Magnus Renaa Kjørseng

Simulation of a coupled system consisting of a non-buoyant remotely operated vehicle and a surface vessel

Specialization Project Report, IP502122
Supervisor: Øivind Kåre Kjerstad
December 2024

Norwegian University of Science and Technology
Faculty of Engineering
Department of Ocean Operations and Civil Engineering

NTNU

ABSTRACT

A Remotely Operated Vehicle (ROV) can be used to find and grab underwater plastic litter. Lifting heavier litter is difficult however. This project simulates an ROV which is negatively buoyant, that is it sinks in water. The project uses Algoryx' AGX as a framework for simulating a coupled system consisting of a surface vessel and an ROV, connected by a flexible tether.

The results of this project show that AGX is a usable framework for simulating this type of problem. Additionally the project has created a simulation which is usable for further work with this configuration of a non-buoyant ROV. Using the simulation, a simple proportional-derivative controller has been implemented and tested.

The report concludes that the simulator created is useful in further exploration on the topic of non-buoyant ROVs.

CONTENTS

ABBREVIATIONS

Abbreviations used in the report listed in alphabetical order.

- **CFD** Computational Fluid Dynamics
- **CoG** Center of Gravity
- **NTNU** Norwegian University of Science and Technology (Norsk Teknisk og Naturvitenskapelig Universitet)
- **PD** Proportional-Derivative [controller]
- **ROV** Remotely Operated Vehicle

CHAPTER
ONE

INTRODUCTION

1.1 About this report

This report is a summary of work done in the course IP502122 - Specialization Project, undertaken in the autumn of 2024 at NTNU in Ålesund as a part of my master's degree in Product and Systems Engineering. The specialization project is intended as an introduction and "head start" on my master's thesis next semester. As such, the work done here is used in large part as a jumping off point for later.

1.2 Marine plastic pollution

Plastic pollution in the oceans has been widely documented, however the amount of plastic currently in the ocean is uncertain. Jambeck et al.[1] estimates that in 2010, somewhere between 4.8 and 12.7 million metric tons(MT) of plastic ended up in the ocean. According to the World Economic Forum[2], there is between 75 and 199 million MT of plastic waste currently in the ocean. Around two thirds of all plastics that end up in the ocean are heavier than seawater [3] meaning that they sink and either drift in the pelagic zone or end up on the seafloor as litter. Removing litter and plastic pollution on a large scale is difficult, removing it under many meters of ocean makes it much more difficult.

It is undesirable to have plastic waste in the oceans. This is because of the health effects the plastics have on marine and terrestrial life. Two points are especially of note: microplastics and leeching. Microplastics are plastic particles smaller than 5mm. Leeching on the other hand, is the plastics' chemical interaction with the seawater surrounding them, leeching harmful chemicals into the water.[4][5][6][7] For both of these issues, the best solution is to remove the litter. This is because plastics' general longevity. For example, Oluwoye et al. [8] found that polyethylene, commonly used as a coating for subsea structures, would take about 800 years to degrade on the ocean floor. Polyethylene is also used in many consumer- and industrially facing applications, for instance in plastic cannisters for liquids, boxes and crates for fishing or other industrial practices, or as plastic bags.

1.3 The Plan Sea Project

The desire to deal with sub-surface marine plastic waste, i.e. litter both in the pelagic zone and on the seabed, was what sparked the Plan Sea project. Plan Sea is a student driven project at NTNU in Ålesund with the goal of finding, developing and testing a potential solution for removing sub-surface plastics. The project is at time of writing still in its early phases and ongoing. At time of writing, a hull has been constructed from carbon fiber sandwich boards. Thrusters have been mounted to the hull and work on controlling them has started. Additionally, an ROV has been acquired for the project.

Since this is not a report focused on the Plan Sea project, the proposed solution arrived at in Plan Sea will not be discussed in detail. However, because of the relationship between this project and Plan Sea, it is necessary to describe the solution at a surface level.

1.3.1 The proposed solution

The solution which the Plan Sea project is aiming for is an ROV-based solution with an unmanned tender-vessel on the surface. The ROV has a gripper attached and will navigate to find litter, grab it and pick it up. The surface vessel exists to provide the ROV with a greater lifting capacity. If the ROV was to lift purely under the force of its own thrusters, as is traditional for ROVs, the total amount of lifting force available would be limited by the vertical thrust available. This would mean that either the ROV would have to have a very large amount of vertical thrust available relative to its size, or that the total lifting capacity would be very small, neither of which are desirable. By connecting the ROV to the surface vessel with a winch and a lifting cable, it is possible to use the ROV to do fine-navigation to find and attach to litter, and then use the lifting force of a winch and the total buoyancy of the surface vessel for lifting heavier objects. Ideally an ROV which originally only has a total lifting force of about 15kg would be able to lift heavy cables or car tires. Additionally, the solution is supposed to have an undersea basket for collection to avoid having to lower the ROV to the seabed and then lift it up to the surface for each piece of waste. Instead the ROV can be lowered down and can pick litter it finds and place it in the basket. Once the basket is full it can then be lifted up and either emptied on deck, exchanged for another basket or taken back to shore for further sorting there. A sketch of the solution can be seen in fig. 1.1.

Using this solution allows for completely ignoring the buoyancy of the ROV, unlike traditional ROVs. Traditional ROVs are generally designed to be neutrally buoyant, meaning that they neither sink nor float, but keep their vertical position in water once placed there. Since the Plan Sea ROV will be attached to a cable to the surface vessel at all times, it can instead hang from the cable. This means that it's possible to attach larger grippers, more battery capacity, more detection/lighting/navigation equipment, and otherwise allows for any desired modifications to be done to the ROV. Additionally, since the ROV doesn't need to provide vertical thrust, it is much easier to not disturb the seabed which will provide a clearer view for detection equipment based on visible or near-visible light. However, having a non-buoyant ROV does come with some drawbacks.

One drawback of this solution is that it will switch between two operating modes, searching/grabbing and lifting. In the search/grab mode the ROV will be near-neutrally buoyant, or somewhat negative. When lifting the ROV might be severely negatively buoyant. These two wildly different operating modes increase the needed complexity of

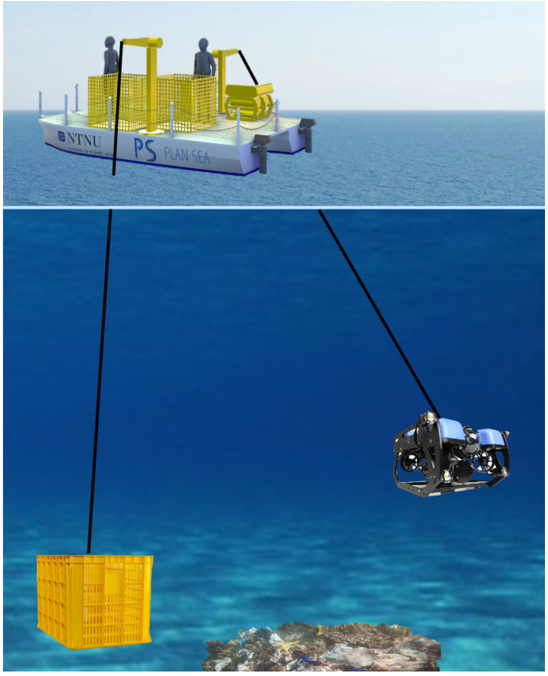


Figure 1.1: A sketch of the proposed solution for Plan Sea showing a surface vessel, a tethered ROV and a collection basket

the system. Another drawback is that as the ROV is hanging from the cable, it creates a coupled system consisting of the surface vessel and the ROV, and necessitates the two moving together as one unit. The forces the surface vessel experiences, such as waves or wind, will impact the ROV, likewise currents or snags the ROV experiences will affect the surface vessel.

1.4 Control systems

A control system commands and regulates the behaviour of other systems automatically. For this project in particular, the control system will be in charge of maintaining and changing positions of the two vessels. A simplified function block diagram of the total system can be seen in fig. 1.2. The goal of the simulator is to function as a drop-in replacement for the vessel, local controllers and environmental impact shown in the figure. This makes it so that the development can happen digitally to then be quickly deployed in the real-world.

1.4.1 Considerations because of a coupled system

In the marine sector, dynamic positioning (DP) is commonly used. DP allows for a vessel to maintain a position or a course automatically despite external effects. This is used for example for offshore supply vessels which need to stay stationary relative to an anchored platform to allow loading and offloading of supplies. DP is also used for applications such as laying subsea fiberoptic cables, where maintaining correct speed and course is important to avoid damaging the cables. For the Plan Sea project too, a DP system is necessary because it consists of vessels that need to maintain specific positions at sea with wind, wave and current forces affecting the vessels

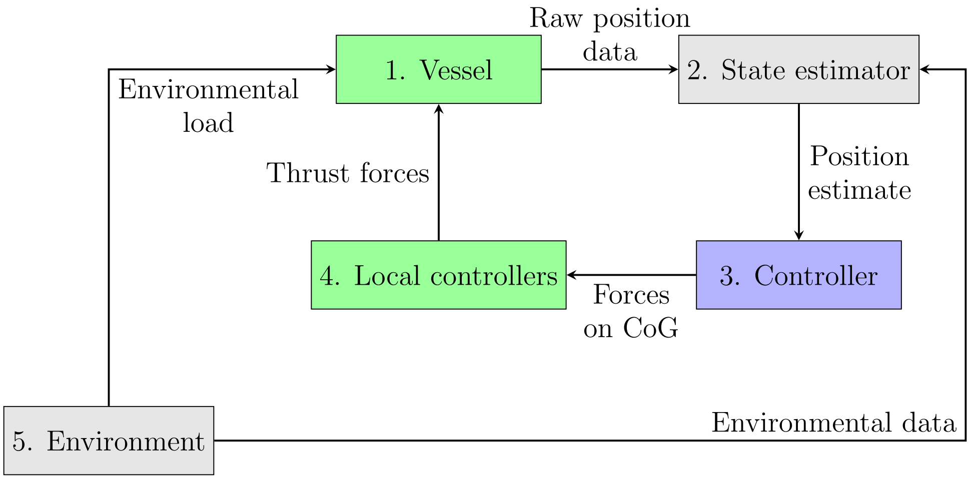


Figure 1.2: A simplified function block diagram of the total system. Grey blocks are not currently implemented, green blocks are simulated and blue is for the controller which is system-agnostic

Normally a DP system only considers one vessel, however for the Plan Sea project it has to be more comprehensive than that because the two vessels are coupled. This is all further discussed in section 2.2

1.4.2 The need for rapid prototyping

Rapid prototyping is becoming increasingly popular with time. The goal of rapid prototyping is to create some simulated environment in which you can test and iterate on a solution until it is acceptable. Then, once you have a solution that works within a certain level of acceptability you can start to put materials and resources into building and implementing the solution in the real world.

For my purposes, rapid prototyping will allow me to experiment with control system tuning and variables without having to deploy the full-scale vessel every time. Ideally, the solution arrived at in the prototyping stage will be directly applicable to the full-scale version, which allows for rapid deployment. The hope is that any issues will be detected and solved while testing digitally, meaning that we hopefully avoid large surprises during deployment.

1.5 Problem description

A simplified sketch from fig. 1.1 can be seen in fig. 1.3. It shows the three main components of this system: The surface vessel, the ROV, and the tether between them. It also shows the forces in the tether. As the tether holds the ROV up, the tether pulls the surface vessel down. This follows from Newton's second law of motion. The figure also shows the coupled nature of the system, since the tether will be taut at all times, both vessels will experience this force from the other at all times.

In fig. 1.4 two scenarios are shown overlaid, one where the ROV is hanging straight below the surface vessel and another where it is at an angle. The result is that the ROV is not at a constant height. If we imagine a desired elevation above the seafloor is constant, then either the ROV needs to have more tether payed out and provide lift

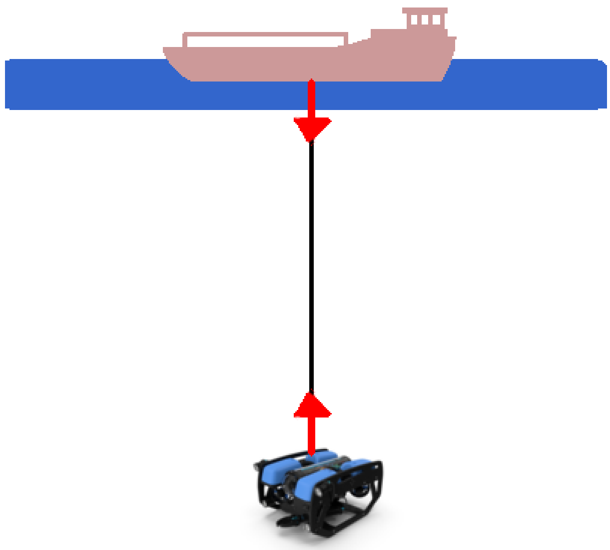


Figure 1.3: A simplified sketch of the problem, forces experienced by each vessel are shown in red.

through its own thrusters, or the surface vessel needs to move to allow the ROV to hang perpendicular to the surface of the sea. If more tether was to be payed out and the system then stabilizes, the ROV will fall to the lowest possible point. It is possible then that the ROV might collide with the seabed or other objects. The ideal solution then becomes that the surface vessel follows the ROV, or the ROV only operates within a given area of operations directly underneath of the surface vessel.

Further, this non-perpendicular arrangement will lead to tangential forces, shown in fig. 1.5. The forces will of course be equal and opposite on the vessel's end, though this has been omitted from the figure for clarity. The horizontal force the tether imparts on the ROV will act as a restoring force, trying to move it back to be perpendicular to the surface vessel. The surface vessel likewise will experience a pull towards the ROV.

Because these two vessels are connected, and therefore dependent on each other's positions, the control system needs to take this into account. Probably, the simplest solution will be to have one large control system handling both, or alternately having one of the vessels take a leading part and the other attempt to follow. This will be touched on later in this report, though not discussed in detail.

1.6 Statement of intent

For this project I want to create a simulation which is able to quantify the effects that the parts of the system have on each other. I want to be able to measure the tension in the wire, the force exerted by the vessels, how well the vessels follow each other or orders given and influence from the environment. The goal of this simulation is to be used as a future design tool for finalizing design of the Plan Sea vessel, its control systems, as well as defining operational criteria.

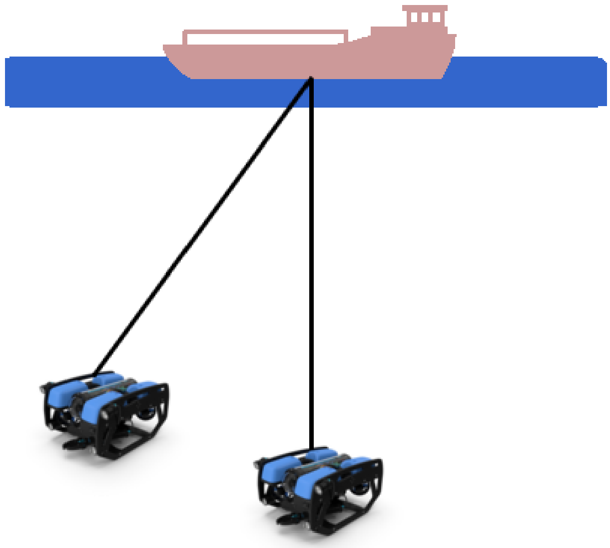


Figure 1.4: The ROV is hanging from the same point on the surface vessel with an equal tether. Note how the height of the ROV has changed because the tether has stayed the same

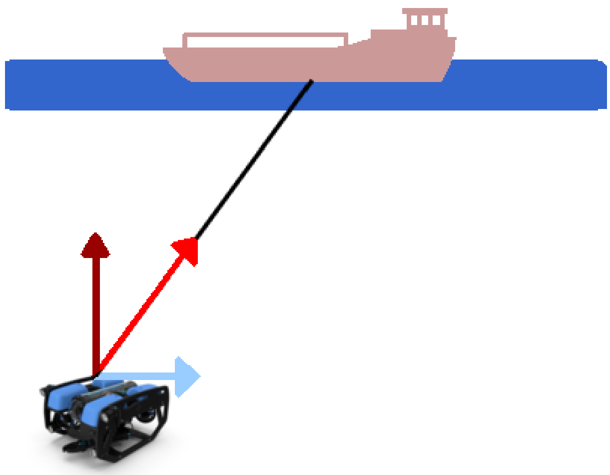


Figure 1.5: The component forces on the ROV resulting from an angled lift, decomposed

CHAPTER

TWO

MODELLING AND CONTROL DESIGN

2.1 State of the art

There is little literature specifically on the topic of coupled powered systems with elements both above and below the surface of the sea. I have however found bits and pieces that are useful. Primarily the offshore oil and gas industry, as well as general ROV operations are able to help a bit. The main issue with this project is that it is fairly unique. Usually when dealing with ROV operations, the surface vessel is intended to stay stationary for longer periods of time, or the ROV is neutrally buoyant and able to do its work below regardless of what the surface vessel does on the surface. This project deals with a powered non-buoyant ROV which needs to be able to do both large motions to sweep larger areas of seabed, as well as precision work to move in on and pick up litter. This is a unique problem from the literature I have found.

There are several papers discussing for instance the effect of currents on deep-submergence suspended tooling or ROVs used for oil and gas installations that can help. For example, Lian and Sortland 1996 [9]. They also performed simulations, however their paper is focused on non-powered remotely operated tooling as opposed to a remotely operated vehicle. Their results can be useful as a kind of sanity check for the results of the simulations of this project, though their working area is down to 1500m below the surface while it's improbable that the Plan Sea project will ever operate below 200m.

The tether is also a consideration for this project, Chen et al., 2021 [10] discuss the hydrodynamic effects on ROV tethers under complex sea conditions. Whether or not this is useful for this implementation is uncertain, as the tether simulations are all done by the simulation software. Additionally the tethers used for this iteration are likely so small in diameter that their effects in the sea are likely small. It might be possible to expand the tether simulations to include Chen et al.'s findings in future.

Enevoldsen et al., 2018 [11] provide one of the more useful documents on the topic. They discuss a simplified modelling strategy for ROVs to allow for both greater control capabilities and simulated efforts. The state of this project has not used this information, but it will be very helpful in later iterations to expand the accuracy of the simulation for the ROV, both for control purposes and for a more accurate simulation. For instance a model of the vessel can be implemented into the controller so that it's able to act predictively and not just reactively like a PID controller would. Anderlini, Parker and Thomas, 2018 [12] discuss control of an ROV carrying an object. They discuss the sudden added mass of the vessel and how to compensate for it from a control perspective, though their paper focuses more on autonomous underwater maintenance vehicles. This paper as well will be very helpful for further implementation work, but has not been used in any great extent here. The Anderlini paper also primarily focuses on self-propelled underwater vehicles, not the externally lifted version that this project considers. Thingstad and Hveding, 1982 [13] have a conference paper on non-buoyant ROVs for performing subsea work. On the surface this sounds perfect, but looking into the paper it is more focused on the physical construction of the ROV rather than the control of it. This makes it less helpful for me. Additionally, Thingstad and Hveding's paper is more than 40 years old at time of writing, and applications of control theory, as well as microelectronics and actuators have evolved a lot since then, making what little control they do discuss less useful.

2.2 Mathematical basis

The root problem can be decomposed into equations 2.1 and 2.2. They show the forces that impact the momentums of both the surface vessel and the ROV. The forces that have an impact are hydrodynamic forces, such as buoyancy, righting moment etc. The propulsive forces that the vessels' thrusters provide. Environmental forces coming from waves, winds and currents. And finally the force the coupling acts with on each vessel. Do note that though the form of the equations is the same for the surface vessel and the ROV, the values both in total and in each individual element are not necessarily equal. Hydrodynamic, propulsive and environmental forces will be entirely individual for each element because of their physical shape and capabilities, and while the coupling forces are linked, their relationship is not necessarily linear. See section 2.2.1 for elaboration.

$$M_{v_{\mathrm{surf}}} = \sum f = f_{hydro} + f_{prop} + f_{env} + f_{coupling} \tag{2.1}$$

$$M_{v_{\mathrm{ROV}}} = \sum g = g_{hydro} + g_{prop} + g_{env} + g_{coupling} \tag{2.2}$$

The equations demonstrate the need for simulation compared to analytical examination of the problem. Hydrodynamic forces and environmental forces contribute to a highly dynamic system for which it is difficult or impossible to find a closed form expression. This means that calculating an expected state for the dynamic system manually is labour and time intensive. Using a simulation instead of analytical methods hides these problems away. The simulation will take care of the complex interactions which allows me to focus on extracting interesting data. I will further discuss simulation options in chapter 3

Using a simulation does introduce a new requirement: that of validating the models' accuracy. There are many ways of doing this, both intuitively and mathematically. I will get talk more about the validation methods I've chosen in section 4.1.

The simulation allowing for "setting and forgetting" whatever parts of the force equations are desired allows the user to focus on whatever specific field they are interested in. For example a user might examine the propulsive force required given a certain seastate, or how hull shape and hydrodynamics affects the stability of the total system. Using simulation gives the user a greater degree of freedom in finding exactly the variables they're interested in.

Figure 2.1: A chain forming a catenary between two posts under the force of gravity. Used under Creative Commons, credit: https://en.wikipedia.org/wiki/File:Kette_Kettenkurve_Catenary_2008_PD.JPG

2.2.1 Catenaries

When a rope or chain is suspended from two points and affected by forces not in-line with the two points, the chain forms a catenary. Catenaries are relevant to this project because the lifting tether will form a catenary whenever the ROV is not directly underneath the surface vessel and also not experiencing any external forces. If there is a current or movement affecting the tether it will form a catenary.

The fact that the lifting tether will form a catenary is helpful to know, as one might intuitively assume that the lifting tether would be straight between the surface vessel and the ROV. If the tether were smooth, finding the position of the ROV could be done by trigonometry, knowing the length of the tether, its angle at the lifting point relative to gravity, and knowing the tether's azimuth, the ROV could be located using Pythagoras' theorem. However, since this simplification can't necessarily be done, a slight complication has to be added. The angle at the lifting point is not necessarily directly linked to the angle of a straight line to the ROV. This can be seen demonstrated in fig. 2.2

In 2 dimensions, a catenary is well defined as

$$y = a \cosh\left(\frac{x}{a}\right)$$

Where a defines the width of the catenary and can be found in relation to the relevant forces acting on the rope. For this project's applications however, the shape of the tether can't necessarily be simplified to 2D. I have been unable to find a simple, closed form of the catenary equation in 3D for a rope (as opposed to a plane), though it may well exist. This further solidifies the necessity of a simulation over using analytical methods. Additionally, the simulator has shown that the tether does not necessarily form a catenary. This is also shown in fig. 2.2 where the shape of the tether is more bowed near the surface than near the bottom.

This simulator can be used to find an error between assuming a straight line between the ROV and the surface vessel, compared to the real situation. This relationship or

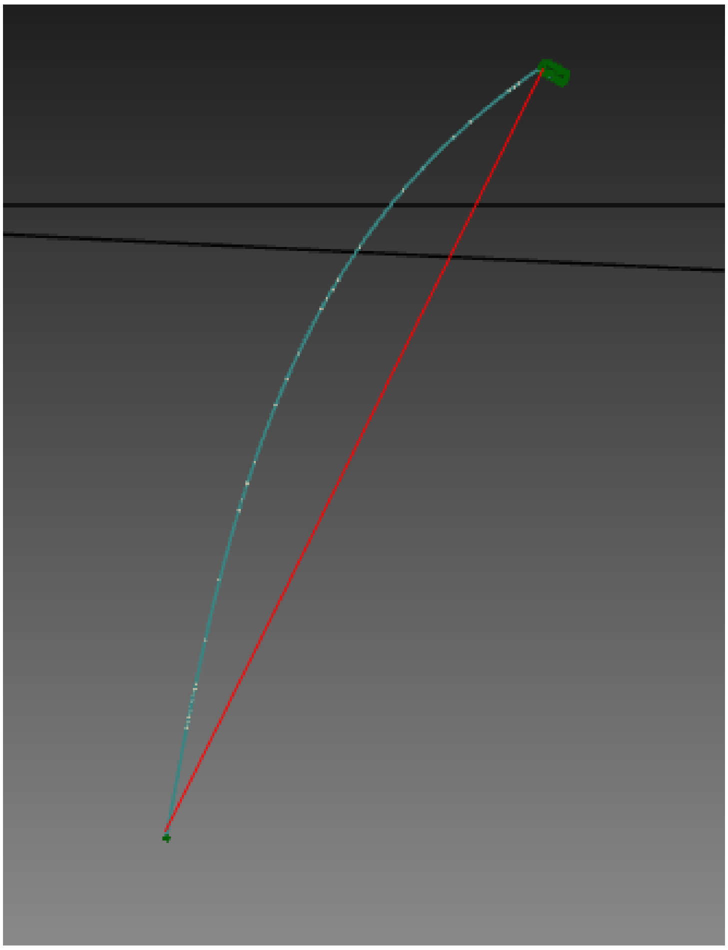


Figure 2.2: The simulation showing the towing line (light blue) and a straight line between the surface vessel and the ROV (red). Note how the towing line is not straight

error can be included as a part of the internal model of the vessel's state estimator and be used as a part of positioning the ROV under water.

CHAPTER
THREE

SIMULATION FRAMEWORK

The goal of this specialization project is to build a simulator which is applicable to the case of the Plan Sea project. There are four elements that need to be simulated for this to be considered a success with a fifth "nice-to-have" element. The elements are:

- Surface vessel
- Subsurface vessel
- Connecting wire
- Control system
- Weather impacts

There are two main paths to take with regards to making the simulation. I could make everything from first principles or I could use an already existing simulation framework and build something on top of that. Both solutions have positives and negatives.

Creating a system from scratch would be an interesting challenge. It would also give me exactly the results I'd want with very little overhead, assuming that my own programming skills and understanding of the relevant physics are up to the task. The bespoke, home-made simulator also wouldn't have any associated licensing costs. On the other hand, to create a simulator which includes buoyancy, fluid dynamics, wire physics, handles a coupled system and also has some form of graphical interface/readout for the user is a large task to undertake, especially in the roughly 4 months available to this project.

Using a commercially available system also has a fair few benefits. I would have a mostly ready-made framework within which I can just configure the simulation as I need it. Changing out parameters and variables would also be very simple, as it's essentially the same operation as configuring the simulation from the beginning. I believe also that the results from a commercial solution would be more reliable than my own attempt. It is reasonable to assume that an industrially used, commercially available simulation solution made by a team of physicists, computer engineers and other specialists is going to be more accurate with regards to its results than me. Commercially available solutions are not all perfect however. There are licensing costs associated with many simulation frameworks. Some having enormous costs for the scale of a student project. There is also likely more computational overhead with a commercially available solution due to them being as wide in their application as possible. This can make the commercially available solution slower or less responsive in use.

After considering the points above for both commercially available and personally crafted simulation frameworks, I decided that using a ready-made solution would be better. This was especially decided because of the time constraints of this project. My goal is to have a working simulation that can be used to provide information, the goal is not to make a simulator. If I was to make it myself, the project would quickly turn into "make a simulator" rather than "make a simulation", simply because of the scale of the undertaking.

3.1 AGX

After considering multiple simulation options and on the advice of a professor, I landed on using Algoryx' AGX as a simulation framework. AGX has a solid wire simulation package, a hydrodynamic simulation package, and allows for scripting and setup using both Python and C++. It has interfaces towards both Unreal Engine and the Unity engine for further graphical display of the results or graphical user interfaces. AGX allows for multi-body simulations, and in addition it has interfaces for ROS2 which I will get into in section 3.3.

My implementation of AGX is based on Python as that's the language I'm most familiar with. I am aware of Python's inferiority to a C++ based approach in terms of speed and efficiency with regards to computational resources. The time investment required for me to get to an acceptable level of C++ proficiency was not worth it for this project, and so I stick with Python.

3.1.1 Limitations of AGX

By using a ready-made simulation platform I am able to quickly implement a simulation without needing to worry about the mathematical models that exist behind the simulation. This allows me to focus on achieving results, although it also necessitates a degree of trust in the simulation. I am able to verify whether the simulator acts as expected or not, but changing the governing equations is not necessarily something I am able to do.

Another limitation of AGX is that it's a licensed software. This means that in order to apply and use the findings of this report, the software needs to be acquired. This is a limitation for further research, and ideally the findings should be based on open and available software or arrived at from first principles.

Algoryx lists some known limitations of the relevant packages in their user manual[14] pp. 95 and 169, as well. For the hydrodynamic module for example, wakes and object-generated waves are not possible, semi-submerged convex objects will also not be water-tight. There is also the possibility of unwanted behaviour when using multiple linked models. I believe this is in reference to multiple rigidly linked models, not loosely linked ones like I'm using. A limitation of the wire module is that wires can't handle torsion forces. This is likely unnecessary to model for the lifting tether, as it will be sensitive to torsion like some cables or wires might be.

3.2 Description of the simulation setup

The simulation starts with defining a water volume. This is done using AGX built-in functions. Currently the volume is 200m along the sides and 100m deep, though this is arbitrary. I've implemented one "water controller" which is the object that handles wave, current and wind forces. Currently this controller is completely still and there is no force inputs, but it is implemented so that adding weather forces is simple for later iterations. In AGX, each element in a simulation has to be individually added to that simulation. Once both the water volume and the water controller have been added to the active simulation, the vessels are made.

The surface vessel and the ROV are both implemented as children of a general vessel parent class. I've done this for later ease of expansion. The ROV is just a simple box with a given density and size. The ROV's size I've taken from BlueROV's websites[15], as an example of the size of ROV we will be working with in this project, though as mentioned it's just implemented as a box. The University does currently have a BlueROV that is intended to be used for the initial steps of the Plan Sea project, as well as for my master's thesis which will be based on this work, that is why I've chosen to use their dimensions. The exact dimensions are $0.45m \times 0.575m \times 0.254m$ for a total volume of $0.0654m^3$. If a more accurate shape for the ROV is desired, it is possible to implement it similarly to the surface vessel, but since this project is mostly a first-order approximation of the problem, I believe a simple box is sufficient. The ROV's density I have wildly exaggerated and arbitrarily chosen to be $2000\frac{kg}{m^3}$. This is an absurd exaggeration and should be replaced with more accurate data later. As a proof of concept it will work fine. This gives a total mass of the ROV of 131kg.

The surface vessel's shape is defined by a wireframe stored in an `.obj` file I've made that approximates the shape of the hull, while its density I've approximated from the density of the carbon-fiber sandwich board used to $600\frac{kg}{m^3}$. The simulation software has provided a mass calculation for the vessel to approximately 2300kg. This gives the ratio of mass between the two vessels of roughly 5%, which is far more than an assumed real-world estimation of less than 1%. This will skew the results to exaggerate the impacts of the ROV's motion.

In addition to a class which handles the shape and properties of the surface vessel, I have made a class that handles controlling the vessels. The controller is just a simple implementation of a PD controller. The controller checks the position of the vessel on each timestep and calculates an error. The error is then controlled using the PD controller and a response is found. The response is clamped within a given authority limit that simulates the vessel's control authority limited by the physical capabilities of its thrusters. I've estimated the authority for right now, but it should be changed later when the real command authority is found.

Currently, the way the controller is acting on the vessel is by simply applying a force in a given direction on the CoG of the vessel. For the surface vessel, any force and error given in Z-direction is zeroed as the surface vessel will not be able to cancel its own heave in waves. This can be changed later and the Z-direction error/control can be implemented as heave-compensation for the ROV. This isn't implemented yet though. I've also implemented a simple proportional controller for heading control which works by adding a torque to the body it's controlling. In a future iteration, both the simplistic control and heading control could be implemented by splitting the force inputs from one single input at CoG to one input at the position of each thruster. Then yawing

Figure 3.1: Illustration image of the BlueROV2 used as a basis for the ROV in this project. Credit: BlueRobotics

motion can be applied with differential thrust and a heading-term can be added to the controller. Whether the complications of simulating force inputs at the thruster locations are necessary is not clear yet. If we consider the ideal future implementation in a physical vessel, the controller designed now is only intended to give the commands to the vessel as force acting on CoG. Thrust allocation and local control will be designed elsewhere and simply be a node in the ROS2 system that will be the final vessel. Either way, the option of making the simulation more true to life exists if it should be desirable.

Finally, the wire which connects the ROV and the surface vessel is created. AGX has two similar wire-like simulation objects: Wires and Cables. The documentation gives a comparison of wires and cables. Because torsion of the wire is irrelevant, we want to be able to winch the wire in and out and we are working on long wires, the Wire module should be used. The way wires work in AGX is as a series of links connected to or passed through a series of nodes. For this implementation only two nodes are necessary, one connecting the wire to the surface vessel and one connecting it to the ROV. More nodes could be used for example if the wire was to be pulled around a corner, or through certain points for routing. The Wire module of AGX allows for winches to be simulated as well, with given speeds, gearings, torques etc.. I have not implemented this in the current simulator.

In addition to all the "necessary" elements, the simulation also has a manual controller for the vessels. It is possible to use the keyboard to give manual force inputs on the vessels. This uses the same framework for adding force as the controller does. I've done this as a way to debug and test the system a bit.

The final simulator can be seen on Github[16]. The URL is `https://github.com/MagnusKjorseng/FordypOgMaster/tree/main/Simulator/V3`

3.3 ROS2

ROS2, short for Robot Operating System 2, is an easily expandable and configurable operating system used primarily for hobbyists and research in control and robotics. The main selling point of ROS2 is its node model where different parts of a control system can be placed in separate, segregated nodes with certain interfaces. Those nodes then either publish data to or subscribe to data from what ROS2 calls topics. This system grants a developer or team of developers flexibility with regards to changing out certain nodes while still allowing the larger system to work. I.e. experimenting and changing out single nodes, so long as the same topics are still used, is extremely simple.

I was not able to implement ROS2 during this project. I believe this is because ROS2 is not designed to be used on Windows and requires a lot of workarounds which I was not able to figure out. This section is in the report more as a reference and a reminder for future iterations based on this report that ROS2 would make the project easily scalable and iterable and should be implemented later.

AGX has the ability to work with ROS2, having built in methods for both publishing and subscribing to topics. Applied to this project it would in theory allow for the control system to be built in ROS2 and connected to the simulator. The control system could then be tested and tuned in different simulated environments until satisfactory results are achieved. When the control system works as intended, it can then be disconnected from the simulator and connected to a physical system and be tested in the real world.

The end result of the process above would be two equivalent systems, one digital and one physical, which would allow for rapid prototyping in the digital space before quick deployment in physical space.

CHAPTER
FOUR

RESULTS

I have implemented a simulated surface vessel and a non-buoyant subsurface vessel, both of which are controllable either manually or by automated controller. The two vessels are bound together by a wire tether. The vessels and the wire are all affected by weather effects such as ocean currents, as well as physical effects like added mass and inertia.

The simulation works as would be intuitively expected. An example of the simulation graphical interface can be seen in fig. 4.1. The figure shows a surface vessel and the ROV under water in green, and the teal tether connecting them. The simulated water is shown as a grey volume, but it's not easy to distinguish it in fig. 4.1 because it takes up the entire screen.

4.1 Model Validation

There are two ways I will attempt to validate the simulator results here. One is intuitively and the other is using analytical methods and a simple case. The intuitive demonstration is difficult to convey in text-form in a report, but I have placed some animations of the results on GitHub[16] alongside the simulator. The intuitive demonstrations consist of starting the simulation, pulling objects around and seeing if they "act right". Human minds are excellent pattern recognition machines, and I will use this to my advantage

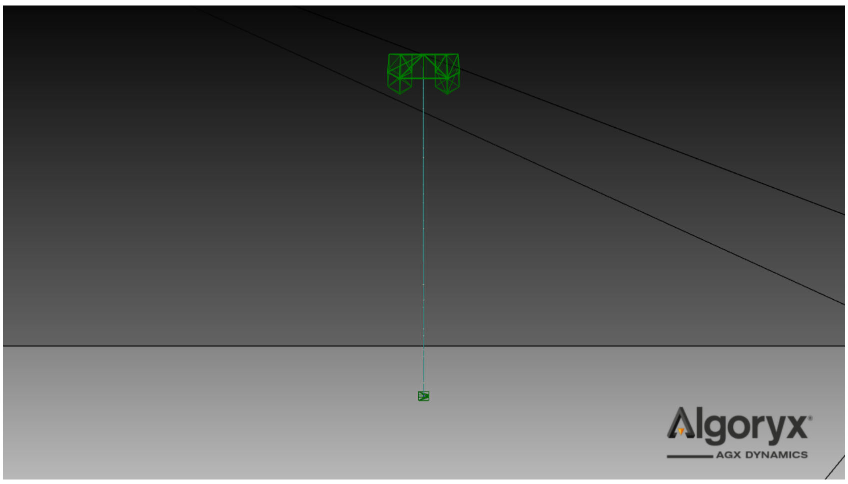


Figure 4.1: The graphical user interface of AGX with the simulation running

here, as if something is "off", it would be noticable.

For the analytical validation I will use analytical methods and simple physics to estimate an expected result, and then simulate the result to compare the values for both.

4.1.1 Intuitive validation

I have done two tests here, one where I pull the surface vessel and look at how the ROV follows, and the other where I pull the ROV and see how the surface vessel follows. The shape of the tether is also relevant here. Looking at the results, shown in the `Results` directory of the simulator on the Github repo, there is nothing that pops out as obviously wrong. Of note is that the surface vessel flips in the ROV pulled test. This is because I pulled the ROV down and given the situation the surface vessel was in it was more stable flipped upside down. This is not a realistic scenario for the real-life project because the amount of force exerted pulling the surface vessel down will never be that large. Looking at the tension in the tether at the ROV it reaches upwards of 20kN at points, which corresponds to roughly 20 tonnes lifted.

In addition to the two tests above I've also tried swinging the ROV as a pendulum underneath the surface vessel. This test also confirms the intuitive assumption that when the surface vessel is far more massive than the ROV, it will be less affected when pulled like this, although it will still be affected.

I will note that in the two first animations the ROV is shown as larger than in the last. This was due to an implementation error on my part. The result of this is that the ROV is far more massive than it's supposed to be, and also larger which gives it greater drag in the water. Neither of these affect the intuitive approach as the physics still "act right".

4.1.2 Analytical validation

For the analytical part of the validation I will use a simple case in which the surface vessel moves forward, towing the ROV behind it. This will be roughly analogous to how the ROV and tether will respond in currents, though not entirely interchangable. The current speed will impart parallel forces on the tether and ROV while the towing case will impart a tension force on the tether. Still, this is a simple case for rough validation of the model and can be further worked or reworked for later use.

It is possible to analytically find an expected tension in the tether and then compare this with the simulated results. I will ignore the effects of the tether in the analytical calculations for simplicity. This will be a source of error on the final result as the tether will have an effect in the simulation.

The tension on the tether in the towing case will be dependent on the resistance of the water around the tether and ROV as well as the effect of gravity. This gives the equation

$$F = F_g - F_b + F_D$$

Where F is the total force pulling on the wire, F_g is the force due to gravity, F_b is the buoyant force and F_D is the drag force.

For the force of gravity I will assume that gravity is in-line with the tether. This is not the case in reality as the ROV will lag behind a bit as can be seen in fig. 5.1. This causes effectively a cosine error. This angle can be quite large, however since the only thing keeping the ROV from sinking is the tether, the gravitational force is still transmitted through the tether. The force of gravity becomes

$$F_g = mg \approx 131kg \times 9.8\frac{m}{s^2} \approx 1300N$$

Buoyant force is given by the volume the ROV displaces and is given by

$$F_b = \rho V$$

Where ρ is density of the fluid, V is volume displaced and g is the acceleration due to gravity. The seawater used in the simulation is defined with $\rho = 1025\frac{kg}{m^3}$ and the volume of the ROV is previously found to be $V = 0.0654m^3$. Using these values, we can find that the buoyant force is

$$F_b = \rho V g = 1025\frac{kg}{m^3} \times 0.0654m^3 \times 9.8\frac{m}{s^2} = 657N$$

When it comes to drag force, the ROV will be the largest influence due to its large size compared to the tether. The resistance for an object in a fluid (drag) is given by the equation

$$F_D = \frac{1}{2}\rho v^2 C_D A$$

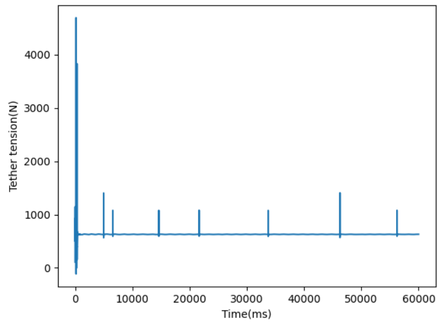
Where v is velocity, C_D is the coefficient of drag and A is the cross-sectional area. For the cross-sectional area I will assume that the ROV doesn't rotate around any axis as it is being dragged. In reality it definitely will, which will change both A and C_D. Including the effect of these rotations will be very complex, and so I will ignore them. I will assume that the cross-sectional area is the front facing area, defined as

$$A = 0.45m \times 0.254m = 0.114m^3$$

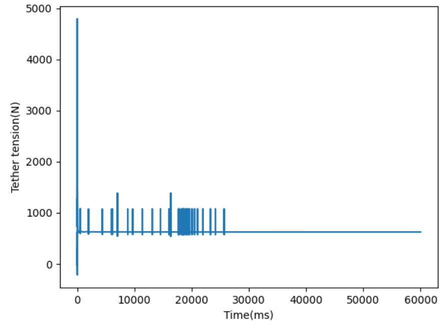
Coefficient of drag is a value found experimentally and normally referenced in tables. The drag coefficient of a cube is according to tables 1.05, while the drag coefficient of a square prism perpendicular to flow is 2.05. The ROV is simulated as a simple box which is roughly half of a cube, divided horizontally. I believe the actual coefficient of drag on the simulated ROV will be somewhere between these and I will use both values to calculate an upper and lower bounds. For best results, either CFD analysis or physical experiments on an equivalent shape could be done.

The dragging of the ROV behind will cause it to no longer be directly beneath the surface vessel. This means that the force of gravity and the force of drag will not be acting in-line with the tether. This is no matter, as the total force will still have to be carried by the tether, so off-axis forces will not impact this. The ROV lagging behind will however lead to it being easier to tumble, changing its forward facing area which would affect the drag calculations.

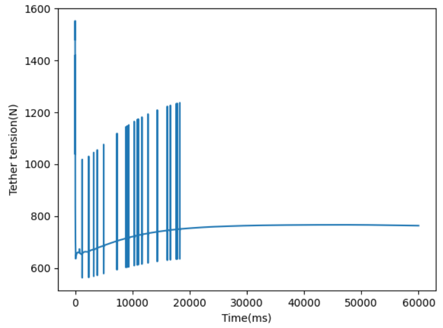
That only leaves the velocity as a variable. I will do simulations in increments of 1m/s from 0 to 5m/s, roughly 10 knots. The simulated tensions will be the average of tensions over a 60s period. This period is chosen so that the system is allowed to stabilize. The calculated and simulated tensions can be seen in table 4.1. Graphs of tensions can be seen in fig. 4.2.



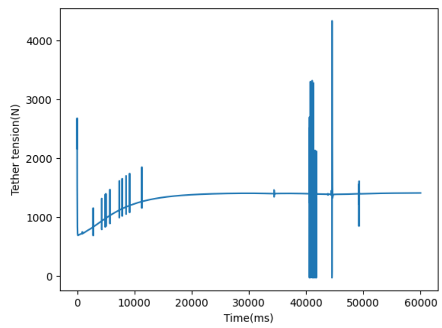
(a) $v = 0$



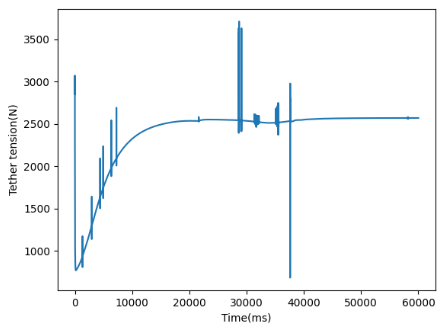
(b) $v = 1\frac{m}{s}$



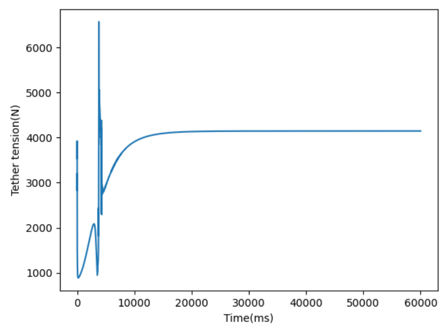
(c) $v = 2\frac{m}{s}$



(d) $v = 3\frac{m}{s}$



(e) $v = 4\frac{m}{s}$



(f) $v = 5\frac{m}{s}$

Figure 4.2: Graphs of tension in the towing tether at different towing speeds. The graphs are spiky at points, I believe this is because of singularities in the simulation or other instabilities though I don't know for sure

Velocity (m/s)	Calculated drag (N) $C_D = 1.05$	$C_D = 2.05$	Calculated tension(N) $C_D = 1.05$	$C_D = 2.05$	Simulated tension (N)
0	0	0	643	643	628
1	61	119	704	763	626
2	245	479	888	1122	747
3	552	1077	1195	1721	1318
4	981	1916	1625	2559	2388
5	1533	2994	2177	3627	3901

Table 4.1: Calculated drag and tensions in the tether between the surface vessel and the hanging ROV. Results are from both calculation and simulation. Simulated tensions are an average over a 60s period.

Velocity	Deviation($C_D = 1.05$)	Deviation($C_D = 2.05$)
0	-2.4%	-2.4%
1	-12.5%	-21.9%
2	-18.9%	-50%
3	10.7%	-30%
4	32.0%	-7.2%
5	55.8%	7.1%

Table 4.2: Discrepancies between the calculated tension and the simulated tension from table 4.1. Negative numbers indicate that simulated tension is lower than calculated tension

4.2 Control system results

The control system has been implemented as a simple PD controller. The weights for the controller are set to

$$k_p = m_{\text{surf}}\omega^2$$
$$k_d = 2m_{\text{surf}}\zeta\omega$$

Where m_{surf} is the mass of the surface vessel, ω is a damping frequency and ζ is the damping ratio. Through experimentation, I found that $\omega = 0.45$ and $\zeta = 1.2$ gave good results for this vessel.

The way the controller is implemented it is able to take in a list of several targets and will treat them as individual targets sequentially. Once the vessel has reached one within a given acceptable error (currently set to 0.1m), the next target in the list is selected and the vessel moves towards it. The target selection finds not only if the current step has achieved the goal but also checks the errors for the last 50 timesteps. This is done so that the vessel couldn't hypothetically run straight through the waypoints, and instead has to actually come to a stop (or close to it) at the target positions. The number 50 was chosen arbitrarily. While there is a simple heading controller implemented, heading error is not taken into account with whether the vessel has reached the target or not.

One simulation with the control system was done with 4 waypoints. The vessel moved from the starting point at (0,0) to (10,15) to (50, -30) to (-20,10) and back to (0,0). These points were chosen arbitrarily and were chosen because they are a fair distance away from each other. In total, the theoretical shortest distance to travel would be roughly 180m.

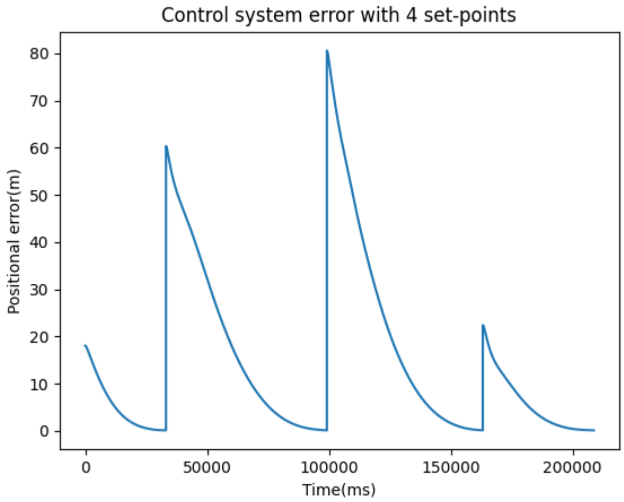


Figure 4.3: Total error for the vessel between the four waypoints

The time taken for the simulated vessel to move was 208s, or approximately 3.5 minutes. The total error can be seen in fig. 4.3

CHAPTER

FIVE

DISCUSSION

5.1 Validation of the simulation

In section 4.1 I presented two methods of validating the simulator. The intuitive and analytical methods. The intuitive validation I have already explained there. The results of the intuitive validation I did I believe show that the simulator at least "acts right". Any flaws that it may have are not so obvious that they are visible without analysis. I will here try to do some further analysis here below of the tension data.

5.1.1 Tension data

Looking at the results from table 4.1 and fig. 4.2, I believe the simulator can be called valid for this use-case. In all cases, except for the fastest one, the simulated tension falls between or below the assumed drag values. The deviations from the calculated values are shown in table 4.2. The cause of the deviations are likely because of the way I've done the calculations; ignoring tumbling and using a "best guess" value for the drag coefficient. Looking at the actual simulation for the highest speed I can confirm that my assumptions will be completely off, as the ROV in that case is practically surfaced as it is dragged behind the surface vessel, see fig. 5.1. The cross-sectional area used in the calculations will be far off from the real values here, as it's almost the "top" surface of the ROV that's facing the direction of travel. Another point to note is that at these high speeds the surface vessel tips so far forward it gets caught up by the hydrodynamic forces at the water's surface and flips over. I believe this happens because the tether is not perfectly centred on the body of the surface vessel, but it is also somewhat a bad sign for such high speeds with this vessel. It indicates that stability analysis at different speeds should be performed. When it comes to the vessel's tipping at higher speeds and validity of the results: the velocity with which the tether is dragged is set and independent of the surface vessel. The vessel is being given a speed with infinite acceleration and is simply along for the ride.

What this dragging simulation tells us is that at this depth, the ROV is not usable. This can be a useful tool to gauge what speeds the surface vessel will be able to move without adversely affecting the ROV underwater. For example, in fig. 2.2 the surface vessel is also moving at $v = 5\frac{m}{s}$, but the ROV is not as affected because the tether damps more when it is longer. The results also show that the analytical method provides an

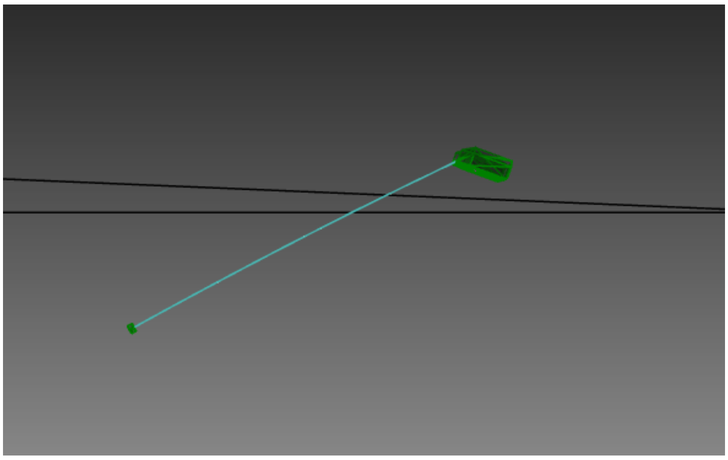


Figure 5.1: The ROV shown dragged behind the surface vessel at $v = 5\frac{m}{s}$

overestimate for the force that the tether will be required to endure under low-speed conditions. This can be used as a tool to find suitable tether materials.

Further, looking at the two first simulation cases they show a very close relationship with the analytical method. At $v = 0$, the simulator has 2.4% less tension than the analytical method expects. This can be because the tether has an impact. If the tether is slightly non-buoyant, the ROV would experience a smaller tension than the surface vessel. This is because the surface vessel would carry the weight of both the tether and the ROV, while the ROV will not "notice" the tether in its connection.

Looking at $1\frac{m}{s} \leq v \leq 4\frac{m}{s}$ the tension is lower than calculated. Looking at the graphical output of the simulation, the suspicion mentioned in section 4.1.2 is confirmed and the ROV is indeed tumbling a bit, presenting a sharp angle towards the direction of travel instead of the blunt front. This would lower the coefficient of drag significantly. For instance, an angled cube, presenting a corner towards the stream rather than a face, has $C_D = 0.80$ compared to a face-on cube with $C_D = 1.05$.

All in all, I believe the differences in results between the analytical method and the simulated method can be seen as a combination of tumbling and angling. By angling I mean presenting a sharper angle towards the stream, and by tumbling I mean presenting the larger top face instead of the smaller front face. Angling would decrease the coefficient of drag while tumbling would increase the forward facing area. I believe these two effects explain the discrepancies seen. Further, I believe that the simulator provides more accurate data than the analytical method could as I have used it. I believe this because the simulator is able to account for angling, tumbling and other effects which are much harder to implement in the closed-form equations I've used.

5.2 Use as a rapid prototyping tool

The simulator shows a lot of promise as a rapid prototyping tool. As it stands at time of writing, it is fairly easy to configure. Changing for example initial conditions, depths of water/length of cable or the type and tuning of controller is very simple. Configuring the seastate is somewhat more complex as it's necessary to describe an elevation function for the sea surface. Currents and waves are not currently implemented, but waves should be fairly simple to implement given an elevation field, and judging by the rest of the framework I doubt currents are much more difficult.AGX allows for currents using the

`WaterFlowGenerator` interface.

5.3 Future work

Currently only the surface vessel has a control system implemented and this control system is a waypoint style system. The ROV is only a hanging weight as implemented right now. In the final implementation of the Plan Sea project, it's likely that one of the vessels will be actively following the other to allow for the desired range of motion or lifting strength. Thus further development on the control side of the simulator is necessary.

The tether connecting the surface vessel and the ROV is currently a fixed length. AGX does have methods of simulating winches, including their internal forces, required torques and other elements. This implementation should be considered a high priority to make the simulation closer to reality and allow for more dynamic states to be simulated.

Further, there are no environmental effects as the simulator stands today. Implementing especially currents would be essential to getting a model that can be instructive about the real-life applicability of the controllers designed. Waves would also be important to implement, along with a heave-compensation system to allow the ROV to maintain altitude even as the surface vessel is moving vertically.

The simulator right now requires a knowledge and understanding of both Python and AGX to configure and run. It would be nice if in the future it is possible to configure basic functions of the simulator through a graphical user interface as opposed to through code. I believe this should be a lower priority to implement, as currently the simulator is a development tool, but it would be nice to have.

Another nice-to-have is a better model of the surface vessel and the ROV. More detailed models would allow for a more accurate hydrodynamic simulation to be run. Higher detail would also increase the complexity of the model though, so it should only be implemented to a certain point. I believe that better models should also be considered a low priority, as the approximations taken right now are likely accurate enough, within one order of magnitude or so.

For my Master's thesis, I would like to work further on this simulator to create a well functioning control system and then apply it to the real-life vessel and ROV. I would implement some of these changes, as well as deal with the physical modifications of the real-life vessel and ROV.

CHAPTER SIX

CONCLUSIONS

The stated goal of the project has been to make a simulation of the Plan Sea project's proposed implementation. The project consists of a small surface vessel and a non-buoyant remotely operated vehicle attached to each other by a lifting wire. The simulation has been made to allow for control system design, as well as to act as an engineering tool for this specific project, allowing different sizes and types of wires to be used, allowing for changing sea-state and currents, or the type and size of vessels.

In this report I've documented why the simulation would be a helpful tool for prototyping. I've also described some of the reasons why I believe simulation is more helpful for this case than finding analytical solutions to the problems posed by the project.

I have created the simulation and run some validation tests on it to see whether it acts close to as should be expected from a realistic simulation. I have also created a simple PD-control system which is able to position the surface vessel at desired points in the world-space. The results found show that the simulation is likely more accurate than the simple analytical methods used to estimate the forces that would act on the system.

I have proposed a list of topics for future iterations of the simulator, as well as stating my goal of continuing work on the simulator to use it for further development of the Plan Sea project.

In all, I would count this project a success. The simulator has been created and validated, and is now ready for further processing and use in the Plan Sea project.

BIBLIOGRAPHY

[1] Jenna R. Jambeck et al. “Plastic waste inputs from land into the ocean”. In: *Science* 347.6223 (Feb. 2015). Publisher: American Association for the Advancement of Science, pp. 768–771. DOI: 10.1126/science.1260352. URL: https://www.science.org/doi/10.1126/science.1260352 (visited on 06/12/2024).

[2] World Economic Forum. *Top 25 recycling facts and statistics for 2022*. en. June 2022. URL: https://www.weforum.org/stories/2022/06/recycling-global-statistics-facts-plastic-paper/ (visited on 06/12/2024).

[3] Atsuhiko Isobe and Shinsuke Iwasaki. “The fate of missing ocean plastics: Are they just a marine environmental problem?” In: *Science of The Total Environment* 825 (June 2022), p. 153935. ISSN: 0048-9697. DOI: 10.1016/j.scitotenv.2022.153935. URL: https://www.sciencedirect.com/science/article/pii/S0048969722010270 (visited on 06/12/2024).

[4] Joana Correia Prata et al. “Environmental exposure to microplastics: An overview on possible human health effects”. In: *Science of The Total Environment* 702 (Feb. 2020), p. 134455. ISSN: 0048-9697. DOI: 10.1016/j.scitotenv.2019.134455. URL: https://www.sciencedirect.com/science/article/pii/S0048969719344468 (visited on 15/12/2024).

[5] Natalia Zolotova et al. “Harmful effects of the microplastic pollution on animal health: a literature review”. en. In: *PeerJ* 10 (June 2022). Publisher: PeerJ Inc., e13503. ISSN: 2167-8359. DOI: 10.7717/peerj.13503. URL: https://peerj.com/articles/13503 (visited on 15/12/2024).

[6] Mariana Segovia-Mendoza et al. “How microplastic components influence the immune system and impact on children health: Focus on cancer”. en. In: *Birth Defects Research* 112.17 (2020). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/bdr2.1779, pp. 1341–1361. ISSN: 2472-1727. DOI: 10.1002/bdr2.1779. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/bdr2.1779 (visited on 15/12/2024).

[7] Gloria Ukalina Obuzor and Uchechi Bliss Onyedikachi. “Chemical Leaching into Food and the Environment Poses Health Hazards”. en. In: *Modernity in Health and Disease Diagnosis: The Account from STEM Women*. Ed. by Eucharia Oluchi Nwaichi. Cham: Springer Nature Switzerland, 2023, pp. 129–148. ISBN: 978-3-031-34963-8. DOI: 10.1007/978-3-031-34963-8_14. URL: https://doi.org/10.1007/978-3-031-34963-8_14 (visited on 15/12/2024).

[8] Ibukun Oluwoye et al. "Degradation and lifetime prediction of plastics in subsea and offshore infrastructures". In: *Science of The Total Environment* 904 (Dec. 2023), p. 166719. ISSN: 0048-9697. DOI: 10.1016/j.scitotenv.2023.166719. URL: https://www.sciencedirect.com/science/article/pii/S0048969723053445 (visited on 06/12/2024).

[9] Walter Lian and Bjem Sortland. "Manoeuvering of Bodies Suspended at Extreme Water Depth". en. In: Los Angeles, May 1996. (Visited on 10/12/2024).

[10] Peng Chen et al. "Dynamic characteristics of deep-sea ROV umbilical cables under complex sea conditions". In: *Ocean Engineering* 239 (Nov. 2021), p. 109854. ISSN: 0029-8018. DOI: 10.1016/j.oceaneng.2021.109854. URL: https://www.sciencedirect.com/science/article/pii/S002980182101204X (visited on 19/09/2024).

[11] Thomas Thuesen Enevoldsen et al. "Simplified Modelling and Identification of an Inspection ROV". en. In: *IFAC-PapersOnLine* 51.8 (2018), pp. 257–262. ISSN: 24058963. DOI: 10.1016/j.ifacol.2018.06.386. URL: https://linkinghub.elsevier.com/retrieve/pii/S240589631830716X (visited on 19/09/2024).

[12] Enrico Anderlini, Gordon G. Parker and Giles Thomas. "Control of a ROV carrying an object". en. In: *Ocean Engineering* 165 (Oct. 2018), pp. 307–318. ISSN: 00298018. DOI: 10.1016/j.oceaneng.2018.07.022. URL: https://linkinghub.elsevier.com/retrieve/pii/S0029801818312654 (visited on 19/09/2024).

[13] Per J. Thingstad and Haakon D. Hveding. "A Nonbuoyant ROV for Performing Heavy Subsea Work". en. In: *Offshore Technology Conference*. Houston, Texas: Offshore Technology Conference, 1982. DOI: 10.4043/4182-MS. URL: http://www.onepetro.org/doi/10.4043/4182-MS (visited on 19/09/2024).

[14] Algoryx. *AGX Dynamix User Manual*. URL: https://www.algoryx.se/download/installers/AGX_Dynamics_UserManual.pdf (visited on 15/12/2024).

[15] *BlueROV2 Heavy Configuration Retrofit Kit*. en-US. URL: https://bluerobotics.com/store/rov/bluerov2-upgrade-kits/brov2-heavy-retrofit/ (visited on 14/12/2024).

[16] *FordypOgMaster/Simulator at main · MagnusKjorseng/FordypOgMaster*. en. URL: https://github.com/MagnusKjorseng/FordypOgMaster/tree/main/Simulator (visited on 14/12/2024).