

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Marine Technology

Christopher Anthony Janjua

Altitude Control System for Acquiring Optimal Underwater Hyperspectral Imagery Data using an Unmanned Underwater Vehicle

Master's thesis in Marine Technology

Supervisor: Martin Ludvigsen

Co-supervisor: Markus Fossdal

February 2023



Norwegian University of
Science and Technology

Christopher Anthony Janjua

Altitude Control System for Acquiring Optimal Underwater Hyperspectral Imagery Data using an Unmanned Underwater Vehicle

Master's thesis in Marine Technology
Supervisor: Martin Ludvigsen
Co-supervisor: Markus Fossdal
February 2023

Norwegian University of Science and Technology
Faculty of Engineering
Department of Marine Technology



Norwegian University of
Science and Technology



NTNU Trondheim
Norwegian University of Science and Technology
Department of Marine Technology

MASTER THESIS IN MARINE CYBERNETICS

2022

FOR

Christopher A. Janjua

Title: Altitude control system for acquiring optimal underwater hyperspectral imagery data using an unmanned underwater vehicle

Work description

Over the last decade the maritime industry has sought to slowly convert much of its manual labor to becoming automated. As operational costs rise, stricter regulations are enforced and construction becomes more costly, efficiency becomes much more important. Mapping and exploration of the ocean is a task that is currently being researched to be fully automated.

This thesis will look into developing an altitude control system that can be deployed on an underwater snake robot. The system's main goal is to accurately position an underwater vehicle at a given distance, set by an operator, from the seafloor. This to acquire high quality underwater hyperspectral imagery in an optimal manner, hereunder the trade-off between coverage area and spatial resolution. This thesis will confine itself to simulating an underwater snake robot using the Robot Operating System II (ROS2) in an underwater environment, Plankton.

Scope of Work

1. Investigate the operational limitations and operation of UHI on snake robots.
 2. Gain an overview through a literature study on underwater snake robots. Further reading upon mathematical modeling and control systems for underwater vehicles. Present prior work and development surrounding underwater snake robots. Use of UHI in regard to ocean exploration and mapping.
 3. Propose a control system that can be applied to a underwater snake robot, giving guidance making it able to follow the curvature of the surface in favor.
 4. Create a simulation of the proposed control system based on Plankton in Gazebo using ROS2 software.
-

Specifications

The student shall at startup provide a maximum 2-page week plan of work for the entire project period, with main activities and milestones. This should be updated on a monthly basis in agreement with supervisor. Every weekend throughout the project period, the candidate shall send a status email to the supervisor and co-advisors, providing two brief bulleted lists: 1) work done recent week, and 2) work planned to be done next week.

The scope of work may prove to be larger than initially anticipated. By the approval from the supervisor, described topics may be deleted or reduced in extent without consequences with regard to grading. The candidate shall present personal contribution to the resolution of problems within the scope of work. Theories and conclusions should be based on mathematical derivations and logic reasoning identifying the steps in the deduction.

The report shall be organized in a logical structure to give a clear exposition of background, problem/research statement, design/method, analysis, and results. The text should be brief and to the point, with a clear language. Rigorous mathematical deductions and illustrating figures are preferred over lengthy textual descriptions. The report shall have font size 11 pts., and it is not expected to be longer than 70 A4-pages, 100 B5-pages, from introduction to conclusion, unless otherwise agreed. It shall be written in English (preferably US) and contain the elements: Title page, project definition, preface (incl. description of help, resources, and internal and external factors that have affected the project process), acknowledgement, abstract, list of symbols and acronyms, table of contents, introduction (project background/motivation, objectives, scope and delimitations, and contributions), technical background and literature review, problem formulation or research question(s), method/design/development, results and analysis, conclusions with recommendations for further work, references, and optional appendices. Figures, tables, and equations shall be numerated. The contribution of the candidate shall be clearly and explicitly described, and material taken from other sources shall be clearly identified. Work from other sources shall be properly acknowledged using quotations and a Harvard citation style (e.g. natbib Latex package). The work is expected to be conducted in an honest and ethical manner, without any sort of plagiarism and misconduct, which is taken very seriously by the university and will result in consequences. NTNU can use the results freely in research and teaching by proper referencing, unless otherwise agreed.

The thesis shall be submitted with an electronic copy to the main supervisor and department according to NTNU administrative procedures. The final revised version of this thesis definition shall be included after the title page. Computer code, pictures, videos, data, etc., shall be included electronically with the report.

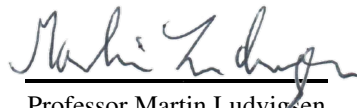
Start Date: 22.08.2022

Due Date: 09.02.2023

Supervisor: Martin Ludvigsen

Co-advisors: Asgeir Sørensen & Markus Fossdal

Trondheim 30.01.2023

A handwritten signature in black ink, appearing to read 'Martin Ludvigsen', written over a horizontal line.

Professor Martin Ludvigsen



NTNU Trondheim
Norwegian University of Science and Technology
Department of Marine Technology

Project Plan

Task / Month, Week	Aug 34,35	Sept 36- 39	Oct 40- 44	Nov 45- 48	Dec 49- 52	Jan 1-5	Feb 6
Investigate the operational limitations and operation of UHI on snake robots.	X						
Gain an overview through a literature study on Underwater Snake Robots. Further reading upon mathematical modeling and control systems for Underwater Vehicles	X	X					
Present prior work and development surrounding underwater snake robots. Use of UHI in regard to ocean exploration and mapping	X	X					
Propose a control system that is able to position underwater snake robots body orthogonal relative to the surface in favor		X	X	X			
Create a simulation of the proposed control system based on Plankton in Gazebo using ROS2 software			X	X	X		
Report Completion						X	X

Abstract

Developing autonomous underwater robot systems that can effectively map the deep sea is a significant challenge. These systems must be able to navigate and maneuver in an environment that is dark, murky, and often hostile, with varying water conditions and terrain that can be difficult to traverse. Despite these challenges, advances in technology, such as the development of Underwater Snake Robots, may open up the possibility to explore and map deeper regions of the ocean than ever before.

This thesis presents the development of an altitude control system, which can be applied to any underwater robot system. The system utilizes a stereo depth camera and a Doppler velocity log to estimate altitude, enabling to maintain a desired altitude relative to the seafloor. The main goal of this system is to favorably position a hyperspectral camera, which is attached to the mainframe of the robot, to capture high-quality data of substances on the seafloor. The system was simulated using the Plankton Simulator, an open-source tool that uses the Gazebo Simulator and the UUV Simulator plugin to recreate a realistic underwater environment for testing. The simulation was built using the Robot Operating System 2 (ROS2) framework.

The system was tested in two sets of simulation cases, first with a traditional ROV and then with a Underwater Snake Robot (USR). The first simulation case involved a chosen mountain section with varying inclines, and the second case was used to create a baseline for comparison. Both robots had their movement in surge manually input by an operator, while the position in altitude was controlled by the automatic system. Between the two sets of cases, improvements were made to the controller and observer, and the simulations were run again.

One of the main challenges faced during the development of the system was the elongated body of the USR, which made it difficult to maintain a stable altitude and resulted in increased oscillations when performing rough contouring. To address this issue, a control algorithm was implemented to minimize these oscillations and improve the stability of the USR. Another issue that arose during development was the difficulty of maintaining a constant altitude with the hyperspectral camera while traversing the irregular seafloor. To address this, a doppler velocity log was used to estimate altitude and maintain a desired altitude relative to the seafloor.

While the current altitude control system is limited to a simulated environment, it has the potential to be a viable prototype for handling course contouring of the seafloor. It serves as a promising baseline for future development in this area and can be used as a stepping stone for ocean mapping and exploration using an USR with a hyperspectral camera.

Sammendrag

Utviklingen av autonome undervannsrobot-systemer for effektiv havkartlegging utgjør en betydelig utfordring. Disse systemene må være i stand til å navigere og manøvrere i et miljø som er mørkt, dystert, og ofte fiendtlig, med varierende vannforhold og et terreng som kan være vanskelig å bevege seg gjennom. Likevel kan teknologiske fremskritt, som utviklingen av Undervannsslange-roboter, gi muligheter til å utforske og kartlegge dypere områder av havet enn noen gang før.

Denne avhandlingen presenterer utviklingen av et høyde-kontrollsystem, som kan brukes av enhver undervannsrobot. Systemet bruker et stereokamera og en Doppler hastighetslogg til å estimere høyden, slik at det kan opprettholde en ønsket høyde i forhold til havbunnen. Hovedmålet med systemet er å plassere et hyperspektralt kamera, som er festet til robotens hovedramme, på en hensiktsmessig måte for å innhente høykvalitetsdata av havbunnsmaterialer. Systemet ble simulert ved hjelp av Plankton Simulator, et åpen kildeverktøy som bruker Gazebo Simulator og UUV Simulator-plugin for å gjenskape et realistisk undervannsmiljø for testing. Simuleringen ble bygget ved hjelp av Robot Operating System 2 (ROS2) rammeverk.

Systemet ble testet i to serier med simuleringer, først med en tradisjonell ROV, og deretter med en Undervanns Slangrobot (USR). Den første simuleringen involverte en valgt seksjon av et fjell med varierende hellinger, mens den andre simuleringen ble brukt som sammenligningsgrunnlag. Begge robotene hadde manuell bevegelse kontrollert av en operatør, mens høyden ble regulert av det automatiske systemet. Mellom de to simuleringene ble kontrolleren og observatøren forbedret, og simuleringene ble kjørt igjen.

En av de største utfordringene under utviklingen av systemet var den forlengede kroppen til USR, som gjorde det vanskelig å opprettholde en høyde uten at systemet begynte å oscillere. For å løse dette problemet ble en kontrollalgoritme implementert for å redusere svingningene og forbedre USR'ens stabilitet. En annen utfordring under utviklingen var å opprettholde en konstant høyde med hyperspektralkameraet mens man traverserte den ujevne havbunnen. For å håndtere dette, ble en Doppler hastighetslogg lagt til for å estimere høyden i forhold til havbunnen.

Det nåværende høydekontrollsystemet, selv om det kun er testet i simuleringer, har muligheten til å bli en fungerende skisse på hvordan roboten kan følge havbunnen. Det er et lovende utgangspunkt for videre utvikling, og kan tjene som et mellomledd for havkartlegging og oppdagelse ved hjelp av USR utstyrt med et hyperspektralkamera.

Preface

This thesis completes the work requirement for course TMR4930. The work makes the author eligible of the degree Master of Science within Marine Technology, with a specialization towards Marine Cybernetics at the Norwegian University of Science and Technology (NTNU). The work presented began in August 2022, and was accomplished in December 2022. This was done independently and was supervised by Professor Martin Ludvigsen with co-supervisors Professor Asgeir Sørensen and Doctoral candidate Markus Fossdal.

The thesis elevates and continues the pre-work done in the course TMR4510 from August to December 2021, "*Development of an underwater hyperspectral imaging payload for AUV in ROS2 Plankton simulator*". Several portions in chapter one, two and three have been repurposed and used. The work in the previous project involved the use of ROS2 and the Plankton simulator accompanied by the use of visual sensors in an underwater environment. The development within this thesis used parts of the code from the previous project, specifically the UHI sensor. The open source unmanned underwater simulator (UUV) package was used with added alterations to the plugins. Also, code from an existing model of the underwater snake robot Eelume was implemented. The problem description was formulated by Martin Ludvigsen, and later modified by the author as the project had its scope and objectives changed over the course of time. In the later stages of development there was collaboration with a colleague, Sayed Emam Ibrahim, to re-factorize the project to a production standard, adding an extension of desired state parameters given by a set of configuration files.

Acknowledgements

I want to show my appreciation towards the people that helped with this thesis. Firstly, I would like to thank my supervisor, Professor Martin Ludvigsen, for providing me with great council and advice. His expertise and knowledge on the many subjects involved has been a great asset. I would also thank my co-supervisor Markus Fossdal, for all his help during the course of the semester. Answering the many questions that I had, and providing me with excellent feedback and tips on the process of writing this thesis. Their enthusiasm and engagement towards this project boosted moral and motivation immensely to apply myself to this thesis.

A special thank you to Ambjørn Waldum, for all his help and for going out of his way to help solve the many technical issues I encountered when developing the simulation. His positive attitude towards every problem motivated me greatly to work hard. Thank you to Einar Kristoffersen for aiding in figuring out infrastructure issues surrounding the ROS2 framework. Thank you also to all of my friends who I have shared an office with during this time.

Trondheim, 09/02-2023

Christopher A. Janjua

Table of Contents

Abstract	i
Sammendrag	ii
Preface	iii
Acknowledgements	iv
Table of Contents	vii
List of Tables	ix
List of Figures	xii
Abbreviations	xiii
1 Introduction	1
1.1 Background and Motivation	1
1.1.1 Autonomous Underwater Robots	2
1.1.2 Visual Mapping of the Ocean Floor	4
1.2 Research Questions and Objectives	6
1.3 Main Contributions	6
1.4 Research Method	6
1.5 Thesis Outline	7
2 Kinetics and kinematics	9
2.1 Kinematics	10
2.1.1 Generalized coordinates	10
2.1.2 Reference frames	11
2.1.3 Transformations	12
2.2 Kinematics of a manipulator	15
2.2.1 Homogenous transformations	15

2.2.2	Forward Kinematics	17
2.3	Kinetics	21
2.3.1	Fossen’s Robot-Inspired Model for Marine Craft	22
2.3.2	Rigid-body Dynamics	23
2.3.3	Control Plant Model	23
2.3.4	Thrust Vector	24
2.3.5	Thrust Allocation	25
2.4	Hydrodynamics	26
2.4.1	Added Mass	26
2.4.2	Damping	26
2.4.3	Restoring- and Hydrostatic-forces	27
2.4.4	Coriolis and Centripetal forces	28
2.4.5	Current	28
3	Guidance, navigation & control	31
3.1	Motion control systems	31
3.2	Sensors	33
3.2.1	DVL	34
3.2.2	Pressure Gauge	34
3.2.3	Stereo Camera	34
3.2.4	UHI	34
3.3	Setup of Sensors on Platform	38
4	Development of a Motion Control for Constant Altitude	41
4.1	Estimating altitude	41
4.1.1	Altitude Approximation by use of a DVL	41
4.1.2	Transformation of DVL beams	44
4.1.3	Altitude approximation by use of a stereo depth camera	45
4.2	System Modules	47
4.2.1	Signal Processing	47
4.2.2	Observer	48
4.2.3	Controller	52
4.2.4	Thruster Allocation	53
4.3	Development Software	53
4.3.1	ROS2	53
4.3.2	Gazebo	54
4.3.3	Plankton	55
4.4	Sensor Carrying Platform Model	55
4.4.1	RexROV	55
4.4.2	Eelume / EELY500	56
4.5	Simulation Cases	57
4.5.1	Case A1 - Altitude Proportional Control RexROV	59
4.5.2	Case A2 - Altitude Proportional Control EELY500	62
4.5.3	Case B1 - Altitude PID Control RexROV	65
4.5.4	Case B2 - Altitude PID Control EELY500	68

5	Discussion	71
6	Conclusion and suggested further development	75
6.1	Conclusion	75
6.2	Suggestions for further development	76
	Bibliography	79
	Appendix	83
A	Simplifications of Hydrodynamics for a Snake Robot with Locked Joints .	83
A.1	Equations for Added mass	83
A.2	Drag forces interacting with a submerged Cylinder	83
A.3	Linear and Quadratic Damping	83
B	USR EELY500	84
B.1	Inspection Module	84
B.2	Ballast Module	84
B.3	Battery Module	84
B.4	Sensor Module	84
B.5	Joint Module	85
B.6	Thruster Module	85

List of Tables

2.1	The notation of SNAME (1950) for marine vessels	10
2.2	Resulting rotation matrix of the kinematic diagram 2.7	20
2.3	Description of the parameters of this sections presented matrices	21
3.1	Overview of current UHI systems for data acquisition including their seafloor mapping capabilities with tradeoffs, [21].	37
4.1	Simulation Case A - Parameters	57
4.2	Simulation Case B - Parameters	57
6.1	Equations for finding added mass elements.	83

List of Figures

1.1	The Royal Navy ROV (Cutlet). Developed and introduced in 1950s to retrieve undetonated torpedoes and mines, [5]	3
1.2	Advantages of using Snake Robots presented by Eelume,[9]	4
2.1	The six-DOF motions in a body-fixed coordinate frame. [35]	10
2.2	Snake Robot modeled as a manipulator arm constricted to links connected to revolute joints, [13].	15
2.3	Homogenous transform between $\{b\}$ and a global frame such as $\{n\}$. . .	16
2.4	Example for homogenous transformation of a rover robot, in four different poses	17
2.5	Actuation of joints between their adjacent links.	18
2.6	Sequence of rotations, from [33]	19
2.7	Kinematic diagram of a manipulator arm	20
3.1	Guidance, Navigation and Control blocks presented in a Motion Control System, where the guidance system makes use of the measured position and velocity for creating a desired state reference model. Referred to as a closed-loop guidance system, [10].	32
3.2	Simplified GNC system for an autopilot. In its most basic form, it consists of a reference model (Guidance system), an observer/gyro-compass (Navigation system) and an autopilot (Control system), [10].	33
3.3	A display of sensors used on UUVs	33
3.4	Amount of information captured when comparing traditional RGB images to Hyperspectral image cube when inspecting a heterogeneous seafloor, [21]	35
3.5	Illustration of the sensor placements onboard the RexROV	38
3.6	Illustration of the sensor placements onboard the Eelume	39
4.1	DVL beam layout. In red, showing the linear approximation of the distance between the sensor bearing platform and the seafloor.	42
4.2	DVL beam vector j^{th} components illustration (left). DVL vector shown in body frame (right).	44

4.3	Illustration for depth estimation using parallel stereo camera system . . .	45
4.4	Illustration of the architecture for the proposed altitude control system . .	47
4.5	Illustration of the Elongated body problem with suggested method of solution	52
4.6	RexROV model details	55
4.7	RexROV model details	56
4.8	The topography of the seafloor captured by the RexROV and EELY500. .	58
4.9	Results of simulation case A1 using a simple proportional controller with the RexROV. Showing the motion control system's altitude variations (left), and the commanded velocities from the system.	60
4.10	Results of simulation case A1 showing produced thrust from the craft. . .	61
4.11	Results of simulation case A2 using a simple proportional controller with the EELY500. Showing the motion control system's altitude variations (left), and the commanded velocities from the system.	63
4.12	Results of simulation case A2 showing produced thrust from the craft. . .	64
4.13	Results of simulation case B using a simple proportional controller with the RexROV. Showing the motion control system's altitude variations (left), and the commanded velocities from the system.	66
4.14	Results of simulation case B1, showing produced thrust from the craft. . .	67
4.15	Results of simulation case B using a simple proportional controller with the RexROV. Showing the motion control system's altitude variations (left), and the commanded velocities from the system.	69
4.16	Results of simulation case B2 showing produced thrust from the craft. . .	70

Abbreviations and Symbols

Abbreviations

AUV	Autonomous Underwater Vehicle
DOF	Degrees of Freedom
COG	Center of Gravity
COB	Center of Buoyancy
CO	Center of Origin
DVL	Doppler Velocity Log
EOM	Equations of motion
GNC	Guidance, Navigation & Control
HI	Hyperspectral Imager
UHI	Underwater Hyperspectral Imager
ROV	Remotely Operated Vehicle
ROS	Robot Operating System

RGB	Red, Green and Blue
USR	Underwater Snake Robot
UUV	Unmanned Underwater Vehicle
API	Application Programming Interface
PID	Proportional Integral Derivative

Symbols

η	Position and orientation of the BODY frame within the Inertial frame
ν	Velocity vector of the BODY frame
ρ	Fluid density
τ	Forces and moments of the Body
B	Bouyancy
\mathbf{T}	Thrust configuration matrix
\mathbf{b}	bias matrix for unmodelled dynamics and slowly varying loads
\mathbf{C}	Rigid-Body Coriolis and Centripetal matrix
\mathbf{D}	Linear damping matrix
\mathbf{g}	Restoring
\mathbf{I}	Inertia matrix
m	Rigid-Body mass
\mathbf{M}	Rigid-body mass matrix
\mathbf{R}	Rotation matrix
\mathbf{q}	Quarternion

Introduction

The following chapter presents background and motivation, as well as thesis research questions. The method will be presented as well as the overall structure.

1.1 Background and Motivation

Ocean discovery has seen a growing interest over the past years. As new technology is introduced, opportunities arise as a result thus enabling new usage aspects and potential industries. The ocean remains today to only have an estimated twenty percent of its sub-surface area as explored and mapped [22]. Exploration has mostly been conducted by ships on the surface. By being confined to only operate at the surface level, the mapping capabilities have been limited to only being able to explore shallow areas of the shoreline. Exploration of the deep sea has been limited and dependent on using Remotely Operated Vehicles (ROVs) to reach the depths required for mapping. Operations using ROVs have been found to be costly and resource inefficient, and this has led to a surge in development towards Autonomous Underwater Vehicles (AUVs). Removing human intervention however opens up to new problems that need to be addressed before AUVs can reach their potential. The seafloor is known to be rough and difficult to maneuver. In a study conducted by [18], an AUV with 4 degrees of freedom (surge, sway, heave, and yaw) was used to search for a sunken vessel from World War II. The terrain of the search area included steep inclines that were challenging to navigate due to the limited maneuverability of the AUV. As a result, the researchers had to exclude certain areas from the search as they were deemed too difficult to map. If the vessel had been resting on a shelf in one of these areas, it would likely have gone undetected. Using a 6-DOF robot would have allowed for additional pitch and roll maneuvers and would have made it possible to map steep inclines, increasing the chances of finding the sunken vessel. Luckily for the researchers they found the ship on a flat area.

Snake robots represents a next step in AUV development, as the body is slender and capable of altering its body configuration, giving it the advantage of doing various ma-

neuers that other underwater vehicles cannot. This gives it an edge in accessibility and maneuvering when traversing the sea bottom. Attaching sensors such as an Underwater Hyperspectral Imager (UHI), has the potential to be able to map more than what the human eye can capture, [7]. The UHI can help to categorize metals, minerals and marine fauna of interest which would otherwise be difficult to identify from a dark murky bottom. As an instrument, it has seen a rise in popularity due to it being so efficient in locating valuable minerals and metals. This could lead to question what is needed to begin deep sea mining?

Deep sea mining has yet to become a fully fledged industry, due to it not being internationally permitted as of yet. Regulations to proceed with mining in international waters has yet to be reviewed, lack of studies and knowledge in what potential harm such an industry can inflict on the marine life is at the core to why this has not been permitted. [3]. Large companies such as BMW and Volvo have stated they will not have dealings with or buy minerals from companies that have mined valuable minerals from the seabed before the environmental risks have been properly assessed in research and documented. This makes it less lucrative for companies to invest, before studies have either developed technology to reduce these risk to an acceptable level or proved that the environmental problem is solved. As of now prospecting is being carried out to find potential mining areas. Cost and efficiency to make this industry profitable is still being looked into and studies are being conducted [24]. As the interest towards deep sea mining continues to rise, it is expected to be an industry in the next decade.

When introducing UHI or any other vision based sensor to an underwater environment, it is important to account for the possible issues that might occur when trying to use such devices in water. Examples of this being, scattering effects, occlusion and light limitations. Keeping a constant distance is important to maintain the quality of the image capture. External forces may intrude and misplace the ROV from its planned operational route. The main drive behind this project, is to look into and try to solve the issue with keeping a constant distance with the UHI when the seafloor is comprised of an irregular plane. As to enable it to achieve capturing quality images with a high resolution.

1.1.1 Autonomous Underwater Robots

Since the 1980's, ROV have been developed into being a practical solution for exploring the depths of the ocean, [4]. These types of machines were first developed by the english Royal Navy, late back in the 1950's, for recovering undetonated ordnance from the seafloor.



Figure 1.1: The Royal Navy ROV (Cutlet). Developed and introduced in 1950s to retrieve undetonated torpedoes and mines, [5]

The usage of this technology saw opportunity and was expanded later when the oil industry began to flourish, to not only cover exploration, but for monitoring of different industrial sites and to perform simple intervention and maintenance tasks. The ROV industry grew to be commercially available as computer science got more advanced, managing to reduce both the initial cost and the size of ROVs. This has resulted in the technology becoming more available to both industries and academic institutions. The general setup of a ROV consists of having a mainframe, with buoyancy elements mounted above on the mainframe. The amount of thrusters varies, but having enough to create desired motions in four degrees of freedom (DOF), surge, sway, heave and yaw is the normal configuration. ROVs operate using tethered cables for power and data-transfer resulting in having a support vessel with it at all times. A new type of ROV was eagerly sought to be developed to lessen the need of having to allocate vast amounts of resources to operate. AUVs aim is to remove the need of having a pilot. Most AUVs are torpedo shaped to be better attuned to movement through water [36]. With limited energy capacity, these are not as well suited for intervention missions.

Underwater Snake Robots

Underwater Snake Robots (USR) such as Eelume [9] aim to expand AUV operations. By combining the range of a AUV with the accessibility of areas and intervention capabilities that small ROVs have, makes snake robots an attractive asset for the industry.

Snake Robots have been inspired by biological snakes. From the amazement of observing these animals, being able to traverse almost any type of terrain, which also includes the ability to swim in water. This has led researchers to try to apply these same robust motion capabilities to robots as they have been displayed in nature [15]. For Snake robots, the body is made up of several links that are connected together by actuated joints. By actuating each joint in an ordered sequence it is possible to create a periodic wave motion. This creates a forward thrust, much like a biological snake. Even though it is possible to create propulsion this way, studies have concluded that it is an inefficient way for forward motion when comparing it to using only thrusters, [13]. The results from the study suggest

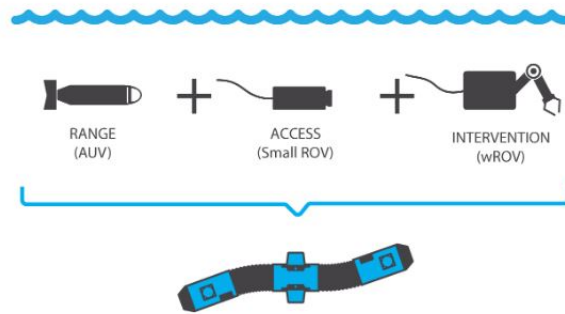


Figure 1.2: Advantages of using Snake Robots presented by Eelume,[9]

using thrusters as the source for propulsion, while using actuation between links to create body shapes that are optimal for a the given operational task. USR are slender by design and their flexibility allows access to otherwise inaccessible areas to common box-shaped ROVs. This makes them especially well suited to conduct inspections, maintenance and repairs. Traditional ROVs have movement only in four degrees of freedom (DOF). This is caused by the a buoyancy block overhead that creates passive restoring moments to counter for movement in pitch and roll. USRs have the ability to additionally move in these two domains, making them capable of moving orthogonal to the given mapping surface.

By being made up of several different links, it creates a higher degree of complexity. Each link can be assumed to be a rigid body which, when connected, creates a chain. This adds multiple degrees of freedom when each link can move in 2-DOFs, able to pan and tilt. It can be challenging to achieve the complex dynamics needed to control joints, thrust and desired behaviour.

1.1.2 Visual Mapping of the Ocean Floor

Poor visibility is a problem when descending into the deep ocean. Visual mapping can become a great challenge to do since there are no natural light sources available. The sun is only able to penetrate the two upper levels, defined as the Euphotic- (Sunlight) 0 to 200 meters, and Dysphotic-zone (Twilight) 200 to 1000 meters. There is rarely any significant light after 200 meters limiting the possibility to use visual tools without an external lightsource, [30]. Due to the poor visibility conditions in the deep sea, mapping platforms must be operated close to the seafloor and equipped with an artificial light source to ensure the capture of high-quality data. The use of an artificial light source is crucial in order to illuminate the seafloor and improve visibility for the mapping sensors. This leads to having small area of coverage and mapping speeds, [32]. The earliest visual systems were designed for inspection or exploration. Since then the mapping systems have evolved to not only have a visual overview of the seabed but also able to use the captured imagery

to be used in a well understood photogrammetry model that can be used to recreate the bathymetry of the seafloor, [25].

Common issues for the early systems and for some today, are still experiencing problems regarding the refraction of light when used in water. When capturing imagery underwater, these issues can cause problems in acquiring good photos. Some of these are loss of contrast and sharpness from scattering effects, distorted colors by attenuation and uneven illumination that is generated from the crafts artificial light source. To account for these effects, studies of image restoration methods have been developed to create high quality underwater mosaics with uniform coloring, [1].

Underwater Hyperspectral Imager

Hyperspectral Imager (HI) differs from how traditional digital cameras operate. This is by them being able to capture more spectral bands than the human eye can perceive. Digital cameras operate with three spectral bands, Red, Green and Blue (RGB). This limits the capacity to detect biochemical properties of sediments, minerals and small ecological process. Further workings of this sensor will be elaborated in section 3.2.4.

1.2 Research Questions and Objectives

The research question that this thesis aims to solve:

Is it possible to design and develop an altitude control system, using an underwater snake robot to favorably position its body relative to the seafloor, thus making it possible to map terrain such as a level, inclined or a precipice surface? The underwater snake robot will be confined to a simple body shape, such as a torpedo. Also it would need to be able to accommodate UHI mounted upon its mainframe

The resulting research questions are:

- Present the mathematical model of the hydrodynamics for a underwater vehicle.
- Investigate prior research and development of relevant unmanned underwater vehicle control systems.
- Describe a design for an altitude control system.
- Simulate vehicle dynamics with the proposed altitude and speed.

1.3 Main Contributions

The Main Contributions from this thesis are:

- Development of an altitude control system for unmanned underwater vehicles. The control system runs in real time, and by choosing a simulation platform with visualization, gives the user a good intuitive overview over how control system responds to external perturbations and control input. The system is set up with configurable mission parameters, making the system able have a setup adaptive for the mission objectives of the operator.

1.4 Research Method

The methodology used for this thesis includes literature reviews, development of simulated sensors and motion control systems. The simulation model used is built by combining mathematical theory and studies with empirical observations. The software chosen to create the simulation is reliant on several open source libraries, including ROS2, Gazebo with relevant plugins. The developed altitude control system is coded to be modular following the development standards of ROS2, and can be applied to any standard modeled underwater vessel. A few simulation case studies were conducted to demonstrate that the altitude control system acted and worked as intended, given an irregular surface.

1.5 Thesis Outline

Chapter 2 - Kinetics and kinematics

This chapter explains and presents how both a single and multi rigid body entity can be modelled. The generalized kinematics and kinetics present for the underwater vehicle are described.

Chapter 3 - Underwater Hyperspectral Imager and Guidance, Navigation & Control

This chapter discusses the requirements and use of UHI. It further describes how motion control systems are commonly structured. The primary sensors that have been applied to the developed motion control system are shown together with a brief overview of their functionality.

Chapter 4 - Development of a Motion Control for Constant Altitude

The proposed altitude control system is presented. The components making up the system, signal processing, observer, controller and thruster allocation are given in detail in this chapter. The results from the case simulations are presented in short.

Chapter 5 - Discussion

This chapter discusses the results of the developed system. It attempts to give insight to the issues that exist within the system and how some of them were solved.

Chapter 6 - Conclusion and Suggested Further Work

This chapter concludes the Master thesis's attempt to answer the research questions that are focused. Suggestions towards further development are summarized.

Chapter 2

Kinetics and kinematics

This chapter aims to give insight into the mathematical model which is used when describing motions of an underwater vehicle. The model can be separated into two main components. Kinematics, which describe geometrical aspect of the model's motions, and kinetics that takes the external forces causing motion to the model. Fossen's robotic-like vectorial model has been found to be an efficient way to be able to express motions of a vehicle with freedom in six degrees, including coherent coupling effects in matrix form.

2.1 Kinematics

To be able to describe the kinematics of a multi rigid-body entity such as an underwater snake robot (USR), the kinematics of a single rigid-body needs to be described first. The following section will present generalized coordinates, the different reference frames which are commonly used and transformations between the different reference frames.

2.1.1 Generalized coordinates

The notations that have been used in this thesis are SNAME convention, presented in Fossen's vectorial model [10]. ROV's in water will be able to have motion in six degrees of freedom (DOF). These degrees are described in table 2.1.

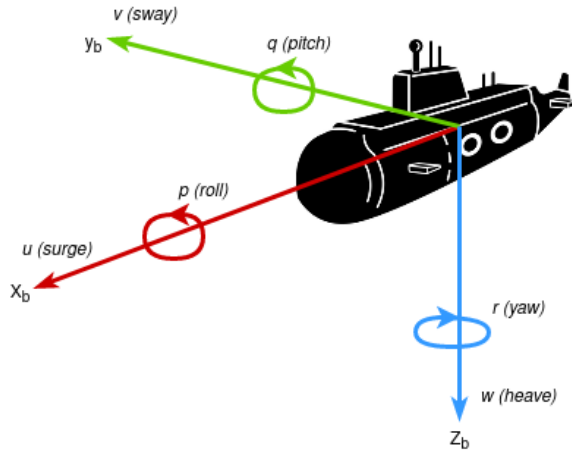


Figure 2.1: The six-DOF motions in a body-fixed coordinate frame. [35]

Table 2.1: The notation of SNAME (1950) for marine vessels

Number	DOF	Forces and moments	Positions and orientations	Linear and angular velocities
1	Surge	X	x	u
2	Sway	Y	y	v
3	Heave	Z	z	w
4	Roll	K	ϕ	p
5	Pitch	M	θ	q
6	Yaw	N	ψ	r

$$\eta = [x \quad y \quad z \quad \phi \quad \theta \quad \psi]^T \quad (2.1)$$

$$\nu = [u \quad v \quad w \quad p \quad q \quad r]^T \quad (2.2)$$

(2.1) is the combined linear and angular position of the system. (2.2) shows combined linear and angular vector of the velocity. These two vectors can be broken down into sub-vectors for linear and angular. These equations portray coordinates in the body frame of the vehicle.

2.1.2 Reference frames

There exists different types of reference frames that each have their advantages and uses. They are commonly used for navigation or to describe motions of a system that can be fixed to a moving point. The three that are most common for navigation are presented here.

ECI , $\{i\}$ Earth-Centered Inertial frame. Axes $\{i\} = [x_i, y_i, z_i]$,

ECEF , $\{e\}$ Earth-centered Earth fixed frame. Axes $\{e\} = [x_e, y_e, z_e]$,

NED , $\{n\}$ North-East-Down frame. Axes $\{n\} = [x_n, y_n, z_n]$,

BODY , $\{b\}$ BODY-Fixed frame. Axes $\{b\} = [x_b, y_b, z_b]$,

Both $\{i\}$ and $\{e\}$ are reference frames that have their origin at the earth's center. The main difference between them being that $\{e\}$ has its axes rotating relative to the earth, meaning that they rotate at the same speed as earth. $\{i\}$ on the other hand has fixed axes meaning that they do not rotate.

$\{i\}$ is an inertial frame which is used when high accuracy is demanded. $\{e\}$, measurement of the vehicle's position and orientation coordinates are done in this frame relatively to some defined origin. This reference system is useful when having to describe motion over longer distances. Each type of position-reference system has its own local coordinate system, this has to be transformed from this local reference system to the Earth-fixed reference system.

Two other reference frames that are better applied to analyze vessel kinematics are the $\{n\}$ - and BODY-frame. The $\{n\}$ reference frame is the most commonly used reference frame, being applied in most GPS systems. It is a geographical reference frame, axes are the tangent plane on the earth surface, having x-axis pointing towards true north, y-axis towards east and z-axis towards the earth's center. The BODY-frame is used to reference the forces relative to the given system. This reference frame is usually used when analyzing the local motions of the vessel. The axes are fixed to the object's body and will rotate correspondingly when the body is set in motion.

2.1.3 Transformations

To be able to set coordinates of a body into one of the reference frames, a conversion is needed. This can be done by using trigonometric relations matrices between two frames. Below are the three trigonometric relation matrices, (2.3), (2.4) and (2.5). The transformations are based on the use of Euler angles of the rigid-body. Presented in the matrices are (ϕ) as roll, (θ) as pitch and (ψ) as yaw.

$$R_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}, \quad (2.3)$$

$$R_{y,\theta} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \quad (2.4)$$

$$R_{z,\psi} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.5)$$

The rotation matrix converting $\{b\}$ to $\{n\}$ is $R_b^n(\Theta) \in SO(3)$. This can be calculated by using (2.6).

$$R_b^n(\Theta) = R_{z,\psi} R_{y,\theta} R_{x,\phi}, \quad (2.6)$$

The following matrices uses the subscript c and s to represent \sin and \cos .

$$R_b^n(\Theta) = \begin{bmatrix} c(\psi)c(\theta) & -s(\psi)c(\phi) + c(\psi)s(\theta)s(\phi) & s(\psi)s(\phi) + c(\psi)c(\phi)s(\theta) \\ s(\psi)c(\theta) & c(\psi)c(\phi) + s(\phi)s(\theta)s(\psi) & -c(\psi)s(\phi) + s(\theta)s(\psi)s(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix}, \quad (2.7)$$

It is important to note that the order of matrix multiplication matters to achieve the correct transformation matrix. In guidance, navigation and control applications, it is common to use the zyx convention when transforming from $\{n\}$ to $\{b\}$. It implies in the matrix transpose that the same result can be obtained by transforming a vector from $\{b\}$ to $\{n\}$, meaning reversing the sequence of transformation to be xyz . The inverse transformation is as shown in (2.8).

$$R_b^n(\Theta)^{-1} = R_b^n(\Theta)^T = R_n^b(\Theta) = R_{x,\phi} R_{y,\theta} R_{z,\psi}, \quad (2.8)$$

Using the time derivative of the Euler angles ($\dot{\Theta}_{nb}$), it is possible to derive an expression for the transformation of the body-fixed angular velocity vector as w_{nb}^b .

$$w_{nb}^b = [p, q, r]^T, \quad (2.9)$$

$$\dot{\Theta}_{nb} = T_{\Theta}(\Theta_{nb})w_{nb}^b, \quad (2.10)$$

$$T_{\Theta}(\Theta_{nb}) = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & \frac{s(\phi)}{c(\theta)} & \frac{c(\phi)}{c(\theta)} \end{bmatrix}, \quad (2.11)$$

The inverse of the angular-velocity transformation matrix $T_{\Theta}(\Theta_{nb})$ is given by

$$T_{\Theta}(\Theta_{nb})^{-1} = \begin{bmatrix} 1 & 0 & -s(\theta) \\ 0 & c(\phi) & c(\theta)s(\phi) \\ 0 & -s(\phi) & c(\phi)c(\theta) \end{bmatrix}, \quad (2.12)$$

From (2.11), it is undefined from values of $\theta = \pm\pi/2$. This is not a problem when considering surface vehicles, but may be problematic when modelling for underwater ones. In these cases, the problem caused by the singularity can be avoided by using unit quaternions, covered in 2.1.3. We can define the 6-DOF kinematic transformation for maneuvering coordinates in equation 2.13.

$$\dot{\eta} = J(\eta)\nu = \begin{bmatrix} R_b^n(\Theta_{nb}) & 0_{3 \times 3} \\ 0_{3 \times 3} & T_{\Theta}(\Theta_{nb}) \end{bmatrix} \begin{bmatrix} \nu_{nb}^b \\ \omega_{nb}^b \end{bmatrix} \quad (2.13)$$

$\eta \in \mathbb{R}^6$ represents the position and orientation within the $\{n\}$ frame. $J_b^n(\Theta_{nb}) \in \mathbb{R}^{6 \times 6}$ represents the Jacobian matrix, describing the rotation of the vehicle in body frame $\{b\}$ to the reference frame. $\nu \in \mathbb{R}^{6 \times 6}$ represents the linear and angular velocity as a vector in $\{b\}$ frame.

Converting euler angles to rotation quaternions

Rotation quaternions are used to represent rotations in three dimensions, and are an alternative to rotation matrices in mathematics. A quaternion is regarded as a complex number, represented by four elements.

$$\mathbf{q} = q_0 + \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3 \quad (2.14)$$

From equation 2.14, q_0 , q_1 , q_2 and q_3 are real numbers, and \mathbf{i} , \mathbf{j} and \mathbf{k} are orthogonal imaginary unit vectors. The first element, q_0 , is referred to as the real part while the other three are imaginary, [29].

$$\mathbf{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}, \quad (2.15)$$

$$q_0 = c\left(\frac{\phi}{2}\right)c\left(\frac{\theta}{2}\right)c\left(\frac{\psi}{2}\right) + s\left(\frac{\phi}{2}\right)s\left(\frac{\theta}{2}\right)s\left(\frac{\psi}{2}\right), \quad (2.16)$$

$$q_1 = s\left(\frac{\phi}{2}\right)c\left(\frac{\theta}{2}\right)c\left(\frac{\psi}{2}\right) - c\left(\frac{\phi}{2}\right)s\left(\frac{\theta}{2}\right)s\left(\frac{\psi}{2}\right), \quad (2.17)$$

$$q_2 = c\left(\frac{\phi}{2}\right)s\left(\frac{\theta}{2}\right)c\left(\frac{\psi}{2}\right) + s\left(\frac{\phi}{2}\right)c\left(\frac{\theta}{2}\right)s\left(\frac{\psi}{2}\right), \quad (2.18)$$

$$q_3 = c\left(\frac{\phi}{2}\right)c\left(\frac{\theta}{2}\right)s\left(\frac{\psi}{2}\right) - s\left(\frac{\phi}{2}\right)s\left(\frac{\theta}{2}\right)c\left(\frac{\psi}{2}\right), \quad (2.19)$$

Converting Quaternion to Euler Angles

(2.20), (2.21) and (2.22) show how to make a conversion between quaternion to euler angles.

$$roll = \phi = \tan^{-1}\left(\frac{2(q_0q_1 + q_2q_3)}{q_0^2 - q_1^2 - q_2^2 + q_3^2}\right) = (atan2)[2(q_0q_1 + q_2q_3), q_0^2 - q_1^2 - q_2^2 + q_3^2], \quad (2.20)$$

$$pitch = \theta = \sin^{-1}(2(q_0q_2 - q_1q_3)) = (asin)[2(q_0q_2 - q_1q_3)], \quad (2.21)$$

$$yaw = \psi = \tan^{-1}\left(\frac{2(q_0q_1 + q_2q_3)}{q_0^2 + q_1^2 - q_2^2 - q_3^2}\right), \quad (2.22)$$

The special case where the pitch Euler angle equals $\pm 90^\circ$, referred to as a "Gimbal Lock", the parameters in 2.20 and 2.22 will all be zero. This is caused by the roll and yaw axes becoming aligned with each other in the global frame ($\{n\}$), and providing no unique solution. Worth to note, while Euler angles are susceptible to this, quaternions are not. This is why introducing them adds the benefit of robustness to the mathematical model.

USRs are made up of multiple rigid-bodies in a chain, where each body has its own coordinate frame attached to it. When calculating the position of each body, there is a high chance that a lock could occur.

2.2 Kinematics of a manipulator

When modelling a USR, another layer of complexity is needed to be able to accurately account for its motions. This kind of robot can be assumed to be a floating manipulator arm. A manipulator would consist of a multitude of links that are connected to each other through joints. Each link is considered a rigid-body, and the joints would commonly be rotary or linear. As this thesis will not involve complex body shapes, the kinematics presented will be restricted to only focus on describing rotary joints. A simplification of a joint is considering it being like a door hinge with a maximum and minimum angle it can perform between the two connected links.

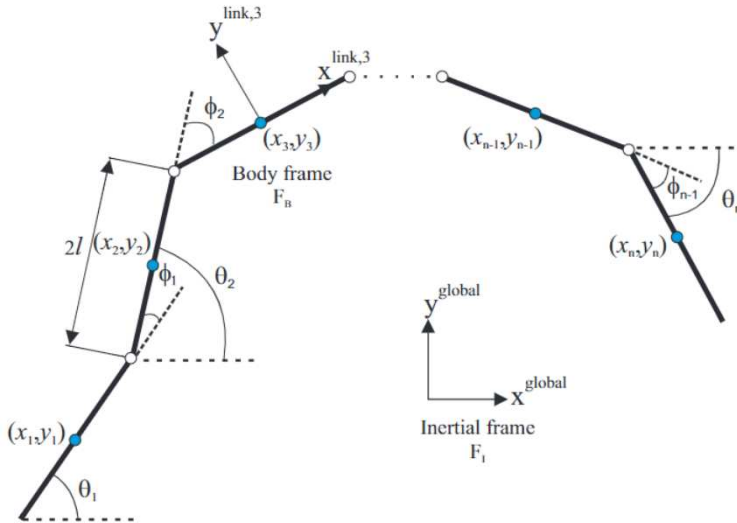


Figure 2.2: Snake Robot modeled as a manipulator arm constricted to links connected to revolute joints, [13].

Each link in the manipulator makes up a chain of links, where each link would have its own reference frame. In order to be able to determine the position and angle of the manipulator, it is necessary to be able to express the position of each link relative to each other. The entire manipulator can then be expressed with its positions and orientations, also known as pose, in a fixed inertial reference frame such as $\{n\}$. As pose is for position and orientation, likewise Twist is used for describing the linear and angular velocities, [14].

2.2.1 Homogenous transformations

Homogeneous transformation matrices (HTM), are used as a tool within robotics to be able to accurately describe both the position and orientation of a rigid-body relative to a global frame.

The configuration of the body can be expressed by \mathbf{R} and \mathbf{p} . $\mathbf{R} \in SO(3)$ is a 3×3 rotation matrix representing the orientation in $\{b\}$ relative to the global frame, $\{n\}$. $\mathbf{p} \in \mathbb{R}^3$ is a 3-vector which represents the bodies origin position in $\{b\}$ relative to $\{n\}$, [20].

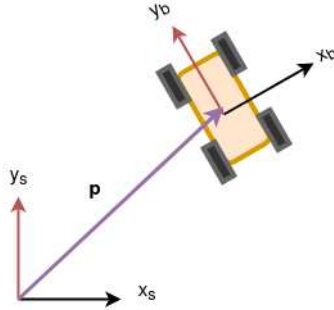


Figure 2.3: Homogenous transform between $\{b\}$ and a global frame such as $\{n\}$

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ 0_{1 \times 3} & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_1 \\ r_{21} & r_{22} & r_{23} & p_2 \\ r_{31} & r_{32} & r_{33} & p_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.23)$$

Example - Two-dimensional Homogenous Pose Transformation

Looking at a two dimensional example of a simple four wheeled robot. Each a to $\{d\}$ frame is attached to the center of the robot.

1. Starts in $\{a\}$. First rotates -90° and travels in surge 3 unit lengths to pose $\{b\}$.
2. Then it rotates again 90° , then travels 3 unit lengths in surge to pose $\{c\}$.
3. Rotates 90° then travels 2 unit lengths in surge ending in pose $\{d\}$.

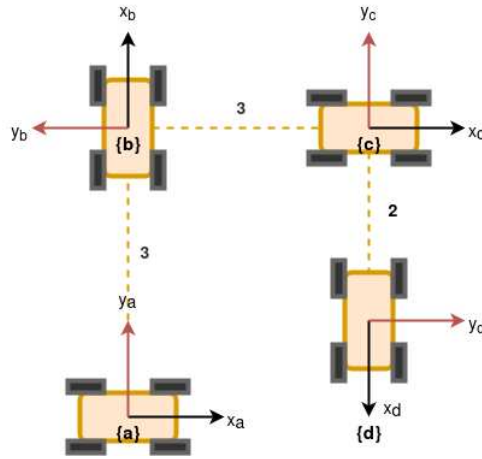


Figure 2.4: Example for homogenous transformation of a rover robot, in four different poses

The homogenous transformation matrices describing the configuration of the robot at each stage was calculated to be:

$$T = \begin{bmatrix} r_{11} & r_{12} & p_1 \\ r_{21} & r_{22} & p_2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & p_1 \\ \sin(\theta) & \cos(\theta) & p_2 \\ 0 & 0 & 1 \end{bmatrix}, \theta \in [0, 2\pi] \quad (2.24)$$

$$T_{ab} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix}, T_{bc} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & -3 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.25)$$

$$T_{cd} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & -2 \\ 0 & 0 & 1 \end{bmatrix}, T_{sd} = \begin{bmatrix} 0 & -1 & 3 \\ 1 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.26)$$

2.2.2 Forward Kinematics

Within robotics, the "end-effector" is the component which is positioned at the end of the arm. This part is usually able to interact with the surrounding environment. Forward kinematics refers to usage of kinematic equations to calculate the pose of the end-effector from the joint parameter values. $\mathbf{q} = [\theta_0, \dots, \theta_n]^T$ represents a vector with the manipulator joint angles. Each joint has its movements limited to 2-DOF, pitch and yaw, and each link i in the arm is connected to the adjacent link $i + 1$. When the joint between link i and link $i + 1$ is actuated, as seen in figure 2.5.

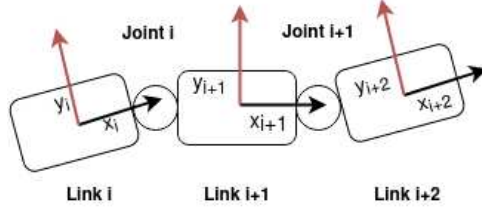


Figure 2.5: Actuation of joints between their adjacent links.

Each link has an adjacent fixed body reference frame. For describing the pose of the end-effector, homogeneous transform can be used as previously presented in section 2.2.1.

Suppose that A_i represents the homogeneous transformation matrix which expresses both the position and orientation of link i with respect to link $i - 1$. From previous, all joints are revolute, making the input argument of the function to be a single joint variable, θ_i

$$A_i = A_i(q_i), \quad (2.27)$$

$$A_i(\theta_i) = A_i(0) \begin{bmatrix} \mathbf{R}_{s_i, \theta_i} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \quad (2.28)$$

$$A_i(0) = \begin{bmatrix} \mathbf{I}_3 & l_i \mathbf{e}_1 \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \quad (2.29)$$

From (2.29), l_i represents the length of link i . $\mathbf{e}_1 = [1 \ 0 \ 0]^T$ represents the local x-axis direction of the body to link i . $\mathbf{R}_{s_i, \theta_i}$ represents the rotation matrices presented in (2.3), (2.4) and (2.5). The subscript used in the rotation matrix, s_i is defines which axis joint i rotates on.

It is possible now to define the pose of a frame j with regard to the pose in frame i , as described in [33].

$$\mathbf{T}_i^j(\mathbf{q}) = \begin{cases} \mathbf{A}_{i+1} \mathbf{A}_{i+2} \dots \mathbf{A}_{j-1} \mathbf{A}_j, & \text{if } i < j \\ \mathbf{I}, & \text{if } i = j \\ (\mathbf{T}_i^j)^{-1}, & \text{if } j > i \end{cases} \quad (2.30)$$

Using this, the position and orientation of link i can be determined in the inertial frame.

$$A_i = \begin{bmatrix} R_i^{i-1} & o_i^{i-1} \\ 0 & 1 \end{bmatrix} \quad (2.31)$$

$$T_j^i = A_{i+1} \dots A_j = \begin{bmatrix} R_j^i & o_j^i \\ 0 & 1 \end{bmatrix} \quad (2.32)$$

o_n^0 denotes the position and orientation of the end-effector, this is with respect to base frame and given by a 3×1 vector. To determine any other link i , o_i^{i-1} is set to be p of the link in favor.

The theory that has been presented so far describes how to determine position and orientation of a manipulator link as long as the connected joints are limited to rotate in one DOF, which was chosen as pitch (θ). To enable a rotary ball joint, which is similar movement to expect when controlling a USR, there is needed to be added rotation also in yaw (ψ). This can be done by first redefining the manipulator joint angles from $\mathbf{q} = [\theta_0, \dots, \theta_n]^T$ to also include yaw to be $\mathbf{q} = [\theta_0, \psi_0, \dots, \theta_n, \psi_n]^T$, [33]. A desired transformation can be made, by combining rotation matrices, as presented in section 2.1.3. It is important to note that the sequence which the rotations are done in matters for the outcome. This is since the current axes is used for when calculating the next rotation. Therefore doing it in any other order will result in the wrong rotation as the current axes used will not be the same.

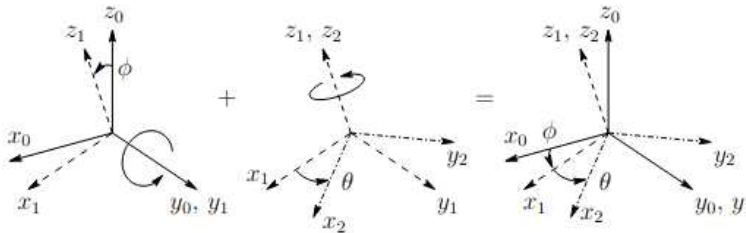


Figure 2.6: Sequence of rotations, from [33]

Example - Rotation matrix in practical application to determine manipulator end-effector

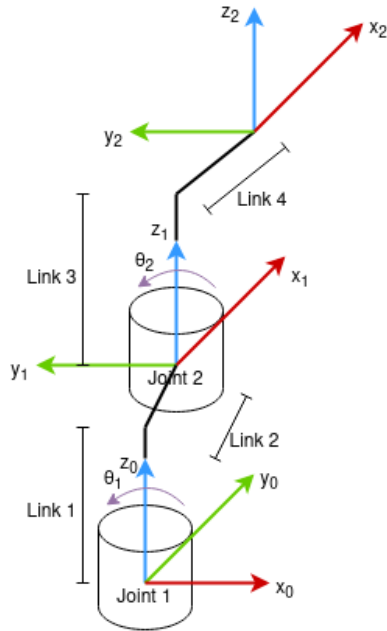


Figure 2.7: Kinematic diagram of a manipulator arm

In practical application of using rotation matrices can be seen in figure 2.7. There are four links making up a manipulator arm. In this case $\theta_1 = 90$ degrees as angle of servo 0, and servo 1 angle at $\theta_1 = 0$.

- If an axis in one body frame points in the same direction as an axis in another, a 1 is assigned in the matrix to account for this.
- If an axis in one body frame points in the opposite direction as an axis in another, a -1 is assigned in the matrix to account for this.
- If an axis in one body frame is perpendicular to an axis of another frame, a 0 is assigned.

The resulting rotation matrix:

Table 2.2: Resulting rotation matrix of the kinematic diagram 2.7

	x_2	y_2	z_2
x_0	0	-1	0
y_0	1	0	0
z_0	0	0	1

2.3 Kinetics

As previously mentioned, an USR is essentially a floating base with a manipulator arm attached to it. It is subjected to external forces which add motions to the body, such as added mass, drag and restoring hydrostatic forces. The kinetics of a multi rigid bodied system can be defined to be solved either recursively using the equations of motion (EOM) for each link, or it can be solved as a matrix, described by Fossen in [10]. This would solve all the equations at the same time. The easiest of the two would be to solve the equations recursively, however this could be difficult to do if the number of links become high. The EOM are computational expensive to solve. If the body in question contains a large sum of links and joints, the computational cost might become to high for it to be a good method to use. Others might be more prudent use in achieving cheap and fast computational results. As for this thesis, the snake modelled was limited to five links and four joints, making the EOM easy enough to solve recursively.

Table 2.3: Description of the parameters of this sections presented matrices

Parameter	Description	Unit
m	Mass	kg
W	Dry weight	$Newton$
B	Buoyancy	$Newton$
r_b	Distance from CB to center of the body frame	m
r_g	Distance from CG to center of the body frame	m
I_x	Inertia moment on x_b	kgm^2
I_y	Inertia moment on y_b	kgm^2
I_z	Inertia moment on z_b	kgm^2
$X_{\dot{u}}$	Added mass in surge	kg
$Y_{\dot{v}}$	Added mass in sway	kg
$Z_{\dot{w}}$	Added mass in heave	kg
$K_{\dot{p}}$	Added mass in roll	m^2/rad
$M_{\dot{q}}$	Added mass in pitch	m^2/rad
$N_{\dot{r}}$	Added mass in yaw	m^2/rad
X_u	Linear damping surge	Ns/m
Y_v	Linear damping sway	Ns/m
Z_w	Linear damping heave	Ns/m
K_p	Linear damping roll	Ns/m
M_q	Linear damping pitch	Ns/m
N_r	Linear damping yaw	Ns/m
$X_{ u u}$	Quadratic dampening surge	Ns^2/rad^2
$Y_{ v v}$	Quadratic dampening sway	Ns^2/rad^2
$Z_{ w w}$	Quadratic dampening heave	Ns^2/rad^2
$K_{ p p}$	Quadratic dampening roll	Ns^2/rad^2
$M_{ q q}$	Quadratic dampening pitch	Ns^2/rad^2
$N_{ r r}$	Quadratic dampening yaw	Ns^2/rad^2

2.3.1 Fossen's Robot-Inspired Model for Marine Craft

For simulation purposes, a model should have as many needed details as system requirements specify. The components should be of high fidelity, as to accurately be able to replicate the system dynamics digitally. Equations describing the ROV's motion by relative velocities in a vectorial form are obtained from [10] and presented in (2.33) and (2.34).

$$\dot{\eta} = J_{\theta}(\eta)\nu, \quad (2.33)$$

$$\underbrace{M_{RB}\dot{\nu} + C_{RB}(\nu)\nu}_{\text{rigid body terms}} + \underbrace{M_A\dot{\nu}_r + C_A(\nu_r)\nu_r + D(\nu_r)\nu_r}_{\text{hydrodynamic terms}} + g(\eta) = \tau + \tau_{ext}, \quad (2.34)$$

Where the terms from (2.33) and (2.34) are,

$M_{RB} \in \mathbb{R}^{6 \times 6}$	represents the rigid body mass matrix.
$C_{RB} \in \mathbb{R}^{6 \times 6}$	represents the rigid body Coriolis and centripetal matrix.
$M_A \in \mathbb{R}^{6 \times 6}$	represents the added mass matrix.
$C_A(\nu_r) \in \mathbb{R}^{6 \times 6}$	represents the added mass Coriolis and centripetal matrix.
$D_{\nu_r} \in \mathbb{R}^{6 \times 6}$	represents the damping matrix.
$g(\eta) \in \mathbb{R}^{6 \times 1}$	represents the hydro-static restoring force vector.
$\tau \in \mathbb{R}^{6 \times 1}$	represents the input force vector.
$\tau_{ext} \in \mathbb{R}^{6 \times 1}$	represents the external forces.
$\nu_r \in \mathbb{R}^{6 \times 1}$	represents the relative velocity vector.
$J_{\theta} \in \mathbb{R}^{6 \times 6}$	represents the rotation matrix from $\{b\}$ to $\{n\}$.

$M \in \mathbb{R}^{6 \times 6}$, represents the mass matrix of the system the including added mass.

$$M = M_{RB} + M_A, \quad (2.35)$$

$C \in \mathbb{R}^{6 \times 6}$, represents the Coriolis-centripetal force matrix including the added mass

$$C(\nu_r) = C_{RB}(\nu_r) + C_A(\nu_r), \quad (2.36)$$

$\nu_r \in \mathbb{R}^{6 \times 1}$, represents the relative velocity vector when compared to the ocean current velocity, decomposed in $\{b\}$.

$$\nu_r = \nu - \nu_c, \quad (2.37)$$

2.3.2 Rigid-body Dynamics

$$M_{RB}^{CG} = \begin{bmatrix} m_i I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_g \end{bmatrix}, \quad (2.38)$$

The rigid-body mass matrix located at the center of gravity (CG) of the object is expressed in equation 2.38.

$$I_g = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{bmatrix}, \quad (2.39)$$

$I_g \in \mathbb{R}^{3 \times 3}$ represents the inertia matrix for the center of gravity (CG). I_x , I_y and I_z are the moments of inertia around the body frame axes. The products of inertia can be expressed as $I_{xy} = I_{yx}$, $I_{xz} = I_{zx}$ and $I_{yz} = I_{zy}$.

$$C_{RB}^{CG} = \begin{bmatrix} mS(w_{b/n}^b) & O_{3 \times 3} \\ O_{3 \times 3} & -S(I_g w_{b/n}^b) \end{bmatrix}, \quad (2.40)$$

S is a matrix which contains the unit rotation vectors. $C_{RB}^{CG} \in \mathbb{R}^{6 \times 6}$ represents the rigid body Coriolis and centripetal matrix in CG. This expression has to be transformed from the point of CG to center of origin (CO) before it can be used in (2.34). This can be done by using 2.41 in (2.42) and (2.43).

$$H(r_g^b) = \begin{bmatrix} I_{3 \times 3} & S^T(r_g^b) \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix}, \quad (2.41)$$

$$M_{RB} = H^T(r_g^b) M_{RB}^{CG} H(r_g^b), \quad (2.42)$$

$$C_{RB} = H^T(r_g^b) C_{RB}^{CG} H(r_g^b), \quad (2.43)$$

2.3.3 Control Plant Model

The control plant model is used as a simplified mathematical description to only having to contain the most important physical properties of a single process. It is commonly used when designing a controller or for determining analytical stability. It accounts for the main dynamics of a underwater vehicle. Complex processes are hard to model, and often some of the processes aspects need to be made easier. It is important to note that it is desired to have the model depicted as accurately as possible, but still able to use in calculations. Such

a simplification can be that for vehicles operating at low speeds, with low speeds defined as < 1 [m/s]. The Coriolis and centripetal forces, as well as the linear damping small can be neglected (2.44), (2.45) and (2.46). Current velocities interacting with the body are slowly varying or constant. (2.34) is given in terms of the vehicles velocities. Other external ocean forces are included in the bias estimate. When pitch and roll motions are small, defined as motion angles being < 10 [degrees], the center of buoyancy (CB) would be positioned directly above CG, making such that the restoring forces are linearized using G .

$$\dot{\eta} = J(\eta)\nu, \quad (2.44)$$

$$M\dot{\nu} + D\nu + G\eta = \tau + J^T(\eta)b, \quad (2.45)$$

$$\dot{b} = -T_b^{-1}b + w_b, \quad (2.46)$$

$D \in \mathbb{R}^{6 \times 6}$ represents the linear damping matrix.
 $b \in \mathbb{R}^{6 \times 1}$ represents the bias vector which accounts for the unmodelled dynamics and slowly varying loads.
 $T_b \in \mathbb{R}^{6 \times 6}$ represents a positive diagonal matrix, including bias time constants.
 $w_b \in \mathbb{R}^{6 \times 1}$ represents the zero mean Gaussian white noise processes.
 $G \in \mathbb{R}^{6 \times 6}$ represents the restoring force matrix.

G needs to be transformed from CG to CO. The calculation for the transformation is shown in equation 2.47.

$$G^{CG} = \text{diag}\{0, 0, 0, -z_b B, -z_b B, 0\}, \quad (2.47)$$

$$G = H^T(r_g^b)G^{CG}H(r_g^b), \quad (2.48)$$

2.3.4 Thrust Vector

The thrust vector is constructed by i amount of thrusters in a system configuration. A system with r thrusters, has it's vector represented as (2.49).

$$f = [f_1, f_2, \dots, f_r]^T, \quad (2.49)$$

The total force from thrusters, in p DOFs, are given by (2.50).

$$\tau = Tf \quad (2.50)$$

$T \in \mathbb{R}^{p \times r}$ represents the thruster configuration matrix.

$$T = [t_1, t_2, \dots, t_r] \quad (2.51)$$

The function is based on the thrusters position on the vehicle, $r_{t_i/b}^b$. Azimuth and elevation angles are expressed as α and β .

$$t_i = \begin{bmatrix} I_{3 \times 3} \\ -S^T(r_{t_i/b}^b) \end{bmatrix} R(\alpha, \beta) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} f_i, \quad (2.52)$$

The rotation matrix of the thruster axes when on the body of the vehicle are calculated as shown in (2.53). The elevation angles are kept constant and fixed, which simplifies thruster allocation 2.3.5.

$$R(\alpha, \beta) = R_{z,\alpha} R_{y,\beta}, \quad (2.53)$$

$$R_{y,\beta} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \quad (2.54)$$

$$R_{z,\alpha} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.55)$$

2.3.5 Thrust Allocation

The thrust vector τ from (2.50), is given by the control system. It is used first for mapping the generalized forces, then to allocate how much thrust force each thruster needs to produce to create movement along a desired path. The input u_i of each thruster, must be translated into revolution speed of the propeller. (2.50) can be rewritten to be as:

$$\tau = TKu, \quad (2.56)$$

$K \in \mathbb{R}^{r \times r}$ represents the diagonal thruster coefficient matrix, where $K_i = K_T(J)\rho D^4$. $u \in \mathbb{R}^r$ is the input, where $u_i = |n|n$. n is the revolution speed of the propeller. To solve for the input in the thruster allocation problem we need to solve equation 2.57

$$u = K^{-1}T^{-1}\tau, \quad (2.57)$$

The revolution speed of each propeller is found by using equation 2.58.

$$n_i = \text{sign}(u_i)\sqrt{|u_i|}, \quad (2.58)$$

2.4 Hydrodynamics

2.4.1 Added Mass

The added mass created of body in a fluid can be expressed as in the added mass matrix 2.59.

$$M_A = - \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix}, \quad (2.59)$$

The added-mass matrix 2.59 is symmetrical when the body is either moving or oscillating in water, meaning that $M_A = M_A^T$. For the intended purpose of seabed mapping at deep depths, it is assumed underwater vehicles has no interaction with forces caused by waves on the free surface. When moving through water, the surrounding water will accelerate relative to the body, which creates a pressure field that is frequency dependent. This pressure field can be integrated across the entire body to find the added mass interacting with the body, this does calculation does not include the hydrostatic pressure. When completely submerged, only restoring forces in pitch and roll exist. For surge, sway, heave and yaw, the added mass is calculated from the zero-frequency.

M_A is symmetrical when moving, meaning that if all 36 terms were deemed necessary it would suffice with the diagonal and the 15 terms above. The diagonal elements describe the added mass in the 6 DOFs. The off-diagonal terms in (2.59) will be small when compared to the diagonal elements. Therefore it is deemed to be satisfactory to only find the diagonal elements when accounting for the added-mass. The matrix can be simplified to (2.60).

$$M_A = -diag\{X_{\dot{u}}(0), Y_{\dot{v}}(0), Z_{\dot{w}}(0), K_{\dot{p}}(w_{roll}), M_{\dot{q}}(w_{pitch}), N_{\dot{r}}(0)\}, \quad (2.60)$$

2.4.2 Damping

For a underwater vehicle, potential and wave related damping effects can be neglected. Even though these are neglected, some damping still interacts with the body. These effects are caused mostly from vortex shedding when moving through water and friction from the surface of the body. It can be difficult to determine the damping properties. Simplifications to get approximations are often done, such as simplifying the shape of the body to either a sphere or a cube, making it easier to calculate. Other objects such as cameras, sensors and thrusters are not usually accounted for in such an approximation. These components will

also create effects towards the damping properties. The simplification that is chosen here are presented in (2.61).

$$D_{\nu_r} = D + D_n(\nu_r) \quad (2.61)$$

$D \in \mathbb{R}^{6 \times 6}$, represents the linear damping matrix. This due to the skin friction.

$D_n(\nu_r) \in \mathbb{R}^{6 \times 6}$, represents the quadratic damping matrix. This comes mainly from vortex shedding of the body moving through water.

$D(\nu_r) \in \mathbb{R}^{6 \times 6}$, represents the damping matrix. This matrix is strictly positive, due to energy dissipation out of the system when damping occurs. The diagonal terms in these matrices are dominant allowing for simplification. These can be reduced to the following state as shown in equation 2.62 and 2.63.

$$D = -diag\{X_u, Y_v, Z_w, K_p, M_q, N_r\}, \quad (2.62)$$

$$D_n = -diag\{X_{|u|u}|u|, Y_{|v|v}|v|, Z_{|w|w}|w|, K_{|p|p}|p|, M_{|q|q}|q|, N_{|r|r}|r|\}, \quad (2.63)$$

2.4.3 Restoring- and Hydrostatic-forces

A body in a fluid will have interaction of both gravitational- and a buoyancy-forces. These forces are commonly associated with ROVs as restoring moments, that brings a tilted body back into a stable position if rightly configured as such. The hydrostatic restoring forces have been calculated in CG.

$$g^{CG}(\eta) = \begin{bmatrix} (W - B)\sin\theta \\ -(W - B)\cos\theta\sin\phi \\ -(W - B)\cos\theta\cos\phi \\ y_b B \cos\theta \cos\phi - z_b B \cos\theta \sin\phi \\ -z_b \sin\theta - x_b B \cos\theta \cos\phi \\ x_b B \cos\theta \sin\phi + y_b B \sin\theta \end{bmatrix}, \quad (2.64)$$

W represents the weight of the body and B the buoyancy when submerged in a fluid. This is calculated as shown in equation 2.65 and 2.66.

$$W = mg, \quad (2.65)$$

$$B = \rho g \nabla, \quad (2.66)$$

From (2.65) and (2.66), the terms are:

- m represents the dry mass of body.
- g represents the gravitational constant.
- ρ represents the density of the fluid the body is submerged in.
- ∇ represents the displaced volume of fluid done by the body.

When in neutral the equation can be set to be $W = B$.

As the restoring forces are calculated in CG, a transformation is needed to convert it into CO state.

$$g(\eta) = H^T(r_g^b)g^{CG}(\eta), \quad (2.67)$$

where H^T is the transposed transformation matrix presented in (2.41).

2.4.4 Coriolis and Centripetal forces

These are the forces experienced when rotational movement is applied on the body. This is determined by the rate of rotation and the mass of the object. The Coriolis force is perpendicular to the objects axis.

$$C_A = - \begin{bmatrix} 0_{3 \times 3} & -S(A_{11}v + A_{12}\omega) \\ -S(A_{11}v + A_{12}\omega) & -S(A_{21}v + A_{22}\omega) \end{bmatrix}, \quad (2.68)$$

C_A represents the Coriolis and centripetal matrix. This representation is valid for a rigid-body moving through an ideal fluid. The matrix can be parameterized to be skew-symmetric:

$$C_A = -C_A^T(\nu), \quad \forall \nu \in \mathbb{R}^{6 \times 1} \quad (2.69)$$

$A_{ij} \in \mathbb{R}^{3 \times 3}$ and includes the same terms as in the added mass matrix M_A .

$$M_A = - \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad (2.70)$$

2.4.5 Current

From Fossen [10], a three-dimensional irrotational model is proposed. (2.71) is expressed in the $\{n\}$ frame.

$$v_c^n = R_{y,\alpha_c}^T R_{z,-\beta_c}^T \begin{bmatrix} V_c \\ 0 \\ 0 \end{bmatrix} \quad (2.71)$$

Expanding (2.71) yields

$$v_c^n = \begin{bmatrix} \cos(\alpha_c) & 0 & \sin(\alpha_c) \\ 0 & 1 & 0 \\ -\sin(\alpha_c) & 0 & \cos(\alpha_c) \end{bmatrix} \begin{bmatrix} \cos(-\beta_c) & -\sin(-\beta_c) & 0 \\ \sin(-\beta_c) & \cos(-\beta_c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_c \\ 0 \\ 0 \end{bmatrix} \quad (2.72)$$

From (2.72), the ocean current speed is declared as V_c . α_c represents the vertical current direction. β_c represents the horizontal current direction. The angles are expressed relative to $\{n\}$. To transform from $\{n\}$ to $\{b\}$, is expressed in (2.73).

$$v_c^b = \begin{bmatrix} u_c \\ v_c \\ w_c \end{bmatrix} = R_b^n(\theta_{nb})^T v_c^n \quad (2.73)$$

If α_c is assumed to be zero. 2.72 can be reduced from three to two-dimensions.

$$v_c^n = \begin{bmatrix} V_c \cos(\beta_c) \\ V_c \sin(\beta_c) \\ 0 \end{bmatrix}, \quad (2.74)$$

$$u_c = V_c \cos(\beta_c - \phi), \quad (2.75)$$

$$v_c = V_c \sin(\beta_c - \phi), \quad (2.76)$$

Ocean currents velocities are arbitrarily changing. For this, a first-order Gauss-Markov process can be used to express this occurrence.

$$\dot{V}_c + \mu V_c = w, \quad (2.77)$$

(2.77) w represents the Gaussian white noise and $\mu \geq 0$ is a constant.

Guidance, navigation & control

To enable mapping capability for a snake robot using Underwater Hyperspectral Imager (UHI), a guidance system is needed to create path that overarches the area that is to be mapped. Fossen [10] defines guidance as a basic methodology to generate references for a desired position, velocity and attitude of a the vehicle which is to be controlled. Navigation can refer to either maneuvering a marine vessel or determining the position and orientation of a vehicle. The navigation system can by using the references draw trajectories for the craft to follow. Motion control, or often referred to as just control, is the process of having to determine the needed control moments and forces that have to be applied to the craft, in order to reach a desired position or state that fulfills a control objective. The following chapter aims to give more insight of how the UHI functions, and what is important to note when operating using this type of sensor. A summary over theoretical key points of guidance, navigation and control (GNC) will be given. Sensors will be shortly presented with their setup on simulation models.

3.1 Motion control systems

In order to achieve autonomy for underwater crafts, it is necessary to have systems in place that can eliminate the need of having pilots attending to the vehicles motions. Several methods need to be developed to handle the different aspects: guidance, navigation and control (GNC).

Guidance refers to the process of determining the desired trajectory or path that a craft is to follow. This can involve having to calculate the optimal route to a destination, or it could be determining the best path for achieving a specified objective, such as avoiding obstacles or maximizing efficiency of travel. To be able to perform guidance, the vehicle may use sensors such as sonar, cameras, LIDAR, IMU and pressure sensors, maps and algorithms to be able to gather information about its environment. The goal for guidance is to provide the craft with an accurate and clear path to follow.

Navigation, involves the ability of estimate the vehicle’s orientation and position in relation to the desired path from the guidance. This would involve using GPS (global positioning system), IMUs (inertial measurement unit) and other instruments with the ability of determining the crafts location, speed and heading. This information is then used to compare with the desired state to the estimate vehicle state.

Control, is the action of determining and applying the control forces needed to reach a set control objective. This can be to nullify external forces acting on the craft making it able to hold its position. Or it could be a combination, having the craft move along a predetermined path and having it follow it precisely by accounting for external forces that could potentially throw it off course. The controller of the vehicle requires a desired state for operation, which is provided by the guidance system. This ensures that the vehicle is able to navigate through its environment effectively. The output data from the navigation system are continuously used as feedback input for the motion control system, and sensor data from the guidance system is used to feed the controller, [10]. The subsequent control objective is to ensure that the craft stays upon its desired trajectory or path, and doesn’t deviate. This can be achieved through a combination of algorithms and sensors which operate with real-time feedback.

Together, these three aspects of GNC work together to enable that a motion control system is able to systematically guide a vehicle along a desired path, navigate to its destination, and make any necessary adjustments to its movement such to ensure a safe and efficient journey.

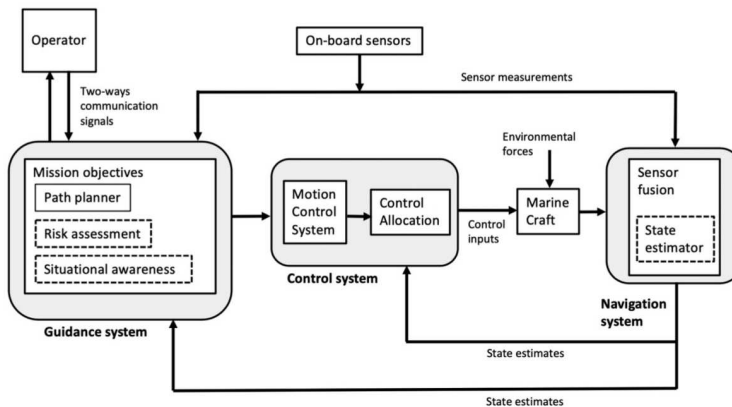


Figure 3.1: Guidance, Navigation and Control blocks presented in a Motion Control System, where the guidance system makes use of the measured position and velocity for creating a desired state reference model. Referred to as a closed-loop guidance system, [10].

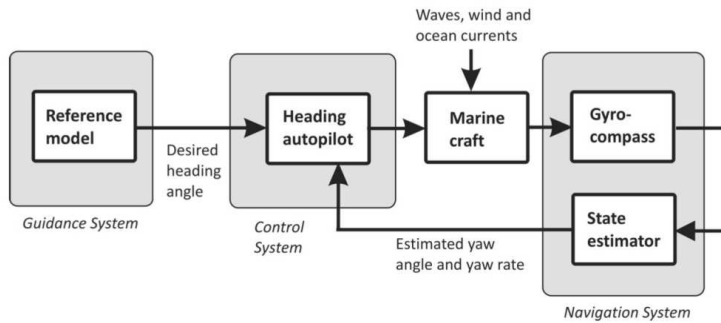


Figure 3.2: Simplified GNC system for an autopilot. In its most basic form, it consists of a reference model (Guidance system), an observer/gyro-compass (Navigation system) and an autopilot (Control system), [10].

3.2 Sensors

As this thesis contributes to non intervention vehicles, the following section will be describing strictly the primary sensors used when subsea monitoring and mapping are in the main scope of objectives. Four of the sensors which were installed onto the sensor platform of the vehicle are shown in figure 3.3.



(a) Doppler Velocity Log (DVL)



(b) Pressure gauge



(c) Stereo depth camera



(d) Hyperspectral Imager (HI)

Figure 3.3: A display of sensors used on UUVs

3.2.1 DVL

Doppler Velocity Log (DVL) is an acoustic sensor instrument. The principle behind its functionality is measuring the Doppler shift in frequency when moving towards or away from a target. By measuring the Doppler shift in an echo from the acoustic signal of a transducer, it can accurately calculate the velocities which can be used to estimate the distance traveled. The DVL commonly operates with four transducer heads, each creating a beam of sonic pulses. From each echo it can calculate the velocity vector of the vehicle with respect to the seafloor or the water column. It gives an output of 3-DOF velocity measurements u , v and w . Each beam's coverage is determined by the angle of each transducer head, [2].

3.2.2 Pressure Gauge

A pressure gauge, is a sensor used to measure the pressure of a liquid or a gas. In addition to monitoring the pressure inside the thrusters, a pressure gauge can be utilized to the its most common purpose, which is to give a pressure reading of the surrounding liquid it is submerged in. This gives valuable information as to if the vehicle is operating within safety limits.

3.2.3 Stereo Camera

A stereo camera is a type of camera which is organized as a setup of two sensors, as to capture images of the same scene but from slightly different angles. This allows the camera to calculate the distance to objects in the scene, a process which is known as stereo triangulation or stereoscopic vision. Within UUVs, such a camera is useful to help the robot avoid obstacles and navigate, as well as to provide accurate depth information for subsea mapping and inspection. Its functionality is more accurately described in section 4.1.3.

3.2.4 UHI

A Hyperspectral Imager (HI) is a camera, with capability of recording parts of the electromagnetic spectrum outside the range as well as the range of visible light. Each pixel in a traditional RGB camera has its image made up of only three spectral bands, being red, green and blue. Images capture by a HI camera has its pixels made up of several hundred different bands, [11].

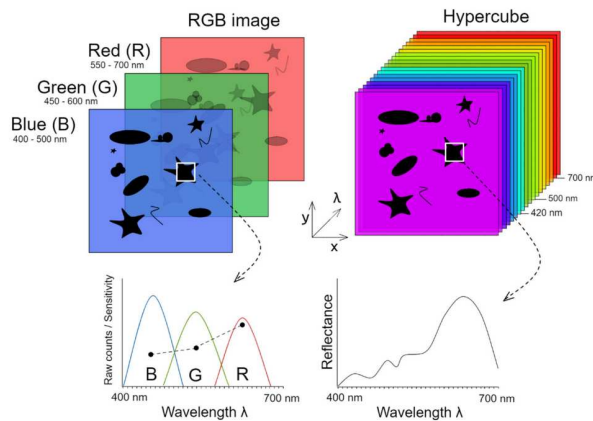


Figure 3.4: Amount of information captured when comparing traditional RGB images to Hyper-spectral image cube when inspecting a heterogeneous seafloor, [21]

Comparing the use of RGB sensors and HI. The "HI-Cube" instead of ranging between RGB values, is represented by (x, y, λ) . Where x and y are the spatial dimension and λ is the spectral dimension. Objects or any surface of interest have a spectral field of absorption and scatter light. From analysis of the spectral reflection, it is possible deduce the molecular and structural composition of the surface or object of interest.

In order to acquire high-quality visual data, several requirements must be met for the Unmanned Underwater Vehicle (UUV) that is carrying the camera. One important requirement is that the depth of the vehicle must be set correctly in order to avoid scattering and clouded vision. This is particularly important for deep sea mapping as the water can be cloudy and the visibility can be low.

Another important requirement is that the attitude or angle of the vehicle must be set according to the target surface plane. This is to ensure that the camera is able to capture images at the correct angle and orientation. The velocity of the vehicle must also be set according to the specifications of the HI sensor. This is to ensure that the camera is able to capture clear and sharp images.

The control system that is being developed in this thesis is focused on creating a system that can be used for mapping at depths where natural sunlight is no longer present. However, the system is not limited to this and can also be deployed to be used in coastal regions where the sun is still causes illumination. To capture images at deep depths, the camera needs to be accompanied by an artificial light source. The use of artificial light is important as the natural light is not present and the image quality can be affected. As this thesis focuses on the motion control system, only the most important requirements will be presented and discussed in order to determine the desired position of the UUV.

Artificial Lighting

Visual sensors are highly dependant on having an artificial light source to be able to capture imagery. At a depth of fifty meters, there is often presence of sunlight, as it is still within the Euphotic layer. However the intensity is reduced by more than 80% and the spectral bands present are limited to only low wavelengths with blue and green hue, [37].

A field study conducted in the Peruvian basin [7], explained how the placement of the artificial light sources determined the quality of the image data. The ROV used had a configuration of ten light sources. These were mounted above the camera on the main-frame of the ROV. This configuration led to shadows that resulted in noise on the captured UHI data. Optimally suggested by [7] a configuration of two light sources beside the UHI would be ideal for the imagery to give the best possible result.

In the study [17], several visible light sources can be used as part of the UHI system such as halogen lamps, LED's and lasers. The light source should cover a broad spectrum (400-700nm or more) and have enough intensity to illuminate the object of interest. The altitude of the ROV from the seafloor varies from 1 [m] to 5 [m]. The light source should be matching this. Uniformity and stable lightning is necessary to avoid any degrading effects on the image quality. From [7], problems related to placement showed the importance of having lights placed in such a way as to compliment the camera. Shadow casting caused by a wrong lighting configuration, can lead to the materials not reflecting light with enough intensity to be captured by the camera.

However, the need for high intensity illumination, and sometimes multiple light sources, adds another restriction to the operational time of the AUV as this is directly connected to the battery capacity.

Desired altitude and area of coverage

The altitude that is preset as a desired state, should be determined on the basis of the mission objectives. High altitudes risk having a reduction of detail while adding scattering effects due to light having to travel further to illuminate the target. The artificial light must fully penetrate the water column for images to be captured. Operating at low altitudes, results in a slow mapping rate, as the area that the sensor is covering is reduced. This results in making it necessary for the underwater vehicle to do multiple passes, that could have been covered if the altitude was optimized. Having to do multiple passes adds an excessive amount of time spent on completing the planned mission, and it also causes additional battery power consumption.

In the review of UHI [21], an overview is presented for the current UHI systems that have been used for data acquisition. It includes the performance recommendations of each platform. This is presented in table 3.1.

Table 3.1: Overview of current UHI systems for data acquisition including their seafloor mapping capabilities with tradeoffs, [21].

System/ Platform	Achieved Transect Length [m]	Possible Survey Area per Deployment [m ²]	Spatial Resolution Achieved [cm/pixel]	Distance to Target [m]	Deployment Depth [m]
Underwater Rail	1 – 5	10	0.1	1	5 – 20
AUV	Not Defined	1×10^9	0.6	8.5	2300
ROV	1 – 20	< 500	0.1	1	30 – 4000
USV	1 – 20	< 500	0.5	1.5	Surface
Under-ice slider	10 – 30	< 40	0.1	1.2	1.5

The AUV data that table 3.1 presents is derived from the conducted survey [34]. The altitude range was recorded to be in the region of 6.5[m] to 9.1[m], with their average being at 8.5 [m]. These values were due to unknown bathymetry of the seabed. An extra safety margin was added to reduce risk of collision. At this average altitude with a 40 degree field of view, the cross-track coverage was estimated to be 6.2[m] when the camera was faced directly towards the surface. This configuration gave excellent results where the captured images cross-track were made up of 960 pixels, with each pixel having a spatial resolution of 0.64cm. The parameters from the study [34] were used as a baseline, of the system development presented in chapter 4.

Mapping velocity

There are a few key factors to consider when choosing velocity of the craft which is intended to be used for deep sea mapping. The resolution of the mapping sensors must be considered. The UUV must move slowly enough as to allow the sensors to capture the information of the mission objective. As well as considering the velocity in surge, it is important as to the speeds in heave motion. Maintaining a constant distance and only make adjustments when needed as to obtain the desired quality of sensor data. It should be able to avoid collisions with obstacles and operate smoothly when moving across the seafloor. This may require a lower velocity being set, especially for areas that consist of complex topography or high density of obstacles. The velocity reference in [34] was suggested to be set according to the image capture frequency of the camera. As the HI camera used a capturing frequency of 10 [Hz], the velocity reference was chosen to be set to 1.8 [m/s]. This was such that the along track spatial resolution was set to 18 [cm] per pixel. The resolution can be altered by using different speeds.

3.3 Setup of Sensors on Platform

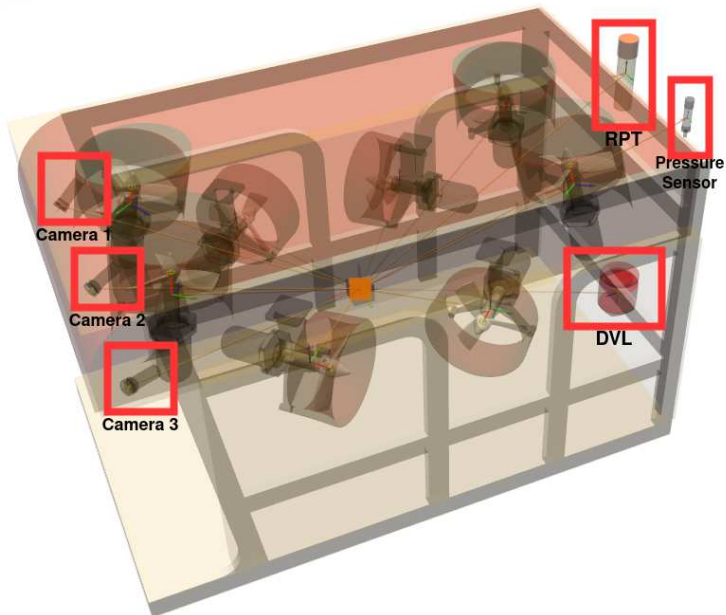


Figure 3.5: Illustration of the sensor placements onboard the RexROV

Figure 3.5 illustrates where the different sensors are positioning according to the main frame. The model indicates that there are three cameras available, however only camera 1 and camera 3 were used when creating the stereo vision module. The ROV Positioning Transponder (RPT) and pressure sensor was part of the default setup of the model, but was not used in the altitude control motion system.

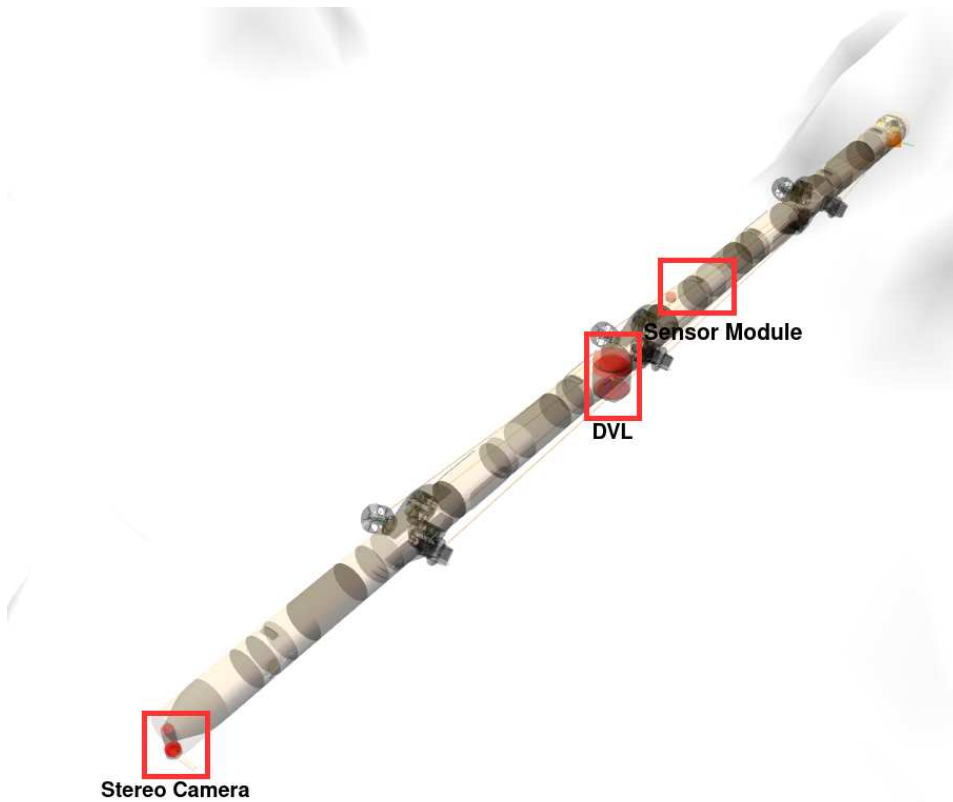


Figure 3.6: Illustration of the sensor placements onboard the Eelume

Figure 3.6 shows the primary sensors that were added to the USR model EELY500. From the illustration it is shown that the stereo camera was placed at the bow and DVL in mid ship. By default, the model comes with a sensor module that can accompany a multitude of sensors that can be activated. Only DVL and the stereo camera was used when running simulations with this model.

Development of a Motion Control for Constant Altitude

The following chapter presents the implemented correcting altitude motion control system as a whole. The first section shows the mathematical concept of estimating the altitude, when using a doppler velocity log (DVL) and when using a stereo depth camera. After, a presentation of the system components and their functionality will be described.

4.1 Estimating altitude

To create an altitude reference model, it was determined that applying a fusion of the measurement data from the DVL and stereo camera would be to prefer. Even though both provide an altitude estimation, their working principles are very different and they can complement each other in doing so. While a camera may be obstructed due to water effects and having to operated at closer distances, DVL can be used at larger distances. In the event of if one of the sensors would fail to supply a measurement, the system would still be able to function.

4.1.1 Altitude Approximation by use of a DVL

DVL was to be added into the system to enhance the range of the altitude measurements. While visual sensors are usually limited to give readings at altitudes of a few meters, DVL can operate at ranges of many tens of meters. For the implemented system, the range of the DVL was chosen to have a max measurement distance of 81 [m]. From [6], it was suggested to use *least squares approximation* (LSA) when determining the local distance between the sensor platform and the seafloor. When using basic linear approximations, only three points of measurement are needed to be able to accurately determine the distance. This can be described as in function (4.1).

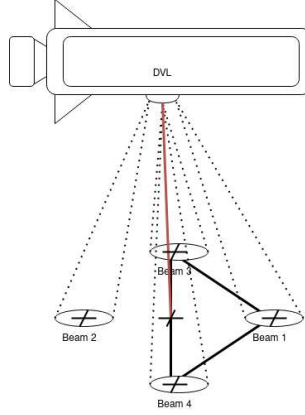


Figure 4.1: DVL beam layout. In red, showing the linear approximation of the distance between the sensor bearing platform and the seafloor.

$$F(x, y, z) = f_j(x, y) - z_j^{CO} = 0, \quad (4.1)$$

$$f_j(x, y) = z_j^{CO} = a_j + b_j x + c_j y, \quad (4.2)$$

x and y represent the position in the two-dimensional plane, where each lobe intersects the seabed. The intersection point between the lobe and the plane is referred to as r_j^n , where n represents $\{n\}$ frame and j is the beam number.

$$r_j^n = \begin{bmatrix} x_j^n \\ y_j^n \\ a_j^n \end{bmatrix}, \quad j = 1, 2, 3, 4 \quad (4.3)$$

$$\begin{bmatrix} 1 & x_j^n & y_j^n \\ 1 & x_{j+1}^n & y_{j+1}^n \\ 1 & x_{j+2}^n & y_{j+2}^n \end{bmatrix} \begin{bmatrix} a_j \\ b_j \\ c_j \end{bmatrix} = \begin{bmatrix} a_j^n \\ a_{j+1}^n \\ a_{j+2}^n \end{bmatrix} \quad (4.4)$$

$$\sum_{j=1}^4 [a_j^n - (a + bx_j^n + cy_j^n)]^2, \quad (4.5)$$

The coefficients a_j , b_j and c_j , are found from solving for the three DVL range measurements. This method can be expanded to include solving for four measurements. This can be calculated by using different combinations of the three out of four beams. The LSA method does provide the best linear approximation when aiming to minimize (4.5). Using approximation methods has unfortunately some drawbacks, as it adds uncertainty to the

system. The altitude which is determined from it, could be greater than the true altitude. In the case of the approximation giving a larger estimate of the altitude than the true. Could add further risk of collision than if it was the other way around.

The system of four beams is represented as $\mathbf{Ax} = \mathbf{b}$ in (4.6).

$$\begin{bmatrix} 1 & x_1^n & y_1^n \\ 1 & x_2^n & y_2^n \\ 1 & x_3^n & y_3^n \\ 1 & x_4^n & y_4^n \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} a_1^n \\ a_2^n \\ a_3^n \\ a_4^n \end{bmatrix}, \quad (4.6)$$

where $x = [a, b, c]^T$, by solving the system $\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}$ to get the minimum altitude from (4.5). The LSA of the seafloor plane is given by (4.2). The altitude approximations is seen in 4.1 as the red orthogonal line.

4.1.2 Transformation of DVL beams

Figure 4.2 indicates how one DVL beam can be decomposed in the local DVL fixed coordinate frame $\{d\}$.

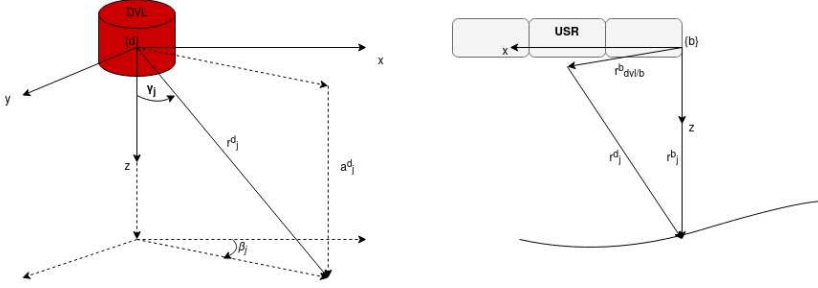


Figure 4.2: DVL beam vector j^{th} components illustration (left). DVL vector shown in body frame (right).

The left part of figure 4.2 shows γ_j as the angle of number j beam (r_j^d), measured from the DVL z-axis. β_j represents the angle between the DVL x-axis and the beam. a_j^d is the vertical component of r_j^d . The r_j^d as well as the constant angles γ_j and β_j , are given as outputs from the DVL. The vector which includes all four altitude measurements is represented as a^d .

$$r_j^d = \begin{bmatrix} x_j^d \\ y_j^d \\ a_j^d \end{bmatrix} = a_j^d \begin{bmatrix} \tan(\gamma_j) \cos(\beta_j) \\ \tan(\gamma_j) \sin(\beta_j) \\ 1 \end{bmatrix}, \quad (4.7)$$

Terms from equation 4.7 are

- γ_j angle of beam j has from $\{d\}$ z-axis.
- β_j angle from $\{d\}$ x-axis.
- a_j^d vertical component of r_j^d , the measured altitude of the beam.

The component r_j^d is transformed and shifted to $\{b\}$ in (4.8), then transformed to $\{n\}$ in (4.9).

$$r_j^b = R_d^b(\Theta_{bd})r_j^d + r_{dvl/b}^b, \quad (4.8)$$

$$r_j^n = R_b^n(\Theta_{nb})r_j^b, \quad (4.9)$$

From (4.8) and (4.9), Θ_{bd} is a vector of Euler angles and representing the orientation of $\{d\}$ relative to $\{b\}$. After transformation, the resulting vector from the USR center of origin to the seabed in $\{n\}$ is given as (4.3).

4.1.3 Altitude approximation by use of a stereo depth camera

The altitude from the stereo camera setup, was determined from having the system create a point cloud. Each pixel in the cloud includes positional data (x, y), depth (z) and intensity (i). A pixel can be described as $\{x, y, z, i\}$.

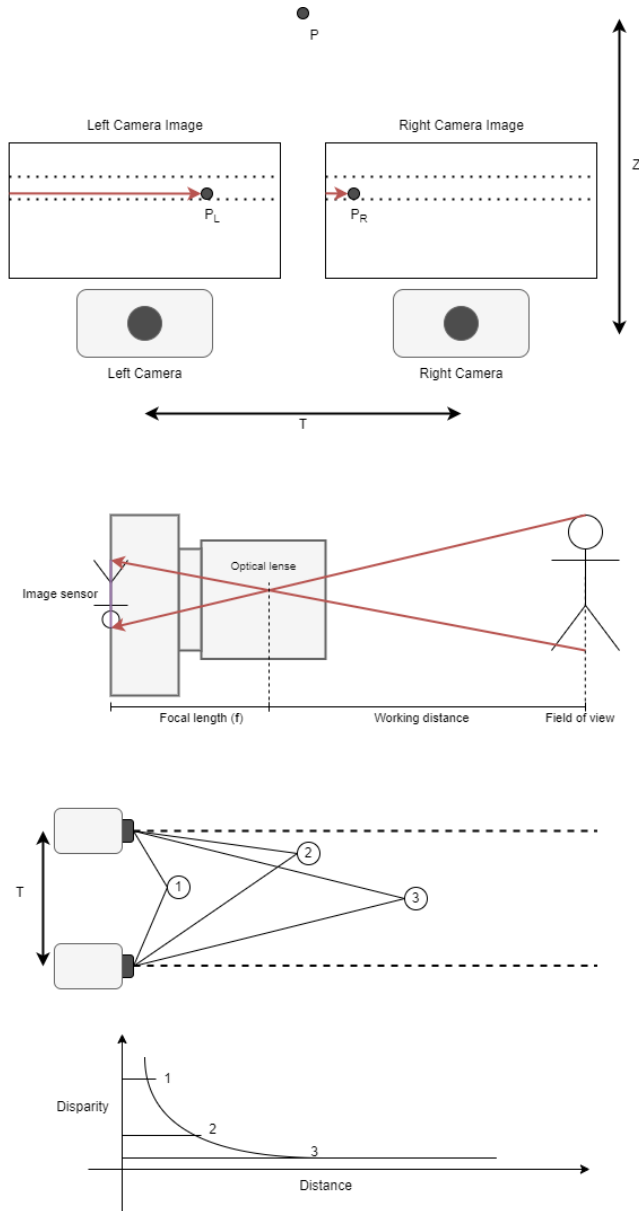


Figure 4.3: Illustration for depth estimation using parallel stereo camera system

From figure 4.3, target point P represents a physical point. P_L and P_R represent the same point P in left and right camera image, and can be described as (x_L, y_L) and (x_R, y_R) . The horizontal pixel distance of point P_L in the left camera image is noted as $n1$, and P_R for the right image, noted as $n2$. T is the baseline distance between the center of the left and right camera. Z is the distance between point P and baseline center of the cameras. Describing the working distance between the camera and target field of view. f is the focal length of the cameras, which is the distance between the optical lens and the image sensor.

$$D = n1 - n2 = x_L - x_R, \quad (4.10)$$

Equation 4.10 gives the disparity between the two cameras. Disparity grows as the target point is closer to the baseline of the camera.

$$Z = \frac{f}{d} \cdot \frac{T}{D}, \quad (4.11)$$

The altitude estimate is done by comparing each altitude (z) tuple in the point cloud to find the shortest orthogonal distance between the seafloor and the platform.

4.2 System Modules

The system components of the proposed correcting altitude motion control system is illustrated in figure 4.4. It shows a simplified overview over the primary modules that the system relies upon to operate as is. The basic components in this system are signal processing, observer, controller and thrust allocation. The illustration is meant to give an insight of the inner workings of the infrastructure of the system. The configuration file shown to the right, is defined by an operator who can adjust this according to best suit the mission.

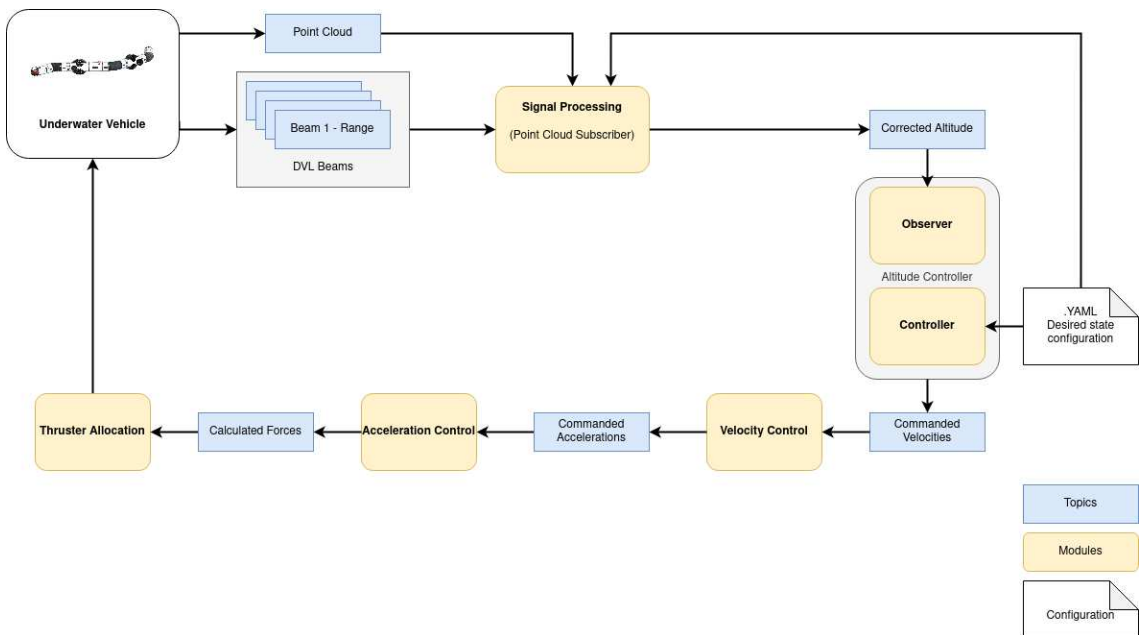


Figure 4.4: Illustration of the architecture for the proposed altitude control system

4.2.1 Signal Processing

The signal processing module's main task, is to acquire and validate the quality of all sensor data before handing it over to the observer. In this process, sensors are checked for wrongful readings such as wildpoints and dead readings. Wildpoints are points in the sensor signal where an amplitude occurs as a discrepancy and is non repetitive, meaning that it is probably a fault. Dead readings, are defined as periods where the signal does not read any new measurements, but is flatlining at the last read value of the signal. In these cases the data sample is disregarded as depending of the probability of them being wrong. When operating within a simulation, the default scenario is that everything works flawlessly. This is far from reality, where systems are surrounded by external interferences that may cause wrong instrument readings and faults. To try to add a layer of realism to the simulator, white noise was added using a Gaussian distribution that created a deviation

on each of the sensors signals. Wildpoints were artificially created by adding amplitudes to the distributions which allowed them to occur at random during operation.

Adding noise and discrepancies to the signal, helps to replicate and tackle issues that are probably to be encountered in the physical world. The observer becomes a necessity to keep the system from reacting to faults occurring from noise and artificial sensor faults.

4.2.2 Observer

The objective of the observer is be able to process the outputted measurements from the signal processing module, and give a smooth and accurate output of the state position and velocity. Sensors can vary in measurement update rates, and it may be tasked to the observer if it is configured to do so. The observer can also fuse several signal data sources into one single state, referred to as sensor fusion. The sensors applied to the implemented system, operated having their update rates within 5-10 [Hz]. The observer model chosen for the system was a single-dimension linear Kalman filter. This added a solid way of handling the problem with multiple measurement readings and fusing them together to one state. By using probability of how reliable each reading is, it is capable of minimizing measurement noise, and producing a smooth output.

Linear Kalman Filter

The problem which the implemented Kalman filter was seeking to solve, was being able to estimate continuously from a set of measurement parameters. Updating of the estimation is achieved by continually combining sets of observations, $\mathbf{z}(t)$ which contain the information about the signal of interest, $\mathbf{x}(t)$. A choice was made as to using either a discrete Kalman filter or a dynamic Kalman filter. For the system implementation a discrete one was chosen, when considering that the thesis had a short development window and most of the system dynamics were simplified as to being constant over time. In a discrete Kalman filter, the state transition- and measurement-model are fixed. The filter algorithm uses these models to manage prediction of the system's next time step state, based upon the current state and input measurements.

The state prediction equation is given by:

$$\hat{\mathbf{x}}_{n+1,n} = \mathbf{F}\hat{\mathbf{x}}_{n,n} + \mathbf{G}\mathbf{u}_n + \mathbf{w}_n, \quad (4.12)$$

Where the terms from (4.12) are

$\hat{\mathbf{x}}_{n+1,n}$	predicted system state (altitude) vector at time step $n + 1$.
$\hat{\mathbf{x}}_{n,n}$	estimated system state (altitude) vector at time step n .
\mathbf{u}_n	control variable.
\mathbf{w}_n	process noise.
\mathbf{F}	state transition matrix.
\mathbf{G}	control matrix.

The system state x_n was set to:

$$\mathbf{x}_n = \begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix} \quad (4.13)$$

The state prediction for the next time step $n + 1$ can be described in the system of equations:

$$\begin{cases} \hat{x}_{n+1,n} = \hat{x}_{n,n} + \hat{\dot{x}}_{n,n} \Delta t + \frac{1}{2} \hat{\ddot{x}}_{n,n} \Delta t^2 \\ \hat{\dot{x}}_{n+1,n} = \hat{\dot{x}}_{n,n} + \hat{\ddot{x}}_{n,n} \Delta t \end{cases} \quad (4.14)$$

Δt is the time between successive measurement.

Covariance extrapolation equation is given by:

$$\mathbf{P}_{n+1,n} = \mathbf{F} \mathbf{P}_{n,n} \mathbf{F}^T + \mathbf{Q}, \quad (4.15)$$

Where the terms from equation 4.15 are:

- $\mathbf{P}_{n,n}$ uncertainty (covariance) matrix, that of the current state estimation.
- $\mathbf{P}_{n+1,n}$ uncertainty (covariance) matrix, that of the next state estimation (prediction).
- \mathbf{Q} process noise matrix.

Choosing the uncertainty (covariance) matrix can be a complex task, as it depends on the specific characteristics of the sensor measurements and the environment. The uncertainty of the sensors was not easily determined and an arbitrary value for the covariance matrix was chosen.

The generalized measurement equation in matrix form is given by:

$$\mathbf{z}_n = \mathbf{H} \mathbf{x}_n + \mathbf{v}_n, \quad (4.16)$$

Where the terms from equation 4.16 are:

- \mathbf{z}_n measurement vector.
- \mathbf{x}_n the true system state.
- \mathbf{v}_n random noise vector.
- \mathbf{H} observation matrix.

\mathbf{z}_n vector contains the measured sensor data, which are given by the related sensors.

$$\mathbf{z}_n = \begin{bmatrix} x_{n,\text{measured stereo}} \\ x_{n,\text{measured dvl}} \end{bmatrix} \quad (4.17)$$

$x_{n,\text{measured stereo}}$ expresses the scalar altitude value of the stereo camera measurement. This was determined by calculating the shortest distance among points in the PointCloud and the sensor.

The measurement covariance matrix is:

$$\mathbf{R}_n = \begin{bmatrix} \sigma_{x_{m,\text{stereo}}}^2 & 0 \\ 0 & \sigma_{x_{m,\text{dvl}}}^2 \end{bmatrix} \quad (4.18)$$

The measurement covariance matrix \mathbf{R} represents the uncertainty of sensor measurements, including the signal-to-noise ratio (SNR). The diagonal elements of the matrix represent the variances of the sensor measurements, which can be used to determine the SNR by dividing the variance of the signal by the variance of the noise. The variance of the signal can be estimated from the signal frequency and the variance of the noise can be estimated from the angle between the sensors. Additionally, The SNR is a measure of the ratio of the signal power to the noise power, it is used to quantify the quality of a sensor measurement, the signal frequency and angle between sensors can both affect the SNR. A higher signal frequency leads to a lower SNR because at high frequencies, the sensor may be more sensitive to noise and the noise power relative to the signal power increases. Similarly, a larger angle between the sensors leads to lower SNR because the sensor measurements may be more affected by noise and the noise power relative to the signal power increases.

The Kalman gain in matrix form is given by:

$$\mathbf{K}_n = \mathbf{P}_{n,n-1} \mathbf{H}^T (\mathbf{H} \mathbf{P}_{n,n-1} \mathbf{H}^T + \mathbf{R}_n)^{-1} \quad (4.19)$$

Where the terms from equation 4.19 are:

- \mathbf{K}_n Kalman Gain.
- $\mathbf{P}_{n,n-1}$ previous estimation of the uncertainty matrix (covariance) of the current state (prediction of the previous state).
- \mathbf{R}_n Measurement uncertainty (measurement noise covariance matrix).

The State Update equation in matrix form is given as:

$$\hat{\mathbf{x}}_{n,n} = \hat{\mathbf{x}}_{n,n-1} + \mathbf{K}_n (\mathbf{z}_n - \mathbf{H} \hat{\mathbf{x}}_{n,n-1}) \quad (4.20)$$

The Covariance Update equation is given as:

$$\mathbf{P}_{n,n} = (\mathbf{I} - \mathbf{K}_n \mathbf{H}) \mathbf{P}_{n,n-1} (\mathbf{I} - \mathbf{K}_n \mathbf{H})^T + \mathbf{K}_n \mathbf{R}_n \mathbf{K}_n^T, \quad (4.21)$$

Elongated Bodies

Algorithm 1, was aimed to solve the problem of maintaining a minimum altitude for elongated-bodied UUVs, such as torpedoes or gliders. By continuously measuring and recording the UUV's altitude, the algorithm can determine the lowest altitude reached by the UUV and publish this information. This information could be used by the UUV's control system to ensure that the UUV does not descend below a certain altitude, for example to avoid collisions with the seafloor or other obstacles. Additionally, by publishing the lowest altitude, the algorithm could provide other systems with information about the UUV's altitude.

Elongated bodies are here defined as UUVs with bodies of significant length, body lengths that exceed the length of their sensors area of coverage. The formulated algorithm was applied to both the DVL sensor module and stereo-camera sensor module of the system.

Algorithm 1 Publisher for minimal altitude for elongated bodied UUVs

Require: f_s , $gridList = [-]$, $lowestAltitude$
while *Activated* **do**
 $gridList[] \leftarrow measuredAltitude$
 if $length(gridList)$ is equal f_s **then**
 $lowestAltitude \leftarrow gridList[0]$
 for i in $length(gridList)$ **do**
 if $lowestAltitude \geq gridList[i]$ **then**
 $lowestAltitude \leftarrow gridList[i]$
 end if
 end for
 delete $gridList[length(gridList) - 1]$
 publish $lowestAltitude$
 end if
end while

This algorithm describes a publisher for determining the minimal altitude for elongated-bodied unmanned underwater vehicles (UUVs). The algorithm takes as input the sampling frequency f_s , a list of measured altitudes $gridList$, and a variable for storing the lowest altitude $lowestAltitude$.

The algorithm then enters a loop that runs as long as the publisher is activated. Inside the loop, the algorithm adds the current measured altitude to the list of altitudes that have been filtered from each grid. If the length of the list is equal to the sampling frequency f_s , the algorithm calculates the lowest altitude by iterating through the list and finding the minimum value. The algorithm then removes the oldest measurement from the list and publishes the lowest altitude. The loop continues until the publisher is deactivated.

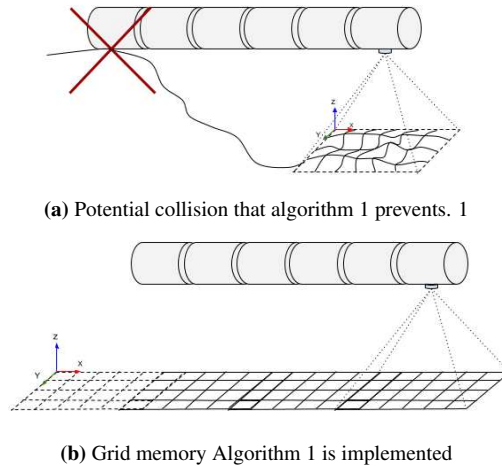


Figure 4.5: Illustration of the Elongated body problem with suggested method of solution

Figure 4.5 shows a practical example of an unwanted situation that could occur. The first figure having the system performing a controlled descent and colliding with terrain as it does not have a complete view of the topography outside its sensor view. The second actively using the suggested algorithm to avoid descending upon an obstacle or uneven terrain formation.

4.2.3 Controller

In the implemented system, the controller module receives the state input from the guidance system. The state error is calculated as the difference between desired and the estimated state, for this system this implies only the altitude state as this is the state that is to be controlled. The output τ is derived from the state error and feedback of the desired state. The control law which was chosen to calculate the heave state velocity, was a Proportional Integral Derivative controller (PID). Algorithm 2 presents a pseudo formulation of the algorithm that was implemented.

Algorithm 2 1-Dimensional PID for velocity in heave

Require: $k_p, k_i, k_d, sat, integral, err, prev_err, t, prev_t, derr_dt$
 $dt = t - prev_t$
if $prev_t > 0.0$ and $dt < 0.0$ **then**
 $derr_dt = (err - prev_err)/dt$
 $integral += 0.5 * (err + prev_err)dt$
end if
 $u = k_p err + k_d derr_dt + k_i integral$
 $prev_err = err$
 $prev_t = t$
if $normalize_matrix(u) > sat$ **then**
 $u = sat * u / normalize_matrix(u)$
end if
return u

4.2.4 Thruster Allocation

The thrust allocation module uses a mapping method to relate the revolution speed of the propellers to the thrust force, as described in [12] chapter 2.3.5. This method remains unchanged throughout the course of the project. The thrust force for each propeller is calculated according to the direction of its thrust vector. The thrust allocation module receives input about the desired motion commands and the current state of the vehicle, and uses this information to calculate the appropriate thrust commands for each thruster. These thrust commands are then sent to the thrusters, which apply the required force to produce the desired motion. The thrust allocation module plays a critical role in enabling a vehicle to move accurately and precisely in response to control commands.

4.3 Development Software

The following section will cover the software used in developing an altitude motion control system, including testing in a simulated underwater environment. The software which was chosen was recommended by supervisors, being a well tested and known software.

4.3.1 ROS2

Robot operating system is a powerful tool which is used for modeling and controlling mechatronic systems (robots), [26, 27]. In the context of marine technology, this is commonly referred to UUVs, such as ROVs and AUVs. It has a flexible and extensible framework which allows developers to create and manage complex systems without much difficulty. The systems in ROS2 are commonly comprised of several interconnected software modules. ROS2 provides a wide range of libraries and tools for working with sensor data, as in this context performing motion control, or setting up high level behaviours such as planning and decision making.

One of the key benefits of using ROS2 for modeling ROVs, is its support of allowing rapid prototyping and testing. The modular architecture of ROS2 allows developers to quickly swap between different modules and allow for experimentation of different control algorithms and behaviors. This makes it possible to quickly iterate on designs and find the best solution for a particular application.

In addition, the use of ROS2 enables UUVs to take advantage of the large and growing ecosystem of software packages and libraries that are available for ROS. This allows developers to leverage existing solutions to common problems, reducing the development time and effort required to create a new ROV system. It also makes it easier to integrate new sensors and other hardware into a UUV, as most common sensors and actuators already have ROS2 drivers available.

To understand the coding and structure of the developed system in this thesis, it is helpful to have a basic understanding of how the ROS2 framework manages its infrastructure.

Publishing and subscribing using Nodes

ROS2 uses nodes to handle communication and data exchange within the framework. This helps to divide complex systems into smaller, more manageable parts, which is very useful for large and complex robotic systems. Publishers can provide data streams to other nodes through topics. Other nodes can then subscribe to these topics to access the data streams. Each node can be thought of as an individual process that can connect and exchange data with other nodes as needed, forming a network of interconnected processes. This modular approach makes it possible to easily add, remove, or modify individual nodes without affecting the overall system.

4.3.2 Gazebo

Gazebo is a versatile robot simulator which is used both in industry and academia, [28]. It provides a physics-based environment, in which it can test and evaluate robotic systems. Gazebo can simulate the motion and interactions of a robot and its environment. This includes things like gravity, friction and collisions. It can also be used to help generate realistic sensor data, such as images from a camera, to allow the robot to perceive and navigate its surroundings.

One of the most significant advantages of using Gazebo is that it allows developers to be able to test their robot designs fast and easy in a realistic environment. This can be especially useful for identifying and addressing potential problems in an early stage of the design process, when it is still relatively easy and inexpensive to make changes. Gazebo also provides also tools for visualization and analyzing results of simulations, which can be useful for understanding the behavior of a robot and identifying areas for improvement. Another clear advantage of using Gazebo is, that it is designed to work seamlessly with the ROS and ROS2 frameworks. This means that developers can use the same techniques and tools for building and controlling a robotic system in Gazebo, as they would use with

a real-world robot. This makes it possible to test and evaluate a system in Gazebo before implementing it on real hardware, reducing the time and effort required to develop a working system.

4.3.3 Plankton

Plankton is a ported version of the Unmanned Underwater Vehicle Simulator (UUV), from ROS1 to ROS2. It is built upon Gazebo and allows users to test and create underwater mechatronic systems in a virtual environment, [16]. It uses a physics engine and advanced graphics to provide a realistic representation of a ocean world, including the effects of buoyancy, drag, and other hydrodynamic forces on underwater vehicles.

It provides a set of plugins and tools for simulating the motion and control of underwater vehicles. The UUV code can be easily customized to suit a user's specific needs for different robotic systems, such as thruster allocation, sensor configuration, and control algorithms. Plankton allows users to use the UUV simulator under the ROS2 framework.

Plankton also includes a number of other features that make it a powerful tool for simulating underwater systems. These include support for sensors such as cameras and DVL, the ability to model complex underwater environments, and tools for analyzing and visualizing the results of simulations. The API is given in [31].

4.4 Sensor Carrying Platform Model

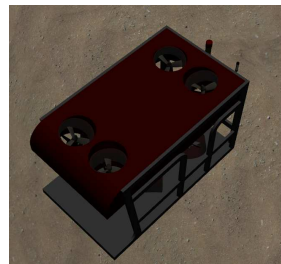
This thesis used previous developed models from [12] (USR - Eelume / EELY500) and [19] (RexROV). Their specifications are summarized in the this section.

4.4.1 RexROV

RexROV is the default model which is included in the UUVs simulator package. It consists of a 4-DOF ROV vessel, that operates using eight fixed angled thrusters. The model is highly configurable, allowing a multitude of sensors to be attached.

Description	Parameter
Depth rating	700 [m]
Robot dry weight	1862.87 [kg]
Robot weight when deployed	positive
Length	2560 [mm]
Width	1500 [mm]
Height	400 [mm]
Thruster count	8

(a) RexROV Specifications.



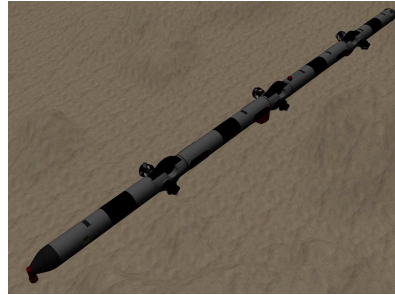
(b) RexRov in Plankton, Gazebo.

Figure 4.6: RexROV model details

4.4.2 Eelume / EELY500

The EELY500 model is based upon NTNU's research version of the USR Eelume. It was developed in association with a master thesis, [12]. The model was based upon the mainframe of the previous NTNU ROV model, Minerva. The joints are in a fixed state, meaning they are locked in place when imported into the Plankton. The default shape which it was configured for was a torpedo shape like configuration. To alter the body to any other setup, the model needs to have its links and joints undergo a homogeneous transformation as in 2.2.1, to have them placed relative to the global frame.

Description	Parameter
Depth rating	500 [m]
Robot dry weight	188 [kg]
Robot weight when deployed	Neutral
Length	6182 [mm]
Ø (without thrusters)	200 [mm]
Links	5
Joints	4
Thruster count	12



(a) Eelume / EELY500 Specifications.

(b) EELY500 in Plankton, Gazebo.

Figure 4.7: RexROV model details

To reduce complexity, the USR model had a few alterations made. These differ from the full scale version. The real robot would be slightly positive buoyant. This was to ensure that at any point of a critical failure, that could cause the propulsion system to stop, the USR would float to the surface and be ready for recovery. On the flip side of this, more energy is needed to create enough thrust to keep it neutral. As for in simulations, the loss of the craft is a non issue as it is always able to be located within the Plankton environment. The USR model is set to be neutral buoyant. Simplifications towards the hydrodynamic forces were panned out by neglecting the thrusters, when the robot was in torpedo shape. This was simplified to being a submerged cylinder, see appendix A.

4.5 Simulation Cases

The purpose of this study was to investigate the behavior of RexROV and EELY500 in a simulated environment with varying degrees of inclined surfaces, specifically examining the performance of the altitude motion control system. To do this, the models were run through a selected section with mountains in the Plankton simulator. During the simulation, the EELY500 model was constrained to using only a simple body shape, as a torpedo, and both models were restricted to only have motions in 4-DOF, with pitch and roll neglected due to the models using restoring moments. Both models were configured as naturally buoyant. The results of the simulation will be used to understand the performance of the altitude motion control system in different terrain types.

Figure 4.8 illustrates the contour of the mountain range for both the EELY500 and the RexROV. The altitude motion control system as inactive and the models were kept at a constant depth, moving only in surge.

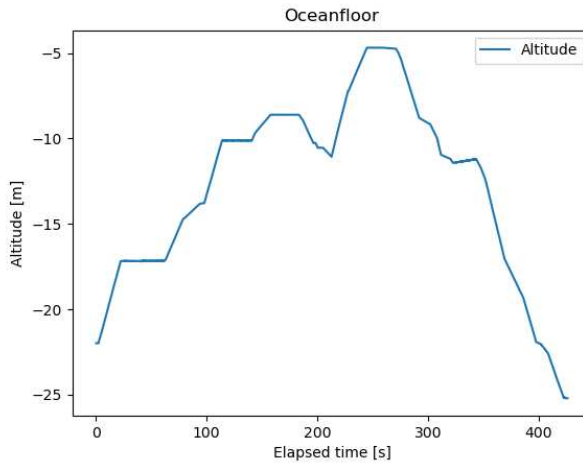
In the simulated scenarios, there were no external forces acting on the robot within the simulator. As a result, the integral term in the PID controller was set to zero, as there was no error to offset. This effectively reduced the PID controller to a PD (proportional-derivative) controller.

Table 4.1: Simulation Case A - Parameters

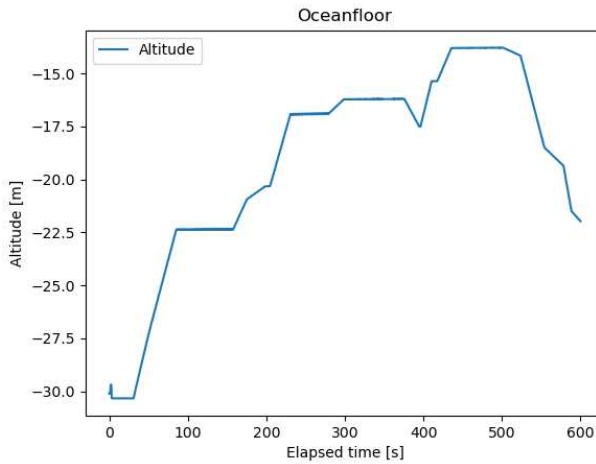
Parameters	RexROV	EELY500
Surge velocity	0.3 [m/s]	0.3 [m/s]
Max Heave velocity	0.5 [m/s]	0.5 [m/s]
Runtime	400 [s]	600 [s]
Measurement sensors	10 [Hz] - Stereo camera	10 [Hz] - Stereo camera
Noise (σ)	None	None
Desired altitude	8.5 [m]	8.5 [m]

Table 4.2: Simulation Case B - Parameters

Parameters	RexROV	EELY500
Surge velocity	0.3 [m/s]	0.3 [m/s]
Max Heave velocity	0.5 [m/s]	0.5 [m/s]
Runtime	400 [s]	600 [s]
Measurement sensors	10 [Hz] - DVL, Stereo camera	10 [Hz] - DVL, Stereo camera
Noise (σ)	0.005	0.005
Desired altitude	8.5 [m]	8.5 [m]



(a) Terrain variation of mountain range in simulation cases using the RexROV, with constant depth maintained and altitude motion control system inactive.



(b) Terrain variation of mountain range in simulation cases using the EELY500, with constant depth maintained and altitude motion control system inactive.

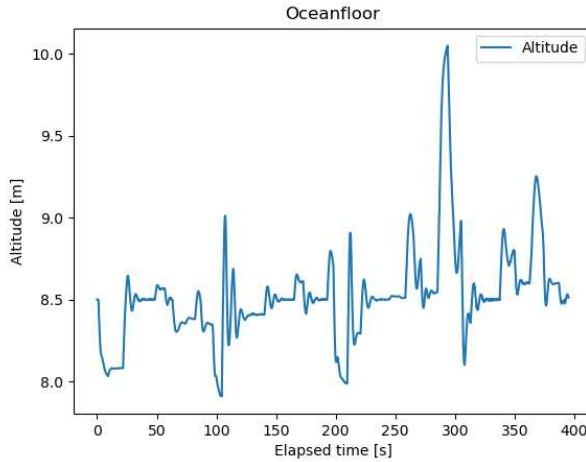
Figure 4.8: The topography of the seafloor captured by the RexROV and EELY500.

4.5.1 Case A1 - Altitude Proportional Control RexROV

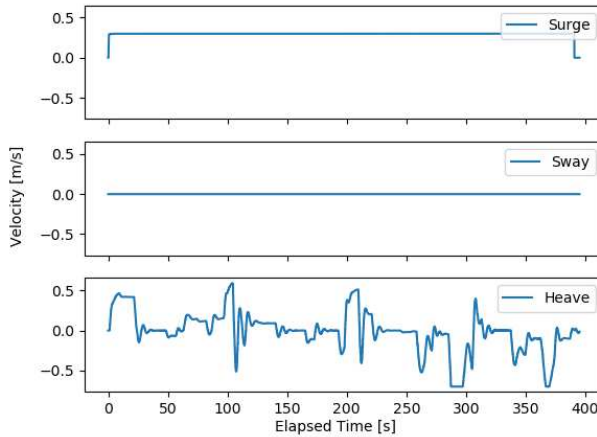
”In this initial simulation, a basic proportional controller was used to control the up-and-down (heave) motion of the altitude control system. As the development was still in early stages, the Kalman filter had not been implemented yet. The system was designed to rely solely on a stereo-vision depth camera for altitude measurements and no noise was added. A constant maximum velocity in the surge direction was set and the altitude controller was activated. The first set of simulations were run to check if the system was performing as expected and using a simple controller and one sensor made it easy to test the whole system.”

The results of the altitude control system showed that the amplitude of the variations ranged from -0.5 [m] to 1.5 [m]. This can be seen in the graph in Figure 4.9a. These amplitude variations were due to the ROV moving over surfaces, as can be seen when comparing the data in Figure 4.9a to the topographic map in Figure 4.8a. The ROV’s altitude control system struggled to maintain a stable altitude when moving over these raised surfaces, causing the observed variations in amplitude.

It was clear that these amplitude variations had a significant impact on the performance of the ROV and it would be incapable of completing tasks effectively. Further testing was conducted to assess the stability of the altitude control system under different conditions. The results of this testing showed that the system struggled to maintain a stable altitude, with significant variations in amplitude. It was evident that improvements were needed to address these issues and optimize the system’s performance.

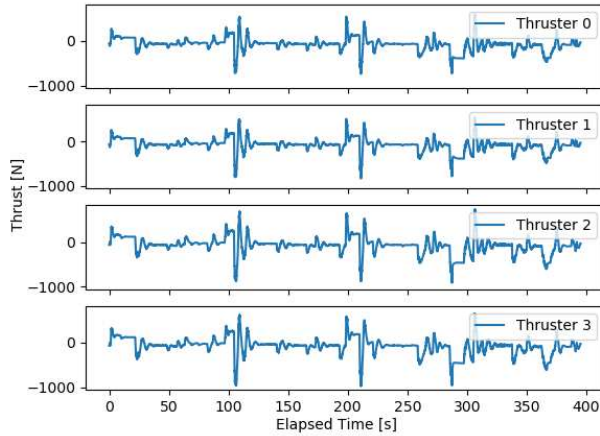


(a) Altitude kept during the altitude control system was active. The desired state altitude was set at 8.5[m].

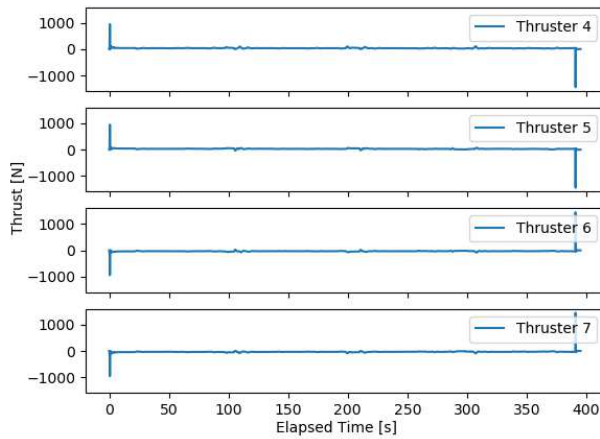


(b) Commanded velocities during the altitude control system was active.

Figure 4.9: Results of simulation case A1 using a simple proportional controller with the RexROV. Showing the motion control system's altitude variations (left), and the commanded velocities from the system.



(a) Thrust force of thruster 0-3 during the altitude control system was active



(b) Thrust force of thruster 4-7 during the altitude control system was active

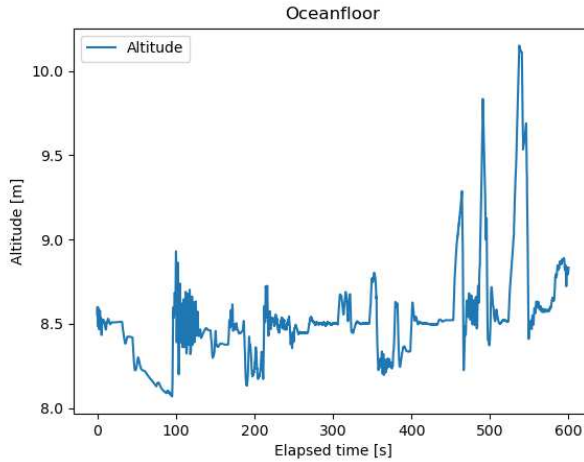
Figure 4.10: Results of simulation case A1 showing produced thrust from the craft.

4.5.2 Case A2 - Altitude Proportional Control EELY500

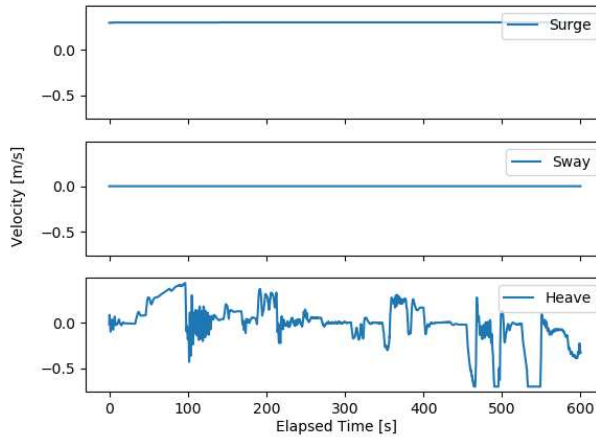
The same tests were also performed using the Eelume robot, EELY500. The results were similar to those obtained with the RexROV, with significant amplitude variations in the altitude control system. It is likely that the additional thrusters on the Eelume robot contributed to the increased oscillations in the system, as more thrusters can result in more complex and sensitive control systems.

The EELY500 model was much slower than the RexROV model in completing the mountain stretch, which may be due to differences in thruster setup or specific design features of the two models.

According to results shown in figure 4.11, the EELY500 model required more control forces to maintain its desired altitude, indicating that it may have had to expend more energy to navigate the stretch.

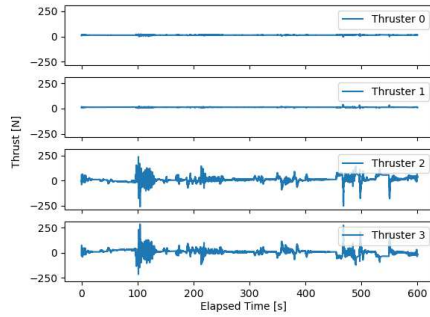


(a) Altitude kept during the altitude control system was active. The desired state altitude was set at 8.5[m].

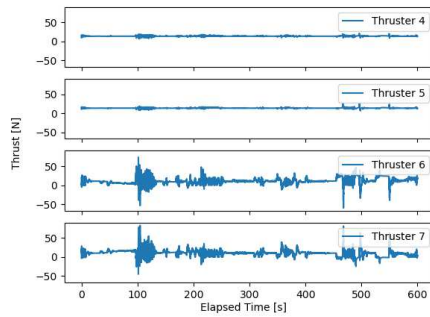


(b) Commanded velocities during the altitude control system was active.

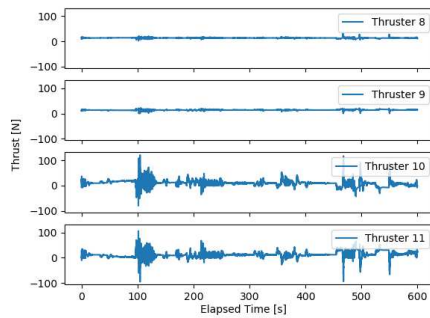
Figure 4.11: Results of simulation case A2 using a simple proportional controller with the EELY500. Showing the motion control system's altitude variations (left), and the commanded velocities from the system.



(a) Thrust force of thruster 0-3 during the altitude control system was active



(b) Thrust force of thruster 4-7 during the altitude control system was active



(c) Thrust force of thruster 8-11 during the altitude control system was active

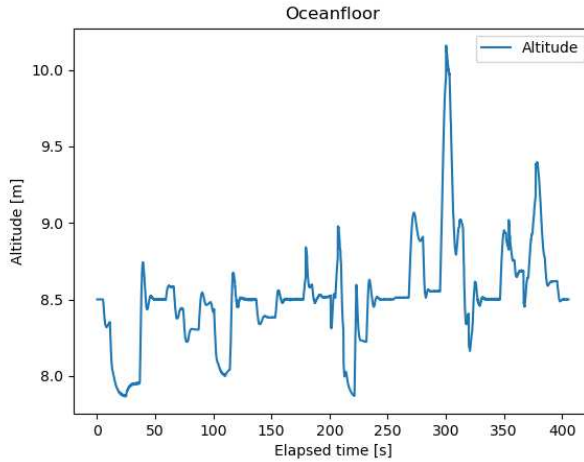
Figure 4.12: Results of simulation case A2 showing produced thrust from the craft.

4.5.3 Case B1 - Altitude PID Control RexROV

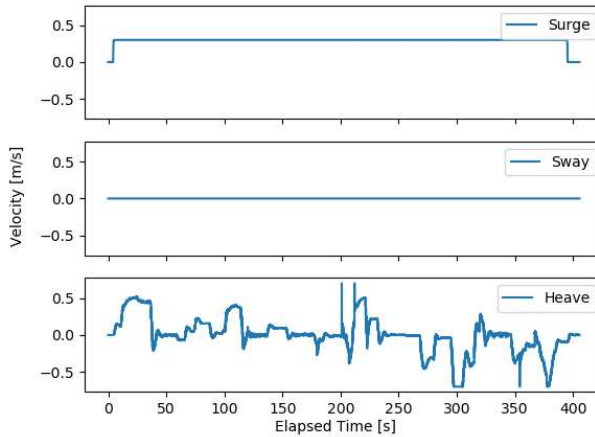
The results of the second simulation using the RexROV, figure 4.13a , showed that the altitude control system was able to successfully maintain the desired altitude within a certain tolerance. The performance of the system was evaluated by analyzing the error between the desired and actual altitude, the rise time, and the settling time.

The error between the desired and actual altitude was found to be within an acceptable range, indicating that the system was able to maintain the desired altitude accurately. The rise time, which is the time taken for the system to reach the desired altitude after a disturbance, was also within an acceptable range. This shows that the system was able to quickly respond to any changes in the altitude and bring the robot back to the desired altitude, figure 4.13b.

The settling time, which is the time taken for the system to stabilize after a disturbance, was within the range of 3 seconds which was acceptable. This indicates that the system was able to quickly return to a stable state after a disturbance.

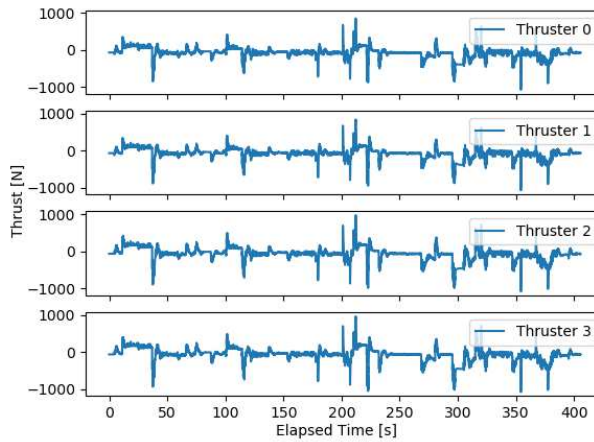


(a) Altitude kept during the altitude control system was active. The desired state altitude was set at 8.5[m].

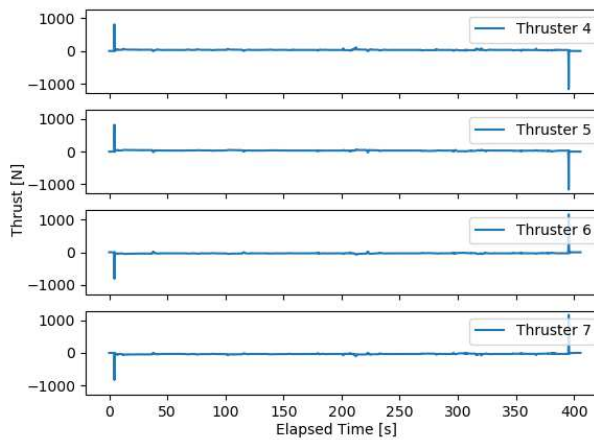


(b) Commanded velocities during the altitude control system was active.

Figure 4.13: Results of simulation case B using a simple proportional controller with the RexROV. Showing the motion control system's altitude variations (left), and the commanded velocities from the system.



(a) Thrust force of thruster 0-3 during the altitude control system was active



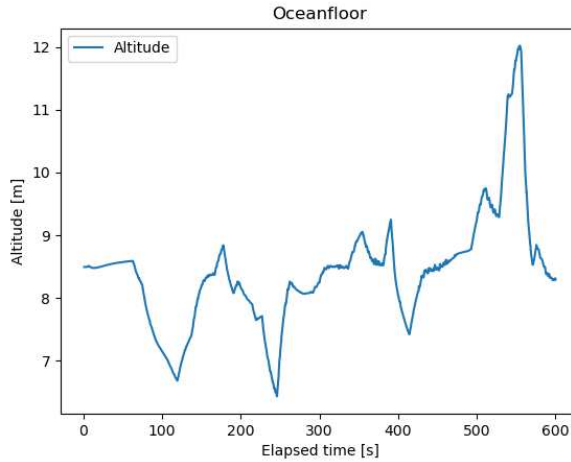
(b) Thrust force of thruster 4-7 during the altitude control system was active

Figure 4.14: Results of simulation case B1, showing produced thrust from the craft.

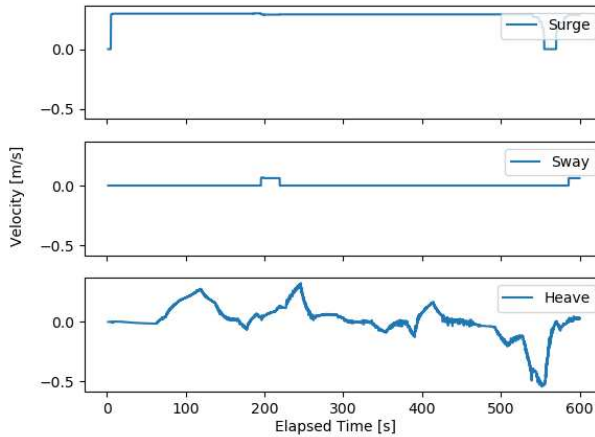
4.5.4 Case B2 - Altitude PID Control EELY500

The EELY500 performed significantly better when using a PID controller in the simulation. The settling times and response times were both much improved compared to the previous simulation using only a proportional controller. In addition, the thrusters were much more active in creating thrust in the heave direction compared to the previous simulation. The PID controller is able to continuously adjust the output of the thrusters based on the current state of the system and the desired state, which allows it to more effectively maintain a stable and controlled position. This improved control provided by the PID controller likely contributed to the EELY500's improved performance in the heave direction.

Despite these improvements, the EELY500 did encounter some difficulties while climbing and descending the steepest parts of the mountain range. This may have been due to the PID tuning being set too low, or possibly due to other factors such as insufficient power or torque, or dynamic instability.

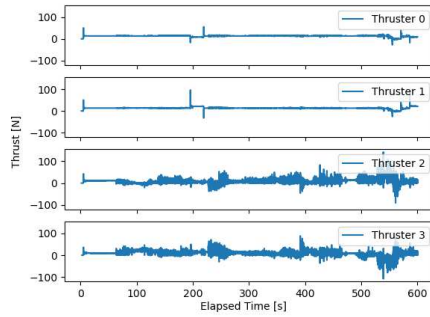


(a) Altitude kept during the altitude control system was active. The desired state altitude was set at 8.5[m].

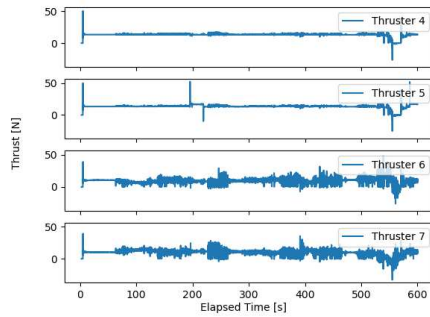


(b) Commanded velocities during the altitude control system was active.

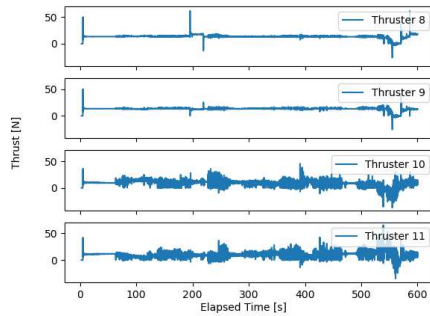
Figure 4.15: Results of simulation case B using a simple proportional controller with the RexROV. Showing the motion control system's altitude variations (left), and the commanded velocities from the system.



(a) Thrust force of thruster 0-3 during the altitude control system was active



(b) Thrust force of thruster 4-7 during the altitude control system was active



(c) Thrust force of thruster 8-11 during the altitude control system was active

Figure 4.16: Results of simulation case B2 showing produced thrust from the craft.

Discussion

The first objective of this thesis was to investigate and understand the operational limitations of using UHI when mapping a subsea area. From [34], it is mentioned that the average mapping altitude was 8.5 [m] when the maximum range variation during operation was between 6.5 [m] and 9.1 [m]. It is worth noting that the campaign was conducted in pristine conditions, with clear water and very high light intensity. The altitude and velocity of a mission are carefully selected based on the objectives of the operation and the details of the mission plan. When flying at lower altitudes, there are several benefits to consider. For example, lower altitudes often result in higher resolution images due to the smaller spatial distance between pixels. This means that objects in the images are captured in greater detail. Additionally, flying at lower altitudes can help reduce diffraction effects from light sources, which can cause distortions or blurriness in the images. Furthermore, in situations where the water is murky or cloudy due to a high concentration of particles, flying at a lower altitude can help maintain visual contact with the target, making it easier to track and monitor its movements. Simulated environments may not take into account the effects of water conditions on UHI operations, potentially leading to unrealistic results. It is important to consider the development of systems or approaches that can effectively handle the challenges and errors that may arise in real-world situations, in order to improve the reliability and effectiveness of UHI operations.

The most significant disadvantage on operating at low altitudes, is that the area of coverage would suffer and the travel velocity will need to match the data capture frequency. This makes the mapping operation a time and resource consuming endeavour, if the area of mapping is many square kilometers large. As the USR is in terms a AUV and operating without a tether. The operational window is determined by the capacity of the battery onboard the platform. Recharging the same battery causes downtime, and if there is no charging platform in the mapping vicinity, it would also be in need of a recovery vessel to recover and redeploy said platform. To use UHI in a deepsea area, artificial lighting is needed. As mentioned in section 3.2.4, several light sources would be expected to be used, to make sure that the entire visualized area is illuminated, and shadow casting is kept at

a minimum. This would add further strain on the electrical system and could deprive the AUV to complete its mission in one setting or constrain it to operate in smaller time windows. The developed altitude control system was designed to be configurable, allowing the system parameters, including the desired state and tuning, to be customized to meet the specific mission objectives. This means that the altitude control system can be used with any type of underwater robot platform, as it can be easily adjusted to the individual characteristics of each platform.

The second thesis objective was to undertake a study of the prior work done on USR as a UHI sensor carrying platform. The EELY500 USR was as mentioned in section 4.4.2, developed as a previous master thesis project. As the model stands, it is not able to actuate its joints, as what would be expected when operating a USR. Instead, all joints are set as a fixed state. The hydrodynamics are inherited from the previous model causing restoring forces in both pitch and roll, this could have been altered, but was not necessary for completing the case simulations, it was not considered to be solved since it would not add anything to the end result. All components in the model were to be homogeneous transformed and rotated according to the local body frame of reference. Making it act as rigid body when movement is applied. To create a USR, the suggestion would be to develop a new model from scratch. Bear in mind, the goal of this thesis was focused on developing an altitude control system, where the goal was maintaining the UHI carrying platform at a desired altitude. The platform carrying the sensor could be arbitrary, since it was expected to perform likewise on any robot, given that some attributes in the system needed to be tuned to compensate for different body shapes. Using the EELY500 is a solid first step to have systems implemented to compensate for elongated bodied crafts. The hydrodynamics were simplified to a submerged cylinder, accomplished as seen in appendix A. It would be preferable to also have added the hydrodynamics of the thruster shapes to the equations, this was not implemented due to time constraint concerns and other priorities were set.

The third and fourth objective of this project were to design and implement an altitude control system for a robot. The system was to allow a robot, such as a USR, to follow the curvature of a surface, but had to be limited to contouring when using fixed body shapes.

Two sensors were applied as means of altitude measurement for the RexROV and EELY500 platforms, a stereo-vision depth camera and a DVL. Redundancy can be pointed out as not as important in a simulation environment as it would for the real endeavour. This adds robustness to the system and reduces possible encountered margin errors. Also mentioned earlier, visual sensors (stereo-vision camera) work poorly when the target is at a great distance. At close range, the thrusters can cause an upheaval of seabed debris and sand. This can cause the depth camera to emit a wrong measurement. This could also happen at greater distances where a fish could cause the same effect. These encounters need to be handled, but have not been valued to be of high importance at this stage of development. In a simulated underwater environment (Plankton), the scenario is always in a pristine state. In the current version of the simulator, it is not possible to include elements of poor visibility and diffraction from artificial light. While it was explored as an option during the development process, a viable solution was not found. This upgrade is expected to

significantly increase the realism of the simulator and should be considered as a goal for anyone planning to continue its development. The DVL is reliant on acoustics to measure velocity and position (altitude), and is unaffected by the visual impairments a camera would experience in deep water. It has superior range of up to 75[m] when operating with four beams, [23], when compared with the common visual range of cameras at 5[m], [8]. Even though it doesn't change the system behaviour within the simulator, it would add a larger depth range of measurement if applied to a real robot.

A few designs were proposed as to solving the collision avoidance problem for elongated bodies for such robots as a USR, while maintaining a trajectory course correction using only input from the two system sensors, DVL and stereo-vision camera. The base concept used a configuration where the trajectory was updated inline with the input sample rate, which was set to be 10 [Hz]. This resulted in the trajectory being corrected every 0.1 [s]. The prototype was developed to be deployed on a traditional ROV (RexROV) which would only move in four degrees of freedom, eliminating possible moments in pitch and roll. Due to the altitude that was preset to be above 6 [m], the PointCloud from the stereo camera which was generated, would cover an area wide enough to safeguard the ROV from colliding from any inclined surface during simulation trials. The DVL would add an extra layer of coverage by its beam layout and added range. This was suitable setup for running the system within the Plankton environment but still has clear weaknesses that need to be addressed before the system could be deployed to any real craft. In a case of a precipice, the system would only create new movement trajectory according to the covered area directly beneath the robot. The depth camera would cover slightly in front of the robot, given that it was at a reasonable altitude (more than 6 [m]). This cannot be relied upon, and is neither convenient when the visual range is described to be 5 [m] and lower. For this scenario, it would be suggested to add another sensor into the loop which has view of the forward motion in surge. The real Eelume robot has another camera available for frontal view, appendix B.1, which could be taken advantage of and added into the system. Adding vision in the direction of movement would also be needed if the system is to be expanded to pitch control or having fully actuated joints. A whole new ordeal would open up if the body shape were to change, the sensors would need to be ideally placed and the system would need some adjustments to incorporate additional sensors.

The system is designed to be strictly conservative when encountering an obstacle or dangerous situation. It will try to elevate itself out of harms way regardless of the obstacle is in its path as it registers by the sensors. Having a defensive behaviour lowers the risk of collision, which surely is important when the robot and equipment is expensive. But the practical application of it will suffer in capture sensor data quality. Having the robot be more environmentally aware of its surroundings will be elaborate more in suggestions for further development in section 6.2.

Simulations allow development and testing to commence without the need of facilities and creation of prototypes. It gives the possibility to verify if the design is sound and sturdy before going into anything costly. This makes simulations a very valuable asset to have. This requires however a great deal of invested time to accurately represent the reality

of the field one wishes to replicate. The payoff is as stated in the long run. The simulation model needs to be as close to a digital replication of a real marine craft, including its adjacent physical and hydrodynamical features.

Looking at the results in simulation case (A1), figure 4.9a, where only a proportional controller was used on the RexROV. It performed much better than what was initially expected at this early stage of development. Oscillations occur at a few points, 100 [s] mark and 280 [s] being the most significant areas. These were at positions that had steep inclines, and the commanded velocities were set to maximum. The combination of damping and thruster control managed to halve the amplitude at each subsequent oscillation before bringing the system to rest.

The same simulated case was completed using the EELY500 model (A2). Like in the RexROV trial, oscillations occurred at the same marked times. These appeared to be much more violent and didn't bring the system to rest before some time. It was discovered that the Eelume model had an incorrect center of gravity (COG), which caused issues with the motion control system. Once this issue was addressed, it was found that there were additional problems with the thruster forces that resulted in excess pitch and yaw movement. While these issues were resolved to a satisfactory level for the current test speeds, further work will be needed to find a more permanent solution. Overall, the results of the initial tests demonstrated that the motion control system had similar performance on both the RexROV and the Eelume robot. This was valuable information, as it allowed issues with the system to be identified and helped in uncovering any potential platform-specific problems. It was clear that the motion control system needed to be adjusted and improved in order to optimize performance and achieve stable control. These issues would be addressed before the next set of simulation cases.

The second set of simulation cases (B1 and B2) demonstrated that the altitude control system was able to effectively maintain the desired altitude within the required tolerance. The use of both DVL and stereo-vision helped to improve the accuracy of the system, while the PID controller was able to effectively control the altitude of the robot. The use of a more complex controller and the addition of additional measurement sensors resulted in a significant overall improvement in performance compared to the previous simulation using only a proportional controller. Hardly any noticeable oscillations occurred during the final run. The thrusters were more active as a result, which could also lead to a energy drain of the onboard battery. This is not something that could be analyzed in the simulation but can be assumed needs to be looked at if system is to be deployed.

Conclusion and suggested further development

6.1 Conclusion

In conclusion, this project aimed to design and implement an altitude control system, not limited to, for elongated robots such as Underwater Snake Robots (USRs). The main objectives were to propose a control system that allows the robot to follow the curvature of a surface while minimizing oscillations, and to create a simulation of the proposed control system using ROS2 software and Gazebo. The proposed control system is a viable prototype that is able to handle the contouring of the seafloor with to a good degree of precision and agility. The system is designed to automate the altitude control, which allows the operator to focus on other aspects of controlling the robot. However, it still requires an operator to control the robot's movement, as the system is not yet fully autonomous.

The system has been tested in a simulated environment using Gazebo as the simulation software. The results of these early tests showed that the system is able to handle the contouring of the seafloor and minimizes oscillations when the robot encounters sudden elevations or drops in the terrain. It performs within acceptable limits. It is worth noting that the system's performance is currently limited to the terrain within the Gazebo simulator, and it is not yet designed to handle walls or other types of obstructions. The system can only apply control to the surface within the sensors' field of view, which means that more sensors or a different sensor layout may be needed for the system to handle a more diverse range of terrain.

While the ideal would have been to test the system on a physical Underwater Snake Robot, such as NTNU's Eelume USR EELY500, to compare the simulation results to sea trials. However, due to the EELY500 being out of commission during the development of this project, it was not possible to conduct such tests. Nevertheless, this project has a lot of

potential for further development and the current altitude control system serves as a solid foundation for future research and development. With additional work, it could be further improved and refined to meet the specific needs of different types of sensor platforms.

6.2 Suggestions for further development

The proposed altitude control system is a strong starting point for future simulations and development in the field of underwater robots. However, there are several areas where the system can be improved to enhance its performance and capabilities.

Velocity, Pitch and Roll Control

One of the main areas of improvement for the system is to add control for the robot's velocity, pitch and roll. Velocity control would enable the robot to collect sensor data at an optimal speed, which would improve its overall performance. Additionally, by implementing control for the robot's pitch and roll, the system would be able to move in six degrees of freedom, which would increase its flexibility and maneuverability. Pitch control would enable the system to follow the contours of the surface more fluently and move parallel to the surface, rather than being limited to rough contouring. Roll control would add the ability for the robot to map walls and ceilings in addition to the seafloor.

Actuated USR Joints

Another area of improvement for the system is the implementation of actuated joints on the USR. Actuated joints would enable the robot to alter its body shape during operation, which would improve its ability to navigate and follow the contours of the surface. Together with pitch control, it could also navigate and turn in confined spaces, such as a wreckage or an underwater cave.

Improved Ocean Simulation

The proposed control system has been tested in a simulated environment using Gazebo, but for more realistic simulation, the ocean environment can be further improved with better scattering effects, turbulence and murky waters. This will enable the system to be attuned to these effects, before being applied to physical entities. Many issues can be addressed at the simulation stage of production, making it easier for the system to work with physical entities.

Environmental Awareness

An additional area of improvement would be to increase the robot's environmental awareness. The proposed system is currently conservative towards its own safety, and relies on the operator to determine the present danger. By increasing the robot's environmental awareness, it would be less reliant on the operator, and would be able to take more appropriate actions based on the data from its sensors. This could be done by using more sensors to cover a larger area around the robot, or by adding new sensors. The suitability

of this will depend on the amount of resources available, as well as the requirements of the mission.

Overall, these suggestions for further development would greatly improve the performance and capabilities of the proposed altitude control system, making it suitable for a wide range of applications in underwater navigation.

Bibliography

- [1] Derya Akkaynak and Tali Treibitz. Sea-thru: A method for removing water from underwater images. *Computer Vision and Pattern Recognition*, pages 1682–1691, 06 2019. doi: 10.1109/CVPR.2019.00178.
- [2] Feyling, Anders. *Tightly Integrated Doppler Velocity Log Aided Inertial Navigational System*. NTNU, 2018.
- [3] Charlie Campbell. A climate solution lies deep under the ocean—but accessing it could have huge environmental costs, 2021. URL <https://time.com/6094560/deep-sea-mining-environmental-costs-benefits>.
- [4] Robert D. Christ and Robert L. Wernli Sr. *The ROV manual (Second Edition)*. BH, 2014. doi: 10.1016/C2011-0-07796-7.
- [5] Wikimedia Commons. Royal navy - cutlet rov. *Cutlet ROV*, 2020-10-12. URL <https://commons.wikimedia.org/wiki/File:Cutletrov.jpg>.
- [6] Fredrik Dukan. *ROV Motion Control Systems*. NTNU, 2014.
- [7] Ines Dumke, Stein M. Nornes, Autun Pursur, Yann Marcon, Martin Ludvigsen, Steinar L. Ellefmo, Geir Johnsen, and Fredrik Søreide. First hyperspectral imaging survey of the deep seafloor: High resolution mapping of the manganese nodules. *Remote Sensing of Enviroment*, 1:19–30, 2018.
- [8] Ecotone. Ecotone hyperspectral ocean vision, 2019. URL https://ecotone.com/wp-content/uploads/2019/10/UHI-OV-2000_3000_6000.pdf.
- [9] Eelume. Reshaping underwater operations. *Eelume*, 2022-10-13. URL <https://eelume.com/>.
- [10] Thor I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. NTNU, 2011. URL <https://doi.org/10.1002/9781119994138>.

-
- [11] Geir Johnsen, Martin Ludvigsen, Asgeir Sørensen, and Lars Martin Sandvik Aas. The use of underwater hyperspectral imaging deployed on remotely operated vehicles - methods and applications. *IFAC-PapersOnLine*, 49(23): 476–481, 2016. ISSN 2405-8963. doi: <https://doi.org/10.1016/j.ifacol.2016.10.451>. URL <https://www.sciencedirect.com/science/article/pii/S2405896316320390>. 10th IFAC Conference on Control Applications in Marine SystemsCAMS 2016.
- [12] Bendik Alexander Urne Jørgensen. *Methods of Visual Inspection and Mapping of Subsea Structures and Rough and Steep Underwater Terrain Using Articulated Underwater Robots*. NTNU, 2022.
- [13] E Kelasidi, K Pettersen, P Liljeback, and J T Gravdahl. Locomotion efficiency of underwater snake robots with thrusters. *Safety, Security and Rescue Robotics*, 2016-10-27. URL [10.1109/SSRR.2016.7784295](https://doi.org/10.1109/SSRR.2016.7784295).
- [14] Frank C. Park Kevin M. Lynch. *Modern Robotics - Mechanics, Planning and Control*. Cambridge University, 2017.
- [15] Pål Liljeback, Kristin Y. Pettersen, Øyvind Stavdahl, and Jan Tommy Gravdahl. *Snake Robots*. Springer, 2011.
- [16] Liquid.ai. Plankton. <https://github.com/Liquid-ai/Plankton.git>, 2021.
- [17] Bohan Liu, Zhaojun Liu, Shaojie Men, Yongfu Li, Zhongjun Ding, Jiahao He, and Zhigang Zhao. Underwater hyperspectral imaging technology and its applications for detecting and mapping the seafloor: A review. *Sensors (Basel, Switzerland)*, 20, 09 2020. doi: [10.3390/s20174962](https://doi.org/10.3390/s20174962).
- [18] Martin Ludvigsen and Asgeir J. Sørensen. Towards integrated autonomous underwater operations for ocean mapping and monitoring. *Annual Reviews in Control*, 42:145–157, 2016. ISSN 1367-5788. doi: <https://doi.org/10.1016/j.arcontrol.2016.09.013>. URL <https://www.sciencedirect.com/science/article/pii/S1367578816300256>.
- [19] Musa Morena Marcusso Manhães, Sebastian A. Scherer, Martin Voss, Luiz Ricardo Douat, and Thomas Rauschenbach. UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation, sep 2016. URL <https://doi.org/10.1109/2F oceans.2016.7761080>.
- [20] mecharithm. Homogeneous transformation matrices to express configurations in robotics, oct 2022. URL <https://www.mecharithm.com/homogenous-transformation-matrices-configurations-in-robotics/>.
- [21] Juan C. Montes-Herrera, Emiliano Cimoli, Vonda Cummings, Nicole Hill, Arko Lucieer, and Vanessa Lucieer. Underwater hyperspectral imaging (uhi): A review of systems and applications for proximal seafloor ecosystem studies. *Remote Sensing*, 13(17), 2021. ISSN 2072-4292. doi: [10.3390/rs13173451](https://doi.org/10.3390/rs13173451). URL <https://www.mdpi.com/2072-4292/13/17/3451>.

-
- [22] NOAA. How much of the ocean have we explored?, 2022-09-12. URL <https://oceanservice.noaa.gov/facts/exploration.html#:~:text=More%20than%20eighty%20percent%20of,unmapped%2C%20unobserved%2C%20and%20unexplored.>
- [23] NORTEK. Doppler velocity log, 2021. URL <https://www.nortekgroup.com/products/dvl-1000-300m>.
- [24] NTNU. Mineralutvinning på havbunnen – den nye oljen?, 2016.
- [25] David M. Price, Katleen Robert, Alexander Callaway, Claudio Lo Iacono, Rob A. Hall, and Veerle A. I.S Huvenne. Using 3d photogrammetry from rov video to quantify cold-water coral reef structural complexity and investigate its influence on biodiversity and community assemblage, oct 2019. URL <https://doi.org/10.1007/s00338-019-01827-3>.
- [26] Open Robotics. Tutorials - ros2 documentation, 2021. URL <https://docs.ros.org/en/foxy/Tutorials.html>.
- [27] Open Robotics. Ros - robot operating system, 2021. URL <https://www.ros.org/>.
- [28] Open Robotics. Gazebo, 2021. URL <https://gazebo.org/home>.
- [29] D. Rose. Rotation quaternions, and how to use them, may 2015. URL <https://danceswithcode.net/engineeringnotes/quaternions/quaternions.html>.
- [30] National Ocean Service. How far does light travel in the ocean?, 2022-09-15. URL https://oceanservice.noaa.gov/facts/light_travel.html.
- [31] UUV simulator. Unmanned underwater vehicle simulator documentation. <https://uuvsimulator.github.io/>, 2022.
- [32] Y. Song, D. Nakatha, and M. She. Optical imaging and image restoration techniques for deep ocean mapping: A comprehensive survey. *Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 2022-06-01. URL <https://doi.org/10.1007/s41064-022-00206-y>.
- [33] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, Inc, 2006.
- [34] Øystein Sture, Martin Ludvigsen, Fredrik Søreide, and Lars Aas. Autonomous underwater vehicles as a platform for underwater hyperspectral imaging. *Autonomous Underwater Vehicles*, 06 2017. doi: 10.1109/OCEANSE.2017.8084995.
- [35] Asgeir Sørensen. *Marine Control Systems - Propulsion and Motion Control of Ships and Ocean Structures*. NTNU, 2013. URL <http://folk.ntnu.no/assor/publications/marcyb.pdf>.
-

-
- [36] Russell B. Wynn, Veerle A.I. Huvenne, Timothy P. Le Bas, Bramley J. Murton, Douglas P. Connelly, Brian J. Bett, Henry A. Ruhl, Kirsty J. Morris, Jeffrey Peakall, Daniel R. Parsons, Esther J. Sumner, Stephen E. Darby, Robert M. Dorrell, and James E. Hunt. Autonomous underwater vehicles (auvs): Their past, present and future contributions to the advancement of marine geoscience. *Marine Geology*, 352:451–468, 2014. ISSN 0025-3227. doi: <https://doi.org/10.1016/j.margeo.2014.03.012>. URL <https://www.sciencedirect.com/science/article/pii/S0025322714000747>. 50th Anniversary Special Issue.
- [37] John Y. Chiang, Ying-Ching Chen, and Yung-Fu Chen. Underwater image enhancement: Using wavelength compensation and image dehazing (wcid). *Advances Concepts for Intelligent Vision Systems*, pages 372–383, 08 2011. doi: 10.1007/978-3-642-23687-7_34.

Appendix

A Simplifications of Hydrodynamics for a Snake Robot with Locked Joints

Eelume consists of several links making it difficult to accurately model the hydrodynamics. To simplify the problem, the body was assumed to be a cylinder with half a sphere attached to each end. The matrices for added mass, linear- and quadratic-damping are presented in this section

A.1 Equations for Added mass

Added mass coefficient were calculated from the equations in table 6.1

Table 6.1: Equations for finding added mass elements.

—	1	2	3	4	5	6
1	$m_{11} = \frac{4}{3}\pi r^3$					
2		$m_{22} = \int_L a_{22} dx$	$m_{23} = -\int_L a_{23} dx$	$m_{22} = \int_L a_{24} dx$		$m_{26} = \int_L x a_{22} dx$
3			$m_{33} = \int_L a_{33} dx$		$m_{35} = -\int_L x a_{33} dx$	
4				$m_{44} = \int_L a_{44} dx$		$m_{46} = \int_L x a_{46} dx$
5					$m_{55} = \int_L x^2 a_{33} dx$	
6						$m_{66} = \int_L x^2 a_{22} dx$

A.2 Drag forces interacting with a submerged Cylinder

Drag forces interacting with the simplified body of Eelume was found by:

$$F_D = \frac{1}{2}\rho C_D DLU^2 \quad (6.1)$$

A.3 Linear and Quadratic Damping

The assumption that the robot will consistently stay at a depth that will not have any interaction with waves, simplifies the linear damping to be a zero matrix.

The quadratic damping is found by:

$$B = \frac{1}{2}\rho DC_D \quad (6.2)$$

B USR EELY500

The USR EELY500 is made up for multitude of different modules. The most significant of these modules are presented in this section.

B.1 Inspection Module

The Inspection module is attached to the bow of the robot. This module is equipped with a HD camera facing forwards. Powerful LED lights are attached for illuminating the surrounding area.



B.2 Ballast Module

The Ballast module incorporates the ability to trim the buoyancy of the USR. This is achieved by adjusting the weight of the module, by adding or reducing a number of coin-shaped weights inside the external slot. This also allows changing the CG position which attributes to a different stability configuration.



B.3 Battery Module

This module houses the battery which drives the entire Eelume system. This contains lithium-ion battery pack producing 868 [Wh].



B.4 Sensor Module

The Sensor module contains the majority of the technical components that are needed. This is such as the acoustic positioning transponder, altimeter, downward-pointing HD camera, and the computer board which handles overall control of the robot.



B.5 Joint Module

The joint module allows the robot to change its body shape configuration. Each joint module houses a motorized 2-DOF joint, which allows for rotation in pitch and yaw. Constraints on the rotation are set as $\pm 80^\circ$.



B.6 Thruster Module

The thruster module contains a set of four thrusters. A pair are fixed horizontally above one another. These two produce thrust in the y-direction relative to the body-frame. Another pair are vertically positioned in alignment on the sides of the body. These thrusters do not set having fixed angles on the real thing. In the simulation module, these are locked to operate at $+45^\circ$ and -45° . When they are at 0° pitch, the thrusters would only produce forces along the x-axis. By having this tilt, they are enabling the production of forces on both the x-axis and z-axis. Mixing all four thrusters together it is possible to achieve moments and motion along all three axes.



