Master's thesis

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

Bendik Alexander Urne Jørgensen

Methods for Visual Inspection and Mapping of Subsea Structures and Rough and Steep Underwater Terrain Using Articulated Underwater Robots

Master's thesis in Marine Technology
Supervisor: Asgeir J. Sørensen
Co-supervisor: Markus Fossdal
June 2022

NTNU
Norwegian University of
Science and Technology

Bendik Alexander Urne Jørgensen

Methods for Visual Inspection and Mapping of Subsea Structures and Rough and Steep Underwater Terrain Using Articulated Underwater Robots

Master's thesis in Marine Technology
Supervisor: Asgeir J. Sørensen
Co-supervisor: Markus Fossdal
June 2022

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

NTNU
Norwegian University of
Science and Technology

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

MASTER OF TECHNOLOGY THESIS DEFINITION (30 SP)

Name of the candidate:	Bendik Alexander Urne Jørgensen
Field of study:	Marine Cybernetics
Thesis title (Norwegian):	Metoder for visuell inspeksjon og kartlegging av undervanns konstruksjoner og krevende og bratt havbunnsterreng ved bruk av artikulerte undervannsroboter
Thesis title (English):	Methods for visual inspection and mapping of subsea structures and rough and steep underwater terrain using articulated underwater robots

Background

With a growing interest in fields like ocean discovery, there is a need for inspection and intervention in confined and demanding areas. Using articulated underwater robots, it is possible to maneuver and operate in confined spaces, that would be inaccessible for a large box-shaped vehicle. An articulated underwater robot consisting of multiple rigid bodies features complex dynamics, which makes control challenging. Therefore, it is necessary to improve methods for guidance, navigation and control. Subsea inspections are often performed using a camara, however, due to a lack of light it may be challenging to achieve good video quality over a large field of view. Improvement of visual inspection can be achieved using Underwater Hyperspectral Imaging sensors, which is a technology for identification and mapping in marine environments based on spectral signatures.

Scope of Work

1. Investigate methods and frameworks for guidance, navigation and control of articulated robots for inspection of structures and rough and steep terrain.
2. Create a simulation platform to be used for supporting experimental work.
3. Familiarize with the Eelume vehicles and describe the set-up for Eelume.
4. Present the simulator and experimental setup.
5. Perform a simulation case study to investigate the behavior of underwater snake robots.
6. Validate the simulator with field experiments.
7. Document the results in a report.

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 about 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, abstract, preface (incl. description of help, resources, and internal and external factors that have affected the project process), acknowledgement, project definition, 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, method, results and analysis, conclusions with recommendations for further work, references, and optional appendices. Figures, tables, and equations shall be numerated. The original contribution of the candidate and material taken from other sources shall be clearly identified. Work from other sources shall be properly acknowledged using quotations and a Harvard citation or Vancouver reference 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 series, etc., shall be included electronically with the report.

Start date: 15 January 2022
Due date: 10 June 2022

Supervisor: Asgeir J. Sørensen
Co-advisor(s): Markus Fossdal

Trondheim, 09.06.2022

Professor Asgeir J. Sørensen
Supervisor

Preface

This Master's thesis completes the work in the course TMR4930, and completes the fulfillment of the Master's degree in Marine Technology, with a specialization in Marine Cybernetics at the Department of Marine Thechnology (IMT) at the Norwegian University of Science and Technology (NTNU). The work was done independently between January and June 2022 under the supervision of Professor Asgeir J. Sørensen and co-advisor Markus Fossdal.

This thesis is a continuation of the pre-work done in the Project-thesis written in the course TMR4510 between August and December 2021. A large portion of the mathematical modeling from the Project thesis is reused in Chapter 2 of this Master's thesis. The work is based on research on underwater snake robots, and is heavily inspired by the Eelume vehicles. Simulations were performed using the open source simulator, Gazebo, with plugins for underwater vehicles. A model based on the Eelume vehicle EELY500 is used for simulation, and was coded by me. Existing code for an ROV has been used as a reference for coding the snake robot model, and a previously existing controller is reused. The problem description was formulated by Asgeir J. Sørensen, and has been modified by me, as the direction of the thesis has changed over time.

The field experiments were performed on May 2, 2022 in the Trondheim Fjord from onboard RV Gunnerus. This trip was organized by Markus Fossdal, and the vehicle was piloted by Gabriele Kasparaviciute. Otherwise, the work presented in this thesis is solely done by me.

Acknowledgements

I am grateful to my supervisor, Professor Asgeir J. Sørensen, for providing excellent guidance through the process of writing this Master's thesis. His enthusiasm and ideas has helped defining this thesis, and his knowledge on the subject has has been motivational for working hard during the semester. I would also like to thank my co-advisor, Markus Fossdal, for all his help throughout the semester, by answering many questions and providing general tips for writing. Having frequent discussions with Sørensen and Fossdal has been essential in improving the quality and shaping the direction of this thesis.

I am thankful to Markus Fossdal for inviting me onboard RV Gunnerus such that field experiments could be carried out. These field experiments would not have been possible without the crew onboard RV Gunnerus. Thank you to Gabriele Kasparaviciute for piloting the Eelume vehicle such that relevant data for the thesis could be gathered. I would also like to thank Pål Liljebäck for providing the files for the 3D model of the Eelume vehicle. Because of this, the final simulation model looks a lot more appealing. Finally I want to thank my friends, whom I have shared an office with during the final year of studying, for making long days and late evenings of work more bearable.

Trondheim, June 10, 2022

Bendik A. U. Jørgensen

Abstract

With a growing interest in fields like ocean discovery, there is a need for inspection and intervention in confined and demanding areas. Today, most operations regarding visual inspection and mapping are carried out by remotely operated vehicles. However, it is difficult to access areas with demanding terrain using these vehicles. One concept with the potential to increase accessibility to demanding areas are underwater snake robots. These are articulated underwater robots inspired by biological snakes. The robot's slender shape, in addition to the ability to change body configurations, make the articulated underwater robots better suited to explore challenging underwater terrain than traditional underwater vehicles.

This thesis presents a simulation model for control of underwater snake robots with thrusters in a visual environment, with the possibility of implementing guidance systems and simulated sensor measurements in the future. The simulation platform of choice is the Plankton simulator, which is an open source simulator for maritime robotics researchers, that is built using Gazebo with the plugin UUV Simulator which has been made compatible with ROS 2. The thesis describes the dynamics of underwater snake robots necessary for creating a mathematical model. The mathematical modeling consists of kinematics and kinetics for marine crafts, combined manipulator kinematics. Simple hydrodynamics for cylindrical bodies are included, while the hydrodynamic effects concerning multiple bodies affecting each other are disregarded. Methods for guiding underwater snake robots for inspection operations are discussed, and one potential method is waypoint following while taking advantage of articulation by rotating the head of the robot to make sure the camera position is always ideal. A model of an underwater snake robot based on the Eelume vehicle EELY500 is implemented in the Plankton simulator. Code for a previously existing ROV is used a reference, while a previously existing speed controller is reused and tuned for better control of the snake robot. The joint angles of the robot can be specified before launch, however they can not be changed dynamically during simulations.

Simulations are performed where the snake robot receives input from a joystick to move in surge, sway and heave, and the resulting positions of the snake robot are shown. Because of the angles of the thrusters placed on the starboard and port side of the robot, moments are induced that need to be compensated for. This may cause more power consumption than necessary, and can potentially be avoided by making sure the thrusters only produce forces in a single direction compared to the body of the vehicle. The simulation results are compared to results from a field experiment where similar tests are performed. These results suggest that the real robot achieved higher velocities than the simulated model, and it is concluded that the most likely cause is insufficient controller gains. One thruster malfunctioned during the field experiments, however the robot could still be controlled without issues, which supports the importance of redundancy in the system. Results also show that there is coupling in the degrees of freedom of the robot which coincides with the theory. Therefore it is necessary to implement a more advanced control system in the simulator to achieve more optimal control of the robot.

Sammendrag

Med økende interesse i områder som havutforskning, er det nødvendig å kunne utføre inspeksjoner og intervensjon i trange og krevende områder. I dag utføres de fleste operasjoner som omhandler visuell inspeksjon og kartlegging ved hjelp av fjernstyrte undervannsfarkoster. Men, det er vanskelig å få tilgang til områder med krevende terreng ved bruk av disse farkostene. Et konsept med potensialet for å utvide tilgang til krevende områder er undervanns-slangeroboter. Dette er artikulerte robotoer inspirert av biologiske slanger. Roboten sin slanke form, i tillegg til muligheten til å endre konfigurasjon, gjøre at artikulerte undervanns roboter er bedre egnet for utforskning av krevende undervannsterreng enn tradisjonelle undervannsfarkoster.

Denne oppgaven presenterer en simuleringsmodell for kontroll av undervanns-slangeroboter med thrustere i et visuelt miljø, med mulighet til å implementere styringssystemer og simulerte sensormålinger i fremtiden. Den valgte simuleringsplattformen er Plankton-simulatoren, som er en open source simulator for forskning på maritim robotikk, og er bygget ved bruk av Gazebo og UUV Simulator som har blitt gjort kompatibel med ROS 2. Oppgaven beskriver den nødvendige dynamikken til undervanns-slangeroboter for å kunne sette sammen en matematisk modell. Den matematiske modellen består av kinematikk og kinetikk for marine fartøy, kombinert med kinematikk for manipulatorarmer. Enkel hydrodynamikk for sylinderformede legemer er inkludert, mens hydrodynamiske effekter som oppstår på grunn av interaksjon mellom flere legemer er sett bort fra. Metoder for styring av undervanns-slangeroboter med tanke på inspeksjon diskuteres, og en mulig metode er å følge veipunkter mens man utnytter artikulering ved å rotere hodet til roboten slik at kameraet alltid er ideelt posisjonert. En modell for en undervanns-slangerobot basert på Eelmue roboten EELY500 blir implementert i Plankton-simulatoren. Kode for en eksisterende ROV er brukt som referanse, og en allerede eksisterende hastighetskontroller er gjenbrukt og innstilt for å tilpasse seg kontroll av slangeroboten. Leddvinklene til roboten kan defineres av en bruker før simulatoren kjøres, men de kan ikke endres dynamisk under simulering.

Simuleringer utføres der slangeroboten mottar kommandoer ved hjelp av en joystick til å bevege seg i jag, svai og hiv, og de resulterende posisjonene til slangeroboten blir vist. På grunn av vinklene til thrusterene som er plassert på styrbord og babord side av roboten, blir det indusert momenter som må kompenseres for. Dette kan føre til større strømforbruk enn nødvendig, og kan potensielt unngås ved å plassere thrusterene slik at de kun produserer kraft i en retning i forhold til fartøyet. Simuleringsresultatene sammenliknes med resultater fra felteksperimenter hvor liknende tester gjennomføres. Ifølge disse resultatene oppnår den virkelige roboten høyere hastigheter enn den simulerte modellen, og det blir konkludert at dette sannsynligvis er forårsaket av utilstrekkelige kontrollparametere. Det oppsto en feil i en av thrusterene under felteksperimentene, men roboten kunne fortsatt kontrolleres uten problemer, som viser viktigheten av redundans i systemet. Resultatene viser også at det er kopling i frihetsgradene til roboten. som stemmer overens med teorien. Derfor er det nødvendig å implementere et mer avansert kontrollsystem i simulatoren for å oppnå mer optimal kontroll av roboten.

Table of Contents

List of Figures

List of Tables

Symbols and Acronyms

Acronyms

AUV	Autonomous Underwater Vehicle
CB	Center of Buoyancy
CG	Center of Gravity
CO	Center of Origin
DOF	Degrees of Freedom
DVL	Doppler Velocity Logger
EOM	Equation of Motion
GNC	Guidance, Navigation and Control
HI	Hyperspectral Imager
HiPAP	High Precision Acoustic Positioning System
IMU	Inertial Measurement Unit
MIMO	Multiple Input Multiple Output
PID	Proportional Integral Derivative
ROS	Robot Operating System
ROV	Remotely Operated Vehicle
UHI	Underwater Hyperspectral Imager
UUV	Unmanned Underwater Vehicle

List of Symbols

$\boldsymbol{\eta}$	Position and orientation of the BODY frame in the inertial frame
ζ	Body and joint velocities
Θ_{nb}	Euler angles
θ_i	1 DOF joint angle
$\boldsymbol{\nu}$	Body velocity vector of the BODY frame
$\boldsymbol{\nu}_c$	Commanded velocities
ρ	Fluid density
$\boldsymbol{\tau}$	Body forces and moments
$\boldsymbol{\tau}_c$	Commanded forces and moments
$\boldsymbol{A}_i(\theta_i)$	Transformation matrix for transformation between link frame $\{i+1\}$ and link frame $\{i\}$
B	Buoyancy
$\boldsymbol{B}$	Thrust configuration matrix
$\boldsymbol{C}$	Rigid-body Coriolis and centripetal matrix
C_D	Drag coefficient
$\boldsymbol{D}$	Hydrodynamic damping matrix
$\boldsymbol{g}$	Restoring matrix
$\boldsymbol{H}$	Homogeneous transformation matrix
$\boldsymbol{I}_b^b$	Inertia dyadic
$\boldsymbol{J}$	Jacobian
$\boldsymbol{K}_d$	Derivative gains
$\boldsymbol{K}_i$	Integral gains
$\boldsymbol{K}_p$	Proportional gains
L_i	Length of link i
L_j	Length of joint module
$\boldsymbol{M}$	Rigid-body system inertia matrix
m_i	Dry mass of link i
N	Number of links
$\boldsymbol{R}(\boldsymbol{\Theta}_{nb})$	Rotation matrix parametrized with Euler angles

r	Number of thrusters
r_i	Radius of link i
$\boldsymbol{T}_{i,j}(\boldsymbol{q})$	Transformation for transformation between frame $\{i\}$ and frame $\{j\}$
$\boldsymbol{q}$	General joint angles or unit quaternions
$\boldsymbol{S}$	Skew-symmetric matrix
$\boldsymbol{u}$	System inputs
W	Gravitational force

Chapter 1

Introduction

The following thesis presents a simulation model for an underwater snake robot, with the intention of using the model to simulate operations regarding visual inspection and mapping in the future. Underwater snake robots may also be used for intervention tasks, however this is not considered in this thesis. This introductory chapter presents the motivation behind the thesis, as well as research questions, methodology and main contributions, followed by an overview of the structure of the thesis.

1.1 Background and Motivation

With a growing interest in fields like ocean discovery, there is a need for inspection and intervention in confined and demanding areas. The majority of underwater vehicles utilized today for subsea operations are Remotely Operated Vehicles (ROVs) characterized by box shapes and tether communication. Rough and confined underwater terrain would be challenging or in many cases impossible to access with such a vehicle. Because of this, underwater images and maps of the seafloor in difficult terrain are limited. For example, Ludvigsen et al. (2013) describe a survey attempting to locate "Falstadbåten" which has been lost since 1945. A large area was covered during the survey, however steep slopes within this area were excluded, which leaves a possibility of discovering the wreck in the slopes. Using articulated underwater vehicles in the form of underwater snake robots could improve accessability to such areas because of their slender and flexible shape. The usage of underwater snake robots for this thesis is inspired by the Eelume vehicles. Additionally, there is a lack of light in deep water which makes it difficult to produce images of the seafloor from ships and airplanes. Attaching a sensor such as a Hyperspectral Imager (HI) with a light system to an underwater snake robot, could allow for the production of images of rough subsea terrain that would otherwise be inaccessible. Advantages of hyperspectral imagers are discussed in Section 1.1.2.

1.1.1 Underwater Snake Robots

An underwater snake robots may combine the range of Autonomous Underwater Vehicles (AUVs), the ability to access challenging areas of small ROVs, and the intervention capabilities of ROVs. This is illustrated in Figure 1.1.

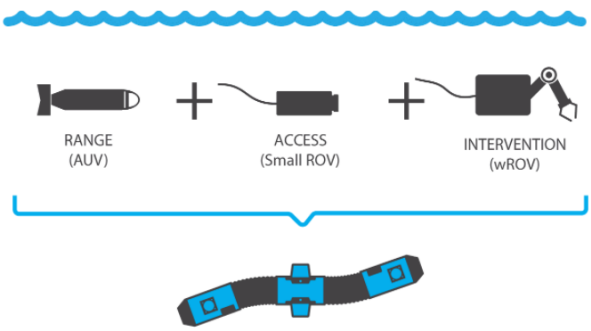


Figure 1.1: Features of Eelume. (Eelume 2022).

Underwater snake robots are articulated vehicles inspired by biological snakes, as they are able to traverse virtually any terrain, and are excellent swimmers. These robots are made up of links interconnected by actuated joints, and are essentially robotic arms that are able to produce locomotion by actuating the joints in order to produce a periodic body wave motion and swim like a biological snake. However, thrusters can also be attached to the vehicle to produce motion. Recent studies, such as a study by Kelasidi et al. (2016), found that locomotion using thrusters is faster and more efficient than using joint actuation for locomotion. These results suggest that it is advantageous to mainly use the thrusters for propulsion, while articulation could be used to change the shape of the body as needed in order to perform certain tasks. For example, straightening all the joints allows the underwater snake robot vehicle to become like a slender torpedo and transit over long distances like an AUV, while changing the body to the shape of the letter "U" could allow the robot to inspect subsea pipelines across a large part of the perimeter at once. The latter example is illustrated in Figure 1.2.

Figure 1.2: Eelume robot inspecting a subsea pipeline. (Eelume 2022).

Additionally, the snake robot's slender and flexible bodies allow them to transit over large distances, and perform inspections, maintenance and repair in areas that would not be accessible to conventional underwater vehicles. Snake robots are well suited to perform inspections of subsea structures, for example detecting leakages or rust in subsea pipelines, in addition to explore and map rough and steep underwater terrain. Another advantage with snake robots is the possibility of modularity. This means that thurster modules, joints, sensor modules, payload modules and intervention tools can be combined in different ways for different tasks. (Eelume 2022)

An underwater snake robot consisting of multiple rigid bodies interconnected by joints gains multiple degrees of freedom and features complex dynamics, which makes controlling joints and thrusters simultaneously to achieve a desired behavior a challenging task. Simulations are a powerful tool to ensure that certain control systems work as intended. Testing algorithms on real hardware may be time consuming, expensive and dangerous, and testing algorithms in a simulator may solve these challenges. Building a simulator for an underwater snake robot is one of the main topics in this thesis. There exist several simulation software for simulating robotics to choose from. For this thesis, the Plankton simulator which is built on Gazebo is used, which will be further described in Chapter 4.

1.1.2 Visual Inspection and Mapping

Visual inspection and mapping operations performed using traditional RGB cameras become difficult when descending to deep parts of the ocean due to a lack of natural light, and must rely on artificial light. Hyperspectral Imagers (HI) have been deployed on boats and airplanes to generate information of the sea surface down to about 20 meters depth during the last decade. For capturing information on the sea floor, Underwater Hyperspectral Imagers (UHI) are currently being deployed on ROVs for more automated identification, mapping and monitoring of the sea floor (Johansen et al. 2016). Hyperspectral imagers create images using multiple bands across the electromagnetic spectrum. Different materials absorb and reflect electromagnetic radiation in different ways, which yields unique spectral signatures. By analyzing the unique spectral signatures reflected by different materials, UHIs are able to classify objects of interest with high spatial and spectral resolution. Another potential application of UHIs is inspection of underwater structures by for instance identifying cracks, leakages and corrosion (Liu et al. 2020).

1.2 Research Questions and Scope of Work

The main goal of this thesis is to set up a simulator that can be used to test different Guidance, Navigation and Control (GNC) methods for an underwater snake robot for the purpose of visual inspection and mapping, in addition to visualize movement. Specifically, EELY500 which is used for research by NTNU will be modeled. EELY500 features a thruster module with several advantages and disadvantages that will be discussed further in Chapter 4. To validate the simulation model, case studies are performed where commands are given to the thrusters using a joystick in order to produce motion in surge, sway and heave, which is used to investigate the step responses of the robot. These simulations are performed for two different body configurations, namely straight-shape and u-shape. For more complex operations it is important that the robot can be controlled precisely. Thus the thesis will also discuss how the thruster module affects the maneuverability of the robot based on the simulation case studies. The main research questions of the following thesis can be summarized as:

1. How can an underwater snake robot be modeled and visualized for a simulator environment.
2. What are the advantages and disadvantages of the current thruster module, and what possible improvements can be made to the thruster module?
3. How is the body-fixed coordinate frame of a vehicle consisting of multiple rigid bodies defined, and how are operations affected by the placement of the body-fixed frame?

Based on the research questions above, the main objectives for gaining knowledge on these questions are presented below:

- Describe mathematical modeling of articulated underwater vehicles.
- Familiarize with EELY500, anf how it is operated.
- Investigate GNC methods for underwater snake robots.
- Build a model of EELY500 in Gazebo and perform simulation case studies.
- Describe the setup for Eelume operations and perform qualitative field experiments in order to validate the simulator.

1.3 Main Contributions

Main contributions are listed below:

- Development of a simulation model of an underwater snake robot in Gazebo which allows for simulations in real time and visualization which makes it intuitive to understand how the system reacts to certain inputs.
- Control methods for underwater snake robots with sensors for visual inspection are proposed.
- Based on the performance of the thrusters during the simulation case studies, alternative thruster modules are suggested.

1.4 Research Method

The research methodology for this thesis includes reviewing literature and theory, simulations and field experiments. Reviewing literature and previous experiments has generated ideas for GNC methods suitable for visual inspection and mapping methods using snake robots. The simulation model is created based on mathematical modelling from theory. Software used for simulation come from several open source libraries, which include Gazebo with plugins and ROS 2. The simulation model for an underwater snake robot is coded based on an existing model of an ROV, and has been reworked several times as bugs and errors were discovered. Since there is limited documentation available for Gazebo with underwater applications, much trial and error has been necessary to build the final model. Simulation case studies are performed in order to make sure that the robot behaves as expected based on basic theory, while also further investigating the behaviour of snake robots with different body configurations. Field experiments resembling the tests performed in the simulation case studies are performed with the intention of discovering inaccuracies in the simulation model.

1.5 Outline

Chapter 2 - Mathematical Modeling of Articulated Underwater Vehicles: This chapter explains how an underwater snake robot can be modeled. General kinematics and dynamics for underwater vehicles, as well as kinematics for manipulator arms are presented. The general equations are then applied to set up the mathematical model for EELY500 that is used for simulation case studies.

Chapter 3 - Methods for Guidance, Navigation and Control: This chapter discusses and proposes methods for maneuvering underwater snake robots. Additionally, thrust configuration matrices for different body configurations are computed.

Chapter 4 - Simulation Case Studies: A model of EELY500 is implemented in the Plankton Simulator. Surge, sway and heave motions are simulated for two different body configurations using methods described in Chapter 3, and the results are presented.

Chapter 5 - Field Experiments: This chapter describes field experiments with EELY500 and presents the results. EELY500 is deployed in the Trondheim Fjord, and are given similar inputs as given in the simulation case studies. The results are compared to results from the simulation case studies.

Chapter 6 - Conclusion and Further Work: This chapter concludes the thesis by attempting to provide answers for the research questions. Suggests ideas for further development of the simulation model are also included.

Chapter 2

Mathematical Modeling of Articulated Underwater Vehicles

The mathematical model is divided into kinematics and kinetics. Kinematics describes the geometrical aspects of motion, while kinetics considers the external forces causing motion. Together, kinematics and kinetics describe the dynamics of the vehicle. Section 2.1.1 is based on modeling and notation from Fossen (2021), and Section 2.2 is based on modeling and notation from Spong et al. (2006) and From et al. (2014).

2.1 Kinematics

An underwater snake robot can be thought of as an underwater manipulator arm which consists of multiple rigid bodies. In order to describe the kinematics of the multi body snake robot, the navigation kinematics of a single body is first discussed, followed by a description of manipulator kinematics.

2.1.1 Navigation Kinematics

Position and orientation of a marine craft moving in six Degrees of Freedom (DOF) can be described using several reference frames. In general, $\boldsymbol{p}_{ij}^{k}$ denotes the position vector of frame $\{i\}$ with respect to frame $\{j\}$ expressed in frame $\{k\}$. The following reference frames are common for navigation of AUVs:

- North-East-Down (NED) frame $\{n\} = (x_n,\ y_n,\ z_n)$
- Body-fixed (BODY) frame $\{b\} = (x_b,\ y_b,\ z_b)$

The NED frame is a flat tangential plane fixed at the Earth's surface used for local navigation, typically an area of 10×10 km, with the origin o_n. It is also assumed that the NED frame is an inertial frame. An inertial reference frame is a non-accelerating reference frame in which Newton's laws of motion apply. The NED frame rotates with the Earth and is technically not inertial, however the rotation is neglected for marine crafts moving in low speed. The BODY frame is a moving coordinate frame that is fixed to the vehicle, and has the origin o_b. Position is usually expressed in the NED frame, and velocities are usually expressed in the BODY frame. Each DOF can be expressed with the notation of SNAME (1950), showed in Table 2.1. Figure 2.1 illustrates velocities in the BODY frame for an underwater vehicle. Equations 2.1 and 2.2 show the generalized position and velocity vectors respectively, where $\boldsymbol{\eta}_1$ represents position, and $\boldsymbol{\eta}_2$ represents attitude.

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_1^\top & \boldsymbol{\eta}_2^\top \end{bmatrix}^\top = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^\top \tag{2.1}$$

$$\boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{\nu}_1^\top & \boldsymbol{\nu}_2^\top \end{bmatrix}^\top = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^\top \tag{2.2}$$

		Forces and moments	$\boldsymbol{\nu}_1$, $\boldsymbol{\nu}_2$	$\boldsymbol{\eta}_1$, $\boldsymbol{\eta}_2$
Motion in the x-direction	Surge	X	u	x
Motion in the y-direction	Sway	Y	v	y
Motion in the z-direction	Heave	Z	w	z
Rotation about the x-axis	Roll	K	p	ϕ
Rotation about the y-axis	Pitch	M	q	θ
Rotation about the z-axis	Yaw	N	r	ψ

Table 2.1: Common notation for marine vehicle's motion

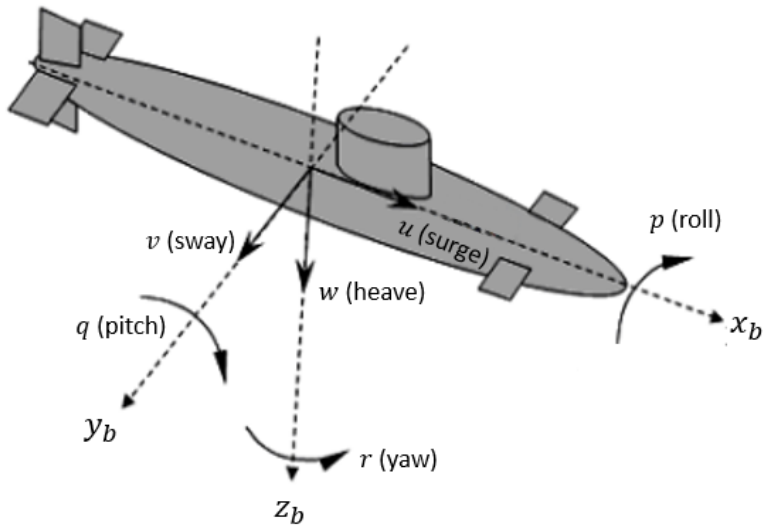


Figure 2.1: Degrees of freedom for an underwater vehicle (Yeo et al. 2014)

In order to transform between BODY and NED frames, a rotation matrix is used. A rotation matrix $\boldsymbol{R}$ is an element in $SO(3)$, which is the special orthogonal group of order 3, shown in Equation 2.3.

$$SO(3) := \{\boldsymbol{R} | \boldsymbol{R} \in \mathbb{R}^{3\times 3},\ \ \boldsymbol{R}^{-1} = \boldsymbol{R}^\top,\ det(\boldsymbol{R}) = 1\} \tag{2.3}$$

The transformation between reference frames are based on the Euler angles of the vessel. These are roll (ϕ), pitch (θ) and yaw (ψ) angles, and they are defined as the orientation of the vehicle with respect to the NED frame. Using the Euler angles, it is possible to express the body fixed linear velocity vector, $\boldsymbol{v}_{nb}^b$ in the NED frame, according to

$$\boldsymbol{v}_{nb}^n = \boldsymbol{R}_b^n \boldsymbol{v}_{nb}^b \tag{2.4}$$

where $\boldsymbol{R}_b^n$ is the rotation matrix from $\{b\}$ to $\{n\}$, $\boldsymbol{v}_{nb}^n = \dot{\boldsymbol{p}}_{nb}^n = [\dot{x}^n, \dot{y}^n, \dot{z}^n]^\top$ is the velocity vector expressed in NED, and $\boldsymbol{v}_{nb}^n$ is the velocity vector expressed in BODY.

$$\boldsymbol{R}_b^n = \boldsymbol{R}(\boldsymbol{\Theta}_{nb}) = (\boldsymbol{R}_n^b)^\top = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\theta s\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{2.5}$$

where $\boldsymbol{\Theta}_{nb} = [\phi, \theta, \psi]^\top$ is the vector containing the Euler angles, and $c(\cdot) = \cos(\cdot)$ and $s(\cdot) = \sin(\cdot)$.

Similar transformations can be performed for angular velocity. Let $\boldsymbol{\omega}_{nb}^b = [p, q, r]^\top$ be the body-fixed angular velocities, and $\boldsymbol{T}(\boldsymbol{\Theta}_{nb})$ be the transformation matrix that relates body-fixed angular velocities and Euler rates as shown in Equation 2.6.

$$\dot{\boldsymbol{\Theta}}_{nb} = \boldsymbol{T}(\boldsymbol{\Theta}_{nb})\boldsymbol{\omega}_{nb}^b \tag{2.6}$$

$$\boldsymbol{T}(\boldsymbol{\Theta}_{nb}) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \tag{2.7}$$

where $t(\cdot) = \tan(\cdot)$. Thus the 6-DOF kinematic equations are

$$\begin{gathered} \dot{\boldsymbol{\eta}} = \boldsymbol{J}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu} \\ \Updownarrow \\ \begin{bmatrix} \dot{\boldsymbol{p}}_{nb}^n \\ \dot{\boldsymbol{\Theta}}_{nb} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{\Theta}_{nb}) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{T}(\boldsymbol{\theta}_{nb}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}_{nb}^b \\ \boldsymbol{\omega}_{nb}^b \end{bmatrix} \end{gathered} \tag{2.8}$$

where $\boldsymbol{\eta} \in \mathbb{R}^6$ is the position and orientation in the NED frame,
$\boldsymbol{J}_\Theta(\boldsymbol{\eta}) \in \mathbb{R}^{6\times6}$ is the Jacobian matrix that relates the linear and angular velocities in the NED frame and the BODY frame. $\boldsymbol{\nu} \in \mathbb{R}^6$ is the velocity in the BODY frame.

Additional tools that are useful for describing kinematics are skew-symmetric matrices and unit quaternions. Skew-symmatric matrices provide a method of calculating the cross-product of two vectors without using the cross-product-operator ($\times$). Let $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]^\top$. A skew-symmetric matrix, $\boldsymbol{S}$, is then defined as

$$\boldsymbol{S}(\boldsymbol{\lambda}) = -\boldsymbol{S}^\top(\boldsymbol{\lambda}) \begin{bmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{bmatrix} \tag{2.9}$$

The cross product of two vectors can now be calculated as

$$\boldsymbol{\lambda} \times \boldsymbol{a} = \boldsymbol{S}(\boldsymbol{\lambda})\boldsymbol{a} \tag{2.10}$$

where $\boldsymbol{a} = [a_1, a_2, a_3]^\top$ is a vector on the same form as λ.

Unit quaternions allow the kinematic equations to be represented without using Euler angles. Notice that when the pitch angle $\theta = \pm 90°$, the transformation matrix $\boldsymbol{T}(\boldsymbol{\Theta}_{nb})$ is not defined. Therefore unit quaternions are introduced to improve robustness. A quaternion, $\boldsymbol{q}$, is a complex number that consists of one real part, η, and three imaginary parts $\boldsymbol{\varepsilon} = [\varepsilon_1, \varepsilon_2, \varepsilon_3]^\top$. The set Q of unit quaternions can be defined as

$$Q := \{\boldsymbol{q} | \boldsymbol{q}^\top \boldsymbol{q} = 1,\ \boldsymbol{q} = [\eta, \boldsymbol{\varepsilon}^\top]^\top,\ \ \eta \in \mathbb{R},\ \boldsymbol{\varepsilon} \in \mathbb{R}^3\} \tag{2.11}$$

In the case for an underwater snake robot consisting of multiple rigid bodies with coordinate frames rigidly attached to each body, there is a high chance of entering a body configuration where at least one of the coordinate frames are oriented with a 90° pitch angle. Thus the use of unit quaternions are necessary for representing the orientation of each body in order to avoid cases where transformations are not defined. On the other hand, Euler angles are more intuitive for describing orientation to a human. Therefore it is useful to be able to represent the relationship between Euler angles and unit quaternions. It is possible to transform unit quaternions to Euler angles with the following equations:

$$\boldsymbol{\Theta}_{nb} = \begin{bmatrix} atan2(2(\varepsilon_2\varepsilon_3 + \varepsilon_1\eta), 1 - 2(\varepsilon_1^2 + \varepsilon_2^2)) \\ -asin(2(\varepsilon_1\varepsilon_3 - \varepsilon_2\eta)) \\ atan2(2(\varepsilon_1\varepsilon_2 + \varepsilon_3\eta), 1 - 2(\varepsilon_2^2 + \varepsilon_3^2)) \end{bmatrix} \tag{2.12}$$

Similarly, unit quaternions can be found from the Euler angles as follows (Janota et al. 2015):

$$\boldsymbol{q} = \begin{bmatrix} c\left(\frac{1}{2}\psi\right) c\left(\frac{1}{2}\theta\right) c\left(\frac{1}{2}\phi\right) + s\left(\frac{1}{2}\psi\right) s\left(\frac{1}{2}\theta\right) s\left(\frac{1}{2}\phi\right) \\ c\left(\frac{1}{2}\psi\right) c\left(\frac{1}{2}\theta\right) s\left(\frac{1}{2}\phi\right) - s\left(\frac{1}{2}\psi\right) s\left(\frac{1}{2}\theta\right) c\left(\frac{1}{2}\phi\right) \\ s\left(\frac{1}{2}\psi\right) c\left(\frac{1}{2}\theta\right) s\left(\frac{1}{2}\phi\right) + c\left(\frac{1}{2}\psi\right) s\left(\frac{1}{2}\theta\right) c\left(\frac{1}{2}\phi\right) \\ s\left(\frac{1}{2}\psi\right) c\left(\frac{1}{2}\theta\right) c\left(\frac{1}{2}\phi\right) - c\left(\frac{1}{2}\psi\right) s\left(\frac{1}{2}\theta\right) s\left(\frac{1}{2}\phi\right) \end{bmatrix} \tag{2.13}$$

Equation 2.13 may be helpful in a case where a user is able to give a setpoint to a control system, as a human will find it more intuitive to define a desired angle using Euler angles.

When paramterized with quaternions, the rotation matrix, $\boldsymbol{R}$, and transformation matrix, $\boldsymbol{T}$, become

$$\boldsymbol{R}(\boldsymbol{q}_b^n) = \boldsymbol{I}_3 + 2\eta\boldsymbol{S}(\boldsymbol{\varepsilon}) + 2\boldsymbol{S}^2(\boldsymbol{\varepsilon}) \tag{2.14}$$

$$\boldsymbol{T}(\boldsymbol{q}_b^n) = \frac{1}{2}\begin{bmatrix} -\boldsymbol{\varepsilon}^\top \\ \eta\boldsymbol{I}_3 + \boldsymbol{S}(\boldsymbol{\varepsilon}) \end{bmatrix} \tag{2.15}$$

where $\boldsymbol{I}_3$ is the identity matrix, and $\boldsymbol{S}(\boldsymbol{\varepsilon})$ is the skew-symmetric matrix shown in Equation 2.9. The 6-DOF kinematic equations can now be rewritten as

$$\begin{gathered} \dot{\boldsymbol{\eta}} = \mathbf{J}_q(\boldsymbol{\eta})\boldsymbol{\nu} \\ \Updownarrow \\ \begin{bmatrix} \dot{\boldsymbol{p}}_{nb}^n \\ \dot{\boldsymbol{q}}_b^n \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{q}_b^n) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{4\times3} & \boldsymbol{T}(\boldsymbol{q}_b^n) \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}_{nb}^b \\ \boldsymbol{\omega}_{nb}^b \end{bmatrix} \end{gathered} \tag{2.16}$$

where $\boldsymbol{\eta} \in \mathbb{R}^7$ is the position and orientation of the NED frame, $\mathbf{J}_q(\boldsymbol{\eta}) \in \mathbb{R}^{7\times6}$ is the Jacobian and $\boldsymbol{\nu} \in \mathbb{R}^6$ is the linear and angular velocity of the BODY frame.

2.2 Manipulator Kinematics

An underwater snake robot can be modeled as an underwater swimming manipulator, which in essence is a floating robotic manipulator arm. Robot manipulators consist of multiple links interconnected by joints. Links are rigid bodies, and joints are typically rotary (revolute) or linear (prismatic). Only revolute joints, will be considered in this thesis. These joints can be thought of as hinges that allow the links to rotate about a given direction. This is illustrated in Figure 2.2. A reference frame is attached to each link, and the joints allow relative rotation between two links and their respective reference frames. Therefore it is necessary to be able to express the position of the reference frames in relation to each other, in order to determine the configuration of the manipulator. Additionally, the positions and orientations of the links of the robot can be described with respect to a fixed inertial frame, which means that the position and orientation of each link can be determined in relation to the NED frame. In robotics, position and orientation are often referred to as the collective term *pose*. Similarly *twist* is the collective term for linear and angular velocities.

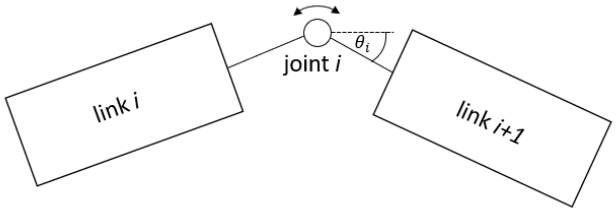


Figure 2.2: Links connected by revolute joint

2.2.1 Homogeneous Transformations

Representations of positions and orientations can be combined to represent rigid motions. A rigid motion is defined as an ordered pair $(\boldsymbol{d}, \boldsymbol{R})$ where $\boldsymbol{d} \in \mathbb{R}^3$ and $\boldsymbol{R} \in SO(3)$. Rigid motions are elements of the special euclidean group defined in Equation 2.17.

$$SE(3) := \left\{ \begin{bmatrix} \boldsymbol{R} & \boldsymbol{d} \\ \boldsymbol{0} & 1 \end{bmatrix} \middle| \, \boldsymbol{R} \in SO(3), \boldsymbol{d} \in \mathbb{R}^3 \right\} \tag{2.17}$$

To illustrate, let the rotation matrix $\boldsymbol{R}_1^0$ specify the orientation of frame $\{1\}$ with respect to frame $\{0\}$. Let $\boldsymbol{d}_1^0$ be the vector from the origin of frame $\{0\}$ to the origin of frame $\{1\}$. Let $\boldsymbol{p}$ be a vector attached to frame $\{1\}$ with the coordinates $\boldsymbol{p}^1$ with respect to $\{1\}$. The coordinates of $\boldsymbol{p}$ can then be expressed in $\{0\}$ using Equation 2.18, where the relationship between $\boldsymbol{p}^0$ and $\boldsymbol{p}^1$ is a rigid motion.

$$\boldsymbol{p}^0 = \boldsymbol{R}_1^0 \boldsymbol{p}^1 + \boldsymbol{d}_1^0. \tag{2.18}$$

In general, rigid motion can be represented by a matrix:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{d} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \tag{2.19}$$

where $\boldsymbol{H} \in SE(3)$ is called a homogeneous transformation matrix. In other words, $\boldsymbol{H}$ can be used to represent the relative poses of two coordinate frames. The homogeneous transformation matrix can then be used to rewrite Equation 2.18 as

$$\begin{bmatrix} \boldsymbol{p}^0 \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_1^0 & \boldsymbol{d}_1^0 \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}^1 \\ 1 \end{bmatrix} \tag{2.20}$$

2.2.2 Forward Kinematics

Forward kinematics considers the problem of determining the pose of the end effector when the joint angles are known. Let $\boldsymbol{q} = [\theta_1, \ldots, \theta_n]^\top$ be the vector containing all the joint angles of a joint with 1 DOF. Link i is connected to link $i+1$ through joint i, and when joint i is actuated, link $i+1$ and the attached coordinate frame rotates with respect to link i. This transformation can be represented with the homogeneous transformation matrix $\boldsymbol{A}_i(\theta_i) \in SE(3)$ defined in Equation 2.21.

$$\boldsymbol{A}_i(\theta_i) = \boldsymbol{A}_i(0) \begin{bmatrix} \boldsymbol{R}_{s_i,\theta_i} & \boldsymbol{0}_{3\times 1} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \tag{2.21a}$$

$$\boldsymbol{A}_i(0) = \begin{bmatrix} \boldsymbol{I}_3 & l_i \boldsymbol{e}_1 \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \tag{2.21b}$$

l_i is the length of body link i, and $\boldsymbol{e}_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\top$ is the local x-axis defined along the length direction of body link i. The subscript s_i in the rotation matrix $\boldsymbol{R}$ denotes which axis joint i rotates about. Rotation matrices about the x, y and z-axis are shown in Equation 2.22.

$$\boldsymbol{R}_{x,\theta_i} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\theta_i & -s\theta_i \\ 0 & s\theta_i & c\theta_i \end{bmatrix} \tag{2.22a}$$

$$\boldsymbol{R}_{y,\theta_i} = \begin{bmatrix} c\theta_i & 0 & s\theta_i \\ 0 & 1 & 0 \\ -s\theta_i & 0 & c\theta_i \end{bmatrix} \tag{2.22b}$$

$$\boldsymbol{R}_{z,\theta_i} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 \\ s\theta_i & c\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.22c}$$

With these tools, the pose of frame $\{j\}$ with respect to frame $\{i\}$ can be defined as a transformation $\boldsymbol{T}_{i,j}$, which is a product of homogeneous transformations $\boldsymbol{A}_i(\theta_i)$. $\boldsymbol{T}_{i,j}$ is defined recursively in Equation 2.23.

$$\boldsymbol{T}_{i,j}(\boldsymbol{q}) = \begin{cases} \boldsymbol{A}_{i+1}(\theta_{i+1})\boldsymbol{A}_{i+2}(\theta_{i+2})\boldsymbol{A}_{j-1}(\theta_{j-1})\boldsymbol{A}_j(\theta_j), & \text{if } i < j \\ \boldsymbol{I}_4, & \text{if } i = j \\ (\boldsymbol{T}_{j,i})^{-1} & \text{if } i > j \end{cases} \tag{2.23}$$

Finally, the pose of link i can be expressed in the inertial frame, here denoted as $\{0\}$, with the following recursive equation:

$$\boldsymbol{H}_0 = \boldsymbol{H} \tag{2.24a}$$

$$\boldsymbol{H}_{i+1} = \boldsymbol{H}\boldsymbol{A}_1(\theta_1)\boldsymbol{A}_2(\theta_2)\ldots\boldsymbol{A}_i(\theta_i) = \boldsymbol{H}\boldsymbol{T}_{1,i}(\boldsymbol{q}) \tag{2.24b}$$

where $\boldsymbol{H}$ is given in Equation 2.19.

Note that for EELY500, each joint has 2 DOF, while the kinematics so far has only been given for manipulators where each joint has 1 DOF. Using theory from Spong et al. (2006), it is possible to describe rotations about multiple axis as compositions of rotations with the matrices shown in Equation 2.22. Although the number of joints remain the same, the vector containing joint angles, $\boldsymbol{q}$, must be redefined. The joint angles can now be given as $\boldsymbol{q} = [\theta_1, \psi_1 \dots \theta_{N-1}, \psi_{N-1}]^\top \in \mathbb{R}^{2N-2}$, where θ is rotation about the y-axis, and ψ is rotation about the z-axis, i.e. pitch and yaw rotations respectively. Similarly to a transformation, $\boldsymbol{T}_{i,j}$, a composition of rotations can be defined as products of $\boldsymbol{R}_{y,\theta}$ and $\boldsymbol{R}_{z,\psi}$. It is important to remember that when one rotation is performed, the next rotation will be performed about an axis that is different from the original axis. This is referred to as rotation about current axes. As a consequence, the order of rotations is important, because $\boldsymbol{R}_{y,\theta}\boldsymbol{R}_{z,\psi} \neq \boldsymbol{R}_{z,\psi}\boldsymbol{R}_{y,\theta}$. In order to get around this, it is possible to define rotations with respect to a fixed frame. In this case, the fixed frame is the frame attached to link i. The difference between composition of rotations about current and fixed axes is illustrated by Figure 2.3 and Figure 2.4. The composition of a rotation about fixed axes is given by

$$\boldsymbol{R} = \boldsymbol{R}_{y,\theta}[\boldsymbol{R}_{y,-\theta}\boldsymbol{R}_{z,\psi}\boldsymbol{R}_{y,\theta}] = \boldsymbol{R}_{z,\psi}\boldsymbol{R}_{y,\theta} \tag{2.25}$$

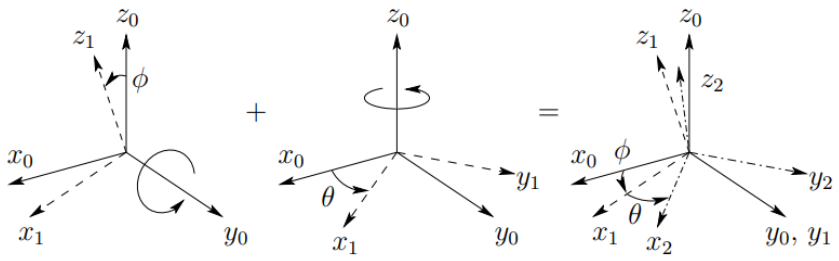


Figure 2.3: Composition of rotation about current axes (Spong et al. 2006)

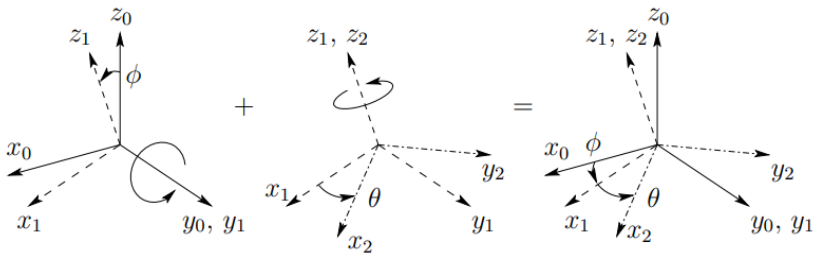


Figure 2.4: Composition of rotation about fixed axes (Spong et al. 2006)

On a side note, $\boldsymbol{R}(\boldsymbol{\Theta}_{nb})$ from Equation 2.5 is found as a composition of rotations about fixed axes, and is defined as

$$\boldsymbol{R}(\boldsymbol{\Theta}_{nb}) = \boldsymbol{R}_{z,\psi}\boldsymbol{R}_{y,\theta}\boldsymbol{R}_{x,\phi} \tag{2.26}$$

which means that alternatively, Equation 2.25 could be found by inserting $\phi = 0$ into $\boldsymbol{R}(\boldsymbol{\Theta}_{nb})$.

2.2.3 Differential Kinematics

The differential kinematics consider the relationships between the twists of the link frames when the joints are actuated. Let $\boldsymbol{V}_i = [\boldsymbol{v}_i^\top, \boldsymbol{\omega}_i^\top]^\top$ denote the generalized linear and angular velocities of link i. $\boldsymbol{V}_i$ is related to the joint velocities in the following equation:

$$\boldsymbol{V}_i = \boldsymbol{J}_i \dot{\boldsymbol{q}} \tag{2.27}$$

where $\boldsymbol{J}_i$ is the Jacobian for body i, and $\dot{\boldsymbol{q}}$ is the joint velocity vector. Featherstone (2008) defines a general Jacobian as:

$$\boldsymbol{J}_i = \begin{bmatrix} \epsilon_{i1}\boldsymbol{S}_1 & \epsilon_{i2}\boldsymbol{S}_2 & \dots & \epsilon_{iN}\boldsymbol{S}_N \end{bmatrix} \tag{2.28}$$

where N is the total number of links, S_i is a matrix containing unit rotation vectors, and ϵ_{ij} is given by Equation 2.29.

$$\epsilon_{ij} = \begin{cases} 1 & \text{if } j \in \kappa(i) \\ 0 & \text{otherwise} \end{cases} \tag{2.29}$$

κ is determined by how the links are connected, in short ϵ_{ij} is nonzero if joint j supports link i. There are several ways to determine the Jacobian presented in the literature. Methods for determining the Jacobian are shown in the following section.

2.3 Snake Robot Kinematics

With the modeling presented in this chapter so far, it is possible to set up a model for an underwater snake robot using the framework presented in Schmidt-Didlaukies et al. (2018). The snake robot is modeled with N rigid links, connected by $N-1$ revolute joints. All links are assigned a coordinate frame $\{i\}, i \in \{1, 2, \dots, N\}$. These frames are typically placed either on the joint axes, or at the Center of Gravity (CG) of their respective links (From et al. 2014). In this case, the frames placed at the CG of each link as shown in Figure 2.6. However, since all the links are rigid, the frames could be placed anywhere on their respective links without having an effect on the model. The frame $\{0\}$ is the inertial reference frame, which in this case is the NED frame. Frame $\{1\}$ defines the head link of the snake, and the end link, $\{N\}$, defines the tail. It is also necessary to define a base link that connects to the inertial frame. The structure of how links and joints are connected, can be represented by a kinematic tree, as shown in Figure 2.5. In this case, the base link is chosen as link 1, which results in a simple straight kinematic tree. If the base link is chosen for example in one of the middle links, the kinematic tree would have two arms attached to the base link. Additionally, it is convenient to define a BODY frame for representing position an orientation of the entire vehicle. Where this frame is placed has consequences for the control system of the vehicle, which will be discussed further in the following chapters. For the time being, the BODY frame is chosen to coincide with frame $\{1\}$ to reduce the amount of coordinate frames to keep track of. The shape, or the configuration, of the vehicle is determined by the joint angles $\boldsymbol{q}$. Each link is modeled as a rigid cylinder with length l_i and radius r_i.

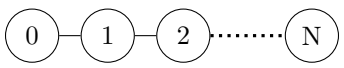


Figure 2.5: Kinematic tree

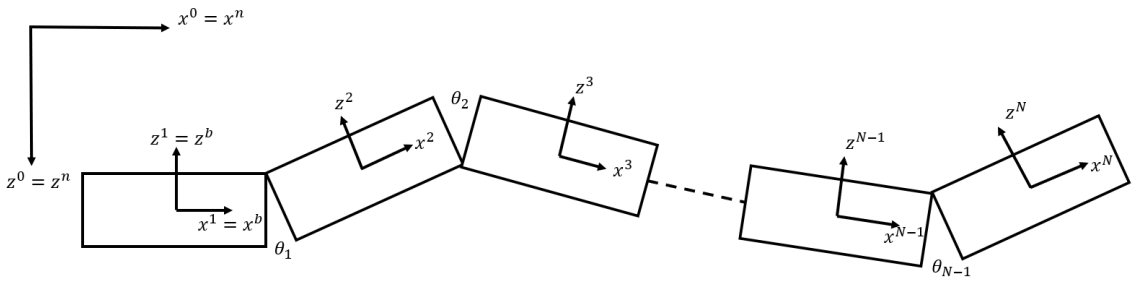


Figure 2.6: Snake robot with coordinate frames

The pose of link i with respect to the inertial frame is represented by the homogeneous transformation matrix $\boldsymbol{H}_i$, which is given in Equation 2.24. Transformation between frame $\{i\}$ and the inertial frame can be described with the map $\mathrm{Ad}(\boldsymbol{H}_i)$:

$$\mathrm{Ad}(\boldsymbol{H}_i) = \begin{bmatrix} \boldsymbol{R}_i & \boldsymbol{S}(\boldsymbol{d}_i)\boldsymbol{R}_i \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{R}_i \end{bmatrix} \in \mathbb{R}^{6\times 6} \tag{2.30}$$

$$\mathrm{Ad}^{-1}(\boldsymbol{H}_i) = \begin{bmatrix} \boldsymbol{R}_i^\top & -\boldsymbol{R}_i^\top \boldsymbol{S}(\boldsymbol{d}_i) \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{R}_i^\top \end{bmatrix} \tag{2.31}$$

The 6 DOF body velocities of link i can be calculated recursively as shown in Equation 2.32.

$$\boldsymbol{\nu}_1 = \boldsymbol{\nu} \tag{2.32a}$$

$$\boldsymbol{\nu}_{i+1} = \mathrm{Ad}(\boldsymbol{H}_i)(\boldsymbol{A}_i(\boldsymbol{\theta}_i))\boldsymbol{\nu}_i + \boldsymbol{a}_i\dot{\boldsymbol{\theta}}_i \tag{2.32b}$$

$\boldsymbol{a}_i \in \mathbb{R}^6$ is the joint twist of joint i. For a joint rotating about the y-axis and the z-axis, $\boldsymbol{a}_i$ is:

$$\boldsymbol{a}_i = \mathrm{Ad}(\boldsymbol{H}_i)\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}^\top \tag{2.33}$$

Define $\boldsymbol{\zeta} = [\boldsymbol{\nu}^\top, \dot{\boldsymbol{q}}^\top]^\top$ as the vector containing the body and joint velocities. The body velocity of link i can be related to $\boldsymbol{\zeta}$ as:

$$\boldsymbol{\nu}_i = \boldsymbol{J}_i(\boldsymbol{q})\boldsymbol{\zeta} \tag{2.34}$$

where $\boldsymbol{J}_i(\boldsymbol{q})$ is the Jacobian, which can be calculated recursively as shown in Equation 2.35.

$$\boldsymbol{J}_1 = \begin{bmatrix} \boldsymbol{I}_6 & \boldsymbol{0}_{6\times 1} & \dots & \boldsymbol{0}_{6\times 1} \end{bmatrix} \in \mathbb{R}^{6\times N+5} \tag{2.35a}$$

$$\boldsymbol{J}_{i+1} = \mathrm{Ad}^{-1}(\boldsymbol{A}_i(\theta_i))\boldsymbol{J}_i + \begin{bmatrix} \boldsymbol{0}_{6\times 1} & \dots & \boldsymbol{0}_{6\times 1} & \boldsymbol{a}_i & \boldsymbol{0}_{6\times 1} & \dots & \boldsymbol{0}_{6\times 1} \end{bmatrix} \tag{2.35b}$$

Alternatively, From et al. (2014) presents non-recursive method of determining the the Jacobian. Here, the Jacobian relating the velocities of link i to $\boldsymbol{\zeta}$ is referred to as the body geometric Jacobian, shown in Equation 2.36.

$$\boldsymbol{\nu}_i = \boldsymbol{J}_i^B(\boldsymbol{q})\boldsymbol{\zeta} \tag{2.36}$$

The body geometric Jacobian is found according to

$$\boldsymbol{J}_i^B(\boldsymbol{q}) = \begin{bmatrix} Ad_{bi}^{-1} & Ad_{bi}^{-1}\boldsymbol{J}_i(\boldsymbol{q}) \end{bmatrix} \tag{2.37}$$

where

$$\boldsymbol{J}_i(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \dots & \boldsymbol{a}_i & \boldsymbol{0}_{6\times(N-i)} \end{bmatrix} \tag{2.38}$$

N is the number of links, and $Ad_{bi}^{-1} = Ad^{-1}(\boldsymbol{T}_{bi})$ similarly to Equation 2.31, where $\boldsymbol{T}_{i,j}(\boldsymbol{q})$ is given in Equation 2.23. The subscript b refers to the BODY frame, and the subscript i refers to link i.

Notice that a snake robot has more degrees of freedom than what is needed for a given task. For example if there is a camera module at the head of the robot, there is an infinite amount of solutions to position the camera a certain way. This may cause problems for dynamics analysis, as decisions must be made for how the redundant degrees of freedom are used. One can combat this by imposing constraints on the system, such as limiting the joints (Josephs and Huston 2002). In the case for an underwater snake robot, thrusters can be used to move the robot and solve given tasks. The joints could be considered fixed, and only change angles to user specified command, and the robot could achieve goals by only using the thrusters. This will be the case for the simulation case studies performed in Chapter 4 of the thesis. Constraint forces will therefore not be further considered.

2.4 Kinetics

The kinetics of a multi body system can be defined either such that the Equations of Motion (EOM) are solved recursively for each body, or in matrix form where the EOM are solved for every rigid body at once. The latter method is arguably the simplest method, but may not be applicable for a system with a high number of links. Both Schiehlen (2006) and Anderson (1990) argue that for a large number of bodies, the equations of motion are computationally expensive, such that the recursive method is more efficient. Additionally, for a large number of bodies, large deformations may appear that can not be neglected such that inaccuracies appear in the system. In this case, the number of links is relatively small, and the kinematic structure is simple. Thus it is sufficient to model the kinetics in matrix form, where the system matrices are configuration dependent. Literature for robotic manipulators where the joints can rotate in multiple DOF is limited. Therefore, this model is simplified such that each joint only rotates in 1 DOF, however each joint can be specified to rotate about either the y-axis or the z-axis. The EOM are given below:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_q(\boldsymbol{\eta})\boldsymbol{\nu} \tag{2.39a}$$

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\zeta}} + \boldsymbol{C}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{g}(\boldsymbol{\eta},\boldsymbol{q}) = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{u} + \boldsymbol{\tau} \tag{2.39b}$$

where

- $\boldsymbol{\eta} \in \mathbb{R}^7$ is the position vector in the NED frame.
- $\boldsymbol{\nu} \in \mathbb{R}^6$ is the velocity vector in the BODY frame.
- $\boldsymbol{J}_q(\boldsymbol{\eta}) \in \mathbb{R}^{7\times 6}$ is the transformation matrix.
- $\boldsymbol{q} = [\theta_1, \theta_2 \dots \theta_{N-1}]^\top \in \mathbb{R}^{N-1}$ is the vector containing joint angles.
- $\boldsymbol{\zeta} = [\boldsymbol{\nu}^\top, \dot{\boldsymbol{q}}^\top]^\top \in \mathbb{R}^{N+5}$ is the vector containing body and joint velocities.
- $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{6\times N+5}$ is the rigid-body system inertia matrix, including added mass.
- $\boldsymbol{C}(\boldsymbol{q},\boldsymbol{\zeta}) \in \mathbb{R}^{6\times N+5}$ is the rigid-body Coriolis and centripetal matrix.
- $\boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta}) \in \mathbb{R}^{6\times N+5}$ is the hydrodynamic damping matrix.
- $\boldsymbol{g}(\boldsymbol{\eta},\boldsymbol{q}) \in \mathbb{R}^6$ is the restoring vector.
- $\boldsymbol{B}(\boldsymbol{q}) \in \mathbb{R}^{6\times N-1+r}$ is the input map for the joint motor and thruster forces.
- $\boldsymbol{u} \in \mathbb{R}^{N-1+r}$ is the vector containing inputs to the joint motors and thrusters.
- $\boldsymbol{\tau} \in \mathbb{R}^6$ is the vector containing the sums of all external forces and moments. In this case $\boldsymbol{\tau}$ comes from environmental forces. For the actual vehicle, a force from the tether should also be included, however this is neglected for the simulation model.
- $N \in \mathbb{N}$ is the number of links.
- $r \in \mathbb{N}$ is the number of thrusters.

The total system inertia matrix is computed as

$$\boldsymbol{M}(\boldsymbol{q}) = \sum_{i=1}^{n} \boldsymbol{J}_i^\top(\boldsymbol{q})\boldsymbol{M}_i\boldsymbol{J}_i(\boldsymbol{q}) \tag{2.40}$$

For each link, the mass matrix, $\boldsymbol{M}_i$, is the sum of the rigid body matrix, $\boldsymbol{M}_{RB_i}$, and the added mass matrix, $\boldsymbol{M}_{A_i}$. $\boldsymbol{M}_{RB_i}$ is modeled as shown in Equation 2.41. The modeling of $\boldsymbol{M}_{A_i}$ is discussed in Section 2.5.

$$\boldsymbol{M}_{RB_i} = \begin{bmatrix} m_i \boldsymbol{I}_3 & m_i \boldsymbol{S}^\top(\boldsymbol{r}^b_{bg,i}) \\ m_i \boldsymbol{S}(\boldsymbol{r}^b_{bg,i}) & \boldsymbol{I}^b_{b,i} \end{bmatrix} \tag{2.41}$$

where m_i is the dry mass of link i, $\boldsymbol{r}^b_{bg,i}$ is the vector from the Center of Origin (CO) to CG expressed in the BODY frame, and $\boldsymbol{I}^b_{b,i}$ is the inertia dyadic. $\boldsymbol{I}^b_{b,i}$ is written as

$$\boldsymbol{I}^b_b = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{bmatrix} \tag{2.42}$$

For a circular cylinder with mass, m, radius, r, length, l, and axes placed as shown in Figure 2.7:

$$I_x = I_y = \frac{1}{12} m l^2 + \frac{1}{4} m r^2 \tag{2.43a}$$

$$I_z = \frac{1}{2} m r^2 \tag{2.43b}$$

$$I_{xy} = I_{xz} = I_{yx} = I_{yz} = I_{zx} = I_{zy} = 0 \tag{2.43c}$$

The second therm in Equation 2.43a appears according to the parallel axis theorem since the cylinder rotates about an axis placed on a joint at the base of the cylinder (Josephs and Huston 2002).

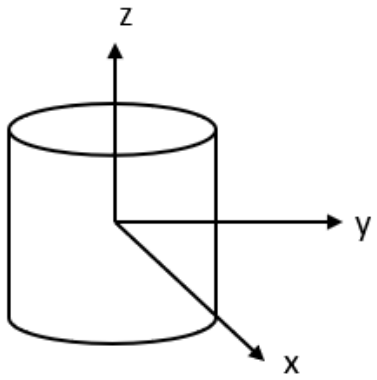


Figure 2.7: Cylinder with axes

The Coriolis and centripetal matrix appears due to rotation of the body-fixed frame about the inertial frame, and is found according to Equation 2.44 for the multi body system.

$$\boldsymbol{C}(\boldsymbol{q}, \boldsymbol{\zeta}) = \sum_{i=1}^{n} \left[\boldsymbol{J}_i^\top(\boldsymbol{q}) \boldsymbol{M}_i \dot{\boldsymbol{J}}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{J}_i^\top(\boldsymbol{q}) \boldsymbol{W}_i(\boldsymbol{q}, \boldsymbol{\zeta}) \boldsymbol{J}_i(\boldsymbol{q}) \right] \tag{2.44}$$

where

$$\boldsymbol{W}_i(\boldsymbol{q}, \boldsymbol{\zeta}) = \begin{bmatrix} \boldsymbol{0}_{3\times 3} & \boldsymbol{S}\left(\{\boldsymbol{M}_i \boldsymbol{\nu}_i\}_{(1:3)}\right) \\ \boldsymbol{S}\left(\{\boldsymbol{M}_i \boldsymbol{\nu}_i\}_{(1:3)}\right) & \boldsymbol{S}\left(\{\boldsymbol{M}_i \boldsymbol{\nu}_i\}_{(4:6)}\right) \end{bmatrix} \tag{2.45}$$

The total damping matrix and restoring vector are shown in Equation 2.46 and Equation 2.47 respectively. Modeling of $\boldsymbol{D}_i(\boldsymbol{\zeta})$ and $\boldsymbol{g}_i(\boldsymbol{\eta})$ is discussed in the following section.

$$\boldsymbol{D}(\boldsymbol{q}, \boldsymbol{\zeta}) = \sum_{i=1}^{n} \boldsymbol{J}_i^{\top}(\boldsymbol{q}) \boldsymbol{D}_i(\boldsymbol{\zeta}) \boldsymbol{J}_i(\boldsymbol{q}) \tag{2.46}$$

$$\boldsymbol{g}(\boldsymbol{\eta}, \boldsymbol{q}) = \sum_{i=1}^{n} \boldsymbol{J}_i^{\top}(\boldsymbol{q}) \boldsymbol{g}_i(\boldsymbol{\eta}) \tag{2.47}$$

2.5 Hydrodynamics

A body submerged in water will experience hydrodynamic forces. The theory behind hydrodynamics is complex, and developing reliable models for hydrodynamic effects is a challenging task. Therefore a considerable amount of simplifications are made for modeling of hydrodynamic effects when simulating control systems. There are several ways of modeling added mass and damping effects, some of which are explored in this section. Note that a cylinder is symmetrical about two axes, thus the hydrodynamic effects in sway and heave can be modelled as equal to each other. This also applies for effects in pitch and yaw. To simplify the hydrodynamic modeling, it is assumed that the links are hydrodynamically decoupled. This means that the hydrodynamic effects on one link caused by the adjacent links are neglected, such that the added mass and damping matrices are independent of the body configuration. Research conducted by Kanso et al. (2005) concludes that models for an articulated body swimming in a perfect fluid achieve larger net locomotion when the links are considered hydrodynamically coupled than models where links are considered hydrodynamically decoupled. This result suggests that some accuracy is lost when the links are assumed hydrodynamically decoupled.

2.5.1 Added Mass

Added mass appears when a rigid body moves in a fluid since the movement of the body accelerates the fluid surrounding the body, creates additional inertia. This effect also causes the appearance of an added Coriolis and centripetal contribution, however this effect is accounted for in Equation 2.44, because the added mass matrix is included in $\boldsymbol{M}_i$. The hydrodynamic forces highly depend on the geometry of the body, and the hydrodynamic force along x_b due to linear acceleration in the x_b-direction is defined as:

$$X_a = -X_{\dot{u}} \dot{u} \tag{2.48}$$

where

$$X_{\dot{u}} = \frac{\partial X}{\partial \dot{u}} \tag{2.49}$$

Similar definitions apply to forces and moments in every direction due to acceleration in every direction, which can be grouped in the added mass matrix $\boldsymbol{M}_A \in \mathbb{R}^{6\times 6}$. For underwater vehicles it is common to only consider the added mass effects on the diagonal, such that the added mass matrix can be modeled as:

$$\boldsymbol{M}_A = -diag\{X_{\dot{u}},\ Y_{\dot{v}},\ Z_{\dot{w}},\ K_{\dot{p}},\ M_{\dot{q}},\ N_{\dot{r}}\} \tag{2.50}$$

There are several ways of modeling the elements in Equation 2.50. Antonelli (2014) provides the following model for a cylindrical rigid body with mass, m, length, L and radius, r, in a fluid with density, ρ:

$$X_{\dot{u}} = -0.1m \tag{2.51a}$$
$$Y_{\dot{v}} = Z_{\dot{w}} = -\pi\rho r^2 L \tag{2.51b}$$
$$K_{\dot{p}} = 0 \tag{2.51c}$$
$$M_{\dot{q}} = N_{\dot{r}} = -\frac{1}{12}\pi\rho r^2 L^3 \tag{2.51d}$$

On the other hand, Fossen (2021) shows a model for a body with a Myring-type hull, which is a hull with a shape akin to a torpedo. The Myring-type hull can then be approximated as an ellipsoid shape. In this case, the elements in Equation 2.50 are modeled as

$$X_{\dot{u}} = -\frac{\alpha_0}{2-\alpha_0}m \tag{2.52a}$$
$$Y_{\dot{v}} = Z_{\dot{w}} = -\frac{\beta_0}{2-\beta_0}m \tag{2.52b}$$
$$K_{\dot{p}} = 0 \tag{2.52c}$$
$$M_{\dot{q}} = N_{\dot{r}} = -\frac{1}{5}\frac{(b^2-a^2)^2(\alpha_0-\beta_0)}{2(b^2-a^2)+(b^2+a^2)(\beta-\alpha_0)}m \tag{2.52d}$$

where

$$\alpha_0 = \frac{2(1-e^2)}{e^3}\left(\frac{1}{2}\ln\frac{1+e}{1-e} - e\right) \tag{2.53a}$$
$$\beta_0 = \frac{1}{e^2} - \frac{1-e^2}{2e^3}\ln\frac{1+e}{1-e} \tag{2.53b}$$

where

$$e = 1 - \frac{b^2}{a^2} \tag{2.54}$$

The ellipsoid shape with the quantities a and b is illustrated in Figure 2.8. a is half the length of the vehicle, and b is the radius.

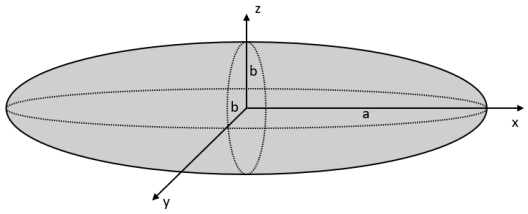


Figure 2.8: Ellipsoid with semi-axes a and b

By examining the model in Figure 4.8, it can be argued that the Eelume robot most resembles a Myring-type hull when all the joints are straightened, however for other body configurations, it might make more sense to model every link except for the head and the tail as regular cylinders. Additionally, the head link and the tail link will have one flat end. Therefore each link will be considered as a regular cylinder, and the added mass for each link will be modeled according to Equation 2.51. It should also be noted than when all the joints are straight, the vehicle could either be considered as one body, or multiple bodies in a straight line. If the vehicle is considered as one body, there will be one added mass matrix for the entire system, but if the vehicle is considered

as multiple bodies, the added mass matrices are added together. This has large implications when considering Equation 2.51d, because the length is cubed, and whether the total length or divided lengths are considered could result in vastly different values for added mass, as shown in the inequality below.

$$\left(\sum_{i=1}^{N} l_i\right)^3 > \sum_{i=1}^{N} l_i^3 \tag{2.55}$$

Of course, this only applies when all the joints are straightened, and for other body configurations the vehicle should without a doubt be considered as a multi body system. However this illustrates the complexity of modeling hydrodynamic effects for this kind of vehicle. To accurately model added mass and other hydrodynamic effects, the best method is likely to perform model tests to receive measurements of hydrodynamic effects.

2.5.2 Damping

Damping effects occur due to the viscosity of the fluid causing dissipative drag and lift forces on the body. It is common to consider linear and quadratic damping and group the terms in a damping matrix. Linear and quadratic damping matrices are assumed diagonal, as shown in Equation 2.56 and Equation 2.57 respectively.

$$\boldsymbol{D} = -diag\{X_u,\ Y_v,\ Z_w,\ K_p,\ M_q,\ N_r\} \tag{2.56}$$

$$\boldsymbol{D}_{NL} = -diag\{X_{u|u|}|u|,\ Y_{v|v|}|v|,\ Z_{w|w|}|w|,\ K_{p|p|}|p|,\ M_{q|q|}|q|,\ N_{r|r|}|r|\} \tag{2.57}$$

Approximating the elements in the damping matrices is a challenging task, and probably impossible to model accurately, as they depend on the fluid conditions, and the geometry, material and velocity of the body. A method of determining damping forces and moments for an underwater manipulator system is shown in McMillan et al. (1995). Here it is assumed that lift forces caused by vortex-shedding are neglectable. Drag forces, $\boldsymbol{f}_b^D$ and moments, $\boldsymbol{n}_b^D$ acting on the body are found by dividing the body into small disk elements, and integrating along the length of the body as shown Equation 2.58 and Equation 2.59.

$$\boldsymbol{f}_b^D = -\rho C_D r \int_0^l ||\boldsymbol{v}^n(x)||\boldsymbol{v}^n(x)dx \tag{2.58}$$

$$\boldsymbol{n}_b^D = -\rho C_D r \int_0^l ||\boldsymbol{v}^n(x)||(\begin{bmatrix} x & 0 & 0 \end{bmatrix}^\top \times \boldsymbol{v}^n(x))dx \tag{2.59}$$

C_D is the drag coefficient, r is the cylinder radius, l is the cylinder length, ρ is the fluid density, dx is the width of each circular disk element and $\boldsymbol{v}^n(x)$ is the transnational velocity vector relative to the fluid and normal to the edge of each disk. Alternatively, Schmidt-Didlaukies et al. (2018) provides a simple model for a constant linear damping matrix:

$$\boldsymbol{D}_L = \rho\pi l C_d v_{ref} \begin{bmatrix} \beta & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \frac{1}{2}l \\ 0 & 0 & 1 & 0 & -\frac{1}{2}l & 0 \\ 0 & 0 & 0 & \gamma r^2 & 0 & 0 \\ 0 & 0 & -\frac{1}{2}l & 0 & \frac{1}{3}l^2 & 0 \\ 0 & \frac{1}{2}l & 0 & 0 & 0 & \frac{1}{3}l^2 \end{bmatrix} \tag{2.60}$$

where C_d, v_{ref}, β and γ are constant coefficients that must be determined. Ferreira et al. (2011) provides a method of calculating the elements in $\boldsymbol{D}_{NL}$ for an AUV as

$$X_{u|u|} = -\frac{1}{2}\rho \int_H l_{yz}(z)dz \tag{2.61}$$

$$Y_{v|v|} = Z_{w|w|} = -\frac{1}{2}\rho \int_L l_{xz}(x)dx \tag{2.62}$$

$$K_{p|p|} = -\frac{1}{2}\rho \int_H |z|^3 l'_{xz}(z)dz \tag{2.63}$$

$$M_{q|q|} = N_{r|r|} = -\frac{1}{2}\rho \int_L |x|^3 l_{xy}(x)dx \tag{2.64}$$

$$\tag{2.65}$$

where the functions inside the integrals are determined based on the geometry and drag coefficients of the body.

2.5.3 Restoring Forces

Restoring forces are the gravitational forces and buoyancy forces acting on a body in a fluid. The weight of the body, W, and the buoyancy force, B are given as

$$W = mg \tag{2.66}$$

$$B = \rho g \nabla \tag{2.67}$$

where m is the mass of the body, g is the acceleration of gravity which is positive downwards, ρ is the fluid density and ∇ is the volume of fluid displaced by the body. In the case where the body is fully submerged, ∇ is equal to the volume of the body. When expressed in the NED frame, the restoring forces are

$$\boldsymbol{f}_g^n = \begin{bmatrix} 0 & 0 & W \end{bmatrix}^\top \tag{2.68}$$

and

$$\boldsymbol{f}_b^n = -\begin{bmatrix} 0 & 0 & B \end{bmatrix}^\top \tag{2.69}$$

Expressing these forces in the BODY frame results in the generalized restoring vector

$$\boldsymbol{g}(\boldsymbol{\eta}) = \begin{bmatrix} (W-B)sin(\theta) \\ -(W-B)cos(\theta)sin(\phi) \\ -(W-B)cos(\theta)cos(\phi) \\ -(y_g W - y_b B)cos(\theta)cos(\phi) + (z_g W - z_b B)cos(\theta)sin(\phi) \\ (z_g W - z_b B)sin(\theta) + (z_g W - z_b B)sin(\theta)cos(\phi) \\ -(x_g W - x_b B)sin(\theta)sin(\phi) - (y_g W - y_b B)sin(\theta) \end{bmatrix} \tag{2.70}$$

where ϕ is the roll angle, θ is the pitch angle, $\boldsymbol{r}_{bg}^b = [x_g, y_g, z_g]^\top$ is the position vector of the CG relative to the CO, and $\boldsymbol{r}_{bb}^b = [x_b, y_b, z_b]^\top$ is the position vector of the Center of Buoyancy (CB) relative to the CO. For a neutrally buoyant body, $W = B$ such that the three first elements in $\boldsymbol{g}(\boldsymbol{\eta})$ are always zero. Further simplifications can be made by assuming the CG is in the same position as the CO. The restoring matrix then becomes

$$\boldsymbol{g}(\boldsymbol{\eta}) = \begin{bmatrix} \mathbf{0}_{3\times 1} \\ y_b B cos(\theta)cos(\phi) - z_b B cos(\theta)sin(\phi) \\ -z_b B sin(\theta) - z_b B sin(\theta)cos(\phi) \\ x_b B sin(\theta)sin(\phi) + y_b B sin(\theta) \end{bmatrix} \tag{2.71}$$

Underwater vehicles are often not equipped for control of roll and pitch, however stability of the vehicle can be achieved by making sure the CB is placed above the CG. When roll and pitch are zero, $\boldsymbol{g}(\boldsymbol{\eta}) = \mathbf{0}$ and the vehicle is in equilibrium and will not move in any direction unless an external force acts upon it. If the roll or pitch angle is nonzero, the gravitational and buoyancy force will induce a righting moment as illustrated in Figure 2.9 which will rotate the vehicle until roll and pitch is zero (Christ and Wernli 2014).

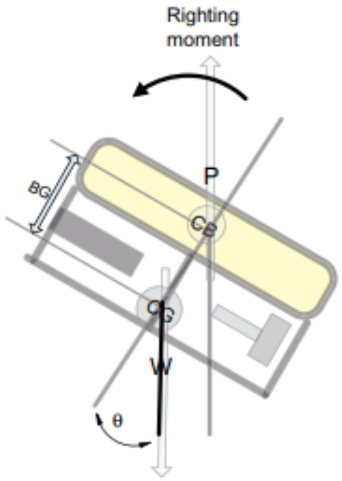


Figure 2.9: Righting moment of an underwater vehicle (Christ and Wernli 2014)

2.5.4 Current

It is assumed that the vehicle is always fully submerged at a depth where it will not be affected by wind or wave forces, thus current is the only environmental force that can affect the vehicle. EELY500 can travel to a depth of around 500 meters. In Faltinsen (1990) it is seen that dynamic pressure, and velocity and acceleration components of a wave in deep water all are proportional to e^z, where e^x is the exponential function, and z is the vertical coordinate which is defined positively upwards. Thus z decreases when the depth increases, and e^z quickly tends to zero. It is therefore valid to assume that the vehicle is unaffected by waves when it is submerged below a certain depth.

A model of a 3D irrotational ocean current can be expressed in the NED frame as

$$\boldsymbol{v}_c^n = \begin{bmatrix} V_c cos(\alpha_{V_c}) cos(\beta_{V_c}) \\ V_c sin(\beta_{V_c}) \\ V_c sin(\alpha_{V_c}) cos(\beta_{V_c}) \end{bmatrix} \tag{2.72}$$

where α_{V_c} is the vertical current direction, β_{V_c} is the horizontal current direction and V_c is the current speed. The vertical and horizontal current directions are expressed relative to the vehicle. Ocean current speed is random, and can be modeled using a first-order Gauss-Markov process as shown in Equation 2.73.

$$\dot{V}_c + \mu V_c = w \tag{2.73}$$

w is the Gaussian white noise and $\mu \geq 0$ is a constant. This current can then be expressed in the BODY frame as

$$\boldsymbol{\nu}_c^b = \begin{bmatrix} u_c & v_c & w_c & 0 & 0 & 0 \end{bmatrix}^\top \tag{2.74}$$

where

$$\begin{bmatrix} u_c & v_c & w_c \end{bmatrix}^\top = \boldsymbol{R}^\top(\boldsymbol{\Theta}_{nb})\boldsymbol{v}_c^n \tag{2.75}$$

The effects of the ocean current on the vehicle is simulated by including the relative velocity between the vehicle and the current, $\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_c^b$, in the EOM. Equation 2.39b is the changed to

$$\boldsymbol{M}_{RB}(\boldsymbol{q})\dot{\boldsymbol{\zeta}} + \boldsymbol{C}_{RB}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{M}_A(\boldsymbol{q})\dot{\boldsymbol{\zeta}}_r + \boldsymbol{C}_A(\boldsymbol{q},\boldsymbol{\zeta}_r)\boldsymbol{\zeta}_r + \boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta}_r)\boldsymbol{\zeta}_r + \boldsymbol{g}(\boldsymbol{\eta},\boldsymbol{q}) = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{u} + \boldsymbol{\tau} \tag{2.76}$$

where

$$\zeta_r = [\boldsymbol{\nu}_r^\top, \dot{\boldsymbol{q}}^\top]^\top \tag{2.77}$$

Chapter 3

Methods for Guidance, Navigation and Control

In order to use an articulated underwater robot for inspections of subsea structures and rough and steep terrain, it is necessary to develop method for Guidance, Navigation and Control. Using definitions from Fossen (2021) verbatim, *Guidance* is the action or the system that continuously computes the reference (desired) position, velocity and attitude of a marine craft to be used by the motion control system. These data are usually provided to the human operator and the navigation system. *Navigation* is the science of directing a craft by determining its position, attitude, course and distance traveled. In some cases velocity and acceleration are determined as well. *Control*, or more specifically motion control, is the action of determining the necessary control forces and moments to be provided by the craft in order to satisfy a certain control objective. Figure 3.1 shows a block diagram for a general control architecture for any marine craft. Methods and sensors useful for controlling an underwater snake robots will be discussed in this chapter. The blocks used for the simulation case study is shown in the next chapter.

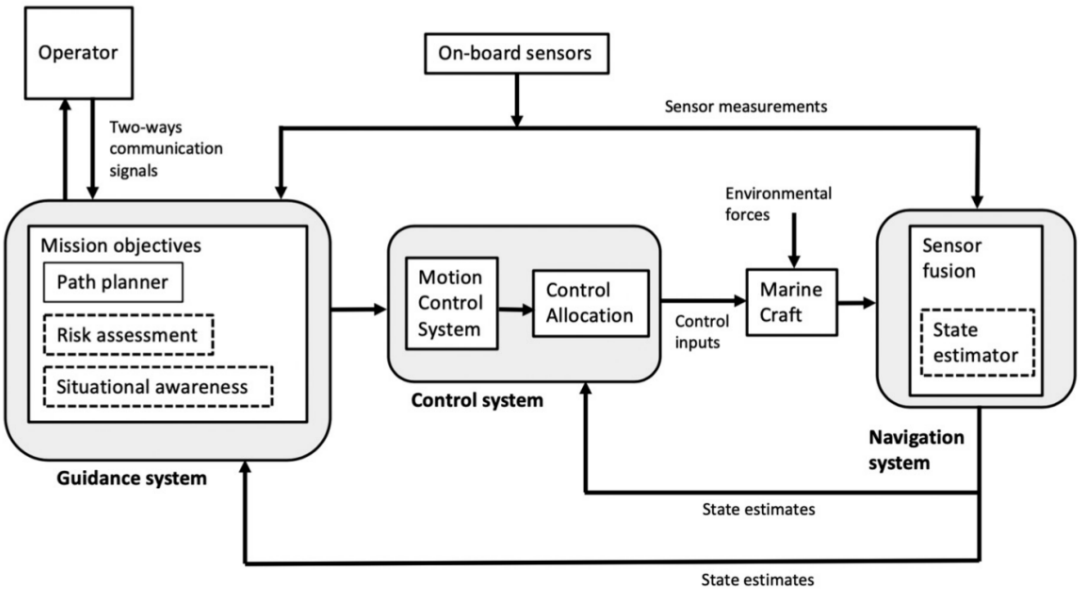


Figure 3.1: General motion control system architecture (Fossen 2021)

An underwater snake robot can be piloted manually by a human operator, or automated using a Dynamic Positioning (DP) system. The optimal methods for guidance and control depend on the objective of the mission or operation. Path following using a regular lawnmower pattern may

be useful to cover large areas (Ludvigsen et al. 2013). For inspection of an underwater structure such as a long pipeline, it is possible to perform path following along the pipeline, given that the location of the pipeline is known, and there are no geohazards along the way. For mapping of steep terrain, Nornes et al. (2017) and Nornes et al. (2016) describe an operation for visual mapping of steep underwater walls using an ROV with a horizontally facing Doppler Velocity Logger (DVL). The horizontally facing DVL is used to measure the distance between the vehicle and the wall, which can allow the vehicle to maintain a safe distance, without having to rely on highly skilled manual operators. Steep underwater walls can then be inspected an mapped by moving the vehicle in a vertical lawnmower pattern while maintaining a constant distance to the wall. The aforementioned papers discuss that changing the angle of the DVL to match the incline of the wall would increase the flexibility of the system and improve result. Flexibility could be increased by using an underwater snake robot and taking advantage of the ability to change body configuration. A DVL could for example be mounted to the head of the snake, which can be rotated to match the incline of a steep area.

3.1 Guidance

During inspections, an underwater snake robot could be controlled by a human operator using a joystick, however according to Johansen et al. (2016) this can degrade the results due to less stable motion. More stable motion could be achieved with a guidance law that feeds desired states to a DP system. As discussed above, path following could be a suitable guidance method for visual inspection and mapping, either by following the path of a pipeline, or a lawnmower pattern. This can be achieved by defining multiple waypoints along the path as described in Mare (2010). Path following can then be achieved by designing a line of sight guidance law as described in Fossen (2021). This method is based on producing forward speed, while the orientation of the vehicle is changed based on the distance to the path. Note that path following is independent of time, which means there are no temporal constraints, and thus no constraints on the commanded velocity of the vehicle. In some cases, such as paths with many sharp turns, it may be necessary to control the vehicle at a certain speed while moving along a curved path. For this purpose, one may define a parameterized path that is time dependent. The maneuvering problem, defined in Skjetne et al. (2004), then becomes converging the position of the vehicle to a desired path, and converging the speed of the vehicle to a desired speed.

3.2 Navigation

During the simulations performed in this thesis, it is assumed that the position of the vehicle is measured perfectly. However for real operations with a marine vessel, the position and other states may be estimated with sensors. This section addresses the sensors present on EELY500. The sensor measurements are included in a Kalman filter in order to estimate the states of the vehicle.

3.2.1 Acoustic Positioning

Surface vessels primarily use global satellite navigation systems for terrestrial navigation, however this is impossible to use under water due to the attenuation of electromagnetic signals. The position of underwater vehicles can however be determined by using acoustic positioning (Antonelli 2014). The position of an underwater vehicle can be determined with the help of a transponder installed on the underwater vehicle, which receives an acoustic signal sent from a transceiver. The transceiver can be mounted on a surface vessel, or installed on the seabed. The acoustic impulse is then returned to the transceiver, and the position of the underwater vehicle relative to the transceiver is computed based on the time of flight of the acoustic impulse (Ludvigsen 2010). During experiments, the position of EELY500 is measured using High Precision Acoustic Positioning System (HiPAP) from Kongsberg Maritime. This system uses the principles of measurements and calculations using acoustic signals as previously described (Kongsberg 2022). Additionally, EELY500 is equipped with

an altimeter, which is an acoustic sensor used to measure the altitude of the vehicle above the sea bottom. Altimeters use the same principles as DVLs to measure the distance from the vehicle to the seafloor. Acoustic energy sent from the vehicle is reflected by the seafloor. The Doppler shift of the reflected signal is measured and used to compute the altitude of the vehicle (Ludvigsen and Sørensen 2016).

3.2.2 IMU

Inertial Measurement Units (IMUs) consist of three-axis accelerometers and three-axis Attitude Rate Sensors (ARS). Measurements from these can be used to continuously calculate by dead reckoning the position, velocity and attitude of a craft without the need for external reference signals.It is assumed that an IMU is mounted on the vehicle in a measurement frame $\{m_I\}$. The accelerometer measures specific force with a bias $\boldsymbol{b}_{acc}$, and the ARS measures angular velocities with a bias $\boldsymbol{b}_{ars}$. The measurement equations for the accelerometer and the ARS are given in Equation 3.1 and Equation 3.3 respectively, with bias models in Equation 3.2 and Equation 3.4.

$$\boldsymbol{f}^b_{imu} = \boldsymbol{R}^\top(\boldsymbol{\Theta}_{nb})(\boldsymbol{a}^n_{nm_I} - \boldsymbol{g}^n) + \boldsymbol{b}^b_{acc} + \boldsymbol{w}^b_{acc} \tag{3.1}$$

$$\dot{\boldsymbol{b}}^b_{acc} = \boldsymbol{w}^b_{b,acc} \tag{3.2}$$

$$\boldsymbol{\omega}^b_{imu} = \boldsymbol{\omega}^b_{nb} + \boldsymbol{b}^b_{ars} + \boldsymbol{w}^b_{ars} \tag{3.3}$$

$$\dot{\boldsymbol{b}}^b_{ars} = \boldsymbol{w}^b_{b,ars} \tag{3.4}$$

$\boldsymbol{w}^b$ is the zero-mean Guassian white noise for modeling noise in each sensor. $\boldsymbol{a}^n_{nm_I}$ is the measured accelerations, and $\boldsymbol{f}^b_{imu}$ is the specific force expressed in the BODY frame. $\boldsymbol{\omega}^b$ is the angular rates. If the initial values are known, these measurements can be used to predict the position and orientation of the vehicle by integrating the linear acceleration twice, and the angular velocity once. The drawback with this method is that the errors caused by bias are also integrated, which over time will cause unbounded drift in the estimates. Compensation for the drift can be provided by combining inertial measurement with other sensors in a feedback interconnection (Fossen 2021).

3.3 Controllers

This section discusses potential motions control systems for underwater snake robots. During field experiments, EELY500 is controlled by a human operator providing control inputs, and a similar controller is implemented for simulations for control using a joystick. Additionally controllers applicable for DP are discussed, as well as joint torque controllers for actuating the joints and controlling the body configuration of the robot.

3.3.1 Speed Controller

The following closed-loop speed controller with joystick inputs is based on Dukan and Sørensen (2012). Suppose the surge, sway, heave and yaw velocities of an underwater vehicle is to be controlled using inputs from a joystick. The 4 DOF position and velocity vectors are given in Equation 3.5 and Equation 3.6 respectively. In this case, the control objective is to guide the vehicle by minimizing the error $\boldsymbol{\nu}_{cmd} - \boldsymbol{\nu}$. $\boldsymbol{\nu}_{cmd}$ are the desired velocities provided by the joystick, which in this case serves as the guidance system.

$$\boldsymbol{\eta} = \begin{bmatrix} x & y & z & \psi \end{bmatrix}^\top \tag{3.5}$$

$$\boldsymbol{\nu} = \begin{bmatrix} u & v & w & r \end{bmatrix}^\top \tag{3.6}$$

The actuator forces are given by

$$\boldsymbol{\tau} = \begin{bmatrix} X & Y & Z & N \end{bmatrix}^\top \tag{3.7}$$

Joystick commands can be represented as $\boldsymbol{\Theta}^{js} = \begin{bmatrix} u^{js} & v^{js} & w^{js} & r^{js} \end{bmatrix}^\top$, where the buttons corresponding to surge commands, u^{js}, sway commands, v^{js}, heave commands, w^{js}, and yaw commands, r^{js}, where $u^{js}, v^{js}, w^{js}, r^{js} \in [-1, 1]$ are illustrated in Figure 3.2.

Figure 3.2: Illustration of commands given by controller

The commanded speed is made proportional to the joystick commands, and can be represented as

$$\boldsymbol{\nu}_{cmd} = \boldsymbol{K}^{\nu}_{js} \boldsymbol{T}^{b}_{js} \boldsymbol{\Theta}^{js} \tag{3.8}$$

where $\boldsymbol{K}^{\nu}_{js} \in \mathbb{R}^{4\times4}$ is a diagonal scaling matrix, where the elements in the diagonal correspond to the maximum command velocity in the corresponding direction. $\boldsymbol{T}^{b}_{js} \in \mathbb{R}^{4\times4}$ transforms the joystick reference to the BODY frame of the vehicle. In order to counteract potential roll and pitch moments that arise from these input, $\boldsymbol{\nu}_{cmd}$ is expanded such that

$$\boldsymbol{\nu}_{cmd} = \begin{bmatrix} u_{cmd} & v_{cmd} & u_{cmd} & 0 & 0 & r_{cmd} \end{bmatrix}^\top \tag{3.9}$$

The speed of the vehicle can now be controlled with the following Proportional Integral Derivative (PID) control law:

$$\boldsymbol{\tau}_c = -\boldsymbol{K}_p(\boldsymbol{\nu}_{cmd} - \boldsymbol{\nu}) - \boldsymbol{K}_i \int_0^{t'} (\boldsymbol{\nu}_{cmd}(t') - \boldsymbol{\nu}(t')dt') - \boldsymbol{K}_d(\dot{\boldsymbol{\nu}}_{cmd} - \dot{\boldsymbol{\nu}}) \tag{3.10}$$

$\boldsymbol{\tau}_c \in \mathbb{R}^6$ contains the commanded forces and moments. $\boldsymbol{K}_p, \boldsymbol{K}_i, \boldsymbol{K}_d \in \mathbb{R}^{6\times6}$ are positive definite diagonal matrices containing the proportional, integral and derivative gains.

In order produce stable and realistic results, the controller must be tuned, which involves choosing values for the gains. The commanded speed is saturated according to Equation 3.11. Proportional gains are chosen to be equal for translation in all directions and rotation in all directions as shown in Equation 3.12. The same applies for integral and derivative gains. This is done in order to speed up the tuning process as there are fewer gains that must be chosen. However the quality of the controller may decrease, as the behaviour of the system is likely different in different directions due to the different dimensions of the vehicle. There exist sever methods for determining the gains.

The PID pole-placement algorithm described in Fossen (2021) is used in order to determine the order of magnitude of the gains, however the final values are chosen through trial and error.

$$\boldsymbol{K}_{js}^{\nu} = K_{js}\boldsymbol{I}_4 \tag{3.11}$$

$$\boldsymbol{K}_p = \begin{bmatrix} K_{p,linear}\boldsymbol{I}_3 & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & K_{p,angular}\boldsymbol{I}_3 \end{bmatrix} \tag{3.12}$$

3.3.2 MIMO Nonlinear PID Control

For DP applications, it may be desirable to control the position and attitude of the vehicle. Let $\boldsymbol{\eta}_d$ denote the desired pose of the vehicle in the NED frame, which can be given by a guidance law. Let $\boldsymbol{e} = \boldsymbol{\eta} - \boldsymbol{\eta}_d$ denote the difference between the actual pose and the desired pose. The control objective is in this case minimizing $\boldsymbol{e}$ with a feedback loop to reach the desired states. This can be achieved with the following nonlinear Multiple Input Multiple Output (MIMO) PID control law.

$$\boldsymbol{\tau}_c = -\boldsymbol{K}_p\boldsymbol{J}_{\Theta}^{-1}\boldsymbol{e} - \boldsymbol{K}_d\boldsymbol{J}_{\Theta}^{-1}\dot{\boldsymbol{e}} - \boldsymbol{K}_i\boldsymbol{J}_{\Theta}^{-1}\int_0^t \boldsymbol{e}(t')dt' \tag{3.13}$$

Nonlinearities arise since it is desired to represent the commanded forces and moments, $\boldsymbol{\tau}_c$, in the BODY frame. Thus $\boldsymbol{e}$ must be transformed from the NED frame to the BODY frame.

3.3.3 Joint Torque Controller

A joint torque controller is used to control the body configuration of an underwater snake robot. Let $\boldsymbol{q}_d = \begin{bmatrix}\theta_{1_d} & \dots & \theta_{(N-1)_d}\end{bmatrix}^\top$ denote the desired joint angles, and let $\boldsymbol{q} = \begin{bmatrix}\theta_1 & \dots & \theta_{N-1}\end{bmatrix}^\top$ denote the actual joint angles. The difference between the actual and desired joint angles is $\tilde{\boldsymbol{q}} = \boldsymbol{q} - \boldsymbol{q}_d$. The joint angles can now be controlled with the following PD control law found from Pettersen (2017).

$$\boldsymbol{\tau}_\theta = -\boldsymbol{K}_{p,\theta}\tilde{\boldsymbol{q}} - \boldsymbol{K}_{d,\theta}\dot{\tilde{\boldsymbol{q}}} \tag{3.14}$$

$\boldsymbol{\tau}_\theta \in \mathbb{R}^{N-1}$ is the commanded joint torques, and $\boldsymbol{K}_{p,\theta}, \boldsymbol{K}_{d,\theta} \in \mathbb{R}^{N-1\times N-1}$ are positive definite diagonal matrices containing the proportional, and derivative gains. In order to control the joints, it is necessary to measure the the joint angles. Magnetic rotary encoders can be used to measure the joint angles of an articulated robot. These encoders can be installed to in each joint motor to measure the relative rotation angle of each joint (Zhang et al. 2019).

3.4 Control Allocation

Control allocation concerns distributing the control signal from a controller to the physical effectors on a vehicle. The generalized 6 DOF control forces and moments $\boldsymbol{\tau}$ can be distributed to the effectors of the vehicle as shown in Equation 3.15, where $\boldsymbol{B}$ is the input matrix, n is the number of inputs, and $\boldsymbol{r}_{bpi}^b = \begin{bmatrix}l_{x_i} & l_{y_i} & l_{z_i}\end{bmatrix}^\top$ are the lever arms. Lever arms refer to the distance from the CO of the body to the line of action of the force.

$$\boldsymbol{\tau} = \sum_{i=1}^{n}\begin{bmatrix}\boldsymbol{f}_i^b \\ \boldsymbol{r}_{bpi}^b \times \boldsymbol{f}_i^b\end{bmatrix} = \sum_{i=1}^{n}\begin{bmatrix} F_{x_i} \\ F_{y_i} \\ F_{z_i} \\ l_{y_i}F_{z_i} - l_{z_i}F_{y_i} \\ l_{z_i}F_{x_i} - l_{x_i}F_{z_i} \\ l_{x_i}F_{y_i} - l_{y_i}F_{x_i}\end{bmatrix} = \boldsymbol{B}\boldsymbol{u} \tag{3.15}$$

Consider an underwater snake robot controlled by r thrusters and $N-1$ joints, such that the total number of inputs to the system is $n = r+N-1$. Let $\boldsymbol{\tau}_c$ be 6 DOF commanded forces and moments the and $\boldsymbol{\tau}_\theta$ be the commanded joint torques. The forces and moments applied to the robot from joint motors and thrusters can be written as follows:

$$\begin{bmatrix} \boldsymbol{\tau}_\theta \\ \boldsymbol{\tau}_c \end{bmatrix} = \boldsymbol{B}_{tot}\boldsymbol{u}_{tot} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{B}_*(\boldsymbol{q}) \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{u}_* \end{bmatrix} \tag{3.16}$$

$\boldsymbol{u} \in \mathbb{R}^{N-1}$ and $\boldsymbol{u}_* \in \mathbb{R}^r$ are the vectors containing joint motor forces and thrust forces respectively. In other words, $\boldsymbol{u}_{tot}$ contains the inputs to the system. $\boldsymbol{B} \in \mathbb{R}^{6\times N-1}$ is the input map for the joint motor forces, and $\boldsymbol{B}_*(q) \in \mathbb{R}^{6\times r}$ is the thruster configuration matrix that maps the desired control forces and moments to the thrusters. Note that for an articulated underwater robot, the relative position and orientation of the thrusters with respect to the base of robots change when the joints rotate. As a result the thruster configuration matrix is dependent of the joint angles, such that $\boldsymbol{B}_*(\boldsymbol{q})$ and must be recomputed when the robot changes its body shape, while $\boldsymbol{B}$ remains constant. This can be achieved with the following algorithm, as described in Sverdrup-Thygeson et al. (2018). It is possible to find geometric Jacobians that maps the forces and moments, $\boldsymbol{\tau}_{t_j}$, from the thruster frames, $\{t_j\}$, to the BODY frame, $\{b\}$, of the vehicle. The mapping is shown in Equations 3.17 and 3.18. Thruster frames are illustrated in Figure 3.3.

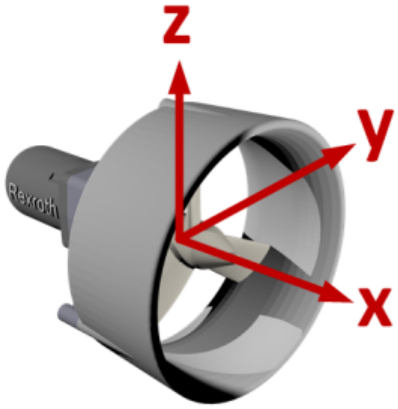


Figure 3.3: Visualization of the thruster frame (Manhães et al. 2016)

$$\boldsymbol{\tau}(\boldsymbol{q}) = \sum_{j=1}^{r} \boldsymbol{J}_{t_j}^\top(\boldsymbol{q})\boldsymbol{\tau}_{t_j} \tag{3.17}$$

$$\boldsymbol{\tau}_{t_j} = \begin{bmatrix} 1 \\ \boldsymbol{0}_{5\times 1} \end{bmatrix} u_{thr_j} \tag{3.18}$$

u_{thr_j} is the scalar force applied by thruster j. In order to find the forces, $\boldsymbol{\tau}_b(\boldsymbol{q})$, acting on the BODY frame which is placed at the head link, the selection matrix $\boldsymbol{H} = \begin{bmatrix} \boldsymbol{I}_{6\times 6} & \boldsymbol{0}_{6\times N} \end{bmatrix}$ is used. $\boldsymbol{\tau}_b(\boldsymbol{q})$ can now be expressed as:

$$\boldsymbol{\tau}_b(\boldsymbol{q}) = \boldsymbol{H}\boldsymbol{\tau}(\boldsymbol{q}) = \sum_{j=1}^{r} \boldsymbol{H}\boldsymbol{J}_{t_j}^\top(\boldsymbol{q})\boldsymbol{\tau}_{t_j} = \sum_{j=1}^{r} (Ad_{bt_j}^{-1})^\top \boldsymbol{\tau}_{t_j} \tag{3.19}$$

where $Ad_{bt_j}^{-1}$ is the inverse map between the BODY frame and the thruster frame defined in Equation 2.31.

Equation 3.19 is then rewritten as:

$$\boldsymbol{\tau}_b(\boldsymbol{q}) = \sum_{j=1}^{r} \boldsymbol{B}_j(\boldsymbol{q}) u_{thr,j} \tag{3.20}$$

where

$$\boldsymbol{B}_j(\boldsymbol{q}) = (Ad_{bt_j}^{-1})^\top \begin{bmatrix} 1 \\ \boldsymbol{0}_{5\times 1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{bt_j} & \boldsymbol{0}_{3\times 3} \\ \hat{\boldsymbol{p}}_{bt_j}\boldsymbol{R}_{bt_j} & \boldsymbol{R}_{bt_j} \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{0}_{5\times 1} \end{bmatrix} \tag{3.21}$$

The complete thruster configuration matrix can hence be expressed as:

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{B}_*(\boldsymbol{q}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{B}_1(\boldsymbol{q}) & \boldsymbol{B}_2(\boldsymbol{q}) & \dots & \boldsymbol{B}_r(\boldsymbol{q}) \end{bmatrix}. \tag{3.22}$$

Equation 3.15 is used to compute the control inputs $\boldsymbol{u}$ as

$$\boldsymbol{u}_{tot} = \boldsymbol{B}_{tot}^{\dagger} \boldsymbol{\tau}_{tot} \tag{3.23}$$

where $\boldsymbol{B}_{tot}^{\dagger} = \boldsymbol{B}_{tot}^{\top}(\boldsymbol{B}_{tot}\boldsymbol{B}_{tot}^{\top})^{-1}$ is the Moore-Penrose pseudoinverse of $\boldsymbol{B}_{tot}$ (Fossen 2021). The psuedoinverse must be used since if the number of actuator forces does not equal the number of DOFs, $\boldsymbol{B}_{tot}$ is not square, and thus $\boldsymbol{B}_{tot}$ is not invertable. In cases where $\boldsymbol{B}_{tot}$ is square, the pseudoinverse is equal to the inverse as $\boldsymbol{B}_{tot}^{\dagger} = \boldsymbol{B}_{tot}^{-1}$. Albert (1972) explains that the pseudoinverse computes a least squares solution to the system to find the best fit solution of $\boldsymbol{u}_{tot}$. In general computation of $\boldsymbol{u}_{tot}$ is an optimization problem with multiple approaches, however this is not considered in this thesis.

3.5 Thruster Model

When the inputs to the thrusters are known, one can model the thruster dynamics in order to find the thrust force produced by each thruster based on the rotation speed of the propeller. Yoerger et al. (1990) presents an energy-based model for the thruster dynamics on the following form:

$$\dot{\Omega} = \beta u - \alpha \Omega |\Omega| \tag{3.24}$$

$$T = C_t \Omega |\Omega| \tag{3.25}$$

Equation 3.24 represents the propeller dynamics and Equation 3.25 is a conversion function. Ω is the propeller angular velocity or shaft speed, u is the input to each thruster computed in Equation 3.23, and T is the thrust produced by the propeller. The produced thrust acts along the x-direction of the thruster frames shown in Figure 3.3. α, β and C_t are constant design parameters. These constants are often found with model tests. The thruster model is illustrated as two simple blocks in Figure 3.4.

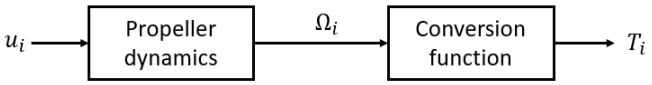


Figure 3.4: Thruster model

The energy-based thruster model can be compared to the bilinear thrust model from Fossen (2021)

$$I_m\dot{\Omega} = Q_m - Q - Q_f \tag{3.26}$$

$$Q = \rho D^5 K_Q(J_a)\Omega|\Omega| \tag{3.27}$$

$$T = \rho D^4 K_T(J_a)\Omega|\Omega| \tag{3.28}$$

I_m is the inertia and added inertia of the rotating propeller. Q_m is the input torque, Q is the propeller torque, and Q_f is the friction torque. ρ is the water density, D is the propeller diameter and $K_Q(J_a)$ and $K_T(J_a)$ are propeller torque and propeller thrust coefficients. The main difference between these two models is that Yoerger et al. (1990) do not include friction torque. Neglecting Q_f and assuming Q_m equals the thruster input, the coefficients from Equation 3.24 and Equation 3.25 can be modeled as

$$\beta = \frac{1}{I_m} \tag{3.29}$$

$$\alpha = \frac{1}{I_m}\rho D^5 K_Q(J_a) = \beta\rho D^5 K_Q(J_a) \tag{3.30}$$

$$C_t = \rho D^4 K_T(J_a) \tag{3.31}$$

Chapter 4

Simulation Case Studies

Simulations are a powerful tool tool for testing and validating robotic systems. If a simulator is to be used to test and develop control and guidance systems for a real vehicle, it is important that the simulator is a good representation of reality. A simulation model for EELY500 is created in the Plankton simulator. In order to verify the simulation model, a simulation case study is performed where the vehicle receives commands to move in surge, sway and heave for two constant body configurations. The body configurations tested were straight-shape, and u-shape where all joints are rotated 45 degrees about the y-axis of the link frames. It is assumed that the joints are fixed during the simulations, which simplifies the thrust allocation problem. The simulations are performed by giving speed commands using a slightly modified version of the speed controller described in Section 3.3.1. Instead of Equation 3.10 computing commanded forces and moments, the controller is defined to produce commanded accelerations $\boldsymbol{a}_c$. Commanded forces and moments are then computed according to $\boldsymbol{\tau}_c = \boldsymbol{M}\boldsymbol{a}_c$, where $\boldsymbol{M}$ is the constant system inertia matrix for each case. This reduces the order of magnitude of the controller gains. The output of the thrusters are computed using the energy-based model described in Section 3.5, and the constants are approximated using Equations 3.29, 3.30 and 3.31. In order to compare the simulation model to the real vehicle, the same tests were performed with the real EELY500 robot in a field experiment. A quick summary of the two cases are given below:

Case 1: Straight-shape

1. Launch the vehicle with all joint angles $\theta_i = [0 \;\; 0 \;\; 0]^\top$.
2. Command surge in one direction for about 20 seconds, wait about 60 seconds, command surge in the opposite direction for about 20 seconds.
3. Repeat the previous step with commands in sway and heave

Case 2: U-shape

1. Launch the vehicle with all joint angles $\theta_i = [0 \;\; \frac{\pi}{4} \;\; 0]^\top$.
2. Command surge in one direction for about 20 seconds, wait about 60 seconds, command surge in the opposite direction for about 20 seconds.
3. Repeat the previous step with commands in sway and heave

In both cases, the position of the vehicle is defined by the position of the BODY frame, which is located at the CO of the head link. Current is not considered in the simulation case studies. One thruster was deactivated in the simulation case study in order to test the performance of the system with a malfunctioning thruster.

4.1 The Plankton Simulator

The simulation model for EELY500 is created in the Plankton simulator[1]. Plankton is open source simulator for maritime robotics researchers, aimed at simplifying robotics research in the maritime domain. The simulator mainly runs using Gazebo and its plugin UUV Simulator[2] in ROS 2. Gazebo is an open source physics engine often used to simulate robots and environments with advanced 3D graphics, which allows users to receive good visual information about how systems behave. It can also be used with several plugins, where UUV Simulator is the most important for this project. UUV Simulator is a package containing necessary tools for simulating Unmanned Underwater Vehicles (UUVs) such as ROVs and AUVs, such as plugins for hydrodynamic forces and thrusters (Manhães et al. 2016). UUV Simulator also has the ability to compute the thruster allocation matrix based on the placement of the thruster frames in relation to the base link. Robot models are described using the Unified Robot Description Format (URDF). All elements of the robot, such as links, joints, thrusters, graphics and coordinate frames are defined in files written on this format, which are the uploaded to Gazebo where the physics of the robot are simulated by solving the equations of motion. The simulation model for EELY500 can be found at the author's git repository[3]. Note that Plankton must be installed first in order to run the snake robot.

4.1.1 ROS 2

The Robot Operating System (ROS) is a set of software libraries and tools for building robot applications. ROS 2 is simply an upgraded version of ROS. One of the most important aspects of ROS is the infrastructure for running code and how messages are passed between processes. In general, the ROS infrastructure is a network of nodes that communicate through topics. Each node is responsible for a single purpose, for example one node could be responsible for velocity control, while another node is responsible for thrust allocation. The ROS infrastructure follows a publish-subscribe architecture, where nodes receive information by subscribing to a topic, and can send information by publishing messages to a topic. This structure allows the program to run multiple processes simultaneously, since none of the nodes need to know about the existence of other nodes (Macenski et al. 2022). Figure 4.1 shows an illustration of the ROS infrastructure used to simulate EELY500. Nodes are contained in boxes, and how they communicate are illustrated by arrows which contain topics. This architecture can be compared with a more classical control architecture shown in Figure 4.2. In both instances, joystick inputs are given to a controller, which generates a desired force, which is received by the thruster allocator. Equations of motion are then solved, and the states of the system, which in the ROS infrastructure is contained within the node $/robot\_state\_publisher$, is updated. Notice that the topic $/tf$ appears in multiple instances. In short, *tf* is a library that keeps track of coordinate frames, and computes transformations between frames similarly to transformations described in Section 2.2, this information is then given and sent by nodes through the $/tf$ topic (Foote 2013).

[1] https://github.com/Liquid-ai/Plankton
[2] https://github.com/uuvsimulator/uuv_simulator
[3] https://github.com/bajorgen/plankton_snake

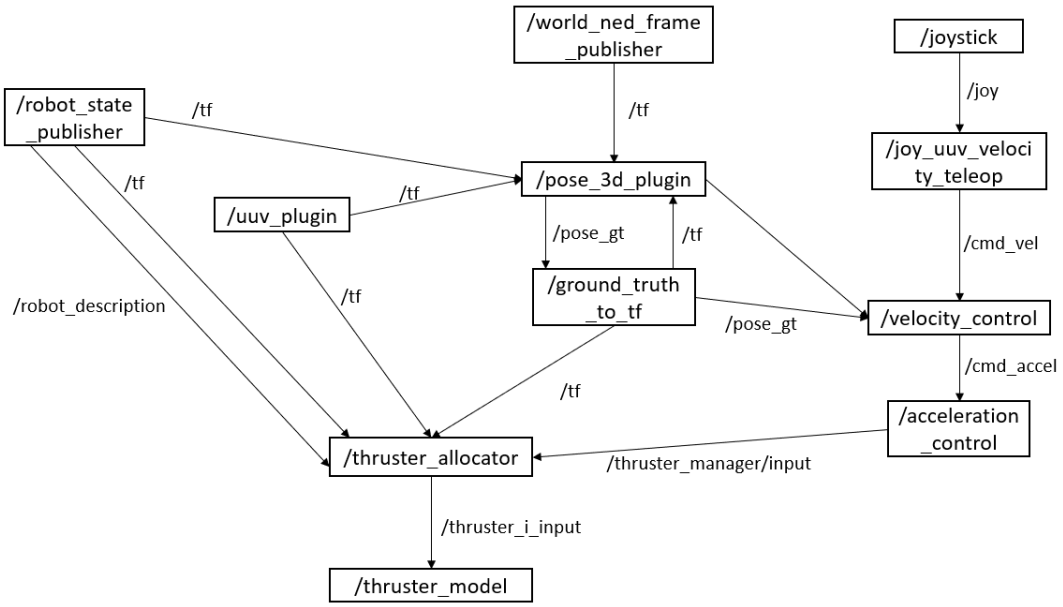


Figure 4.1: ROS infrastructure

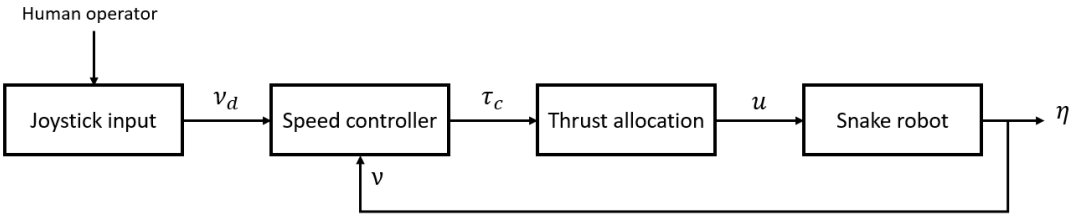


Figure 4.2: Simple control architecture

4.1.2 Model of EELY500

A simulation model for the Eelume vehicle EELY500 can now be set up. The simulations will be performed using Plankton. All properties used for the model are found from the EELY500 user manual. Parameters for the simulation model are given in Appendix B. EELY500 made up of several modules. The most noteworthy modules for this thesis are shown below.

Inspection module The inspection module is pictured in Figure 4.3. This module is placed at the fornt of the robot and is equipped with an HD camera and LED lights.

Figure 4.3: Inspection module

Ballast module
The ballast module is pictured in Figure 4.4, and allows for trimming robot bouncy and achieving a desired weight by inserting weights in an external slot in the module. This also allows the CG to be set to a desired location such that stability as described in Section 2.5.3 can be achieved.

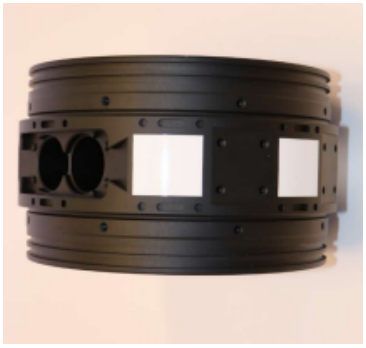


Figure 4.4: Ballast module

Sensor module
The sensor module is pictured in Figure 4.5. Overall control of the robot is handled in this module, as it contains a computer board, in addition to sensors such as acoustic positioning transponder, altimeter and Inertial Measurement Unit (IMU).

Figure 4.5: Sensor module

Joint module
The joint module is pictured in Figure 4.6, and allows the robot to change body configuration. The module contains a motorized 2 DOF joint that allows for rotation in yaw and pitch. Constraints are given in the robot by only allowing the joints to rotate $\pm 80°$. When fully assembles, EELY500 has four joint modules.

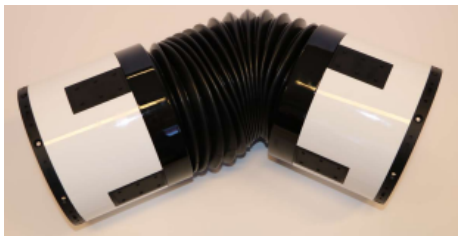


Figure 4.6: Joint module

Thruster module
The thruster module is pictured in Figure 4.7, and contains four thrusters. Two thrusters are fixed to the top and the bottom of the module, and can produce forces in the y-direction of the body. The remaining two thrusters extend from two sides of the body, and can be manually adjusted to a desired angle, which allows them to produce forces in the x-direction and the z-direction. Because of the placement of the thrusters, the thruster module is also able to produce moments about all three axes. When fully assembled, EELY500 has three joint modules.

Figure 4.7: Thruster module

The full EELY500 robot is assembled with one or more of the modules described above, as well as other modules that will not be further explained. A model of the fully assembled robot with all modules is shown in Figure 4.8.

Figure 4.8: 3D view of EELY500 with all modules

In order to model the robot, each section separated by a joint is considered one rigid link. The joints are placed in the middle of the joint module. The length of the joint module is considered as empty space to give the links room to rotate. In other words, each link is placed with a distance l_{joint} relative to each other, while each joint is placed at a distance $\frac{l_{joint}}{2}$ from the base of the previous link. Thus EELY500 is modeled as an underwater snake robot with $N = 5$ links interconnected by $N - 1 = 4$ revolute joints, and $r = 12$ thrusters. An illustration of the modeling of the links and joints is shown in Figure 4.9. The connectivity of each link is illustrated in a kinematic tree shown in Figure 4.10, where frame $\{0\}$ is the NED frame, and the superscript $*$ indicates a thruster frame. Physical parameters for the vehicle are shown in Table 4.1. Note that in this model, the x-axis on the frame of each link points towards the next link. This was done because in Gazebo, the position of other links, joints and thrusters are defined relative to the base link, which is most intuitive to determine as a positive distance away from the base link, which is defined as the head link, i.e. link 1. This orientation of the frames is also the default option in Gazebo. As a consequence, positive surge motion is defined as negative x-direction. Positive pitch and heave correspond to positive y-direction and z-direction respectively.

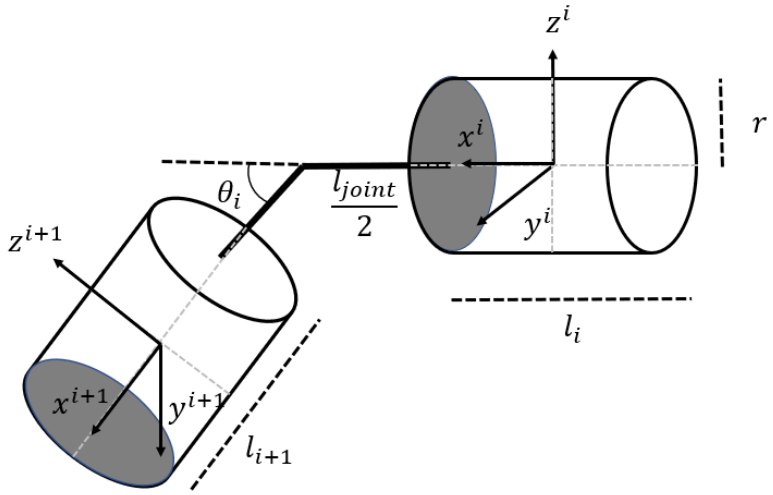


Figure 4.9: Model of links and joints

Parameter	Value	Unit
radius	100	mm
total length	6182	mm
total dry mass	199	kg
link 1 length	494	mm
link 2 length	1185	mm
link 3 length	1435	mm
link 4 length	1185	mm
link 5 length	740	mm
joint module length	285	mm

Table 4.1: Robot properties

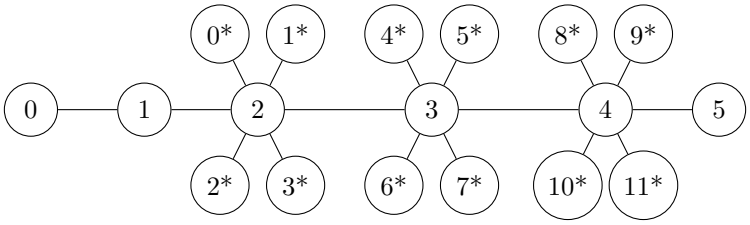


Figure 4.10: Connectivity of EELY500

Hydrodynamic forces are modeled according to Section 2.5. The CO of each link is defined as the middle of the cylinder. The CB is placed above the CO, and the CG is placed at the CO in order to achieve stability in roll and pitch. This is done because in Gazebo, each link is modeled as a simple geometrical shape with uniformly distributed mass, which places the CG in the middle of the cylinder by default, while the location of the CB can be specified by the user. In reality it would probably be more efficient to achieve stability by positioning the CG below the CO using ballast. This could be achieved in Gazebo by placing additional masses inside the links of the robot, however this would require additional links to the system. Additionally the actual EELY500 robot has a slightly neutral buoyancy. If the vehicle is assumed to be neutrally buoyant, the total dry mass could be found to be equal to the weight of seawater displaced by the volume of the vehicle, such that the mass could be determined by $m = \Delta = \rho_w \nabla = \rho_w \pi r^2 l_{tot}$, where ρ_w is the density of seawater, Δ is the displaced water mass, and ∇ is the volume displaced by the vehicle. This would result in a mass of $m = 199$kg, while the dry mass found in the EELY500 user manual states that $m = 188$kg. For underwater robots, positive buoyancy may be an advantage in case of thruster failure. If this occurs, the vehicle will float to the surface, which reduces chances of losing the vehicle. This does however come at the cost of needing to spend energy to keep the vehicle submerged using thrusters. Loss of vehicle is not an issue in simulation, and thus it is decided that the model should be neutrally buoyant such that the implementation of a depth controller is not needed, keeping the simulation model simple. A summary of assumptions and simplifications are listed below:

- All links are solid, rigid cylinders with uniformly distributed mass.
- All links are hydrodynamically independent.
- The vehicle moves at low speed.
- The vehicle will never experience forces from wind or waves.
- The tether is ignored.
- The thrusters experience no drag, and do not contribute to any external moments.
- For every link, CG is placed on CO, and CB is placed above CO.
- Each joint has one DOF, either pitch or yaw rotation.
- States can be measured perfectly with no noise.

In order to compute the thruster configuration matrix, one must know the placement of the thruster frames in relation to their respective links. In the case of EELY500, there are $r = 12$ thrusters in total, with 4 thrusters placed on link 2, link 3 and link 4, and no thrusters placed on link 1 and link 5. Figure 4.11 shows the simulation model with the thrusters numbered. Thruster 2, 6 and 10 are rotated 45 degrees, and thruster 3, 7 and 11 are rotated -45 degrees, such that each thruster produces equal force components along the x-axis and z-axis of the body. The sign of the force components along the x-axis is the same for all the thrusters on the side of the robot, while the sign of force components along the z-axis are opposite for the thrusters on the starboard side and the port side.

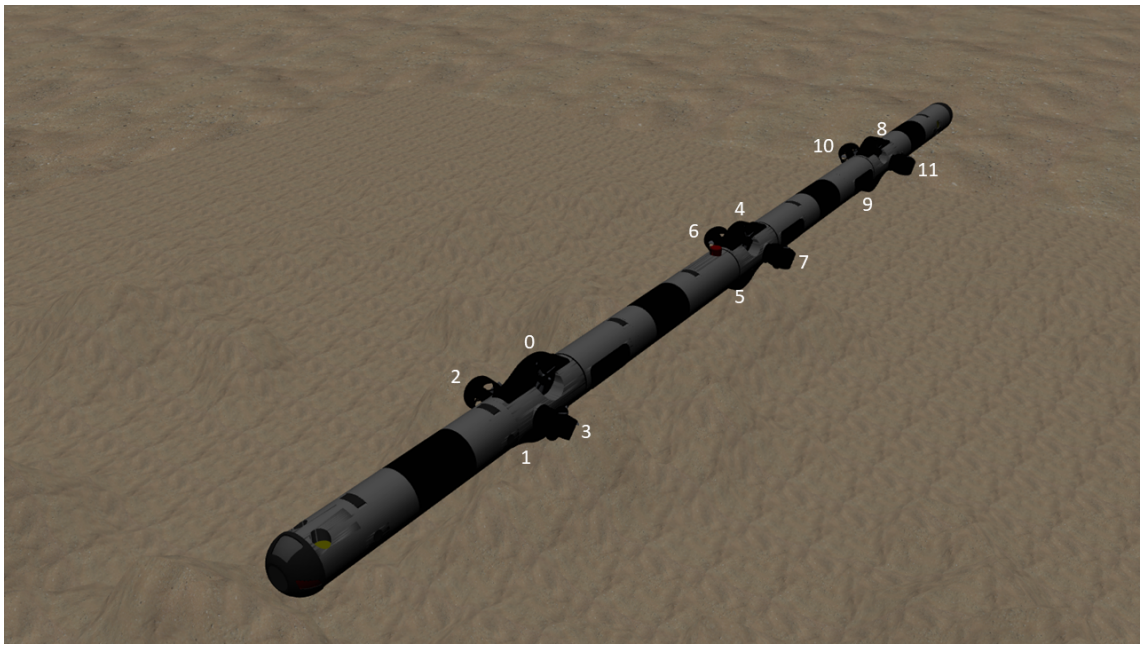


Figure 4.11: Numbering of thrusters

The placement of each thurster relative to the CO of their respective links can be found by examining the 3D model of EELY500. Link 2 and link 4 are identical, where link 4 is rotated 180 degrees about the z-axis. Link 2, link 3 and link 4 are identical when seen in the yz-plane. The positions of the thrusters on the relevant links are shown in Figures 4.12, 4.13 and 4.14.

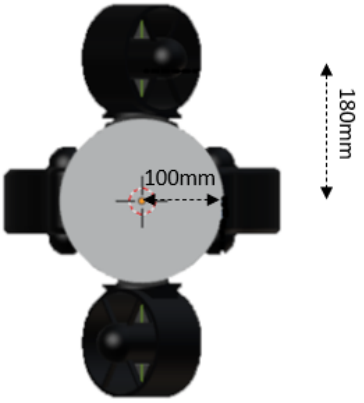


Figure 4.12: Thruster module seen in the yz-plane

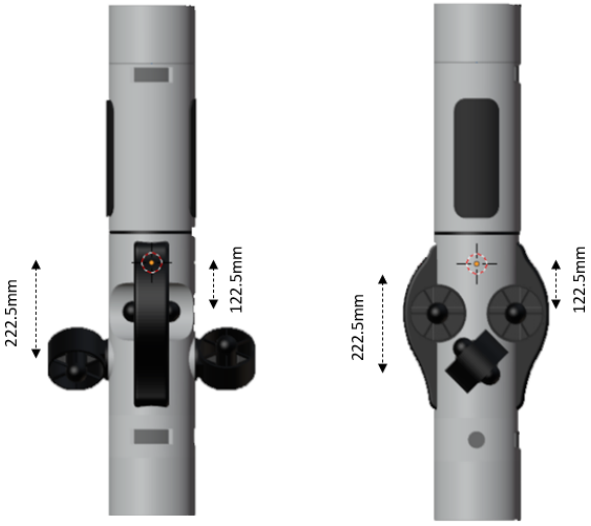


Figure 4.13: Placement of thrusters on link 2

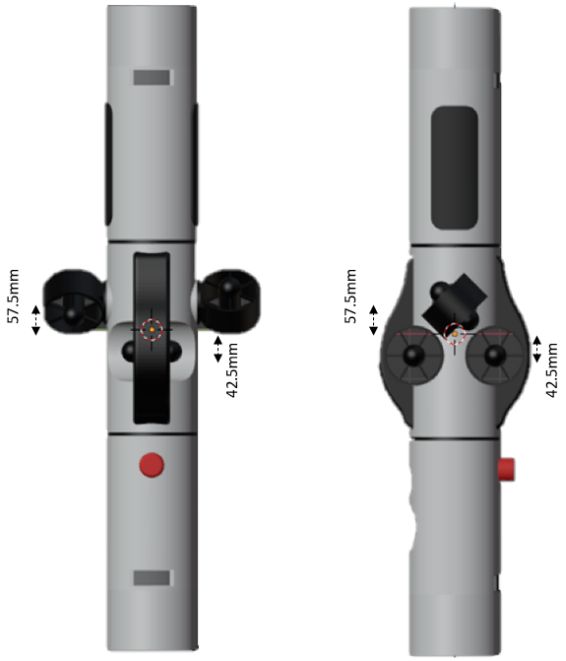


Figure 4.14: Placement of thrusters on link 3

Since there are 12 thrusters placed on EELY500, the number of control forces is larger than the number of DOFs of the vehicle, such that the system is over actuated. This means that all DOFs can be controlled, and that there are an infinite amount of solutions to the thrust alloction problem. Because of this, the solutions computed by Equation 3.23 may not always be the optimal solution in terms of power consumption. In fact, if one thruster module were removed, there would still be 8 thrusters on the robot, such that the system would remain over actuated. However the extra thrusters provide redundancy to the system, such that in case of multiple thruster failures, all degrees of freedom can still be controlled.

4.1.3 Limitations

There are a few limitations and disadvantages with using the Plankton simulator that must be worked around. ROS 2 is still relatively new, with limited documentation. Therefore further development of the simulator with simulated sensors and environmental loads may be challenging as there is little available information. Additionally, during the process of programming the snake robot model, the library for dynamically controlling joints, *ros2_control*, was still in development and had a few issues. It should also be mentioned that even though Gazebo is useful for simulating multi body systems, the plugin UUV Simulator was built mainly with focus on UUVs consisting of one rigid body, although ROVs with manipulator arms are supported. This is apparent as it is currently only possible to place thrusters on the base link, and not any other links. Additionally, UUV Simulator was created for ROS 1, and not all features have been upgraded for ROS 2 in the Plankton simulator. Furthermore, the code for the thruster allocator only considers thrusters placed on the base link. However this can be worked around when all the joint angles are constant. In this case all the link frames do not move relative to the base link, and thus the thruster frames can be placed in the appropriate place in relation to the base link for constant body configurations. It is conceivable that the the thruster frames could be transformed to match the orientation of the frames of their respective links to allow for dynamic computation of the thruster allocation matrix, however this has not been attempted for this project.

It is also relevant to note that according to Manhães et al. (2016), Gazebo simulates rigid body dynamics by integrating the rigid body equations of motion given below:

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}_0 = \boldsymbol{\tau}_g \tag{4.1}$$

Hydrodynamic effects are not handled by Gazebo itself, but are instead handled by the plugin UUV Simulator. Hydrodynamic effects are included by considering the effects as external forces in the right hand side of Equation 4.1 as shown below:

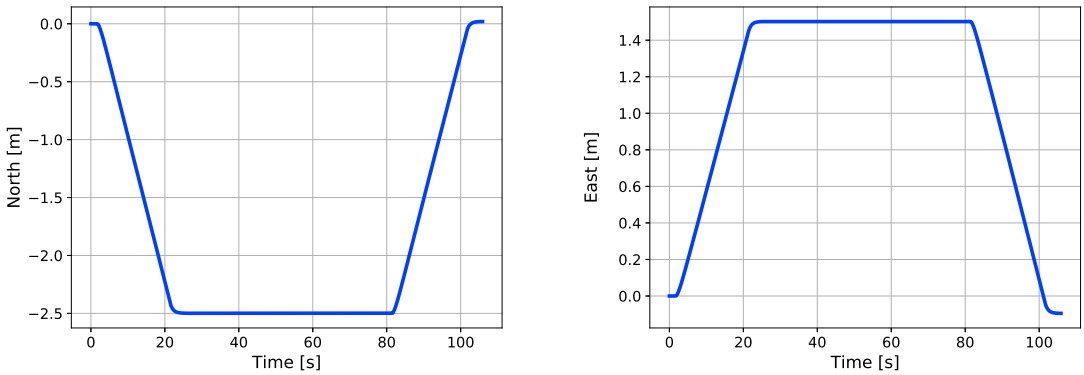
$$\boldsymbol{\tau}_g = -\boldsymbol{M}_A\dot{\boldsymbol{\nu}}_r - \boldsymbol{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r - \boldsymbol{D}(\nu_r)\nu_r - \boldsymbol{g}(\boldsymbol{\eta}) \tag{4.2}$$

Acceleration is needed as an input to compute the hydrodynamic effects, however the accelerations are computed in Equation 4.1. This means that for each time step of the simulation, there is no acceleration available as input in Equation 4.2, and as a consequence the hydrodynamic effects used to compute accelerations in Equation 4.1 can not be computed. As a workaround of this problem, the accelerations for the previous time step is used to compute hydrodynamic effects for the current time step. As a consequence, the system may become unstable if the added mass coefficients are sufficiently large. This is because acceleration at the previous time step can lead to a slightly bigger reaction due to the large added mass at the current time step, which can cause increasingly bigger accelerations at subsequent time step. Unstable simulations are currently avoided by applying a low-pass filter to the previous accelerations used to compute hydrodynamic effects. Manhães et al. (2016) do not discuss how this affects the simulation results further, but it can be hypothesized that since the accelerations are filtered, they will be smaller than expected, resulting in slower reaction times for a simulated system compared to a real system.

4.2 Case 1: Straight-Shape

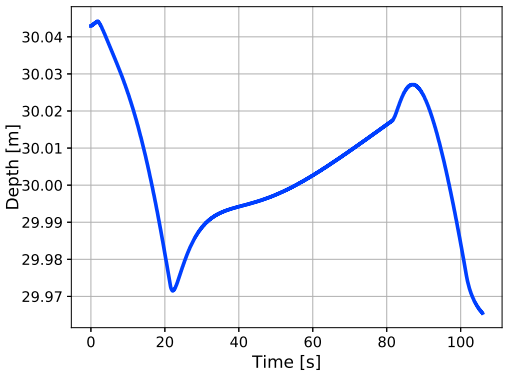
Simulations are performed when all the joints are straightened such that the vehicle has a straight shape. The vehicle receives commands for positive and negative surge, sway and heave motions. Resulting 3D motion for each command is presented, in addition to the heading of the vehicle. A north-east plot is also given to show the vehicle's planar motion. Screenshots for each test are shown in order to visualize the behaviour of the vehicle. Remaining Euler angles, linear velocities and commanded thrust for each case are given in Appendix A. Figures 4.15, 4.16 and 4.17 show results for commanded surge, sway and heave motion respectively. Movement in four degrees of freedom is shown for each case for investigating the behaviour of the system. Change can be seen in all degrees of freedom due to coupling. Some of the plots for the Euler angles show large jumps. This is because the equations used to compute the Euler angles from quaternions are not defined outside of -180 and 180 degrees.

4.2.1 Results

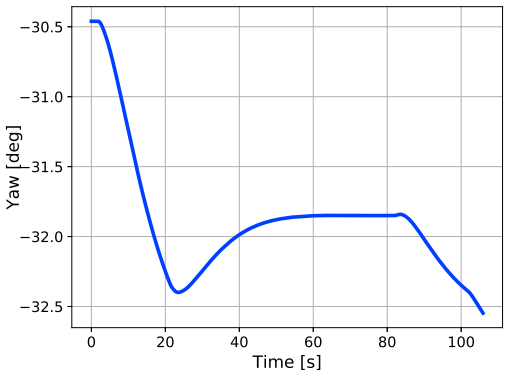


(a) North

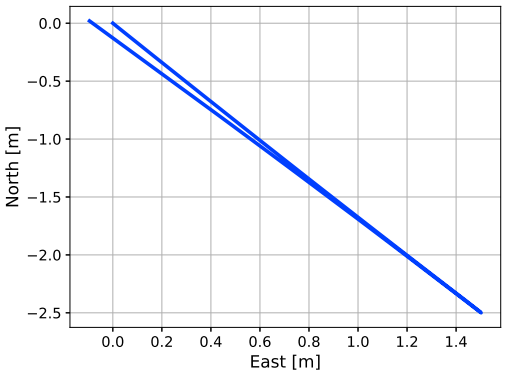
(b) East



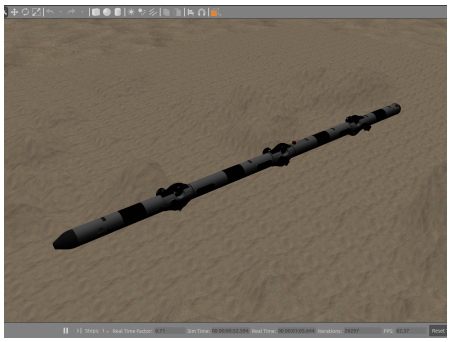
(c) Depth



(d) Yaw



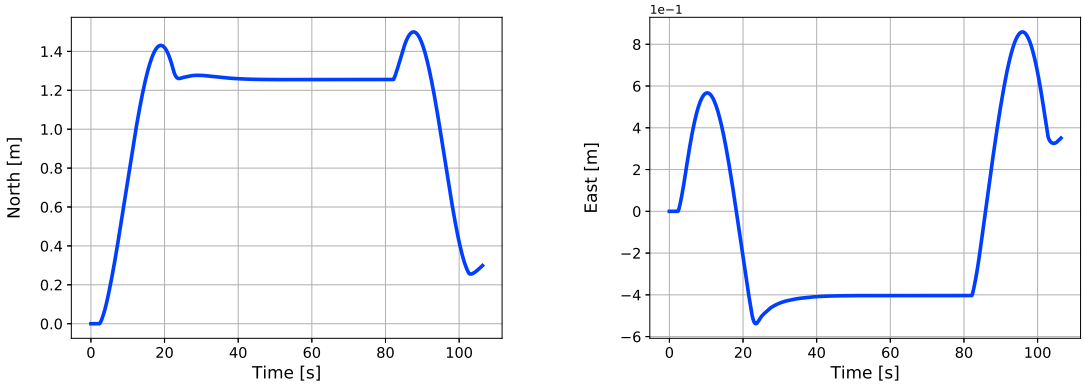
(e) North-East plot



(f) Visual

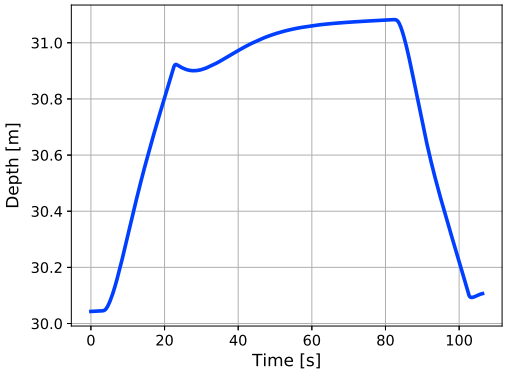
Figure 4.15: Straight-shape surge motion

Figure 4.15a, 4.15b and 4.15c shown the position of the robot in the north, east and depth axes respectively. Figure 4.15d shows the heading. Figure 4.15e is a north-east plot that shows the position in the north-east plane. Finally, Figure 4.15f shows a screenshot from the simulator when the robot receives the inputs. No change in orientation can be seen, which is supported by the negligible change in Euler angles.

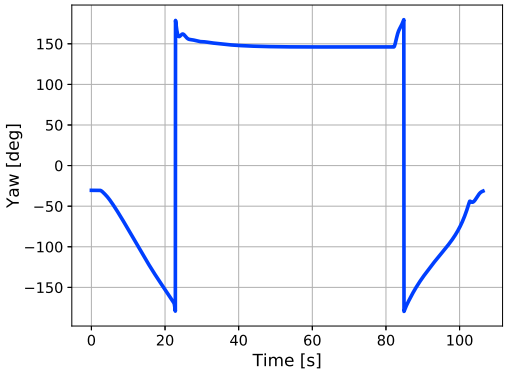


(a) North

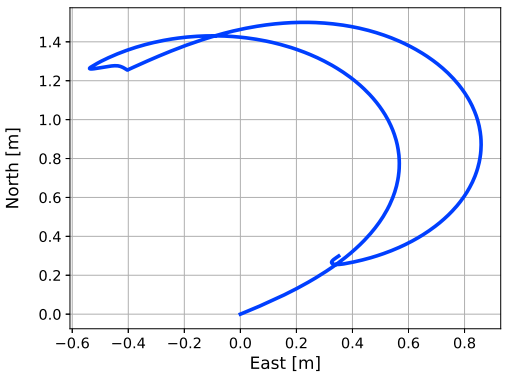
(b) East



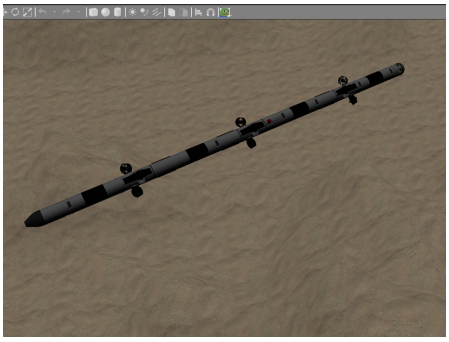
(c) Depth



(d) Yaw



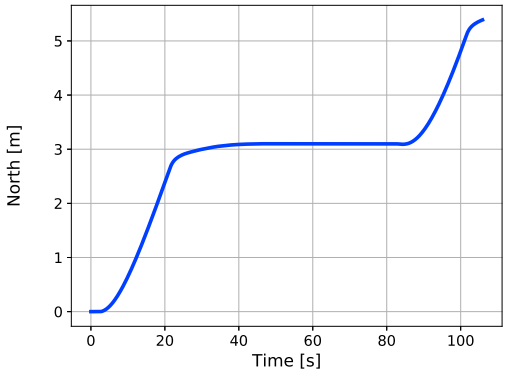
(e) North-East plot



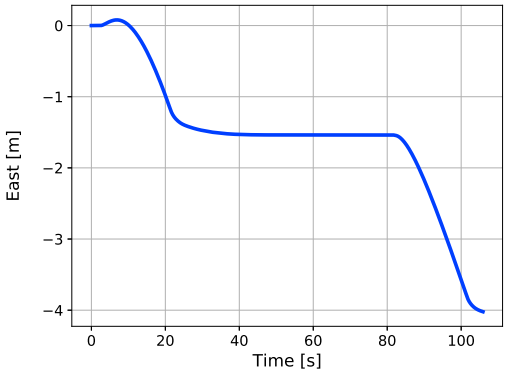
(f) Visual

Figure 4.16: Straight-shape sway motion

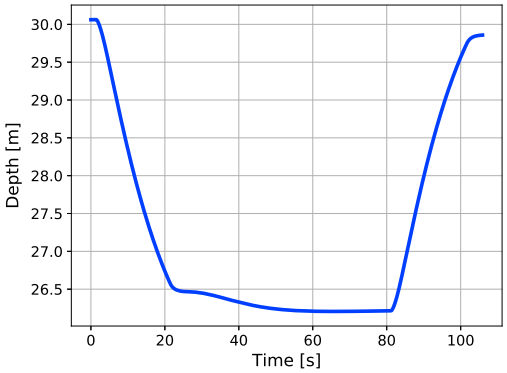
Figure 4.16a, 4.16b and 4.16c shown the position of the robot in the north, east and depth axes respectively. Figure 4.16d shows the heading. Figure 4.16e is a north-east plot that shows the position in the north-east plane. Finally, Figure 4.16f shows a screenshot from the simulator when the robot receives the inputs, such that the change in orientation can clearly be seen.



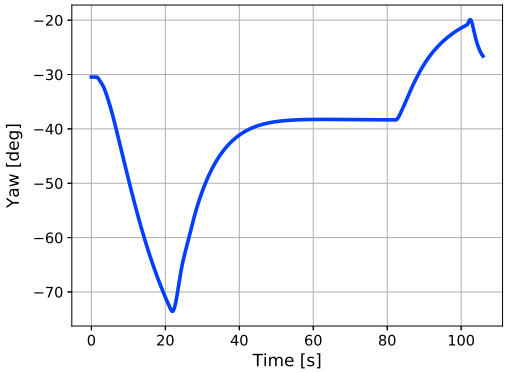
(a) North



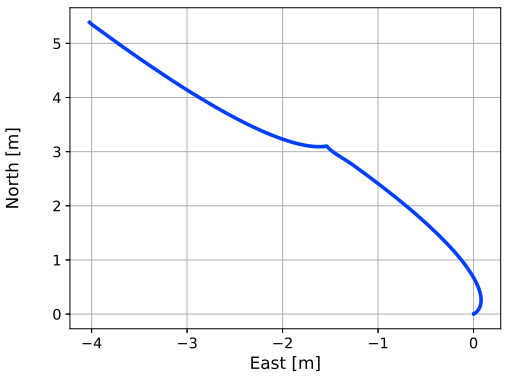
(b) East



(c) Depth



(d) Yaw



(e) North-East plot

(f) Visual

Figure 4.17: Straight-shape heave motion

Figure 4.17a, 4.17b and 4.17c shown the position of the robot in the north, east and depth axes respectively. Figure 4.17d shows the heading. Figure 4.17e is a north-east plot that shows the position in the north-east plane. Finally, Figure 4.17f shows a screenshot from the simulator when the robot receives the inputs, such that the change in orientation can clearly be seen.

4.3 Case 2: U-Shape

The same simulation case study as case 1 is performed when all the joints are rotated 45 degrees about the y-axis of the link frames, which gives the robot a u-shape. In this configuration, the BODY frame, which is rigidly attached to link 1, has a 90 degree pitch angle compared to when the joints are straightened. This means that when the head points towards the sea floor, commanded surge motion will result in movement towards the sea floor, as illustrated in Figure 4.18. It may however be more appropriate to redefine surge, sway and heave motions depending on the situation. For case 2, surge, sway and yaw motions are defined as shown in Figure 4.19 in order to more closely resemble motions as they are defined in case 1. Although motions are redefined, the BODY frame still remains rigidly attached to the head link. Therefore surge motions are achieved by giving a command for heave, and vice versa. In this case, the heading of the vehicle is now defined by the roll angle.

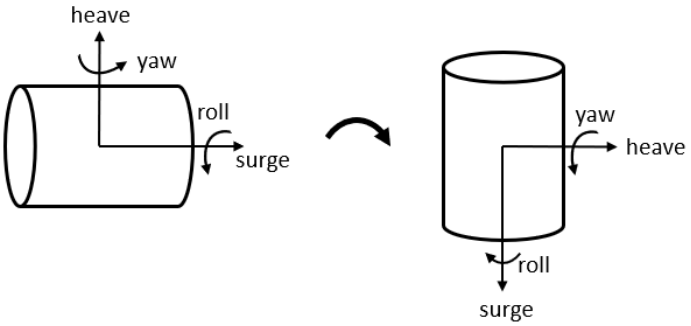


Figure 4.18: Rotated BODY frame

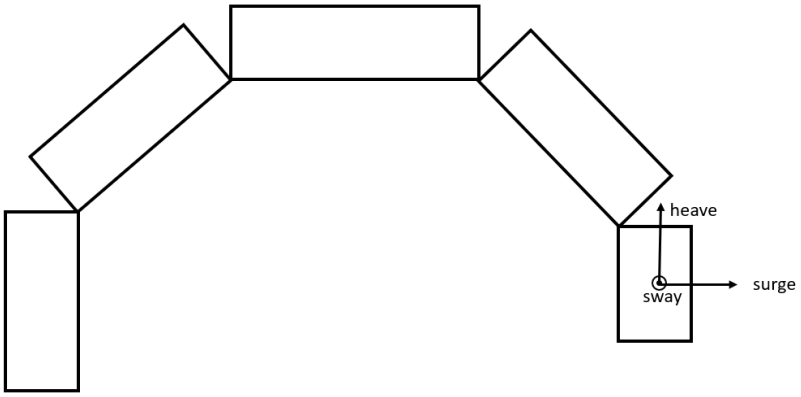


Figure 4.19: Surge, sway and heave in u-shape

Note that the gravitational force and the buoyancy of link 5 are bigger than the gravitational force and buoyancy of link 1. When the robot is in u-configuration, there would be symmetry about the z-axis if link 1 and link 5 were not present. However, the inclusion of link 1 and link 5 results in a positive pitch moment, which is shown in Equation 4.3. Let W_i and B_i be the weight and buoyancy of each link, where $W_i = B_i$ and $W_5 > W_1$. Let L be the distance from the CO of link 3 to the CO of link 1 and link 5. Assume that for each link, CG is placed in CO, while CB is placed a distance δ above CO. These forces and distances are illustrated in Figure 4.20. The moment about the origin of link 3 becomes:

$$\Sigma M = B_5(\frac{L}{2} + \delta) - W_5\frac{L}{2} - B_1(\frac{L}{2} + \delta) + W_1\frac{L}{2} = \delta(W_5 - W_1) > 0 \tag{4.3}$$

This is significant, because the robot will as a result be slightly tilted in equilibrium without pitch control. The BODY frame is fixed to link 1, and will therefore also tilt slightly towards the seabed.

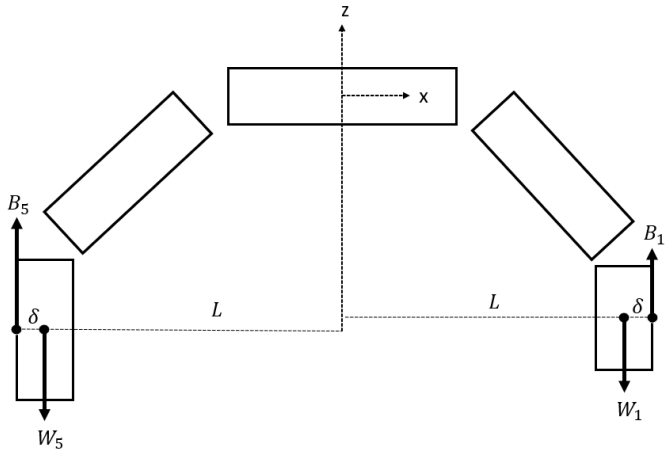
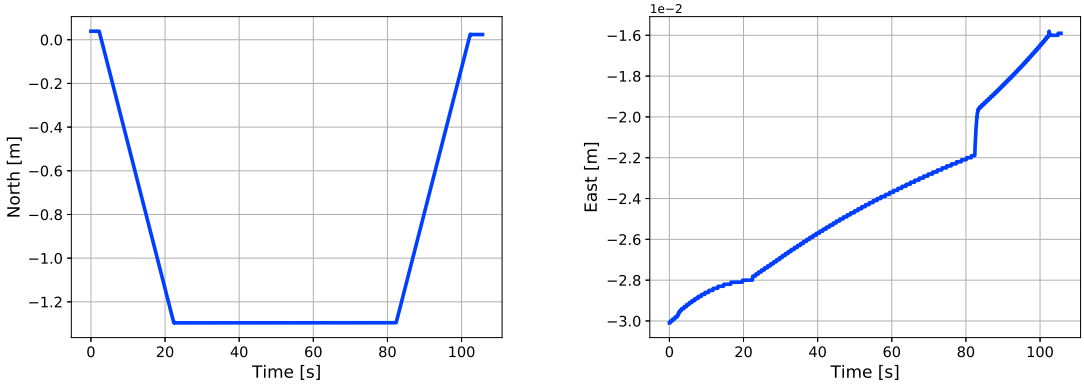


Figure 4.20: Righting moment in u-shape

Since the body frame is not orthogonal to the NED frame, a commanded surge motion will result in descent in addition to forward motion. There are several ways of achieving a orthogonality between the BODY frame and the NED frame. One solution would be to implement a pitch controller that ensures a 90 degree pitch angle. However a simpler solution would be to add extra ballast to link 5, or extra buoyancy to link 1, or slightly change one or more of the joint angles. These options were not considered during the simulations.

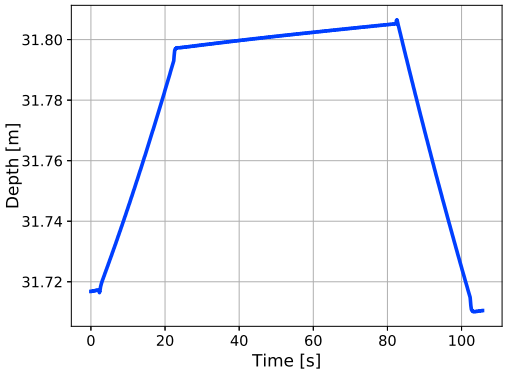
Figures 4.21, 4.22 and 4.23 show results for commanded surge, sway and heave motion respectively. Plots for north, east and depth are included, in addition to the roll angle, which represents the heading when in u-shape, as well as screenshots from the simulator in each case to provide visual information.

4.3.1 Results

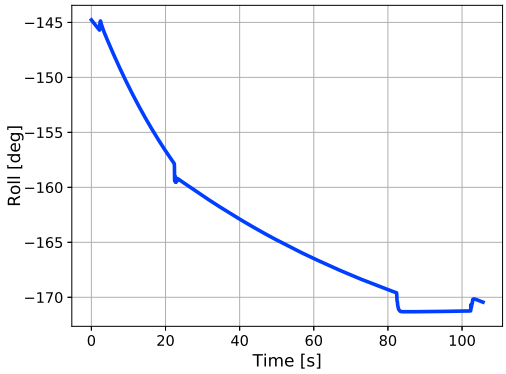


(a) North

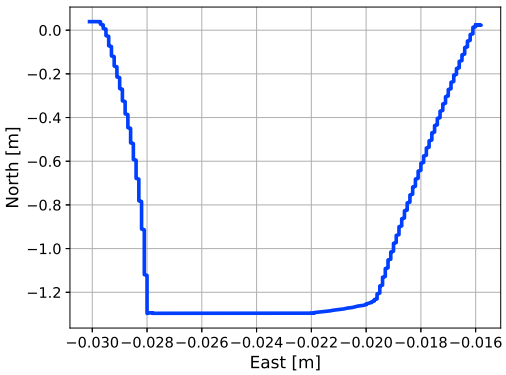
(b) East



(c) Depth



(d) Roll

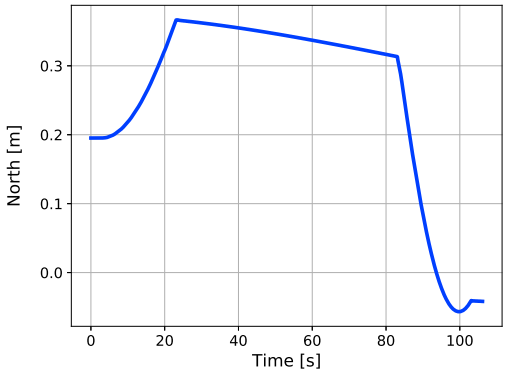


(e) North-East plot

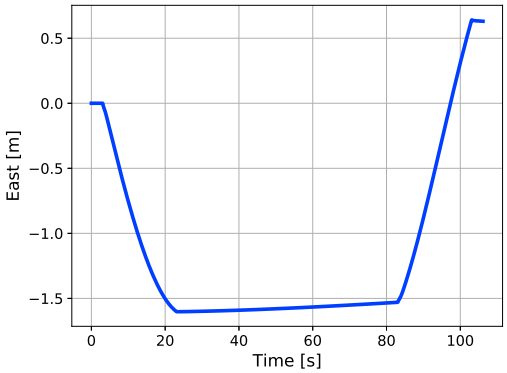
(f) Visual

Figure 4.21: U-shape surge motion

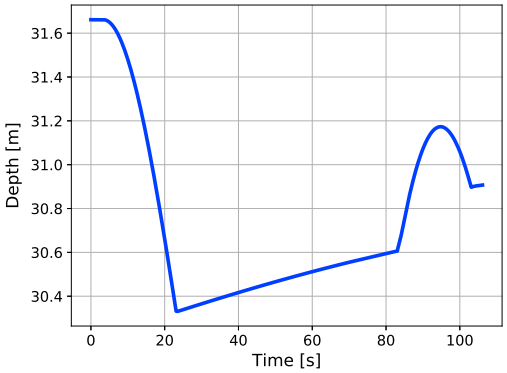
Figure 4.21a, 4.21b and 4.21c shown the position of the robot in the north, east and depth axes respectively. Figure 4.21d shows the heading. Figure 4.21e is a north-east plot that shows the position in the north-east plane. Finally, Figure 4.21f shows a screenshot from the simulator when the robot receives the inputs. No significant change in orientation can be seen.



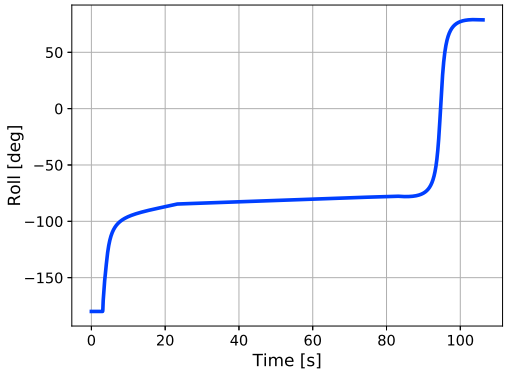
(a) North



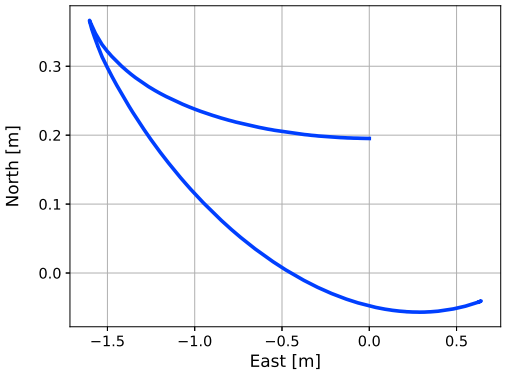
(b) East



(c) Depth



(d) Roll

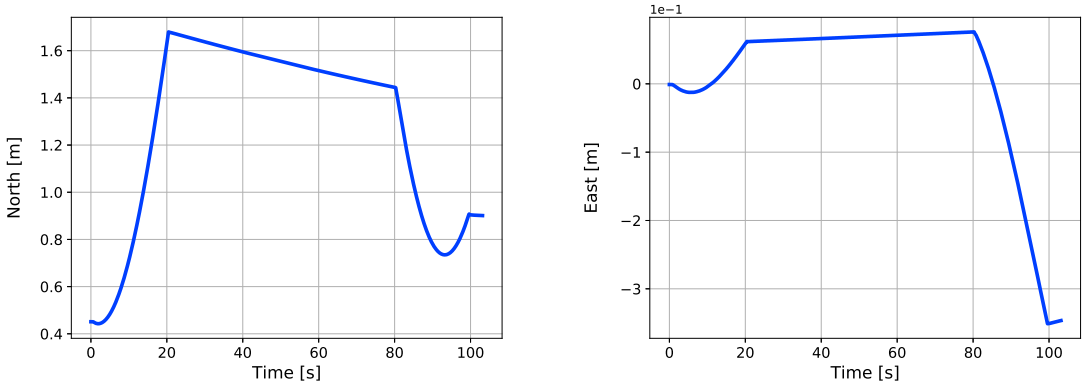


(e) North-East plot

(f) Visual

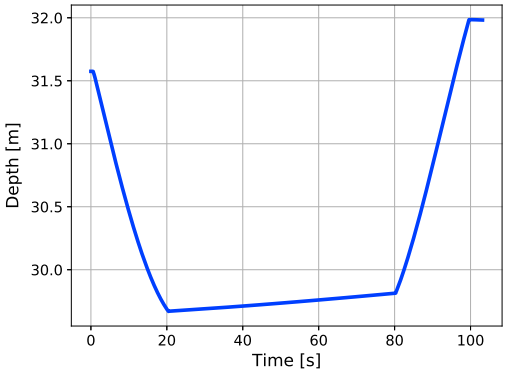
Figure 4.22: U-shape sway motion

Figure 4.22a, 4.22b and 4.22c shown the position of the robot in the north, east and depth axes respectively. Figure 4.22d shows the heading. Figure 4.22e is a north-east plot that shows the position in the north-east plane. Finally, Figure 4.22f shows a screenshot from the simulator when the robot receives the inputs, such that the change in orientation can clearly be seen.

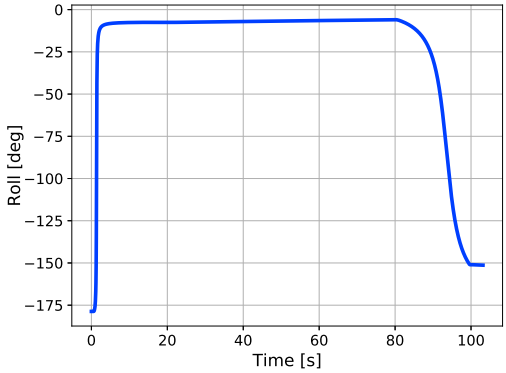


(a) North

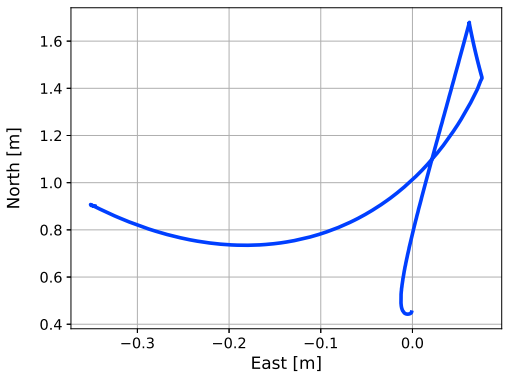
(b) East



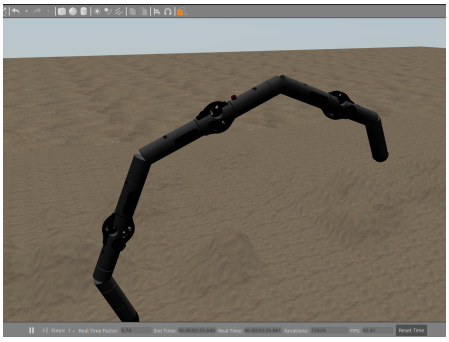
(c) Depth



(d) Roll



(e) North-East plot



(f) Visual

Figure 4.23: U-shape heave motion

Figure 4.23a, 4.23b and 4.23c shown the position of the robot in the north, east and depth axes respectively. Figure 4.23d shows the heading. Figure 4.23e is a north-east plot that shows the position in the north-east plane. Finally, Figure 4.23f shows a screenshot from the simulator when the robot receives the inputs, such that the change in orientation can clearly be seen.

4.4 Discussion

It can be seen from the results of commanded surge during a straight-shaped vehicle that the system behaves as expected. Figure A.7a shows that the surge velocity resembles the step inputs given to the vehicle, while Figure A.7b and Figure A.7c show that the sway and heave velocities are very small compared to the surge velocity. Figure 4.15e shows that the vehicle is able to move in a straight line back and forth, with a slight change in heading. Surge motions are produced by the thrusters placed on the starboard and port side of the vehicle. Because of the angle of these thrusters, a roll moment is also produced as a consequence of the surge commands. The speed controller tries to maintain zero velocity in roll. This must be compensated for by the thrusters on the top and bottom of the vehicle. The roll moment produced by the thrusters on the sides of one link is $M_{side} \approx 0.25T$ Nm. The roll moment produced by the top and bottom thrusters producing the same amount of thrust is $M_{top} = 0.2T$ Nm. This means that the top and bottom thrusters must produce a slightly larger thrust force in order to maintain zero velocity in roll. Figure A.13 shows that this is case during the simulations, since the thrust forces produced by thruster 0 and 1 are larger than the thrust forces produced by thrusters 2 and 3, which suggests that the simulator works according to theory. It can also be seen that since thruster 7 produces zero thrust, the other thrust forces are distributed to maintain a straight forward movement. This illustrates the benefit of having redundant thrusters, since the vehicle can be controlled as desired despite a malfunctioning thruster. Although the current thruster module is able to maintain zero roll velocity, the angle of the thruster on the sides can be disadvantageous, as extra power is needed in order to counteract the moments produced by the thrusters. If the thrusters only produced thrust in one direction, one could allocate more power to move forward, rather than using extra power to maintain the attitude of the vehicle.

Only the top and bottom thrusters produce thrust in the y-direction of the links, and thus these are the only necessary thrusters for producing sway motions. Figure A.14 shows that there are thrust forces present for the side thrusters, although they are significantly smaller than the thrust produced by the top and bottom thrusters. These thrust forces may be the result of poor optimization in the thrust allocation, however the forces could also be produced in an attempt to maintain a constant yaw angle. It can be seen from Figure 4.16d that there is a large change in heading when sway motion is commanded, suggesting that the speed controller is not able to maintain the heading angle. Notice that Figure A.14 shows that the thrust produced by thruster 0 and 1 is larger than the thrust produced by the other top and bottom thrusters. In other words, there is more thrust near the head link or the robot, which causes a yaw moment. The speed controller is tuned with surge motion in mind, and not optimized for sway and heave motions. Added mass and damping effects are larger in the y and z-directions, which may imply that the controller is not able to command enough force , such that the controller gains must be increased. Similar results can be found for commanded heave motions for a straight shaped vehicle, as Figure A.3b shows a large change in pitch, and Figure A.15 shows that the thrusters closest to the head link produce significantly more force than the other thrusters. Coupled motion also appear for the robot in u-shape. One of the suggested operations for a u-shaped vehicle is following along a straight pipeline. The results from Figure 4.22e show that when the robot receives a commanded sway motion, the robot moves in a curved line rather than a straight line because of rotations of the vehicle. This further enhances the point that a more sophisticated control system is needed to control the robot as desired.

The base link which coincides with the BODY frame is located in the same position for both of the simulated body configurations, which is at the CO of the head link. Where the base link is placed can have consequences for how the vehicle is controlled. As explained in Section 4.3, when the vehicle has a u-shape, commands for surge motion results in movement towards the sea floor, which means that commands in surge can be defined as resulting in heave motion. This can be confusing for a human controlling the robot, as the orientation of the BODY frame is rotated compared to how it is defined for straight-shape. Different individuals may have different interpretations of what should be considered surge and sway motion for a vehicle that can change body configuration. Thus it is necessary clearly define these motions. In the case of u-configuration, it may be more intuitive for a human to place the BODY frame on the center link, which has an unchanged orientation compared to the vehicle when all joints are straightened. On the other hand, for inspection and

mapping operations where the camera is placed on the head, it may be more intuitive design a control system where the BODY frame is placed on the head link, as one could wish to control the camera position to be close to the position of an object of interest. One consequence of defining the BODY frame at the head link can be seen in Figure 4.22. It can be seen that the north and east position of the BODY frame changes. However this might not be the case for the frame attached to link 3. Consider that link 3 rotates about the x-axis, the north and east position of the frame attached to link would remain the same, while the north and east position of the head link changes. Because of this, control of the robot with the BODY frame may not produce the intended results, as the head my change position while other links do not. There is no clear answer as to what should be defined as the base link, or how the BODY frame of a robot that can change body configuration should be defined. Pettersen (2017) defines the position of a snake robot as the canter of mass, and the orientation as the average of the joint angles relative to the global or NED frame. Letting the BODY frame be located at the CG of the robot, means that the BODY frame will change position based on the BODY configuration. This eliminates the need to have a predefined BODY frame for each body configuration, however the thrust allocation problem is further complicated, as it may be more difficult to determine the position of the thruster frames relative to a BODY frame that that can change position relative to all the links of the vehicle.

Operations concerning visual inspection and mapping using cameras would require precise control of the vehicle, and therefore it is desirable to control each DOF separately. By investigating the equations of motion for the snake robot, it is obvious that the rigid motions are coupled. This can for example be seen by noting that the rigid body mass matrix in Equation 2.41 is not diagonal, such that change in one DOF can cause changes in other DOFs. According to Fossen (2021), slender symmetrical bodies can be decomposed as a forward speed subsystem, and a pitch-depth system, that are decoupled from each other. This is supported by the simulation results, as commended surge motion for both body configurations produce insignificant change in other DOFs, while sway motion produces a large change in heading, and heave motion produces a large change in pitch. It is therefore necessary that the control system incorporates MIMO control. The speed controller used for the simulation case study attempts to maintain zero velocity for the states that are not commanded to change, however when the controller is unable to maintain for instance no change in pitch, no attempt is made to return to the original pitch angle. In order to maintain a desired attitude it is necessary to develop a controller akin to the MIMO nonlinear controller described in Section 3.3.2, and a guidance system that provides the desired states based on the task. To be able to control all DOFs it is important that the robot has a sufficient amount of control forces. For most UUVs such as torpedo-shaped AUVs and ROVs it is not necessary to be able to control the roll or pitch angle. However for underwater snake robots, the definition of roll and pitch may be changed when the body configuration changes, such that control over all DOFs may contribute to a more flexible system. The current thruster module used for EELY500 is able to produce thrust such that all DOFs can be controlled however as mentioned earlier, the thrusters produce moments that must be repressed by other thrusters, which can consume more power than necessary. It is therefore discussed whether the thrusters can be attached to the robot in a different way in order to control the robot more efficiently. One example is presented in Kelasidi et al. (2016), where experiments are performed for the underwater snake robot *Mamba* which only features two thrusters on the tail link that induce linear forces along the body. In other words, the two thrusters only produce forces along the x-axis of the tail link. Forward motion is in this case caused by the thrusters, while direction control is achieved by actuating the joints. This thruster module greatly simplifies the thrust allocation problem while also taking advantage of the snake robot's ability to swim like a snake. On the other hand, since there are only thrusters that produce thrust in i single direction, the ability to make minor adjustments in each direction is lost, which may be important for inspection and mapping operations. Especially mapping of steep terrain would be difficult, as the robot would lack the ability to produce heave motions. Other possibilities for changes may be reducing moments that need to be repressed. This can be achieved by reducing the lever arms of each thruster, however the ability to produce moments is important for direction control. One option may be thruster modules resembling tunnel thrusters, where the thrusters are placed at the CO if each link. Then multiple of these modules cold be placed on the robot, to gain the ability to produce moments for directional control. Such thruster modules are featured in the Eelume robot described in Liljebäck and Mills (2017) where they are referred to as lateral thruster modules. On the other hand, when the configuration of the robot changes, the lever arms of the thrusters will

change, such that the thrusters could produce moments that need to be repressed. It may therefore be advantageous to design multiple thrust modules, where different modules are advantageous for different types of operations.

Chapter 5

Field Experiments

Field experiments with EELY500 were performed on May 2, 2022. The robot was launched from RV Gunnerus in the Trondheim Fjord. The purpose of the field experiments is performing a qualitative experiment in order to compare simulated behaviour of the robot with experimental results. Tests resembling those of the simulation case studies were performed. Commands to move in surge, sway and heave were given to the robot a straight and u-shaped body configuration. The commands were given by a human operator using a computer onboard RV Gunnerus, using a similar speed controller as used for simulations. In order to achieve heave motion when the robot was in u-configuration, a command for surge had to be given, which suggests that for EELY500 the base link is defined as the head link, and the thrust allocation is computed with a similar method as for the simulation model. Figure 5.1 shows a picture of EELY500 taken during the field experiments. During the field experiments, one of the thrusters on EELY500 malfunctioned and could not produce any thrust. The thruster in question corresponds to thruster 7 in figure 4.11. This was accounted for in the simulation case study by not producing thrust from thruster 7. This can be achieved by setting all the elements in the corresponding column of the thruster configuration matrix to zero, or simply by removing the thruster from the model.

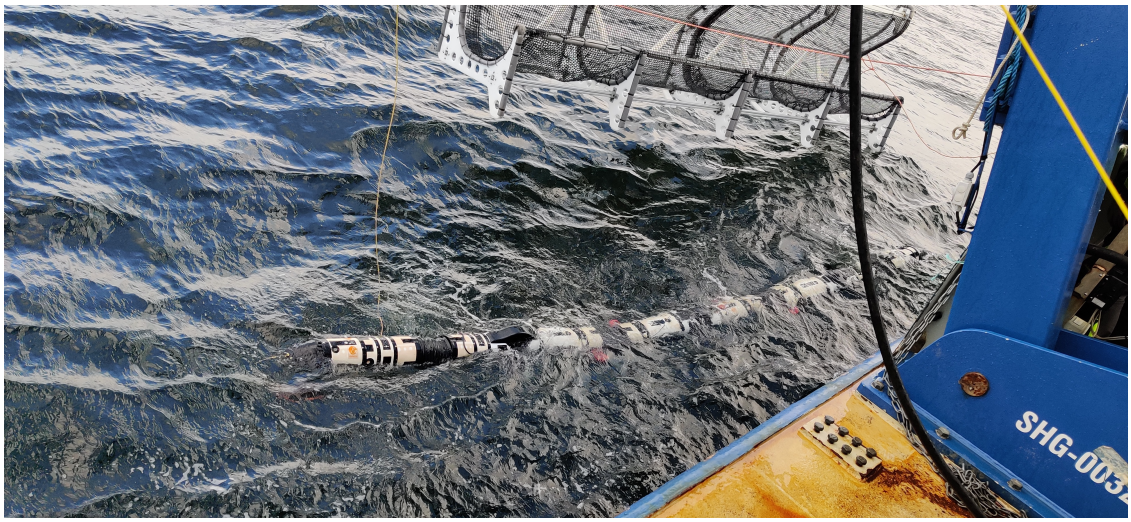


Figure 5.1: EELY500 in the Trondheim Fjord during the experiments

The position of the robot is measured with a HiPAP installed on RV Gunners, which means that the acoustic position measurements are given relative to RV Gunnerus. The global position is in turn measured with a GNSS system. This allows the north and east position of EELY500 to be given in Universal Transverse Mercator (UTM) coordinates. This coordinate system divides the world into multiple zones that can be projected on a flat plane (Toohey et al. 1988). Each of these zones can be compared to a NED frame. Resulting north and east positions from the field experiments are given in this coordinate system, where the axes are scaled to make the results more readable.

5.1 Signal Processing

Sensor measurements produces signals of varying quality, and sometimes there may be faults in a measured signal. For a system relying on measured signals for guidance and control, it is important to control the signal quality to detect and remove errors. A few errors are detected in the measured signals from the field experiments. For example, Figure 5.7c display a few sudden jumps or spikes in the measurements, which is known as wild points. During the field experiments, the robot did not rely on position measurements for guidance or control, however large spikes in the measurements of the north and east position affects the readability of the north-east plots. Therefore measures are taken to remove or reduce the wild points that appear in the north and east measurements, such that north-east plots are easier to understand. A wild point is detected by checking if a measurement deviates significantly from the previous measurements. If a sampled value of a signal is within a band about the estimated mean signal, it may be accepted as shown in Equation 5.1

$$x[k] \in [\bar{x}_k - a\sigma, \bar{x}_k + a\sigma] \tag{5.1}$$

$x[k]$ is a sample of a signal, $\bar{x}_k$ is the estimated mean signal, a is a scaling factor, and σ is the standard deviation. If the sample $x[k]$ is rejected, it is replaced by an estimated mean value. Wild point detection is illustrated in Figure 5.2 (Sørensen 2018).

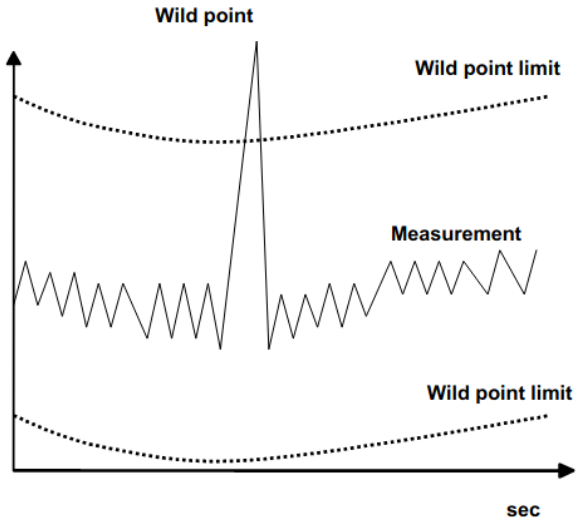
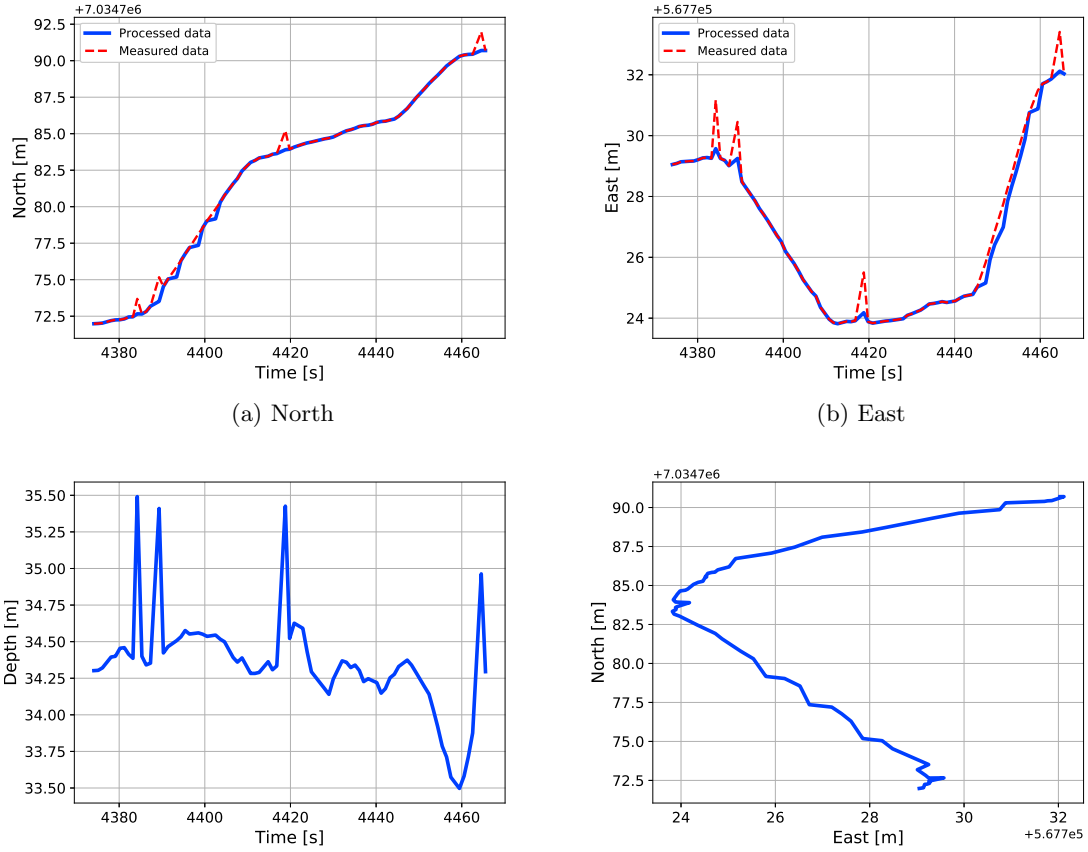


Figure 5.2: Wild point test (Sørensen 2018)

5.2 Results

Results for measured north, east, depth are shown for commanded surge, sway and heave motion for a straight-shaped and u-shaped body configuration. North and east measurements have been processed in order to reduce wild points. The processed data is used to create north-east plots. Figures 5.3, 5.4 and 5.5 show results for commanded surge, sway and heave motions respectively for straight-shape. Figures 5.6, 5.7 and 5.8 show results for commanded surge, sway and heave motion respectively for u-shape. Information about Euler angles and velocities was unfortunately not available, so analysis of the behaviour of the vehicle is only based on the changes in position.



(a) North

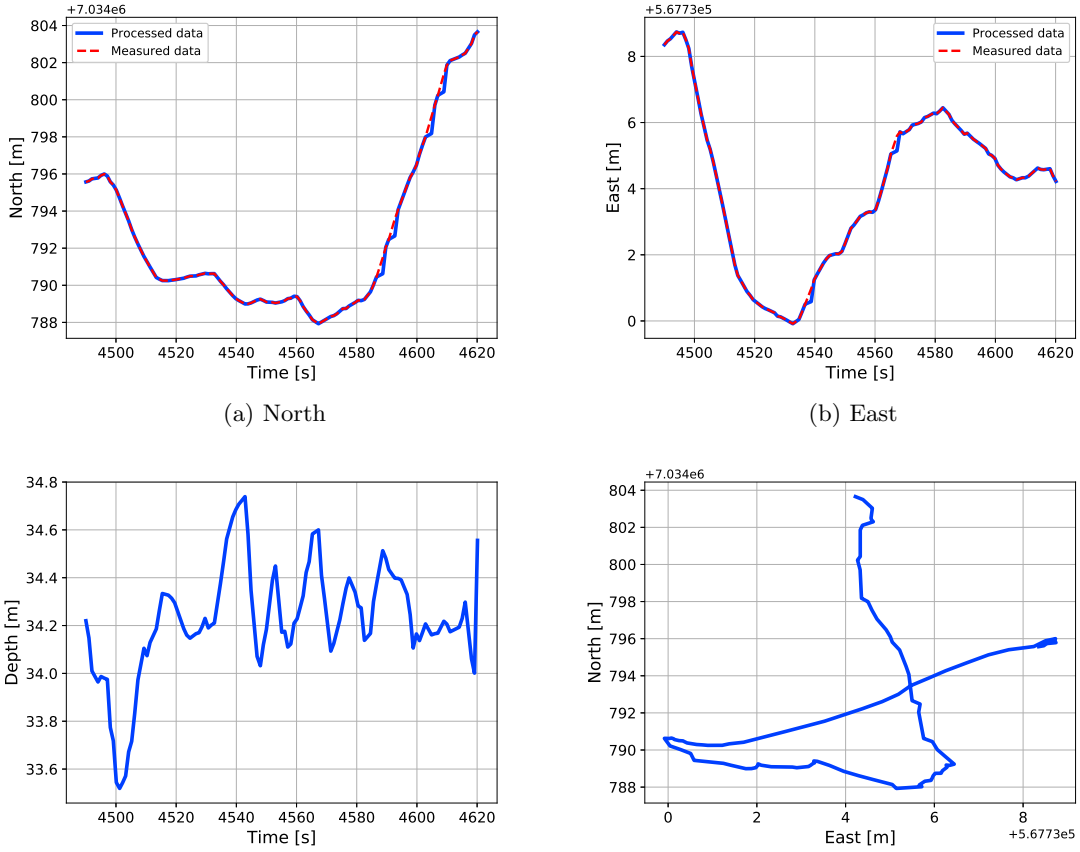
(b) East

(c) Depth

(d) North-East plot

Figure 5.3: Straight-shape surge motion

Figure 5.3a, 5.3b and 5.3c shown the position of the robot in the north, east and depth axes respectively. Wild point removal has been performed for the north and east data. Figure 5.3d is a north-east plot that shows the position in the north-east plane.



(a) North

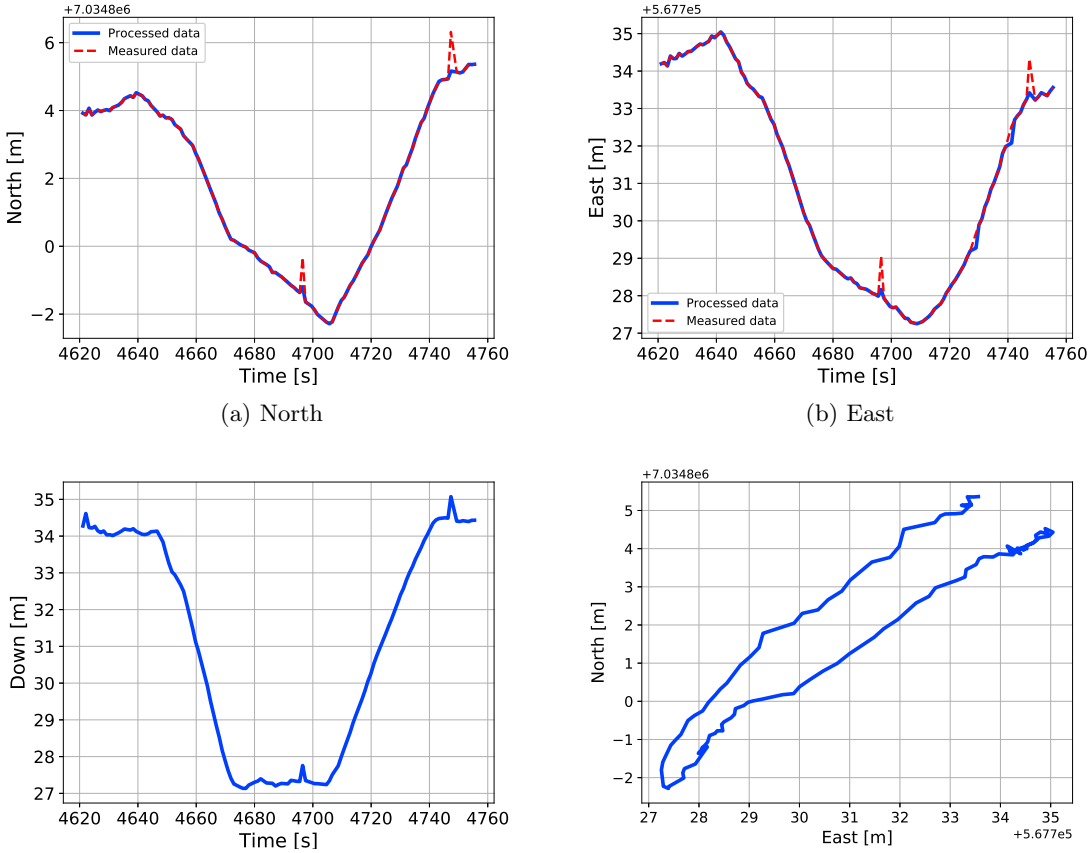
(b) East

(c) Depth

(d) North-East plot

Figure 5.4: Straight-shape sway motion

Figure 5.4a, 5.4b and 5.4c shown the position of the robot in the north, east and depth axes respectively. Wild point removal has been performed for the north and east data. Figure 5.4d is a north-east plot that shows the position in the north-east plane.

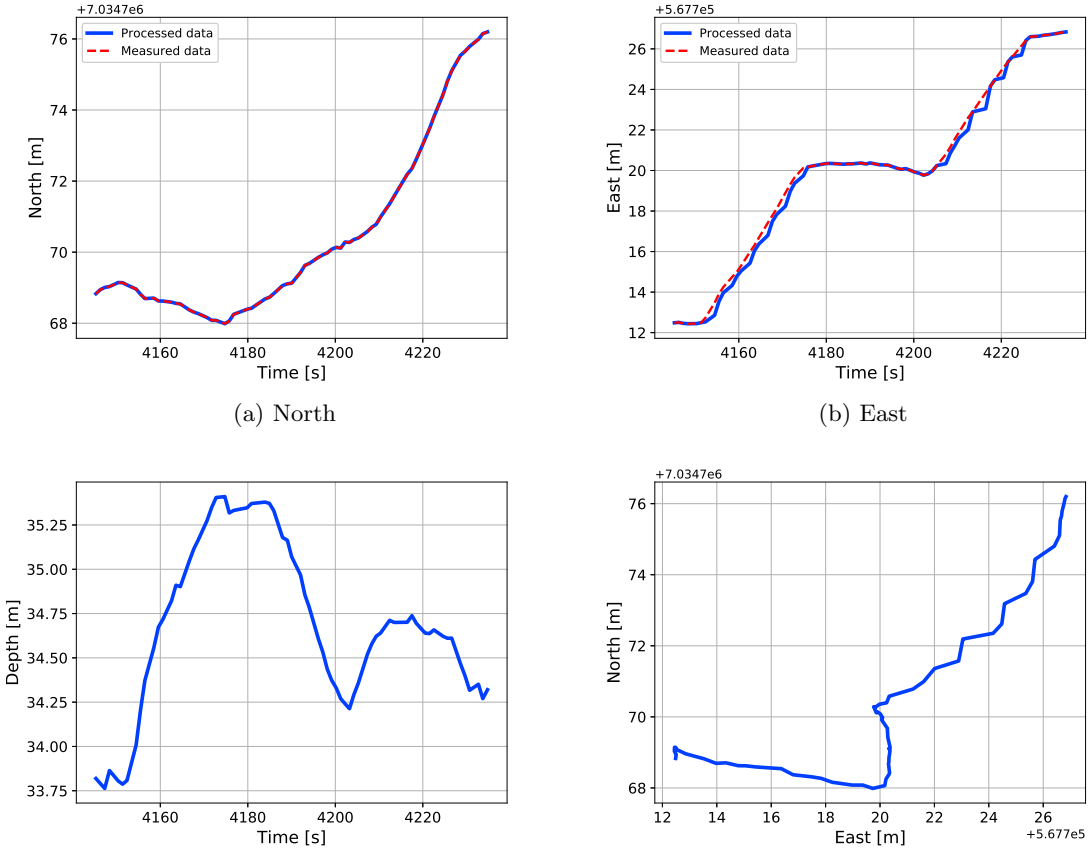


(a) North (b) East

(c) Depth (d) North-East plot

Figure 5.5: Straight-shape heave motion

Figure 5.5a, 5.5b and 5.5c shown the position of the robot in the north, east and depth axes respectively. Wild point removal has been performed for the north and east data. Figure 5.5d is a north-east plot that shows the position in the north-east plane.



(a) North

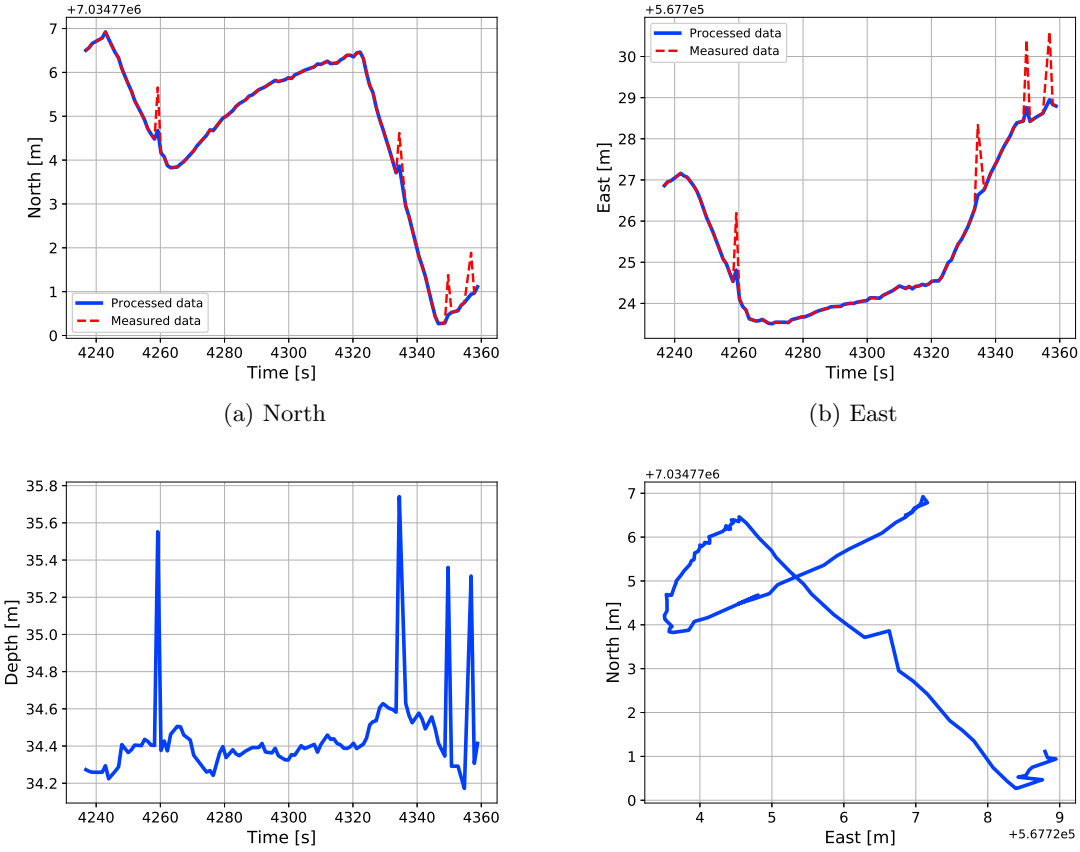
(b) East

(c) Depth

(d) North-East plot

Figure 5.6: U-shape surge motion

Figure 5.6a, 5.6b and 5.6c shown the position of the robot in the north, east and depth axes respectively. Wild point removal has been performed for the north and east data. Figure 5.6d is a north-east plot that shows the position in the north-east plane.

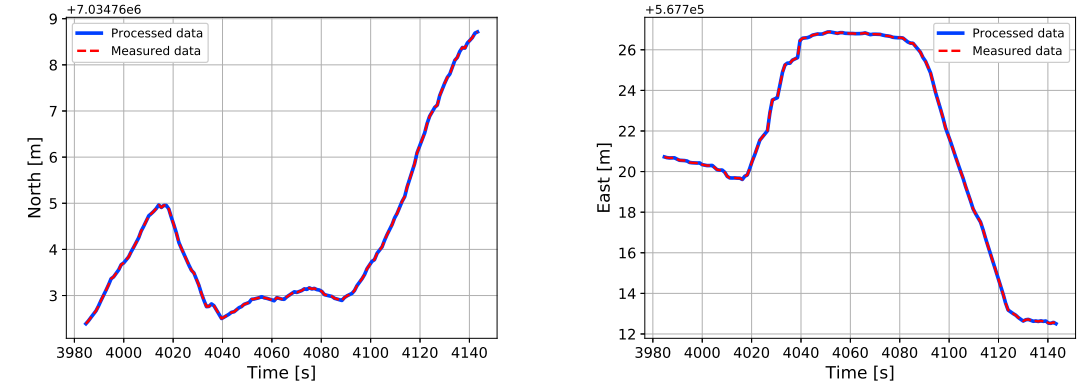


(a) North (b) East

(c) Depth (d) North-East plot

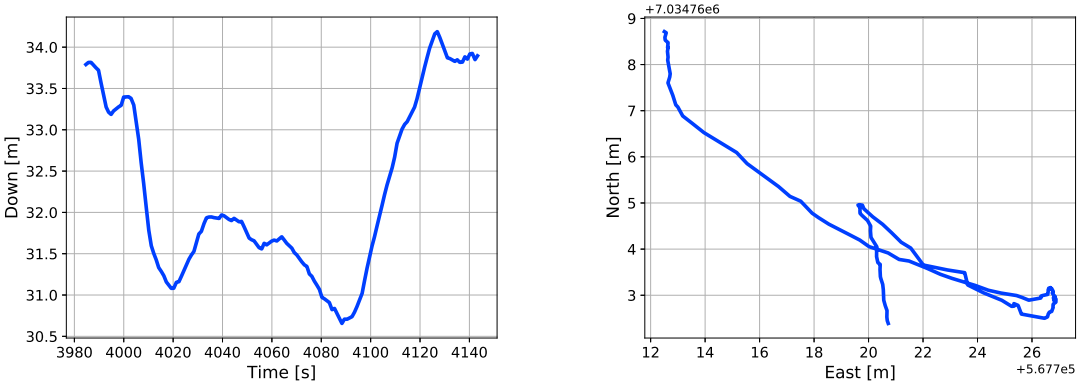
Figure 5.7: U-shape sway motion

Figure 5.7a, 5.7b and 5.7c shown the position of the robot in the north, east and depth axes respectively. Wild point removal has been performed for the north and east data. Figure 5.7d is a north-east plot that shows the position in the north-east plane.



(a) North

(b) East



(c) Depth

(d) North-East plot

Figure 5.8: U-shape heave motion

Figure 5.8a, 5.8b and 5.8c shown the position of the robot in the north, east and depth axes respectively. Wild point removal has been performed for the north and east data. Figure 5.8d is a north-east plot that shows the position in the north-east plane.

5.3 Discussion

The goal of the field experiments is to investigate how the simulation model can be improved to more closely resemble the real system by comparing the simulated responses to the real responses when the vehicle is given certain inputs. Firstly, it should be noted that when compared to simulations in an idealistic environment, there will arise differences and difficulties in a field experiment that are not accounted for in simulation. For instance, the timing and duration of each input is not an exact match compared to the simulation case studies. Additionally it is necessary to manage the tether, and sometimes it may be necessary to perform maneuvers in order to avoid entanglement, which is not an issue in the simulator. A large change in heading can for example be seen in Figure 5.3d. This could be caused by an error made by the operator, or by other unknown external factors. The simulated vehicle is modeled as rigid body, that are rigidly attached to each other. In reality, the vehicle is much more flexible, as a thin, elongated body will experience deformation when subjected to external forces. Additionally, the simulated joints are at a constant fixed angle, while in reality the joints are controlled using a similar controller as shown in Section 3.3.3, and as a consequence the joint angles are constantly adjusting. Figure 5.9 shows the target angle versus the actual angle of joint 1 for the entire duration of the field experiments. This can cause motions and behaviour that are not present in the simulator. Similar behaviour could potentially be captured in the simulator by implementing a similar joint controller, instead of using fixed joints. If one wishes to use Plankton to simulate control systems for an underwater snake robot, it is important to model flexibility in the system, as a flexible system may react differently to control inputs compared to a rigid body.

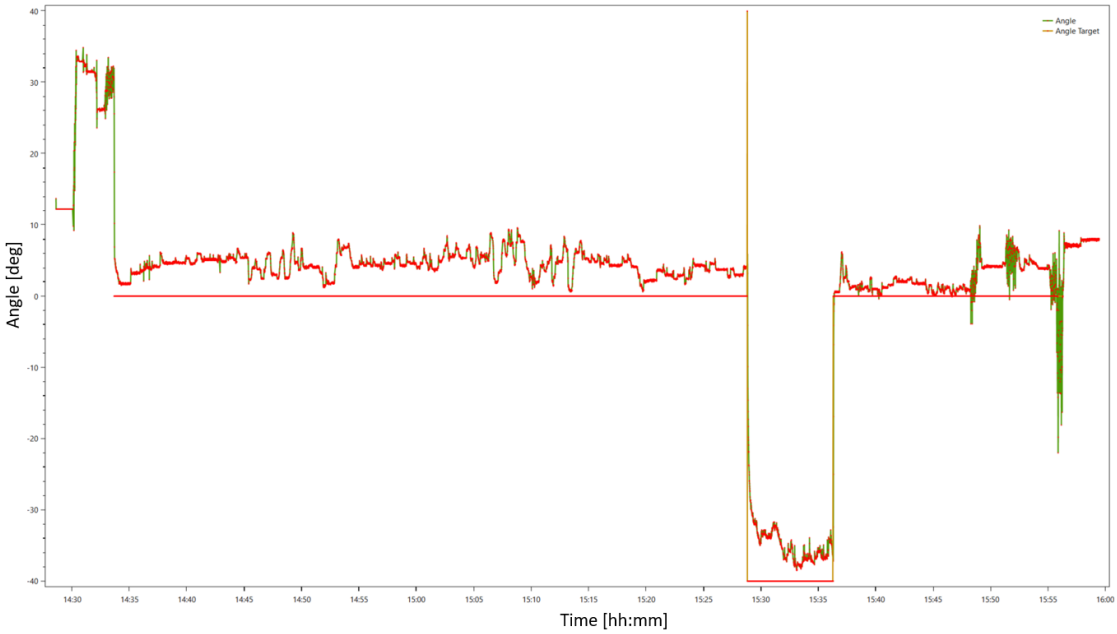


Figure 5.9: Target angle and actual angle of joint 1

One significant difference that can be seen when comparing field experiment results to the simulated results, is that the real vehicle achieves higher velocities than the simulated vehicle. A simple estimate using the change in north and east seen in Figure 5.3a and Figure 5.3b, suggests that the vehicle achieves an average forward speed of around 0.5 m/s when the vehicle is in straight-shape and is commanded to move in surge. The zenith of Figure A.7a shows that the vehicle achieves a max forward velocity of 0.15 m/s when the same test is performed in the simulation case study. Similar differences can be found when comparing the velocities for commanded surge in u-shape. One of the main reasons that the simulated vehicle moves slower, could be that the controller gains are too small, such that not enough force is exerted by the thrusters. Since the vehicle became unstable in sway and heave when the controller gains were set higher, a more sophisticated controller should be implemented for the simulator, such that different gains can be chosen for

each direction. The low-pass filter used to compute hydrodynamic effects as explained in Manhães et al. (2016) may also have an effect on the velocity of the simulation model. Figure 5.5 and Figure 5.8 show that for both cases, commanded heave motion results in significant change in north and east position in addition to change in depth. The same can be seen from the simulation results in Figure 4.17 and Figure 4.23, although the simulated changes are smaller. This suggests that similar coupled motions appear for the real robot and the simulated model.

One of the biggest uncertainties of the simulation model is the modeling of the hydrodynamic damping effects, and the higher velocity achieved by the robot in the simulation model may suggest that more work is required for modeling hydrodynamic effects in order to create a more realistic model. The drag coefficients chosen for the linear drag matrix Equation 2.60 mostly stem from guesses. How much detail is necessary for modeling of hydrodynamic effects depend on the purpose of the model. For the purpose of model based control design, it is not necessary with a detailed model (Kelasidi et al. 2014), and the current model for the damping matrix may be sufficient. However for a simulation model, or process plant model, with the intention of investigating the real plant dynamics, it is important with high accuracy in the model (Sørensen 2018). Therefore more work need to be done to increase the accuracy of the model, while also including other viscous effects. Since measurement noise and errors may occur, it is important that the control system is robust and able control the vehicle despite measurement noise. In order to use the simulator to test control systems in such conditions, it is necessary to be able to simulate noisy measurements. Irregularities appear in almost all of the results from the field experiments, and the plots may be described as rough or rugged. This may be caused by measurement noise, however vibrations in the vehicle may also be a cause. Vibrations may be caused by the joint motors constantly working to maintain the desired joint angles, and vibrations induced by vortex shedding is also not outside the realm of possibility, as the vehicle has a long cylindrical shape (King 1977). Modeling all hydrodynamic effects of the vehicle is a challenge involving highly nonlinear behaviour, and more research and testing is required to create an accurate hydrodynamic model.

Chapter 6

Conclusions and Further Work

6.1 Conclusions

The objective of this thesis has been developing a simulation model for an underwater snake to test and visualize the behaviour of the robot when subjected to certain inputs, with the ability to further develop control systems for the purpose of visual mapping and investigation. The necessary theory for mathematical modeling of underwater snake robots has been presented, in addition to examples of controllers, guidance systems and sensors for the applications discussed in this thesis.

A model for an underwater snake robot based on the Eelume vehicle EELY500 has been successfully implemented in the Plankton simulator, which runs using Gazebo with the plugin UUV Simulator with ROS 2 as middle ware. Using Gazebo to simulate robots and multi body systems is an advantage as the simulator handles transformation between reference frames, such that the user does not need to model this when designing robots. The user may define the joint angles and positions of the thruster before launch, and maneuver the robot with a joystick. It is currently not possible to change the joint angles during a simulation. The simulation model is able to reproduce results known from theory, namely that there is coupling between sway and yaw motions, and heave and pitch motions, which can be considered decoupled from the surge motions for a slender and symmetrical vehicle. Results from the simulations show that the speed controller was able to suppress moments induced by angled thrusters to a degree, however the controller was not able to suppress coupled behaviour. Malfunctioning thrusters can be simulated, and both the real vehicle and the simulation model could be maneuvered without issue with one missing thruster, which supports the need for redundancy in the system. Further development such as implementing a guidance system, improving the controller and simulating sensor measurements is possible.

Due to their modularity, and ability to dynamically change the shape of their bodies, it can be concluded that underwater snake robots are well suited for visual inspection and mapping of difficult terrain and subsea structures, and the model created in the Plankton simulator can be used to test and develop methods for this application. The real vehicle achieved higher velocities during field experiments, than the simulation model was able to achieve. One probable cause is that the controller gains used for simulations are too small such that the commanded thrust is insufficient. There were vibrations present when the real vehicle was piloted that were not present in the simulator, which suggests that there is still physical phenomena that should be modeled for a more complete and realistic simulation model.

6.2 Further Work

There is much potential for future development of the simulator. Firstly, more time could be spent tuning the speed controller, while also expanding the controller such that different gains can be chosen for different directions. This could hopefully allow the vehicle to move at a higher velocity. Additionally a controller should be implemented such that the vehicle can maintain desired states. For more realistic behaviour in the simulator, more time could be spent on hydrodynamic modeling, such that the damping coefficients more closely resembles reality. This could be achieved by performing model tests or CFD analyzes. The control system can also be expanded to include a guidance system that provides desired states to the controller based on trajectory tracking.

Gazebo has the ability to simulate cameras, which can be useful to simulate inspections regarding visual inspection and mapping. There is also the ability to model the simulated world and insert objects, such that for example a pipeline could be added to the simulated world. Several sensors can also be simulated, to give more realistic measurements of the states of the robot, rather than assuming the ground truth is known. Other important functionalities would be implementing the ability to dynamically control the joints. A joint torque controller, like the controller used for the real Eelume vehicle during the field experiments, could be used to maintain the desired joint angles rather than assuming fixed joints. If the vehicle has the ability to dynamically change body configurations during simulations, a dynamic thrust allocation matrix must be implemented as well.

More cases could have been simulated to further understand the behaviour of the robot. Additional tests could include testing more body configurations, adding current forces, or changing the placement and angles of the thrusters to investigate more optimal behaviour to potentially reduce power consumption.

Bibliography

Albert, A. (1972). 'Chapter III Geometric and Analytic Properties of the Moore-Penrose Pseudoinverse'. In: *Regression and the Moore-Penrose Pseudoinverse*. Vol. 94. Mathematics in Science and Engineering. Elsevier, pp. 15–42. DOI: https://doi.org/10.1016/S0076-5392(08)62919-7.

Anderson, K. S. (1990). *Recursive derivation of explicit equations of motion for efficient dynamic/control simulation of large multibody systems*. Doctoral thesis, Stanford University.

Antonelli, G. (2014). *Underwater Robots*. Springer.

Christ, R. D. and Wernli, R. L. (2014). *The ROV Manual*. Elsevier Ltd.

Dukan, F. and Sørensen, A. J. (2012). 'Joystick in Closed-loop Control of ROVs with Experimental Results'. In: *IFAC Proceedings Volumes* 45.5. 3rd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles, pp. 293–300. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20120410-3-PT-4028.00049.

Eelume (2022). *Reshaping Underwater Operations*. Last accessed 8 April 2022. URL: https://eelume.com/.

Faltinsen, O. M. (1990). *Sea Loads on Ships and Offshore Structures*. Cambridge University Press.

Featherstone, R. (2008). *Rigid Body Dynamics Algorithms*. Springer.

Ferreira, B. M., Matos, A. C. and Cruz, N. A. (2011). 'Modeling and control of TriMARES AUV'. In.

Foote, T. (Apr. 2013). 'tf: The transform library'. In: *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on*. Open-Source Software workshop, pp. 1–6. DOI: 10.1109/TePRA.2013.6556373.

Fossen, T. I. (2021). *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, Inc.

From, P. J., Gravdahl, J. T. and Pettersen, K. Y. (2014). *Vehicle-Manipulator Systems*. Advances in Industrial Control. Springer.

Janota, A., Šimák, V., Nemec, D. and Hrbček, J. (2015). 'Improving the Precision and Speed of Euler Angles Computation from Low-Cost Rotation Sensor Data'. In: *Sensors* 15.3, pp. 7016–7039. DOI: 10.3390/s150307016. URL: https://www.mdpi.com/1424-8220/15/3/7016.

Johansen, G., Ludvigsen, M., Sørensen, A. J. and Aas, L. M. S. (2016). 'The use of underwater hyperspectral imaging deployed on remotely operated vehicles - methods and applications'. In: *IFAC-PapersOnLine* 49.23. 10th IFAC Conference on Control Applications in Marine SystemsCAMS 2016, pp. 476–481. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2016.10.451. URL: https://www.sciencedirect.com/science/article/pii/S2405896316320390.

Josephs, H. and Huston, R. L. (2002). *Dynamics of Mechanical Systems*. CRC Press.

Kanso, E., Marsden, J., Rowley, C. and Melli-Huber, J. (Aug. 2005). 'Locomotion of Articulated Bodies in a Perfect Fluid'. In: *Journal of Nonlinear Science* 15, pp. 255–289. DOI: 10.1007/s00332-004-0650-9.

Kelasidi, E., Pettersen, K. Y., Gravdahl, J. T. and Liljebäck, P. (2014). 'Modeling of underwater snake robots'. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4540–4547. DOI: 10.1109/ICRA.2014.6907522.

Kelasidi, E., Pettersen, K. Y., Liljebäck, P. and Gravdahl, J. T. (2016). 'Locomotion efficiency of underwater snake robots with thrusters'. In: *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 174–181. DOI: 10.1109/SSRR.2016.7784295.

King, R. (1977). 'A review of vortex shedding research and its application'. In: *Ocean Engineering* 4.3, pp. 141–171. ISSN: 0029-8018. DOI: https://doi.org/10.1016/0029-8018(77)90002-6.

Kongsberg (2022). *HIPAP OPERATING PRINCIPLES*. Last accessed 23 May 2022. URL: https://www.kongsberg.com/no/maritime/support/themes/hipap-principles/.

Liljebäck, P. and Mills, R. (2017). 'Eelume: A flexible and subsea resident IMR vehicle'. In: *OCEANS 2017 - Aberdeen*, pp. 1–4.

Liu, B., Liu, Z., Men, S., Li, Y., Ding, Z., He, J. and Zhao, Z. (2020). 'Underwater Hyperspectral Imaging Technology and Its Applications for Detecting and Mapping the Seafloor: A Review'. In: *Sensors* 20.17. ISSN: 1424-8220. DOI: 10.3390/s20174962. URL: https://www.mdpi.com/1424-8220/20/17/4962.

Ludvigsen, M. (2010). *An ROV toolbox for optical and acoustical seabed investigations*. Doctoral thesis, Norwegian University of Science and Technology.

Ludvigsen, M., Johnsen, G., Lagstad, P., Sørensen, A. and Odegard, O. (June 2013). 'Scientific Operations Combining ROV and AUV in the Trondheim Fjord'. In: vol. 48, pp. 1–7. ISBN: 978-1-4799-0000-8. DOI: 10.1109/OCEANS-Bergen.2013.6608194.

Ludvigsen, M. and Sørensen, A. J. (2016). 'Towards integrated autonomous underwater operations for ocean mapping and monitoring'. In: *Annual Reviews in Control* 42, pp. 145–157. ISSN: 1367-5788. DOI: https://doi.org/10.1016/j.arcontrol.2016.09.013. URL: https://www.sciencedirect.com/science/article/pii/S1367578816300256.

Macenski, S., T., Foote, Gerkey, B., Lalancette, C. and Woodall, W. (2022). 'Robot Operating System 2: Design, architecture, and uses in the wild'. In: *Science Robotics* 7.66. DOI: 10.1126/scirobotics.abm6074. URL: https://www.science.org/doi/abs/10.1126/scirobotics.abm6074.

Manhães, M. M. M., Scherer, S. A., Voss, M., Douat, L. R. and Rauschenbach, T. (Sept. 2016). 'UUV Simulator: A Gazebo-based package for underwater intervention and multi-robot simulation'. In: *OCEANS 2016 MTS/IEEE Monterey*. IEEE. DOI: 10.1109/oceans.2016.7761080.

Mare, José (2010). 'Path following algorithm for minimally specified lawn-mower type AUV missions'. In: *OCEANS'10 IEEE SYDNEY*, pp. 1–5. DOI: 10.1109/OCEANSSYD.2010.5603826.

McMillan, S., Orin, D.E. and McGhee, R.B. (1995). 'Efficient dynamic simulation of an underwater vehicle with a robotic manipulator'. In: *IEEE Transactions on Systems, Man, and Cybernetics* 25.8, pp. 1194–1206. DOI: 10.1109/21.398681.

Nornes, S. M., Ludvigsen, M. and Sørensen, A. J. (2016). 'Automatic relative motion control and photogrammetry mapping on steep underwater walls using ROV'. In: pp. 1–6. DOI: 10.1109/OCEANS.2016.7761252.

— (2017). 'Motion Control of ROVs for Mapping of Steep Underwater Walls'. In: *Sensing and Control for Autonomous Vehicles: Applications to Land, Water and Air Vehicles*. Ed. by T. I. Fossen, K. Y. Pettersen and H. Nijmeijer. Springer International Publishing, pp. 51–69. ISBN: 978-3-319-55372-6. DOI: 10.1007/978-3-319-55372-6_3. URL: https://doi.org/10.1007/978-3-319-55372-6_3.

Pettersen, K. Y. (2017). 'Snake robots'. In: *Annual Reviews in Control* 44, pp. 19–44. ISSN: 1367-5788. DOI: https://doi.org/10.1016/j.arcontrol.2017.09.006.

Schiehlen, W. (2006). 'Computational dynamics: theory and applications of multibody systems'. In: *European Journal of Mechanics - A/Solids* 25.4. 6th EUROMECH Solid Mechanics Conference, pp. 566–594. ISSN: 0997-7538. DOI: https://doi.org/10.1016/j.euromechsol.2006.03.004.

Schmidt-Didlaukies, H. M., Sørensen, A. J. and Pettersen, K. Y. (2018). 'Modeling of Articulated Underwater Robots for Simulation and Control'. In: *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*, pp. 1–7.

Skjetne, R., Fossen, T. I. and V., Kokotović. P. (2004). 'Robust output maneuvering for a class of nonlinear systems'. In: *Automatica* 40.3, pp. 373–383. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2003.10.010.

SNAME (1950). 'Nomenclature for Treating the Motion of a Submerged Body Through a Fluid'. In: *Technical Research Bulletin No.*, pp. 1–5.

Sørensen, A. J. (2018). *Marine Cybernetics, Towards Autonomous Marine Operations and Systems, Lecture Notes*. Norwegian University of Science and Technology, pp. 245–273.

Spong, M. W., Hutchinson, S. and Vidyasagar, M. (2006). *Robot Modeling and Control*. John Wiley & Sons, Inc.

Sverdrup-Thygeson, J., Kelasidi, E., Pettersen, K. Y. and Gravdahl, J. T. (2018). 'The Underwater Swimming Manipulator—A Bioinspired Solution for Subsea Operations'. In: *IEEE Journal of Oceanic Engineering* 43.2, pp. 402–417. DOI: 10.1109/JOE.2017.2768108.

Toohey, R. E., Brown, W. and Stebbings, J. H. (1988). 'Systematic geographic sampling with UTM coordinates'. In: *Environment International* 14.3, pp. 207–212. ISSN: 0160-4120. DOI: https://doi.org/10.1016/0160-4120(88)90140-7.

Yeo, K. B., Wong, T. H. and M., Ong. C. (2014). 'Modelling and Manoeuvrability Design of Autonomous Underwater Vehicle'. In: *Journal of Applied Sciences* 14.10, pp. 991–999. DOI: 10.3923/jas.2014.991.999.

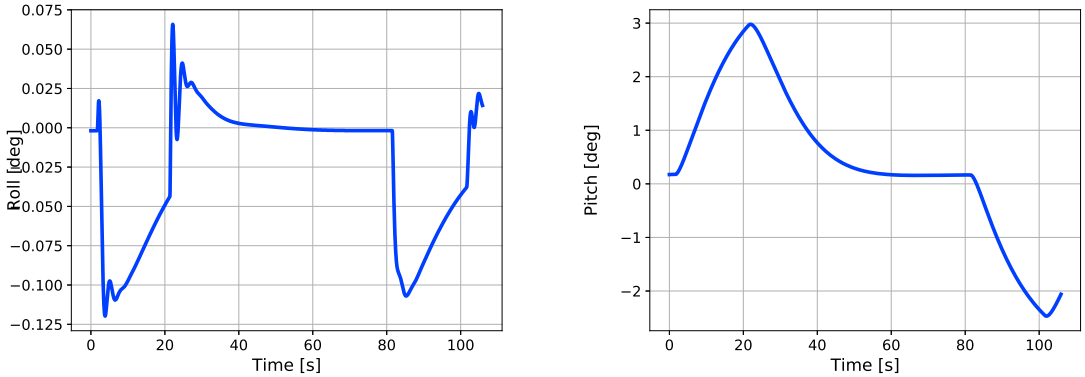
Yoerger, D.R., Cooke, J.G. and Slotine, J.-J.E. (1990). 'The influence of thruster dynamics on underwater vehicle behavior and their incorporation into control system design'. In: *IEEE Journal of Oceanic Engineering* 15.3, pp. 167–178. DOI: 10.1109/48.107145.

Zhang, T., Chen, S., Xu, G., Wang, C. and Tan, C. (Dec. 2019). 'Joint angle measurement of manipulator and error compensation based on an IMU sensor'. In: *The Journal of Engineering* 2019. DOI: 10.1049/joe.2018.9167.

Appendix A

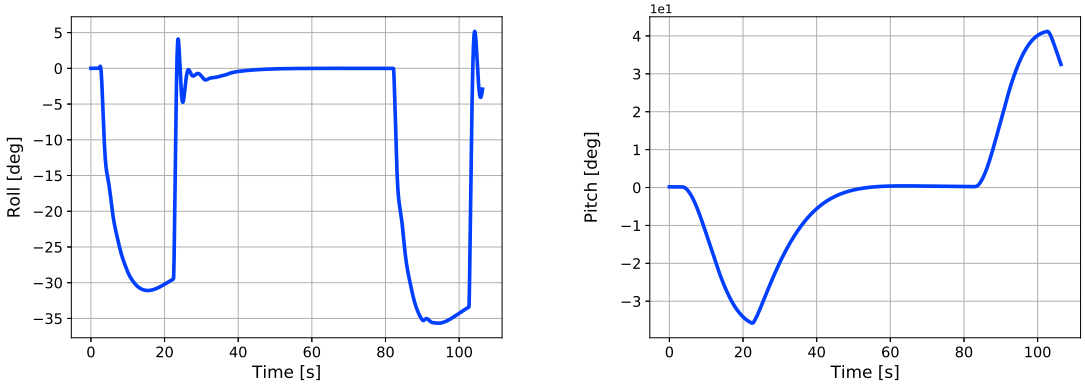
Additional Simulation Results

A.1 Euler Angles



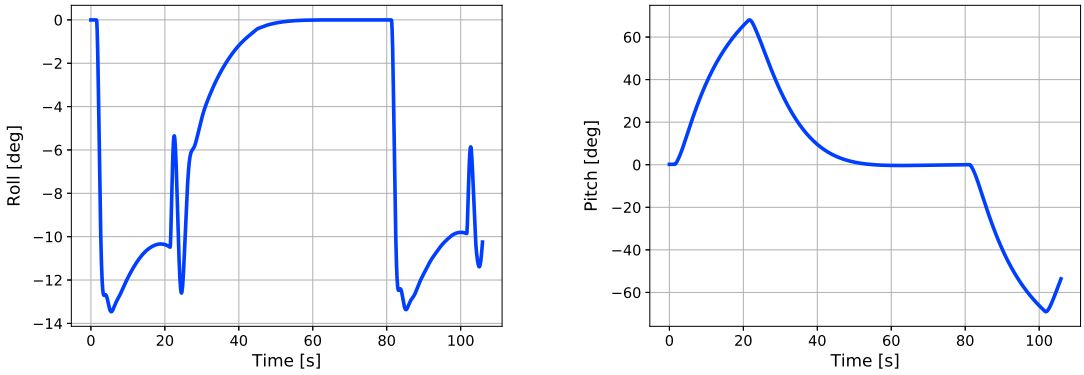
(a) Roll (b) Pitch

Figure A.1: Straight-shape surge



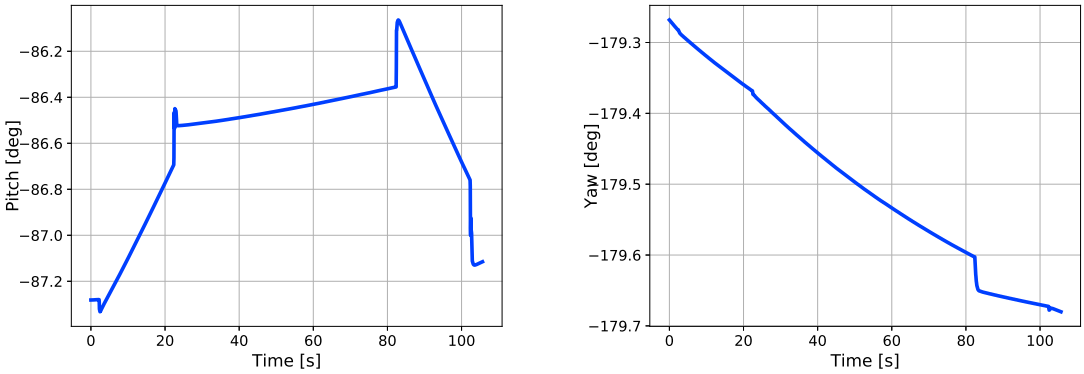
(a) Roll (b) Pitch

Figure A.2: Straight-shape sway



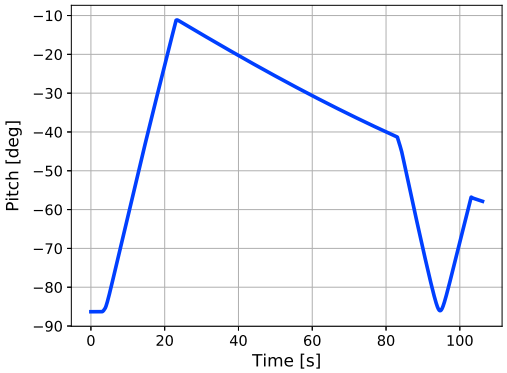
(a) Roll (b) Pitch

Figure A.3: Straight-shape heave

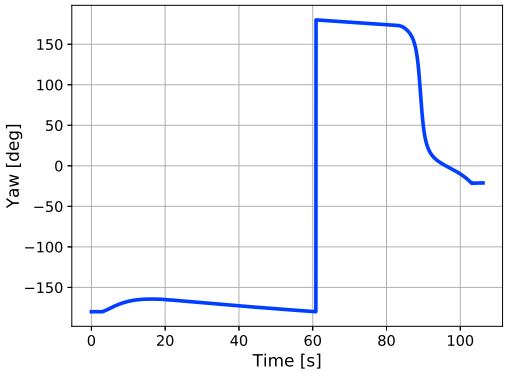


(a) Pitch (b) Yaw

Figure A.4: U-shape surge

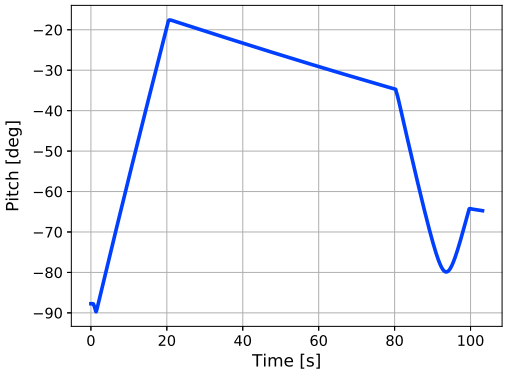


(a) Pitch

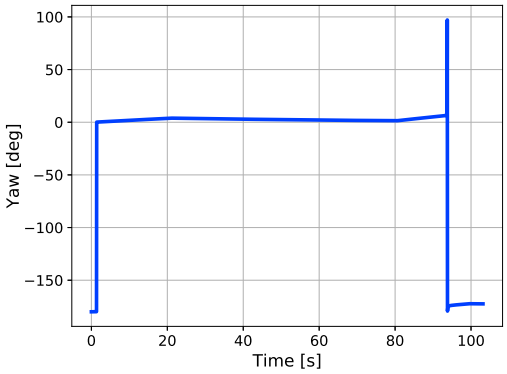


(b) Yaw

Figure A.5: U-shape sway



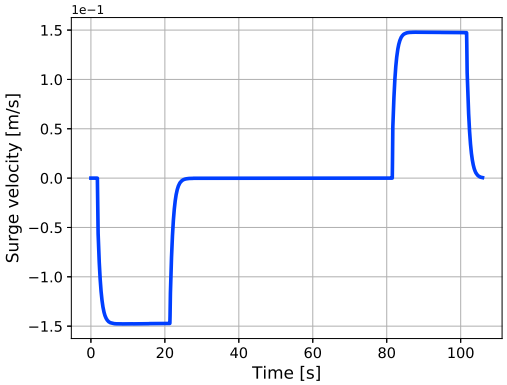
(a) Pitch



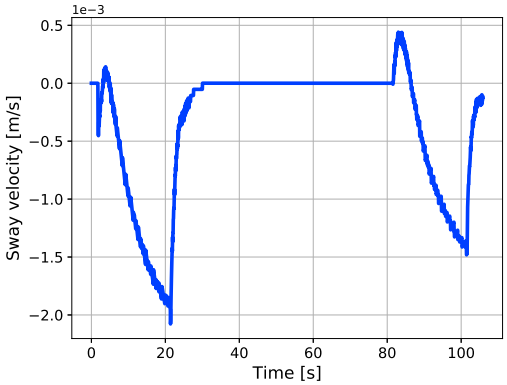
(b) Yaw

Figure A.6: U-shape heave

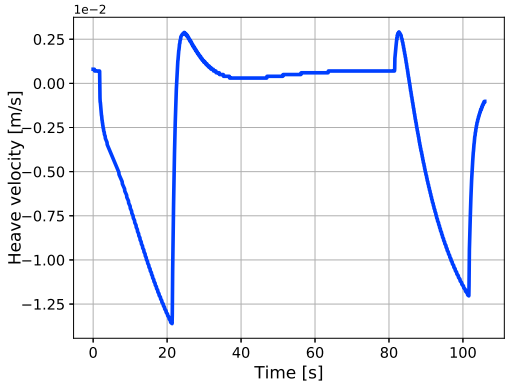
A.2 Velocity



(a) Surge

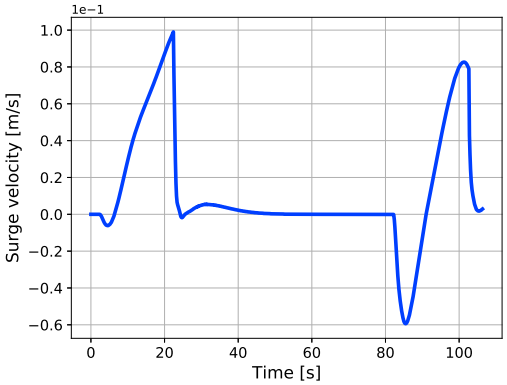


(b) Sway

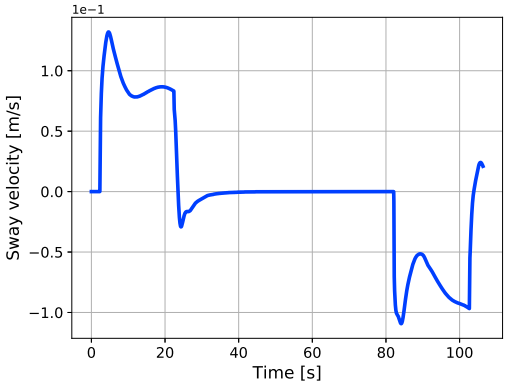


(c) Heave

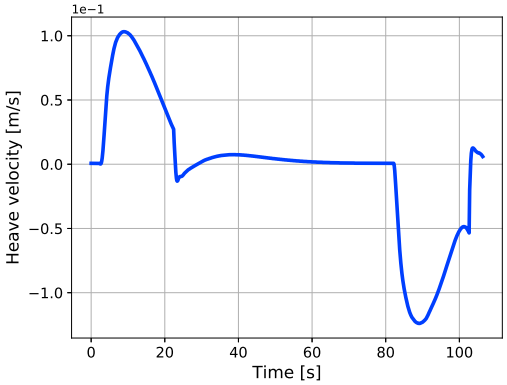
Figure A.7: Velocities for commanded surge in straight-shape



(a) Surge

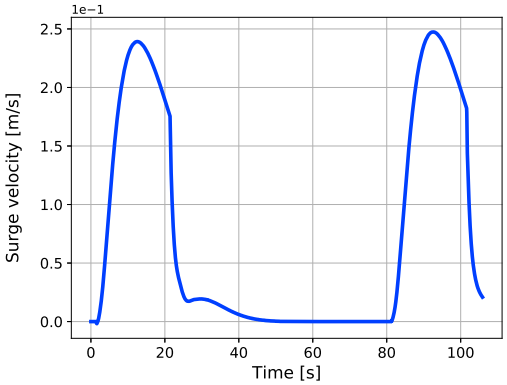


(b) Sway

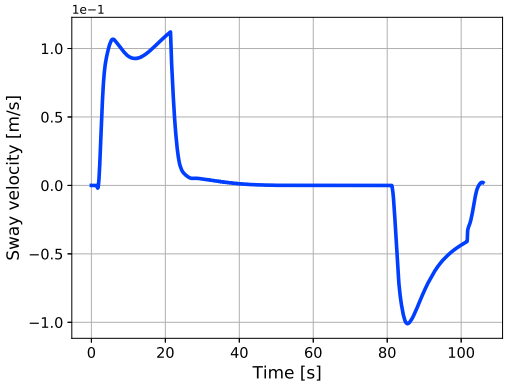


(c) Heave

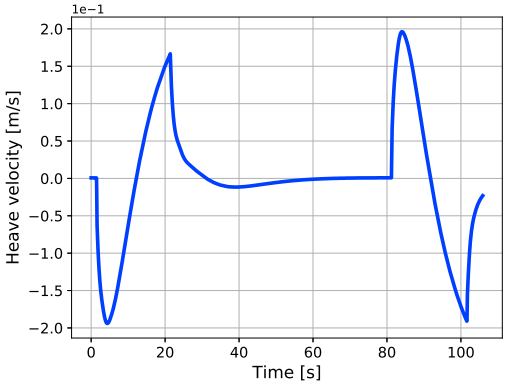
Figure A.8: Velocities for commanded sway in straight-shape



(a) Surge

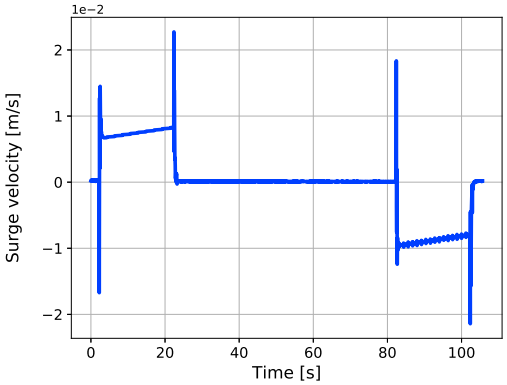


(b) Sway

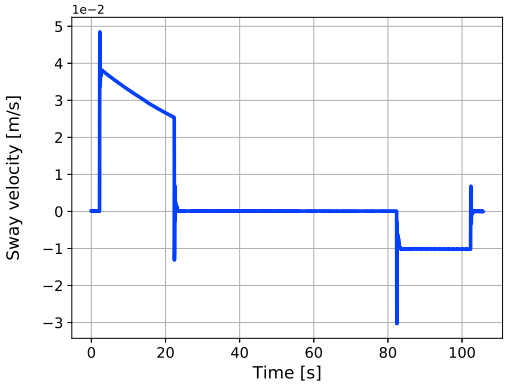


(c) Heave

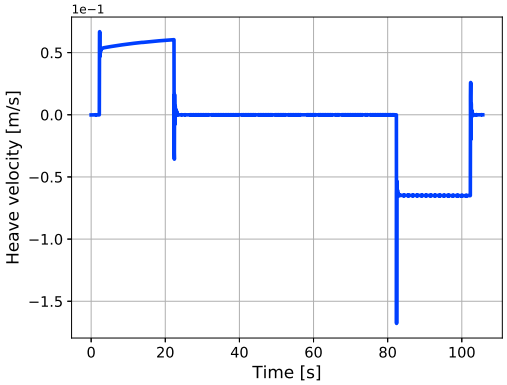
Figure A.9: Velocities for commanded heave in straight-shape



(a) Surge

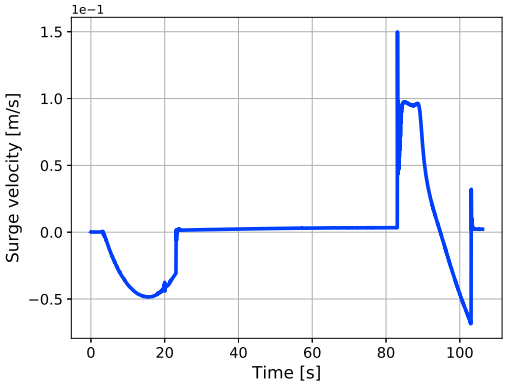


(b) Sway

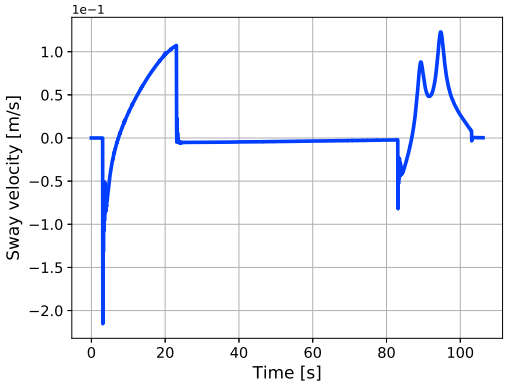


(c) Heave

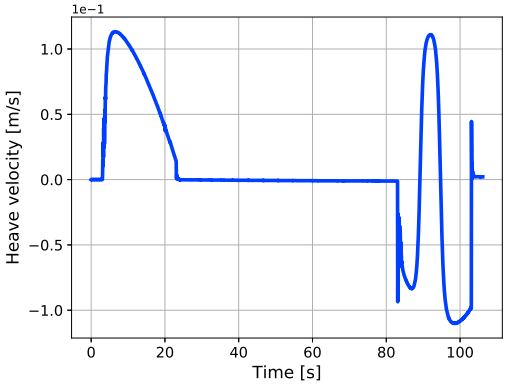
Figure A.10: Velocities for commanded surge in u-shape



(a) Surge

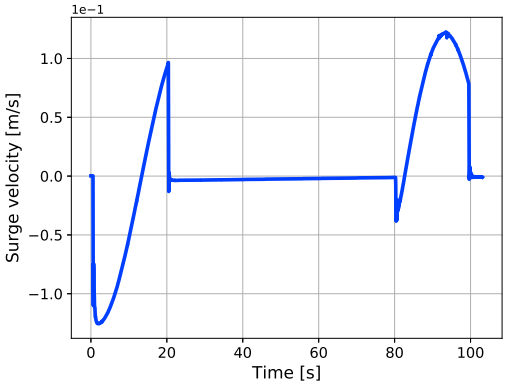


(b) Sway

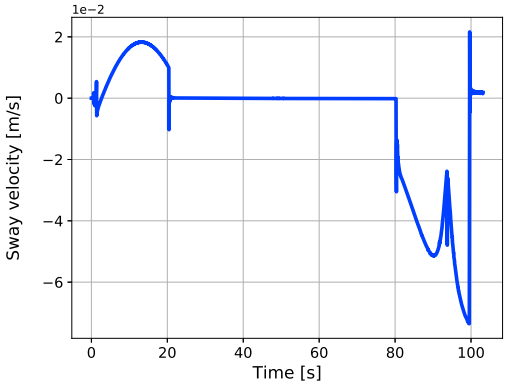


(c) Heave

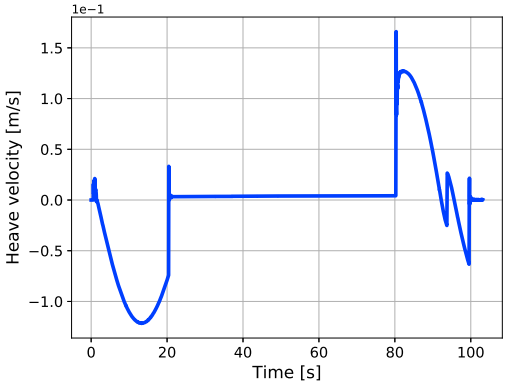
Figure A.11: Velocities for commanded sway in u-shape



(a) Surge



(b) Sway



(c) Heave

Figure A.12: Velocities for commanded heave in u-shape

A.3 Thrust

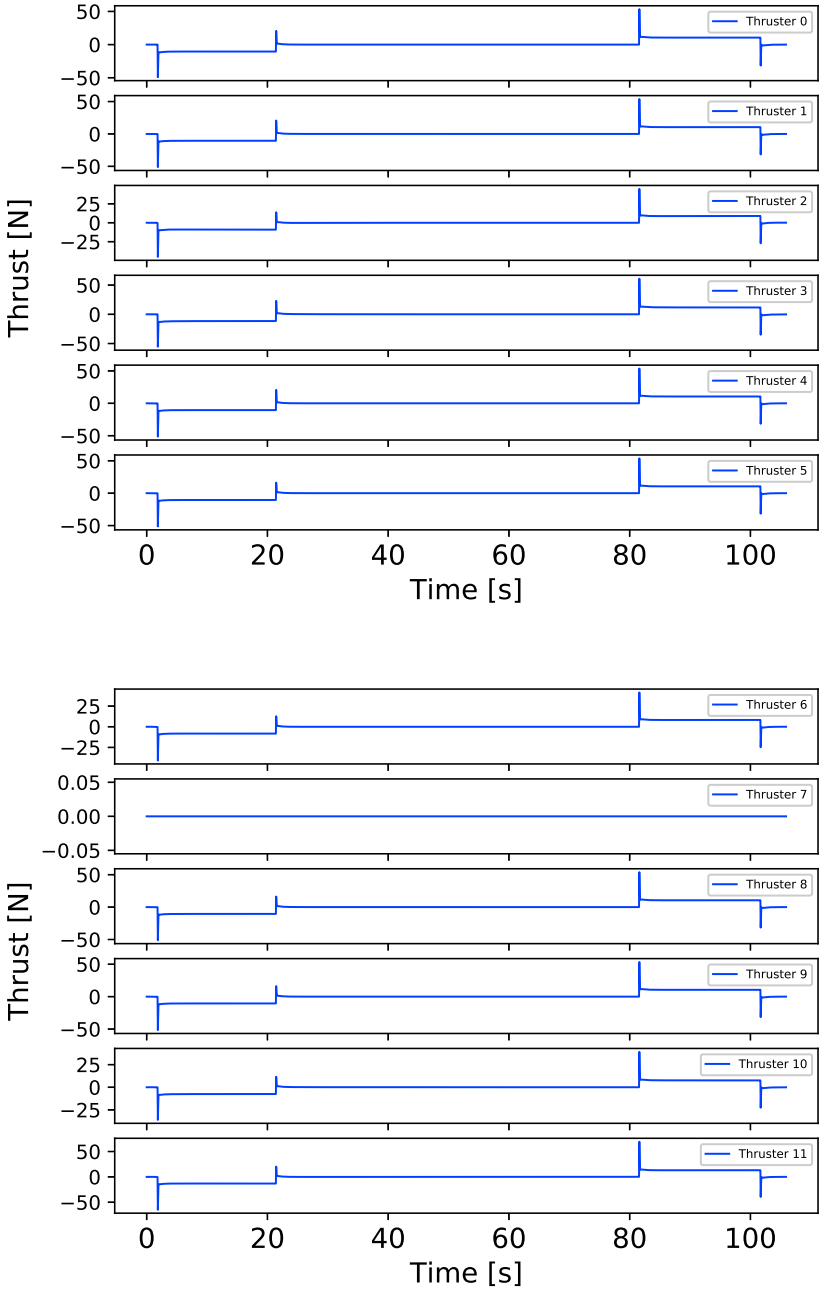


Figure A.13: Thrust straight-shape surge

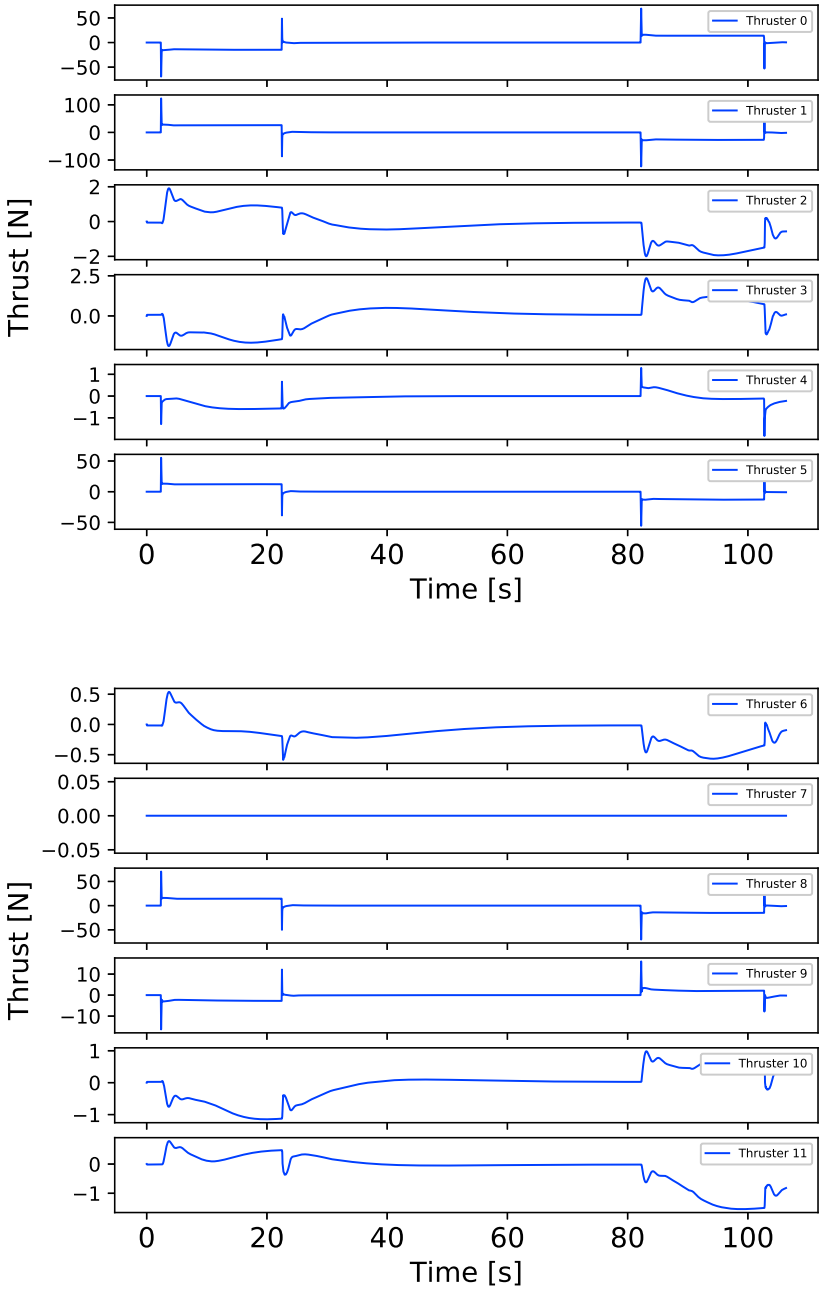


Figure A.14: Thrust straight-shape sway

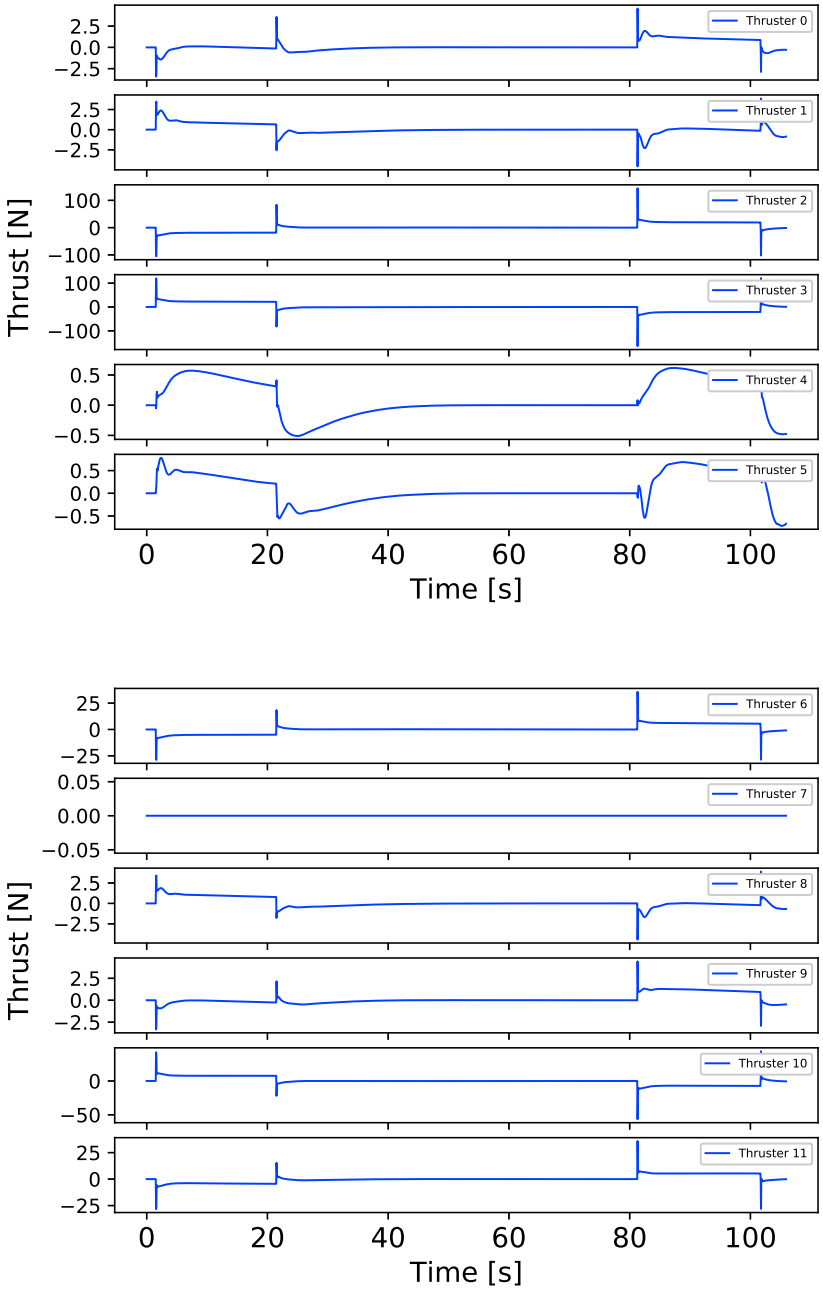


Figure A.15: Thrust straight-shape heave

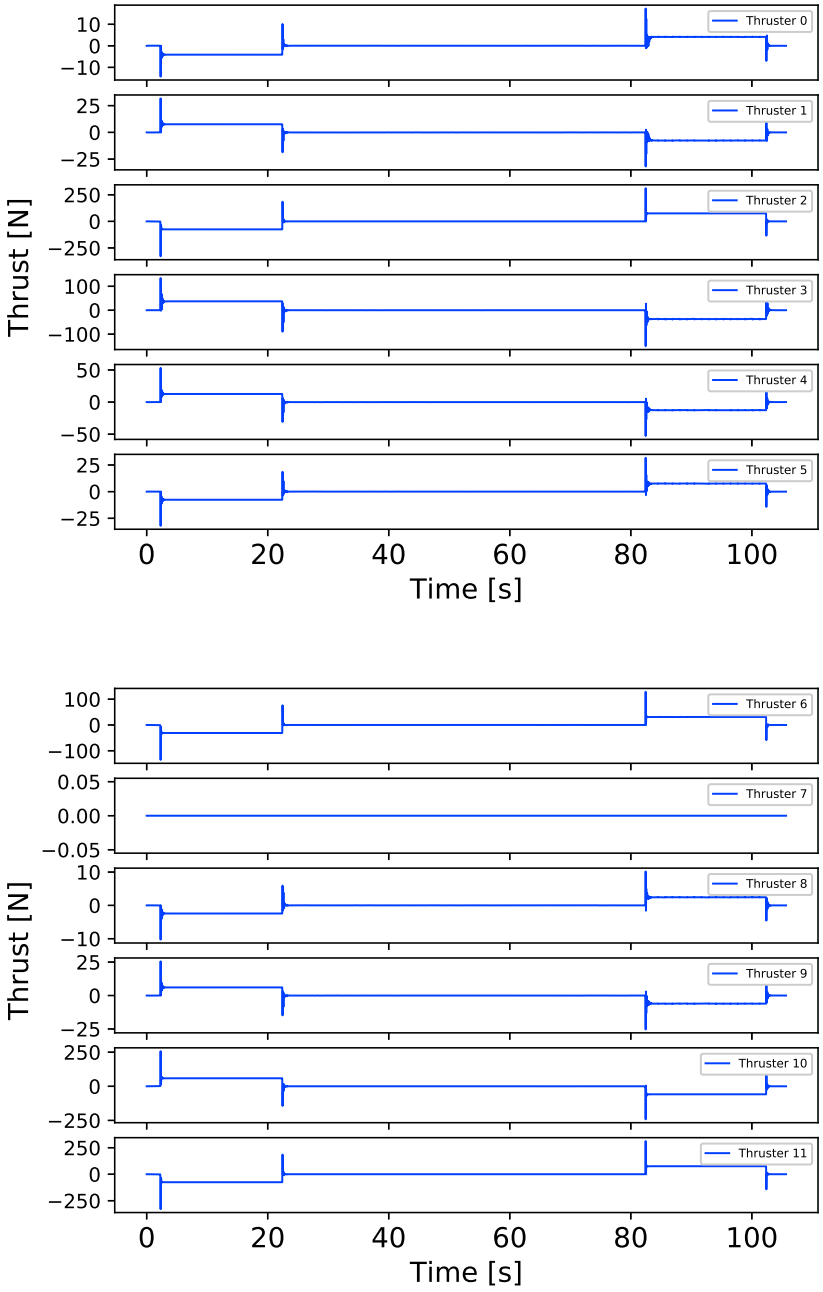


Figure A.16: Thrust u-shape surge

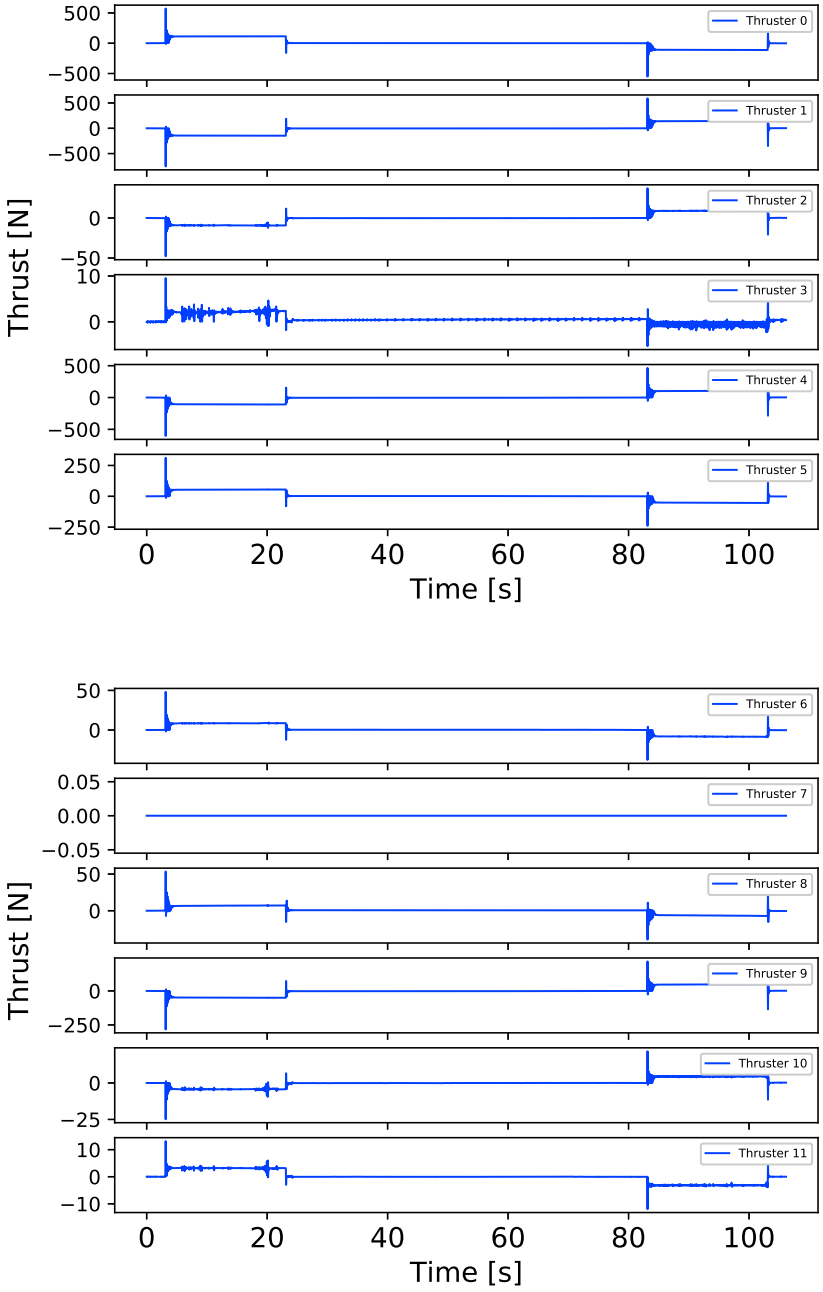


Figure A.17: Thrust u-shape sway

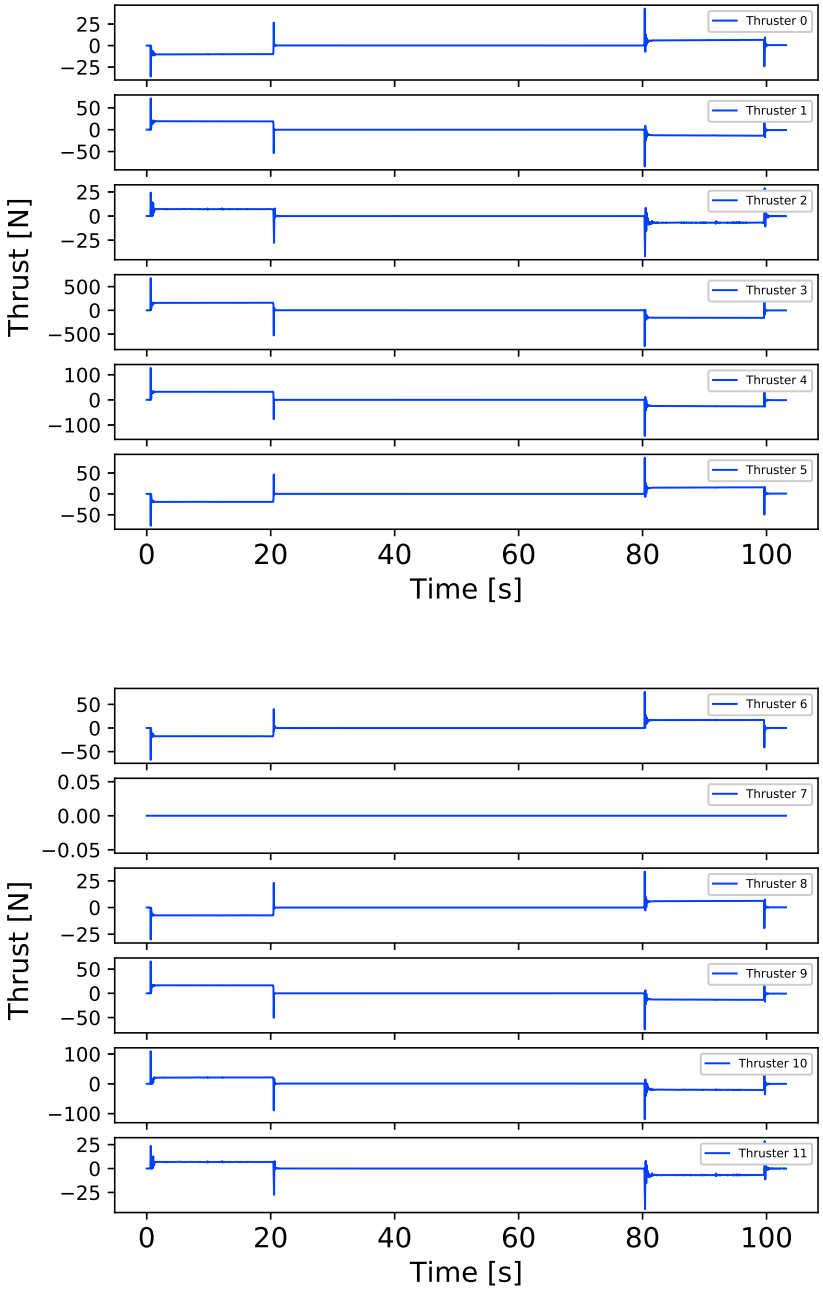


Figure A.18: Thrust u-shape heave

Appendix B

Simulation Parameters

Parameter	Value	Unit	Description
N	5	[-]	Number of links
r	12	[-]	Number of thrusters
r_i	0.1	[m]	Radius of each link
l_1	494	[mm]	Length of link 1
l_2	1185	[mm]	Length of link 2
l_3	1435	[mm]	Length of link 3
l_4	1185	[mm]	Length of link 4
l_5	740	[mm]	Length of link 5
l_j	286	[mm]	Length of joint module
m	199	[kg]	Total dry mass
$r^b_{bb_i}$	(0, 0, 0.025)	[m]	Distance from CO to CB
ρ	1025	$[\frac{kg}{m^3}]$	Density of sea water
C_D	0.3	[-]	Drag coefficient
v_{ref}	1	[-]	Reference velocity
β	0.1	[-]	Linear drag parameter in surge
γ	0.2	[-]	Linear drag parameter in roll
$\frac{\alpha}{\beta}$	0.0003	[-]	Ratio of thruster model parameters
C_t	0.5	[-]	Proportionality constant
γ	0.2	[-]	Linear drag parameter in roll
K_{js}	0.5	[-]	Scaling factor for commanded velocity
$K_{p,linear}$	0.5	[-]	Linear proportional gains
$K_{p,angular}$	0.1	[-]	Angular proportional gains
$K_{i,linear}$	0	[-]	Linear integral gains
$K_{i,angular}$	0	[-]	Angular integral gains
$K_{d,linear}$	0.2	[-]	Linear derivative gains
$K_{d,angular}$	0.05	[-]	Angular derivative gains

Table B.1: Simulation parameters

Bendik Alexander Urne Jørgensen

NTNU
Norwegian University of
Science and Technology