conjugate quaternion defined as $\boldsymbol{q}^* = [\eta \quad -\boldsymbol{\epsilon}^T]^T$ and the product of two quaternions is a quaternion given by the Hamiltonian product:

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \begin{bmatrix} \eta_1\eta_2 - \boldsymbol{\epsilon}_1^T\boldsymbol{\epsilon}_2 \\ \eta_2\boldsymbol{\epsilon}_1 + \eta_1\boldsymbol{\epsilon}_2 + \boldsymbol{\epsilon}_1 \times \boldsymbol{\epsilon}_2 \end{bmatrix} \tag{101}$$

It is of interest to note that the rotation achieved through the quaternion double product is factorable into a traditional rotation matrix that multiply the vector to be rotated. The quaternion rotation matrix mapping the quantities measured in the body-fixed frame to the n-frame is given by:

$$\mathbf{R}_b^n(\boldsymbol{q}) \coloneqq \mathbf{R}_{\eta,\ \boldsymbol{\epsilon}} = \mathbf{I}_{3\text{x}3} + 2\eta\ \mathbf{S}(\boldsymbol{\epsilon}) + 2\ \mathbf{S}^2(\boldsymbol{\epsilon})$$

$$\mathbf{R}_b^n(\boldsymbol{q}) = \begin{bmatrix} 1-2(\epsilon_2^2+\epsilon_3^2) & 2(\epsilon_1\epsilon_2-\epsilon_3\eta) & 2(\epsilon_1\epsilon_3+\epsilon_2\eta) \\ 2(\epsilon_1\epsilon_2+\epsilon_3\eta) & 1-2(\epsilon_1^2+\epsilon_3^2) & 2(\epsilon_2\epsilon_3-\epsilon_1\eta) \\ 2(\epsilon_1\epsilon_3-\epsilon_2\eta) & 2(\epsilon_2\epsilon_3+\epsilon_1\eta) & 1-2(\epsilon_1^2+\epsilon_2^2) \end{bmatrix} \tag{102}$$

The rate of change of the rotation matrix $\mathbf{R}_b^n(\boldsymbol{q})$ is given by

$$\dot{\mathbf{R}}_b^n(\boldsymbol{q}) = \mathbf{R}_b^n(\boldsymbol{q})\ \mathbf{S}(\boldsymbol{\omega}_b) \tag{103}$$

where $\mathbf{S}(\boldsymbol{\omega}_b)$ is the skew-symmetric matrix formed from the angular velocity components and $\mathbf{R}_b^n(\boldsymbol{q})$ is the previously detailed quaternion rotation matrix.

Equating the diagonal components of $\dot{\mathbf{R}}_b^n(\boldsymbol{q})$ to their corresponding components from $\mathbf{R}_b^n(\boldsymbol{q})\ \mathbf{S}(\boldsymbol{\omega}_b)$ and differentiating the unit quaternion constraint $\boldsymbol{q}^{\mathrm{T}}\boldsymbol{q} = \mathbf{1}$ yields

$$\eta\dot{\eta} + \epsilon_1\dot{\epsilon}_1 + \epsilon_2\dot{\epsilon}_2 + \epsilon_3\dot{\epsilon}_3 = 0 \tag{104}$$

which will give to the relation between the quaternion rate of change as a function of the quaternion value and the angular velocity of the body.

The reader is referred to [71] for an extensive discussion of quaternions.

Appendix B: Discretization Routine

An Euler integration was used to discretize the slow dynamics of the AUV. When the dynamics are characterized by higher velocities, however, the simple Euler-based approach will eventually diverge at the selected sampling period. Another discretization can be performed through:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{\eta}} \\ \dot{\boldsymbol{\nu}} \end{bmatrix} = \boldsymbol{\zeta}(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} \mathbf{J}_{\boldsymbol{q}}(\boldsymbol{q})\,\boldsymbol{\nu} \\ \boldsymbol{M}^{-1}\boldsymbol{f}(\boldsymbol{x}, t) \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{6x1} \\ \boldsymbol{M}^{-1}\boldsymbol{u}(t) \end{bmatrix}$$

$$\ddot{\boldsymbol{x}} = \frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} + \frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{u}}\dot{\boldsymbol{u}} \tag{105}$$

It is a fair assumption to consider the input to be constant as a result of the zero-order hold in a digital-to-analog converter. Consequently, the rate of change in input is zero, $\dot{\boldsymbol{u}} = 0$. That reduces the nonlinear continuous equation into the following homogeneous form, which utilizes the Jacobian, $\xi(\boldsymbol{x})$.

$$\ddot{\boldsymbol{x}} = \begin{bmatrix} \frac{\partial \zeta_1}{\partial x_1} & \frac{\partial \zeta_1}{\partial x_2} & \cdots & \frac{\partial \zeta_1}{\partial x_n} \\ \frac{\partial \zeta_2}{\partial x_1} & \frac{\partial \zeta_2}{\partial x_2} & \cdots & \frac{\partial \zeta_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \zeta_n}{\partial x_1} & \frac{\partial \zeta_n}{\partial x_2} & \cdots & \frac{\partial \zeta_n}{\partial x_n} \end{bmatrix} \dot{\boldsymbol{x}} = \xi(\boldsymbol{x})\dot{\boldsymbol{x}} \tag{106}$$

The solution of the homogeneous equation is given by:

$$\dot{\boldsymbol{x}} = e^{\xi(x_0)(t-t_0)}\dot{\boldsymbol{x}}(0) \tag{107}$$

And the discrete form is given by the forward Euler integration:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \int_0^h e^{\xi(x_k)(\tau)}\,\dot{\boldsymbol{x}}_k\,d\tau$$

$$= \boldsymbol{x}_k + \left(e^{\xi(x_k)\Delta t}\dot{\boldsymbol{x}}_k\right)\Delta t \tag{108}$$

This approach was found to provide numerical error that is in the same order of magnitude as a Runge-Kutta-based integration on the original dynamics equation, which requires more evaluations of the dynamics function than the proposed discretization.

Appendix C: Dynamics Equation Jacobian

Columns 1 → 3 of the Jacobian of the dynamics model is:

$$\left[\begin{array}{ccc}
-\dfrac{22.56*(\mathrm{abs}(\mathrm{u})+\ \mathrm{u}*\mathrm{sign}(\mathrm{u})-\ 0.0004)}{\mathrm{m}} & r & -q \\
-r & -\dfrac{25.8576*(\mathrm{abs}(\mathrm{v})+\ \mathrm{v}*\mathrm{sign}(\mathrm{v})+\ 0.0168)}{\mathrm{m}} & p \\
q & -p & -\dfrac{25.8576*(abs(w)+\ w*sign(w)+\ 0.0168)}{m} \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
-2*e2^2-2*e3^2+1 & 2*e1*e2-2*e3*n & 2*e1*e3+2*e2*n \\
2*e1*e2+2*e3*n & -2*e1\hat{}2-2*e3\hat{}2+1 & 2*e2*e3-2*e1*n \\
2*e1*e3-2*e2*n & 2*e2*e3+2*e1*n & -2*e1\hat{}2-2*e2\hat{}2+1 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{array}\right.$$

Columns 4 → 6 of the Jacobian of the dynamics model is:

$$\begin{bmatrix}
0 & -w & v \\
w & 0 & -u \\
-v & u & 0 \\
-\frac{0.1533 * \left(abs(p) + p * sign(p) + 0.3525\right)}{Ixx} & \frac{Iyy - Izz}{Ixx} * r & \frac{Iyy - Izz}{Ixx} * q \\
-\frac{Ixx - Izz}{Iyy} * r & -\frac{0.0471 * \left(abs(q) + q * sign(q) - 0.465\right)}{Iyy} & -\frac{Ixx - Izz}{Iyy} * p \\
\frac{Ixx - Iyy}{Izz} * q & \frac{Ixx - Iyy}{Izz} * p & -\frac{0.0471 * \left(abs(r) + r * sign(r) - 0.4647\right)}{Izz} \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
-0.5 * e1 & -0.5 * e2 & -0.5 * e3 \\
0.5 * n & -0.5 * e3 & 0.5 * e2 \\
0.5 * e3 & 0.5 * n & -0.5 * e1 \\
-0.5 * e2 & 0.5 * e1 & 0.5 * n
\end{bmatrix}$$

Columns 7 → 13 of the Jacobian of the dynamics model is:

$$\left.\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\frac{0.001322 * e1}{Iyy} & -\frac{0.001322 * e2}{Iyy} & 0 \\
0 & 0 & 0 & -\frac{0.000661 * e1}{Izz} & -\frac{0.000661 * n}{Izz} & -\frac{0.000661 * e3}{Izz} & -\frac{0.000661 * e2}{Izz} \\
0 & 0 & 0 & \begin{pmatrix} 2 * e2 * w - \\ 2 * e3 * v \end{pmatrix} & \begin{pmatrix} 2 * e2 * v + \\ 2 * e3 * w \end{pmatrix} & \begin{pmatrix} 2 * e1 * v - \\ 4 * e2 * u + \\ 2 * n * w \end{pmatrix} & \begin{pmatrix} 2 * e1 * w - \\ 4.0 * e3 * u - \\ 2 * n * v \end{pmatrix} \\
0 & 0 & 0 & \begin{pmatrix} 2 * e3 * u \\ - 2 * e1 * w \end{pmatrix} & \begin{pmatrix} 2 * e2 * u - \\ 4.0 * e1 * v - \\ 2 * n * w \end{pmatrix} & \begin{pmatrix} 2 * e1 * u + \\ 2 * e3 * w \end{pmatrix} & \begin{pmatrix} 2 * e2 * w \\ - 4.0 * e3 * v \\ + 2 * n * u \end{pmatrix} \\
0 & 0 & 0 & \begin{pmatrix} 2 * e1 * v - \\ 2 * e2 * u \end{pmatrix} & \begin{pmatrix} 2 * e3 * u - \\ 4.0 * e1 * w + \\ 2 * n * v \end{pmatrix} & \begin{pmatrix} 2 * e3 * v - \\ 4.0 * e2 * w - \\ 2 * n * u \end{pmatrix} & \begin{pmatrix} 2 * e1 * u + \\ 2 * e2 * v \end{pmatrix} \\
0 & 0 & 0 & 0 & -0.5 * p & -0.5 * q & -0.5 * r \\
0 & 0 & 0 & 0.5 * p & 0 & 0.5 * r & 0.5 * q \\
0 & 0 & 0 & 0.5 * q & -0.5 * r & 0 & 0.5 * p \\
0 & 0 & 0 & 0.5 * r & 0.5 * q & 0.5 * q & 0
\end{matrix}\right]$$

Appendix D: Computer Codes

This appendix links to the GitHub repository where python scripts that can be used to proof test most of the functionality presented in this work are permanently stored. The python module depends on the following packages:

1. Math
2. Numpy
3. Scipy
4. Matplotlib

The ROS functionality depends on:

1. uuv_simulator
2. ViSP

The relevant GitHub repositories can be found at: https://github.com/ali94wadi

Vita

Ali Wadi was born in 1994, in Gaza, Palestine. He received his primary education and secondary education in Khan Younis, Palestine, and Abu Dhabi, UAE, respectively. He received his B.Sc. degree in Mechanical Engineering from the American University of Sharjah in 2015. Throughout 2015, he worked as an Inspection Engineer in Khatib & Alami, Abu Dhabi.

In January 2016, he joined the Mechanical Engineering master's program in the American University of Sharjah as a graduate teaching assistant. During his master's study, he co-authored two papers which were published in international journals and co-authored two papers which appeared in international conferences. His research interests include Robotics, Autonomous Vehicles, Chemotherapeutic Drug Delivery Systems, Dynamics and Control, and Estimation Theory in a multidisciplinary setting.