Øyvind Reppen Lunde

# USV Control System Development for Autonomous AUV Recovery

Master's thesis in Cybernetics and Robotics
Supervisor: Kristin Y. Pettersen
Co-supervisor: Glenn Bitar, Else-Line Malene Ruud
June 2022

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Øyvind Reppen Lunde

# USV Control System Development for Autonomous AUV Recovery

Master's thesis in Cybernetics and Robotics
Supervisor: Kristin Y. Pettersen
Co-supervisor: Glenn Bitar, Else-Line Malene Ruud
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

# Preface

This thesis, written in the spring of 2022, concludes my master's degree in Cybernetics and Robotics at the Norwegian University of Science and Technology (NTNU). The thesis, concerning the autonomous recovery of an Autonomous Underwater Vehicle (AUV) using an Unmanned Surface Vehicle (USV), has been written in cooperation with the Norwegian Defence Research Establishment (FFI), as a part of the Norwegian navy's ongoing project of making maritime mine countermeasures autonomous [1]. I would like to thank my supervisors, Kristin Y. Pettersen, Glenn Bitar, and Else-Line Malene Ruud, for providing invaluable help during the development of the thesis. I am also grateful for the help from Jarle Sandrib and the other FFI employees in Horten.

The reader of the thesis is assumed to have a background in cybernetics, where especially knowledge concerning modeling and control of surface vessels might be needed. The thesis is a continuation of a project thesis concerning the same task, written in the fall of 2021 by the same author. With the problem, vessel, and other parts of the two theses being similar, certain paragraphs and subsections from the project thesis are also partly or fully used here. This will be specified at the corresponding places in the thesis.

Øyvind Reppen Lunde

Trondheim, May 31, 2022

# Abstract

The Norwegian Defence Research Establishment (FFI) has an ongoing project of developing autonomous marine mine countermeasures. Aiming to solve a subtask in the project, this thesis considers the development of a control system for an unmanned surface vehicle, equipped with a Launch-and-Recovery system, that lets it navigate to and autonomously recover an autonomous underwater vehicle.

Since physical access to the vehicles was limited, the use of an existing simulation environment was essential. The simulator was further improved by modeling and implementing extensions such as time delay, measurement noise, and environmental disturbances to make it more similar to the physical system. Using an existing control system, the impact of these extensions was tested to see how they affect the system, and measures were taken to counteract their unwanted effects.

Two different control systems with slightly different properties were developed, both aiming to accomplish the recovery task. The first focuses on being simple and reliable, while the other uses a more complex heading controller with the goal of achieving increased robustness. Both were tested and evaluated on the physical system.

After the physical testing, one system was chosen. It was then further upgraded with a revised strategy for the recovery, a more robust guidance law, and better-tuned controllers. Finally, the upgraded control system was tested on the physical system. The final results showed a control system that, under certain conditions, seemed able to accomplish the recovery task.

# Sammendrag

Forsvarets forskningsinstitutt har et pågående prosjekt for minemottiltak, med mål om å gjøre hele operasjonen autonom. Denne oppgaven forsøker å løse en deloppgave av prosjektet. Oppgaven tar for seg utviklingen av et kontrollsystem for et overflatefartøy, utstyrt med et opphentingssystem, som lar den autonomt navigere til og plukke opp en autonom undervannsfarkost som ligger i vannoverflaten.

Siden tilgang til det fysiske systemet var begrenset, har bruk av en eksisterende simulator vært avgjørende. Simulatoren har blitt ytterligere forbedret ved å modellere og implementere elementer som tidsforsinkelse, målestøy og miljøforstyrrelser for å gjøre den mer lik det fysiske systemet. Ved å bruke et eksisterende kontrollsystem har effekten av disse utvidelsene blitt testet for å se hvordan de påvirker systemet, og det har blitt gjort tiltak for å motvirke deres uønskede effekter.

To forskjellige kontrollsystemer med litt forskjellige egenskaper har blitt utviklet, begge med mål om å klare å plukke opp undervannsfarkosten. Det første fokuserer på enkelhet og pålitelighet, mens det andre bruker en mer avansert regulator med mål om å være mer robust mot forstyrrelser. Begge ble testet og evaluert på det fysiske systemet.

Etter den fysiske testen ble ett av systemene valgt. Det ble så oppgradert med en revidert strategi for å løse oppgaven, en mer robust metode for generering av referanser, og bedre tilpassede regulatorer. Til slutt ble det oppgraderte kontrollsystemet testet på det fysiske systemet. De endelige resultatene viser et kontrollsystem som, under visse betingelser, ser ut til kunne klare opplukkings-oppgaven.

# Contents

# List of Figures

# List of Tables

# Acronyms

API - Application Programming Interface

AUV - Autonomous Underwater Vehicle

CG - Center of Gravity

CO - Center of Origin

DOF - Degree of Freedom

FFI - Norwegian Defence Research Establishment (Forsvarets forskningsinstitutt)

ILOS - Integral Line-of-Sight

LARS - Launch-and-Recovery System

LOS - Line-of-Sight

NED - North-East-Down

NLLS - Non-linear least squares

NTNU - Norwegian University of Science and Technology

RBB - Rigid Buoyancy Boat

RPM - Revolutions Per Minute

ROS - Robot Operating System

RV - Rendezvous

SMC - Sliding Mode Control

SSA - Smallest Signed Angle

USV - Unmanned Surface Vehicle

WP - Waypoint

# 1 Introduction

## 1.1 Motivation

Autonomous vessels are a growing field of research and commerce, with an increasing number of companies researching and developing them for various purposes. Autonomous ships give the shipping industry advantages such as increased efficiency, reduced costs and carbon emissions, and reduced personnel risk [2]. The use of AUVs is increasing in several parts of the industry, such as in oil drilling and inspection of fish farms [3] [4]. In Trondheim, a small, autonomous ferry is being developed, with the goal of transporting up to 12 passengers across the canal [5]. Such autonomous ferries have the advantages of increased availability and longer operating hours than manned ones. They can also replace the more expensive option of building bridges, which either block larger ships or would need to be opened and closed in order to let them through. Scaling this up to larger ferries could give inhabitants of small islands a cheap, accessible, and potentially zero-emission way of transportation, and may help revitalize such remote communities.

## 1.2 The Maritime Mine Countermeasures Project

Autonomous vessels are also of significant interest for military purposes. The Norwegian navy has an ongoing project for maritime mine countermeasures, aiming to make the operation of removing mines from the ocean fully autonomous [1], meaning that there will be no personnel in the minefield. To achieve this, both USVs and AUVs play a vital role. The Hugin-class AUVs [6] are tasked with searching for, detecting, and identifying the mines, while the Odin-class USVs will perform the mine sweeps that trigger the mines to explode. Both are shown in fig. 1.1. The project may also require the vehicles to cooperate - after the AUV has finished the mine search it will return to the surface, where the USV might be tasked with recovering it using a Launch-and-Recovery system (LARS), before returning it to port, the mother ship, or a new operation.

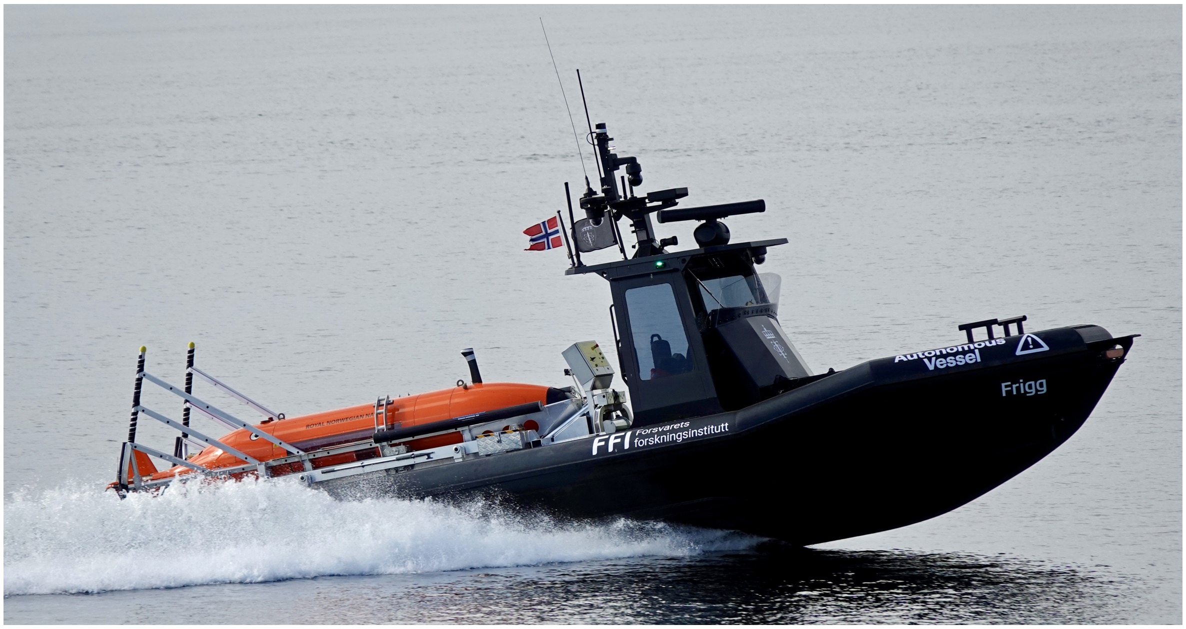


Figure 1.1: The USV, and the AUV on top of the attached LARS. Courtesy of FFI.

### 1.2.1 The Recovery Task

The task of recovering the AUV, which is a potential subtask of the maritime mine countermeasures project, is the main topic of this thesis. For the recovery, the LARS attached to the USV is a Stinger [7]. The Stinger is fixed to the USV and can be submerged into the water when the USV is approaching the AUV for recovery. To recover, the AUV has to slide onto the Stinger, where the nose hook of the AUV attaches to a transverse rope attached to the Stinger. In this thesis, the AUV recovery task is solved by dividing it into three main parts:

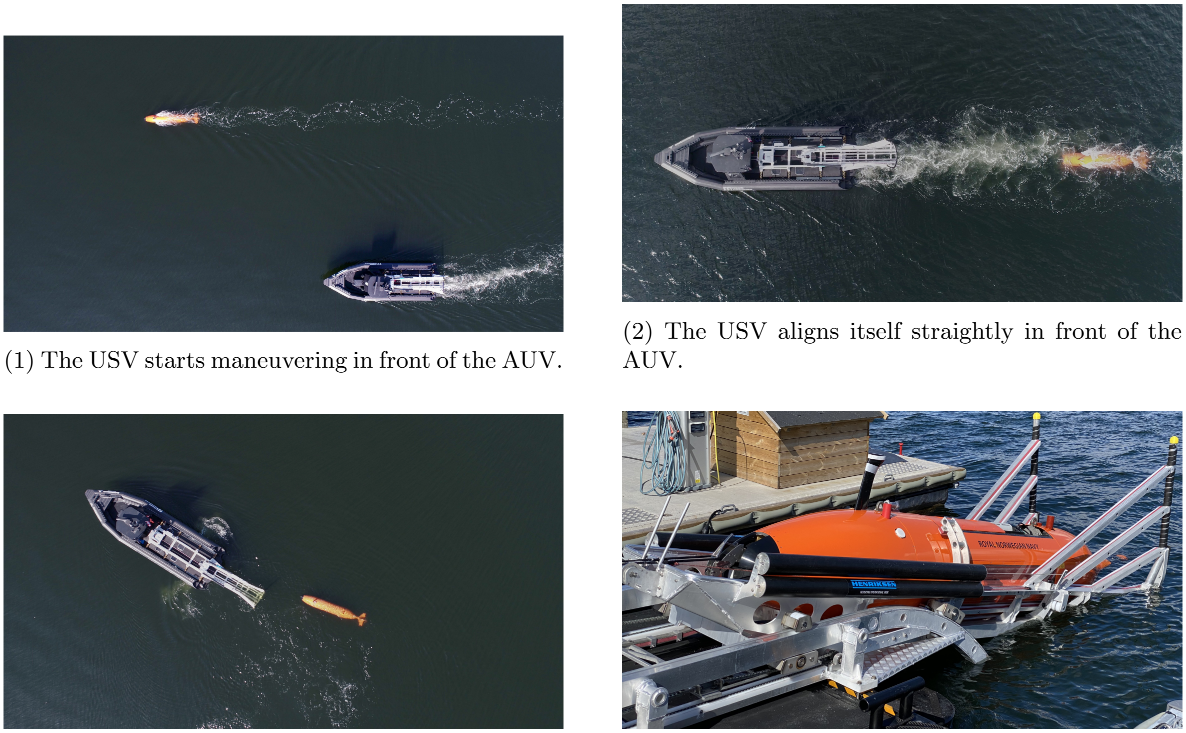
1. With the AUV having finished its search and returned to the surface, the USV maneuvers in front of the AUV.

2. The Stinger is submerged in the water. The USV slowly reduces the distance to the AUV.

3. The USV docks into the AUV, such that the hook on the AUV connects to the Stinger. The Stinger lifts the AUV up onto the USV.

Part 1 is referred to as the initial phase, parts 2 and 3 as the recovery phase, and the entire operation is referred to as the recovery task. A visualization of how the recovery task can be performed can be seen in fig. 1.2.

At the surface, the AUV can either lie still (dead), or maintain a low, constant speed. The former is considered the harder case and will be the focus of this thesis.



(1) The USV starts maneuvering in front of the AUV.

(2) The USV aligns itself straightly in front of the AUV.

(3) The USV submerges the Stinger and reverses towards the AUV.

(4) The AUV is attached to the Stinger and can be lifted onto the USV

Figure 1.2: A visualization of the recovery task. Courtesy of FFI.

## 1.3 Problem Description

The main objective of this thesis is to develop a control system that autonomously accomplishes the recovery task described in section 1.2.1. Since access to the USV and especially the AUV is severely limited by time and cost constraints, further development of an existing simulator and testing the developed control system in it will be important. Since it is not straightforward to know whether or not a solution that is only tested in a simulator will work in the real world, several solutions to the recovery task should be researched, developed, and tested in the simulator in order to have multiple possibilities for the real-world testing. Summarized, the objectives of this thesis will be:

- Extend the existing simulator environment to include time delay, noise, and environmental disturbances.
- Develop and test multiple control systems that can:
  - Plan and follow a path that positions the USV in front of the AUV.
  - Steer the USV accurately towards the AUV and make contact, such that the AUV attaches to the Stinger.
  - Avoid water jet-AUV interaction by minimizing thrust output when the USV is close to the AUV.
  - Handle some degree of external disturbances to the USV/AUV system, such as wind, waves, and ocean currents.
- Perform real-world testing of the developed control systems.

Since a recovery attempt may fail, the control systems should be able to reset and re-try the recovery a number of times. This thesis assumes open waters with the AUV and the USV as the only elements present, such that there are no external obstacles.

## 1.4 Related Work

In this thesis, the AUV is considered to be a passive component and the USV will be the active one for the recovery, whereas in most of the research concerning this topic, the roles are the opposite.

Sarda and Dhanak study different methods for launch and recovery of an AUV using a USV [8], where their USV, a catamaran, is very different from the one in this thesis. They test modifications of both a funnel-shaped docking station and a latching system using trap doors to connect the USV and the AUV, where the latter is based on the same method that the Stinger in this thesis uses. The same authors conduct simulations to test recovery for their final solution [9], which is accomplished by having the AUV home onto a moving dock before being recovered onto the USV. For recovery, acoustic positioning is used to align the two vessels. The USV then lowers a thin line with a depressor wing from a winch, which the AUV latches on to. The line can then be reeled in to recover the AUV.

Proctor et al. [10] attempt to solve the AUV-USV recovery by aligning the vessels on the same path with matching speed. As in this thesis, [10] uses the Hugin-class AUV. Here the AUV is an active component that for the final part of the recovery drives into the back of the USV to dock into it.

Breivik and Loberg [11] consider docking a USV with a larger mother ship. The USV operates in a manner that is relevant to the recovery task in this thesis, while the mother ship is a passive component maintaining a constant velocity, similar to how the AUV may operate in the recovery task. The USV, using constant bearing guidance, tracks the position and velocity of a target point. This point is first located on a safety circle around the mother ship. Having reached this point, the target point then starts moving towards the desired docking point such that docking is achieved.

## 1.5 Contributions

The main contributions of this thesis are:

- Development of two state-space models of the USV. One where the Stinger is raised and one where it is submerged in the water. Both have been implemented in the simulator.
- The modeling and implementation of time delay, measurement noise, and environmental disturbances in the simulator.
- Two control systems that include:
    - Transmission and reception of measurements, control inputs, and other relevant information in order to generate references for and control the USV, both in the simulator and for the physical system.
    - A path planner generating waypoints for the USV based on the pose of the AUV.
    - Controllers that follow the waypoints and finally steer the USV slowly and accurately towards the AUV.
- Planning and execution of physical tests of the control systems.

## 1.6 Preliminary Work

The preliminary work for this thesis is a project thesis, written in the fall of 2021, that concerns the same task. The focus of the project thesis was to, based on gathered experimental data, develop state-space models of the USV system in order to have a realistic simulator for the USV, and design a simple proof-of-concept control system that could accomplish the recovery task in the simulator. Two 3 degrees of freedom (DOFs) state-space models that described the dynamics of the USV as a function of its current velocity and the control inputs from the water jets, were developed. One model described the USV with the Stinger raised and the other with the Stinger submerged in the water. The control system was developed in a simple environment, with the main purpose of testing the models in action, and show that the recovery task could be accomplished in principle.

Section 1.1, section 1.2, and section 1.4 are largely copied from the project thesis, as the motivation, the maritime mine countermeasures project, and related work have been similar for both theses.

## 1.7 Outline

Section 2 presents the Odin-class USV and other key components of the recovery system, the simulation environment, and relevant parts from the preliminary work for this thesis. Section 3 concerns the extension of the simulation environment to make it more realistic. Section 4 describes the development of the new control systems. Section 5 covers the physical testing of the control systems, as well as the further development made between the tests. Finally, section 6 concludes the thesis with a summary and discussion of the key results, as well as a proposition for further work.

# 2 The USV System

This section addresses information regarding the USV system, which includes both the physical system and its corresponding simulation environment. The physical system consists of the Odin-class USV (section 2.1), the AUV Hugin (section 2.2), and the Stinger (section 2.3), while the simulation environment (section 2.4) has functionality for modeling, simulation, and visualization of these. Section 2.5 concludes the section by presenting the control system which was developed in the project thesis. This section is largely copied from the project thesis and is included the completeness of the thesis. Apart from subsections being re-written or extended, the main new content is section 2.4.1.

## 2.1 The Odin-Class USV

FFI has two USVs of the Odin-class, named *Frigg* and *Odin* [12]. Frigg is depicted in fig. 1.1, and Odin is almost identical. Frigg is the one that will carry the Stinger and perform the AUV recovery and is thus the one this thesis will consider. Frigg is a 10.5 m long and 2.5 m wide rigid buoyancy boat (RBB). It is equipped with two Hamilton HJ292 water jets as its thrusters. In order to operate autonomously, Frigg is also equipped with a multitude of sensors, such as radar, Lidar, camera, GNSS, and IMU. For this thesis, measurements of the position, orientation, and velocity of Frigg will mainly be used.

A ship has in general a total of 6 DOFs, as depicted in fig. 2.1, as it can have linear and angular motion along each of the three principal axes. However, for surface vessels such as *Frigg* and *Odin*, one typically neglects motion in *roll*, *pitch*, and *heave* as these are often small or uncontrollable, or both. Thus, the system can have linear motion on the ocean surface and angular motion around the vertical axis, giving us 3 DOFs with *surge*, *sway*, and *yaw* as the control states.

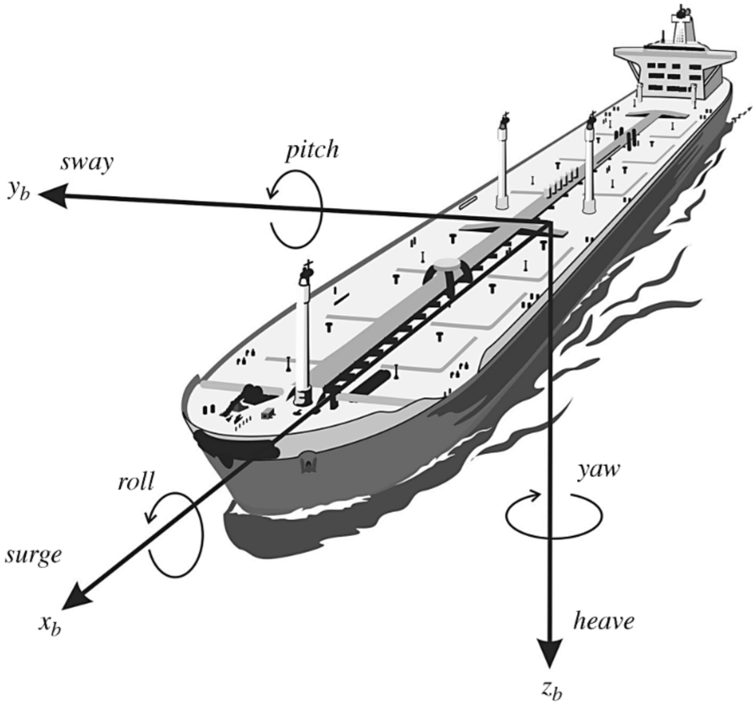


Figure 2.1: Illustration of motion in 6 DOFs for a marine vessel from [13]. ©John Wiley & Sons.

The two main coordinate frames used to describe the motion of the USV are BODY and NED. NED

(North-East-Down) denotes the world frame and gives the position and orientation of the vessel relative to a fixed frame, with the origin as a fixed point, typically on the ocean surface. In NED, the x-axis points toward north, the y-axis points toward east, and the z-axis points downwards, normal to the Earth's surface. BODY denotes a body-fixed reference frame with origin as a fixed point on the vessel, and the axes aligned with the ship, as seen in fig. 2.1. The position and orientation of the vessel are described relative to the NED-frame, while the velocity of the vessel is described relative to the BODY-frame. This thesis uses the notation defined in SNAME [14], which can be found in table 2.1.

| | | BODY | | NED |
|---|---|---|---|---|
| DOF | | Forces and moments | Linear and angular velocities | Positions and Euler angles |
| 1 | Motions in the $x_b$-direction (surge) | $X$ | $u$ | $x$ |
| 2 | Motions in the $y_b$-direction (sway) | $Y$ | $v$ | $y$ |
| 6 | Rotation about the $z_b$-axis (yaw) | $N$ | $r$ | $\psi$ |

Table 2.1: The notation for marine craft from SNAME [14], here for the 3 degrees of freedom for a surface vessel.

### 2.1.1 The Water Jets

Water jets were chosen over the more commonly used propellers, mainly for their excellent maneuverability properties, giving them an edge over propellers in docking situations and when repeatedly switching between forward and reverse thrust [15]. They also provide a strong pulling force, and the manufacturer, *HamiltonJet*, supplies a great application programming interface (API) for control of the jets.

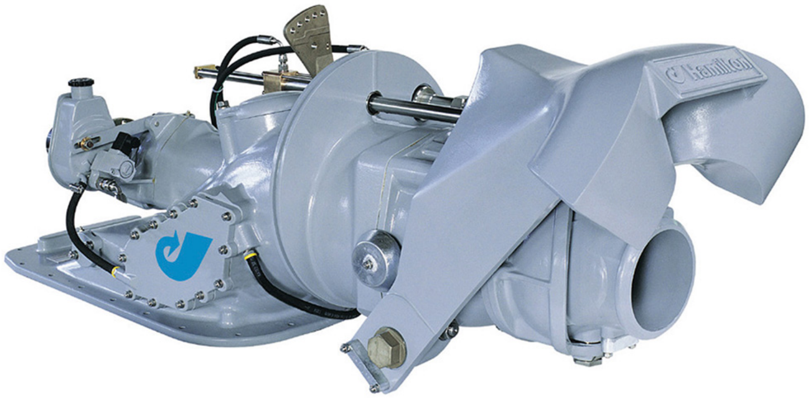


Figure 2.2: The HJ292 water jet. ©Boatswain's Locker [16].

Each water jet has the control inputs throttle, steering (the angle of the water jet), and bucket (a bucket that can be lowered in front of the jet to reverse the direction of the water stream). The water jet is depicted in fig. 2.2, where the bucket that can be lowered to reverse the water stream is located to the right, mounted on the main body. With a total of six control inputs and assuming a 3-DOF model for a surface vessel, the system is over-actuated. The water jets can be controlled in two ways: Using *Helm mode* where the three inputs of each jet are controlled directly, or with a system supplied by *HamiltonJet* called *Thrust mode*.

Thrust mode lets us set desired forces directly on the USV by setting desired demands in surge, sway, and yaw as a percentage from -100 to 100 in each state, meaning that this mode has three states. The system then calculates and sets the water jet inputs that achieve this. While Helm mode quite easily can be used to perform traditional maneuvers, it can be challenging to find the correct set of inputs to perform the more complex maneuvers that are required for the recovery task. To be able to use this in a controller, both modes must have a corresponding transfer function that maps their inputs to forces and moments, a task that is far easier for Thrust mode as the transfer function for Helm mode would be much more complex. With Thrust mode being assumed to be the easiest one to use for control of the USV, it is the one that will be used in this thesis.

Since this mode only has three inputs, each controlling their individual state, the 3-DOF system is, in principle, fully actuated. However, the Thrust mode system is based on an open-loop allocation, meaning that it is not perfect and will have some errors, such as a demand not producing forces in only the intended state. This is especially the case for demands in sway which also give a large yaw moment, in addition to the corresponding sway forces not being large. The latter is also due to the design of the hull of the USV, which is formed for forward movement along its x-axis. Though it has some flaws, Thrust mode is quite good and can, combined with a controller, give us the tools needed to obtain robust control of the USV and execute the recovery task. But due to the limitations regarding the movement in sway, it will often be beneficial to treat the system as under-actuated, with demands in surge and yaw as our control inputs.

### 2.1.2 Water Jet-Induced Flow

The water jets induce a water flow when used, and this can be problematic as the water flow may push the AUV around and make it harder to perform the recovery task. The induced water flow increases with the throttle use, meaning that the throttle use might not be chosen freely. Too much water flow from the jets is problematic as it can push the AUV away, making the recovery task much harder. It is also worth noting that the buckets, which can reverse the direction of the water stream, may direct the stream towards or away from the AUV, depending on their values. Figure 2.3 shows the revolutions per minute (RPM) of the water jets for increasing surge demands. We see that the jets are idling for demands in the range [-60, 60], and steadily increase the RPM outside of this range. Inside the idling range, the system only changes the bucket values to change the forces acting on the USV, meaning that being within this range minimizes the water flow.

Figure 2.4 shows a similar plot, but now for sway demands. We see that the idling range for pure sway demands is reduced down to [-30, 30] and that the sway demands, in general, require more throttle use.

For the yaw demands, the dataset has a slightly worse resolution than for surge and sway, with few pure yaw demands outside of the range [-50, 50]. Yet, looking at fig. 2.5, it seems that we with some certainty can assume the idling range of the yaw demands to be [-50, 50]. The buckets do not change for yaw demands in Thrust mode with only nozzle and rpm being changed, and as the nozzle maxes out in this range, throttle is the only value left to change, suggesting that it must change everywhere outside of the range [-50, 50].

## 2.2 The AUV Hugin

The AUV Hugin, depicted in fig. 1.1 and fig. 2.6, is the vehicle the USV shall recover. Hugin has high maneuverability, can operate at depths of 3-6000 meters, and has several applications in both commerce and defense [6]. In the maritime mine countermeasures project, the task of Hugin is to detect, localize, and identify mines in the ocean.

In this thesis, Hugin will strictly be a passive component. It will lie still in the water, waiting to be recovered. Measurements of the pose of Hugin will continuously be received. While Hugin is quite stable when lying still in the water, its pose may still change due to environmental disturbances, collisions with the USV, or water flow from the jets, as covered in section 2.1.2.

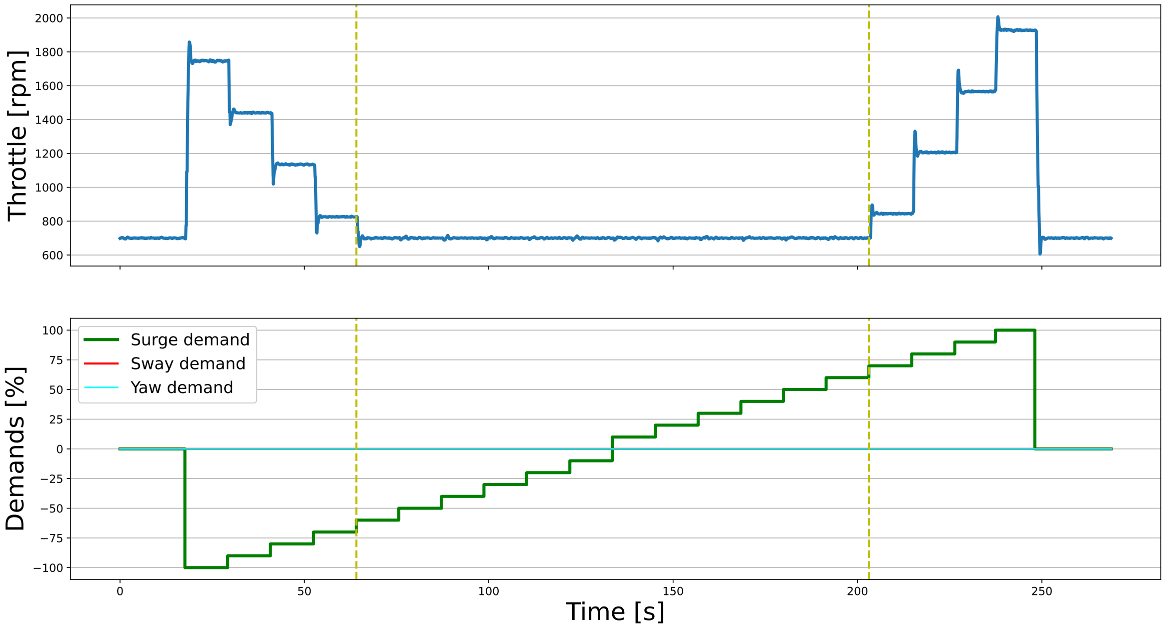


Figure 2.3: RPM values for the water jets for different surge demands.

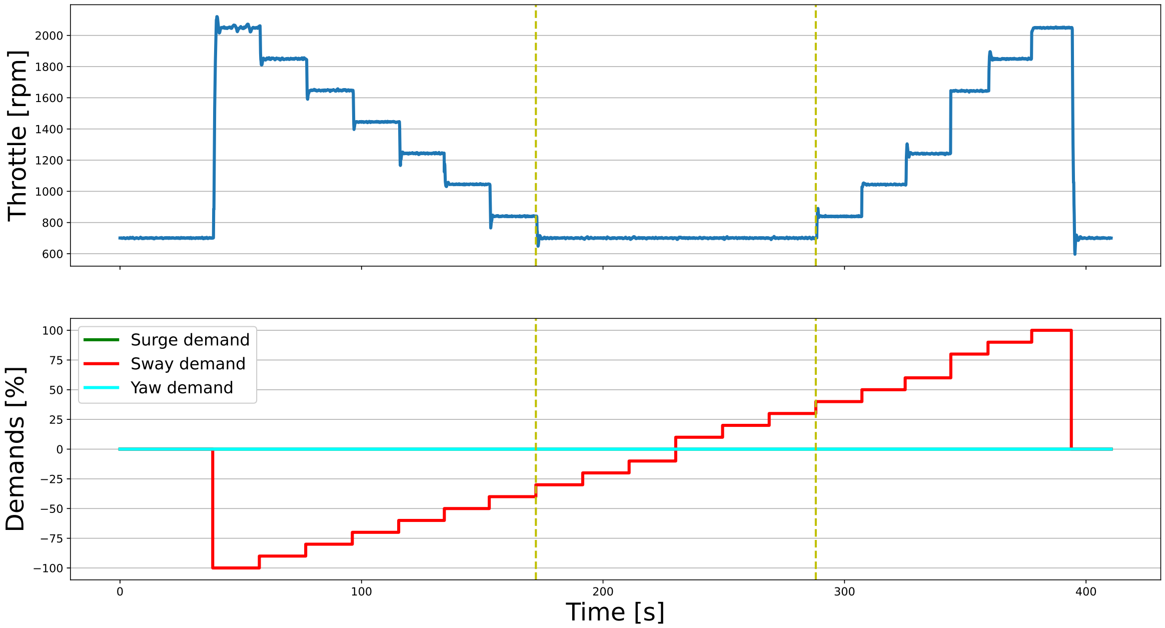


Figure 2.4: RPM values for the water jets for different sway demands.

## 2.3 The Stinger

*Henriksen's* Launch-and-Recovery system, called the Stinger [7], is the component of the USV system that will physically interact with Hugin by attaching a transverse rope to the nose hook of Hugin in order to recover it. Figure 2.7 depicts the Stinger mounted on the USV, while fig. 2.6 shows how the Stinger attaches to and lifts Hugin up onto the USV.

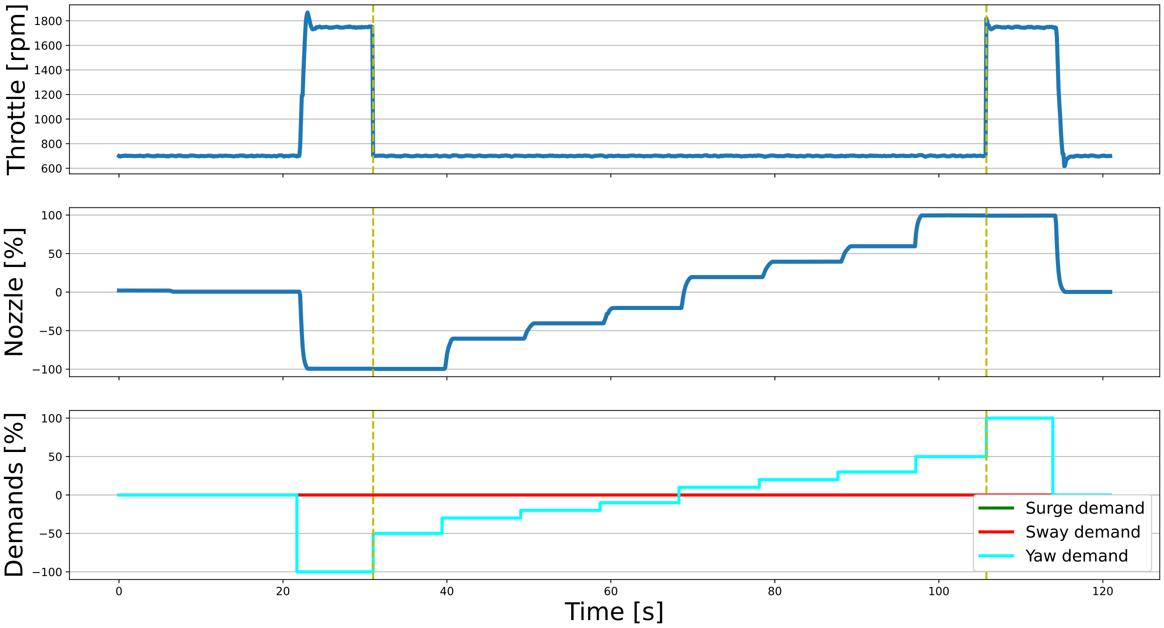


Figure 2.5: RPM and nozzle values for the water jets for different yaw demands.

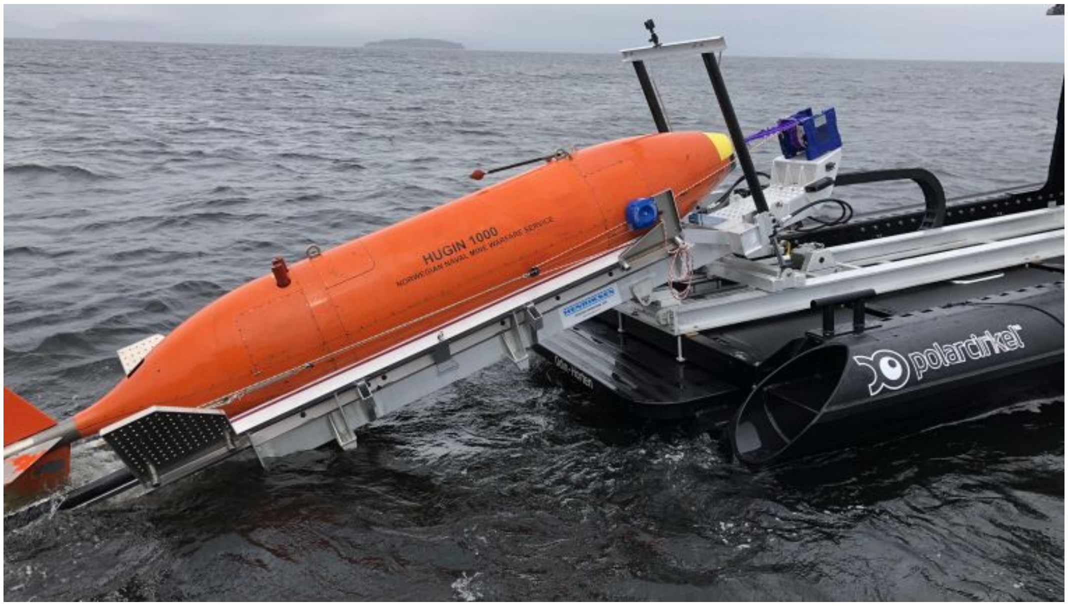


Figure 2.6: The Stinger, mounted on the USV, recovering Hugin. ©Hydro International [17].

In order to recover the AUV, it must enter the shaft of the Stinger, such that it can be caught there and connect to the transverse cable of the Stinger. This means that to successfully recover the AUV, the USV must approach it at some speed at a relatively straight angle, with little error sideways. This sideways error can be a maximum of ± 20 cm.

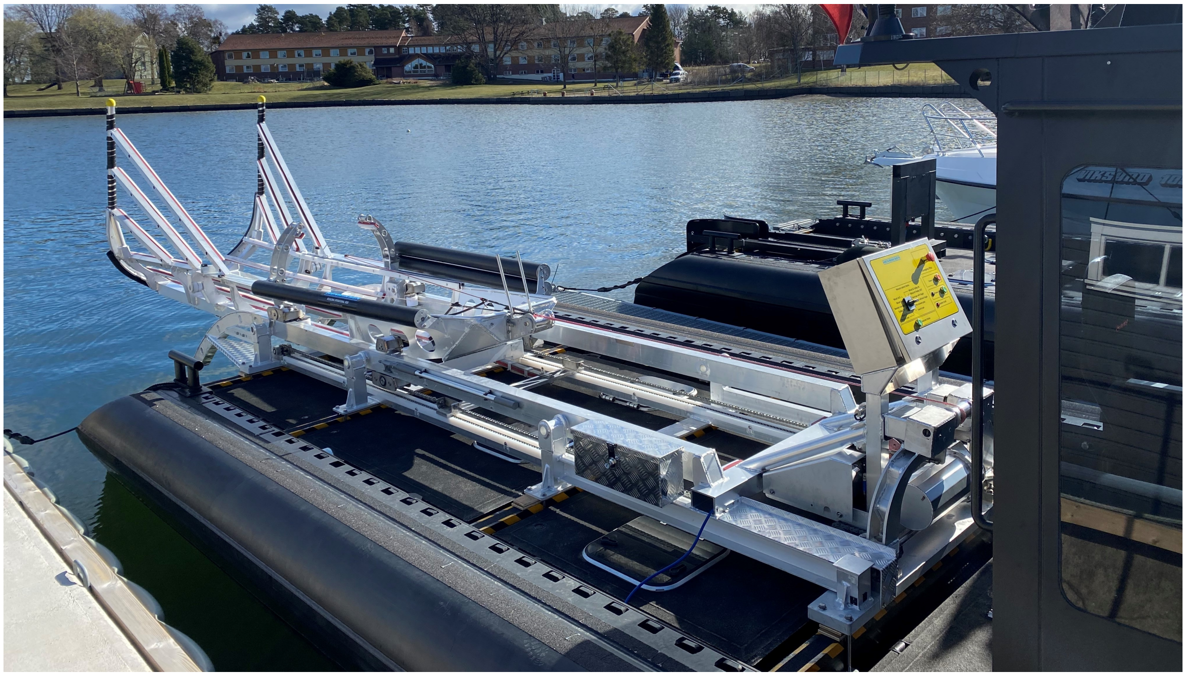


Figure 2.7: The Stinger, mounted on the USV. Courtesy of FFI.

### 2.3.1 USV-Stinger Dynamics

Mounting the Stinger also alters the dynamics of the USV. For the case where the Stinger is raised onto the USV, the dynamics do not change much, except for the USV being slightly heavier and thus making it somewhat slower. When the Stinger is submerged into the water, however, the dynamics drastically change. The Stinger can then be seen as an extension of the hull of the USV, increasing the hydrodynamic damping and moment of inertia around the vertical axis. The latter is drastically increased, but in general, both the acceleration and velocity decrease significantly for all the three states. The altered dynamics with the submerged Stinger induce a need for two separate state-space models of the USV: One for when the Stinger is raised, and one for when it is submerged.

**Stinger-Induced Asymmetry**

Since the Stinger is mounted manually on the USV each time it is to be used, its position and orientation can be slightly different each time. These small differences, in combination with the two water jets being quite close to each other, can potentially yield significantly different dynamics for separate tests, which presents a challenge for both model identification and control of the USV.

Figure 2.8 and 2.9 show two examples from our experimental data where the resulting asymmetry of a skewed Stinger is evident. In fig. 2.8 the USV is given zig-zag demands where it reverses while altering between a positive and negative turning rate. One would expect the USV geometry to be symmetric in the XZ-plane and thus give a yaw rate response that is symmetric around zero. However, the yaw rate is almost entirely positive. This is also the case for the sway motion, which for these maneuvers is expected to be centered around zero as well. This is likely due to the Stinger not being mounted exactly at the center of the USV's y-axis and parallel to the x-axis, in addition to the Thrust mode system being open-loop and thus using a predetermined allocation of the motor inputs. In fig. 2.9 the USV is given pure surge demands. We see that the largest negative surge demands induce a positive spike in both sway and yaw rate, both of which should have been close to zero. The operator could also confirm that the USV felt far easier to turn one way than the other when driving normally.

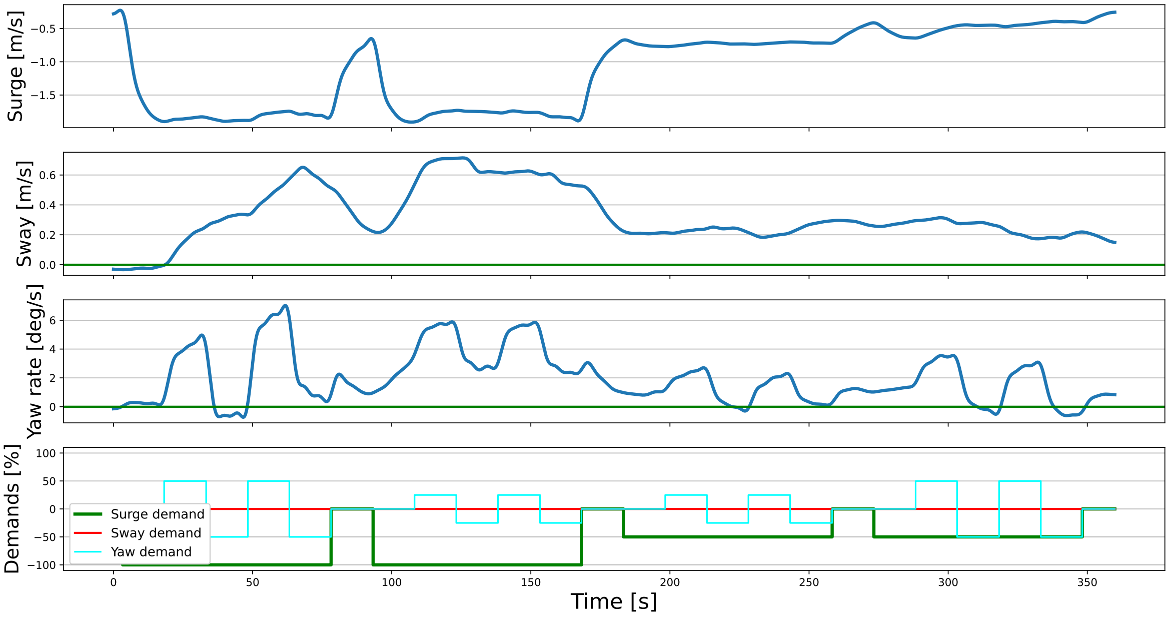


Figure 2.8: USV velocity when given zig-zag demands, with the Stinger submerged in the water.

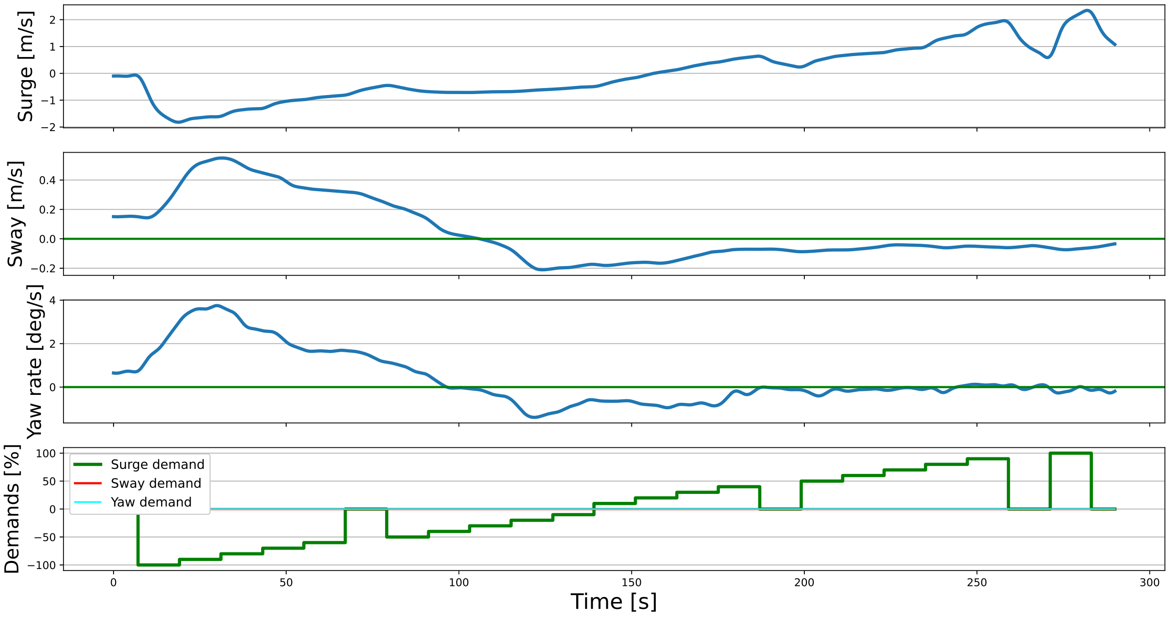


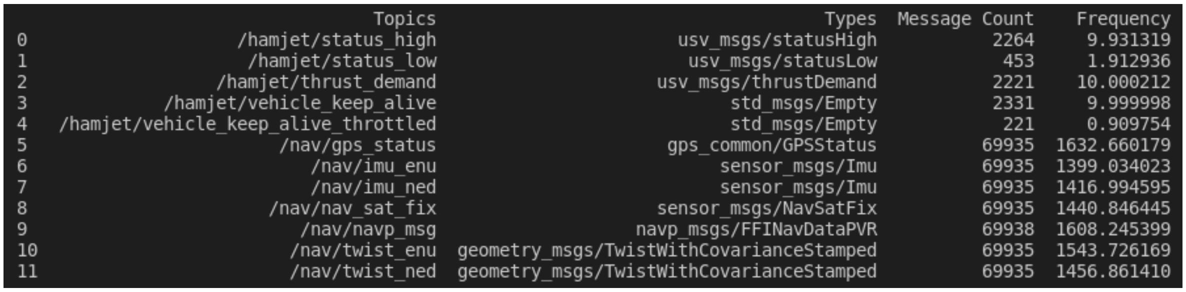
Figure 2.9: USV velocity when given pure surge demands, with the Stinger submerged in the water.

This behavior is not desirable, but is a problem that cannot be removed. The solution will be to find ways to minimize and circumvent the issue. From the two examples, it is evident that the asymmetry in yaw (and sway) becomes larger for larger (negative) surge demands. This might mean that we for the final part of the recovery task have to introduce constraints to the controller on how much force can be used. However, this restriction already exists due to the previously mentioned induced water flow problem, meaning that the asymmetry may not cripple our controller much more than it would be without the issue. As for the asymmetry itself, a controller that utilizes feedback should be able to handle some degree of error while enabling the USV to perform the desired maneuvers.

## 2.4 Simulator

As mentioned in the problem description, there will be few opportunities to test the control system in real life. This creates a need for a simulator that can provide realistic simulations of the dynamics of the USV. Having such a simulator would be desirable even if we had constant access to physical testing, as using controllers that are not tuned properly, have errors, or simply aren't stable to control the vessels could yield fatal errors. Development and testing is also much easier and faster in a simulator.

FFI has an existing 3-DOF vessel simulator, written in C++, with the Robot Operating System (ROS) [18] as its middle-ware. ROS is an open-source set of software libraries and tools for robot applications. Most of the software modules on the USV communicate using ROS, which means that developing the simulator environment using ROS makes it easy to transfer the control system from the simulator to the USV. The modules communicate with each other using *topics* and *nodes*. A ROS topic is a bus used to send and receive messages, and a ROS node is a process, which communicates with other nodes using topics. Figure 2.10 shows a selection of ROS topics available on the USV, which have similar counterparts in the simulator. Most important is the topic */hamjet/thrust_demand*, which is the topic that will be used to send the desired control inputs. Since this topic operates at approximately 10 Hz on the USV, our control system will also operate at this frequency, both in the simulator and in the physical world.



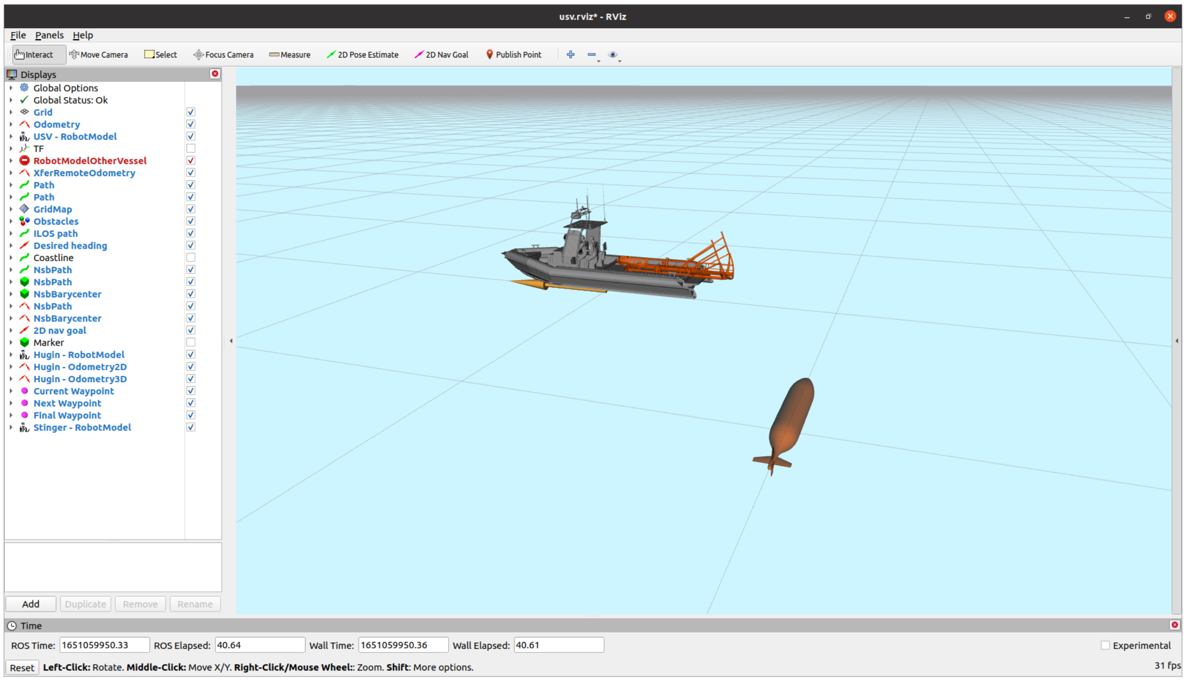
```
                                  Topics                                     Types  Message Count    Frequency
0                    /hamjet/status_high                       usv_msgs/statusHigh           2264     9.931319
1                     /hamjet/status_low                        usv_msgs/statusLow            453     1.912936
2                  /hamjet/thrust_demand                     usv_msgs/thrustDemand           2221    10.000212
3             /hamjet/vehicle_keep_alive                            std_msgs/Empty           2331     9.999998
4   /hamjet/vehicle_keep_alive_throttled                            std_msgs/Empty            221     0.909754
5                        /nav/gps_status                      gps_common/GPSStatus          69935  1632.660179
6                           /nav/imu_enu                           sensor_msgs/Imu          69935  1399.034023
7                           /nav/imu_ned                           sensor_msgs/Imu          69935  1416.994595
8                       /nav/nav_sat_fix                     sensor_msgs/NavSatFix          69935  1440.846445
9                          /nav/navp_msg                   navp_msgs/FFINavDataPVR          69938  1608.245399
10                        /nav/twist_enu  geometry_msgs/TwistWithCovarianceStamped          69935  1543.726169
11                        /nav/twist_ned  geometry_msgs/TwistWithCovarianceStamped          69935  1456.861410
```

Figure 2.10: An selection of the ROS topics available on the USV that we can send and receive messages from.

ROS also supports the Python programming language, which is beneficial as Python has a lot of useful packages and functionality, enabling us to program the control system in Python and directly run it both in the simulator and on the USV. In the preliminary work, the developed models of the USV with the Stinger attached were implemented in the simulator to obtain a fully functional simulation environment. However, further developing the simulator to be more realistic by adding components such as time delay, measurement noise, and environmental disturbances is a central part of this thesis and is covered in section 3. The simulations are visualized using Rviz, which is one of the ROS open-source visualization tools. A screenshot from Rviz is shown in fig. 2.11, displaying the USV with the Stinger attached, and the AUV.

### 2.4.1 Vessel Models

While FFI had an existing model of the USV, this captured its dynamics without the Stinger and used Helm mode control inputs. We require models *with* the Stinger, with Thrust mode control inputs. Thus, the main focus of the preliminary work was to develop models that accurately captured the dynamics of the USV with the Stinger attached, both when it was raised and when it was submerged into the water. To develop the models, experimental data from the USV was gathered, and an



Global Options
Global Status: Ok
Grid ☑
Odometry ☑
USV - RobotModel ☑
TF ☐
RobotModelOtherVessel ☑
XferRemoteOdometry ☑
Path ☑
Path ☑
GridMap ☑
Obstacles ☑
ILOS path ☑
Desired heading ☑
Coastline ☐
NsbPath ☑
NsbPath ☑
NsbBarycenter ☑
NsbPath ☑
NsbBarycenter ☑
2D nav goal ☑
Marker ☐
Hugin - RobotModel ☑
Hugin - Odometry2D ☑
Hugin - Odometry3D ☑
Current Waypoint ☑
Next Waypoint ☑
Final Waypoint ☑
Stinger - RobotModel ☑

ROS Time: 1651059950.33 ROS Elapsed: 40.64 Wall Time: 1651059950.36 Wall Elapsed: 40.61 ☐ Experimental

**Left-Click:** Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options. 31 fps

Figure 2.11: Visualization of the simulator in Rviz.

optimization-based approach was developed to train the models. The general approach and method for the model development is described in appendix A.

In this thesis, revised versions of the models from the preliminary work are used. The original ones, with a structure inspired by [19], were developed with few constraints in terms of parameter values, and an overall goal of being as accurate as possible. However, when introducing new elements such as environmental disturbances, there was some unrealistic behavior, such as wind influencing the USV equally in sway as in surge. It seems that the original models had been developed a bit too freely in terms of sign and value for the parameters, a problem that may be connected to the problem of overfitting [20]. The revised models are somewhat simpler than their original counterparts, and had more constraints during their development, to ensure realistic behavior at all times. The result is models that are slightly less accurate when run on the gathered datasets, but are more realistic in a broader context. The final models, based on the standard model from [13], had the form:

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{f}(\boldsymbol{\tau}_d) \tag{2.1}$$

where $\boldsymbol{\nu}$ is the velocity in surge, sway, and yaw, $\mathbf{M}$ is the inertia matrix, $\mathbf{C}$ is the Coriolis matrix, $\mathbf{D}$ is the damping matrix, and $\boldsymbol{f}(\boldsymbol{\tau}_d)$ is a transfer function converting thrust demands to forces. As a simplification, the added mass and Coriolis is set to zero here. Note that the influence of environmental disturbances will be added in section 3.4.

### Vessel Model for Submerged Stinger

The following describes the model that was used for simulating the USV with the Stinger submerged. It had the form eq. (2.1), with its separate components defined as:

$$
\begin{aligned}
\mathbf{M} &= \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{11} & m_{23} \\ 0 & m_{32} & m_{33} \end{bmatrix} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix} \\
\mathbf{C}(\boldsymbol{\nu}) &= \begin{bmatrix} 0 & 0 & -m_{22}v - m_{23}r \\ 0 & 0 & m_{11}u \\ m_{22}v + m_{23}r & -m_{11}u & 0 \end{bmatrix} \\
\mathbf{D}(\boldsymbol{\nu}) &= -\begin{bmatrix} X_u + X_{|u|u}|u| & 0 & 0 \\ 0 & Y_v + Y_{|v|v}|v| & 0 \\ 0 & 0 & N_r + N_{|r|r}|r| \end{bmatrix} \\
\boldsymbol{\tau} = \boldsymbol{f}(\boldsymbol{\tau}_d) &= \begin{bmatrix} X \\ Y \\ N \end{bmatrix} = \begin{bmatrix} k_0 x_d + k_1 |x_d|\, x_d \\ k_2 y_d + k_3 |y_d|\, y_d + k_4 n_d \\ k_5 n_d + k_6 |n_d|\, n_d + k_7 x_d + k_8 y_d \end{bmatrix} \\
x_d &= \begin{cases} k_9 x_d, & x_d \geq 0 \\ x_d, & x_d < 0 \end{cases} \quad k_7 = \begin{cases} k_{7,pos}, & x_d \geq 0 \\ k_{7,neg}, & x_d < 0 \end{cases}
\end{aligned}
\tag{2.2}
$$

where $m$ is the mass of the USV, $x_g$ is the distance from the center of origin (CO) to the center of gravity (CG), $I_z$ denotes the moment of inertia about the vertical axis, and $u$, $v$, and $r$ denote the velocity in surge, sway, and yaw, respectively. The hydrodynamic coefficients in $\mathbf{D}$ follow the notation from SNAME [14]. The transfer function $\boldsymbol{f}(\boldsymbol{\tau}_d)$ was designed under the assumption that a demand mostly affects its corresponding state, modeled with a second-order polynomial, as well as sway and yaw also being somewhat affected by the other demands. $k_9 > 1$ is added due to the water jets producing more thrust forward than backward, and $k_7$ is split into two cases since the yaw moment induced by the surge demand is significantly different for positive and negative surge demands, due to the asymmetry explained in section 2.3.1. All the gains in $\boldsymbol{f}(\boldsymbol{\tau}_d)$ have the unit [$\text{kg} \cdot \text{m/s}^2$], except $k_9$ which is unit-less.

The final parameter values can be found in table 2.2. The performance of the model is shown in fig. 2.12. The testset consists of several different maneuvers, separated by yellow dotted lines.

| **Parameter** | **Value** | **Unit** |
|---|---|---|
| $m$ | 7787 | kg |
| $I_z$ | 51688.3 | $\text{kg} \cdot \text{m}^2$ |
| $x_g$ | 0.294 | m |
| $X_u$ | -220.0 | kg/s |
| $X_{\|u\|u}$ | -207.9 | kg/m |
| $Y_v$ | -2004.3 | kg/s |
| $Y_{\|v\|v}$ | -2186.9 | kg/m |
| $N_r$ | -4403.9 | $\text{kg} \cdot \text{m/s}$ |
| $N_{\|r\|r}$ | -9.3 | $\text{kg} \cdot \text{m}$ |

| **Parameter** | **Value** |
|---|---|
| $k_0$ | 5.275 |
| $k_1$ | 0.092 |
| $k_2$ | 1.104 |
| $k_3$ | 0.013 |
| $k_4$ | -4.246 |
| $k_5$ | 4.692 |
| $k_6$ | -0.002 |
| $k_{7,pos}$ | -0.043 |
| $k_{7,neg}$ | -0.229 |
| $k_8$ | 1.462 |
| $k_9$ | 1.134 |

Table 2.2: Model coefficients for the Submerged Stinger model.

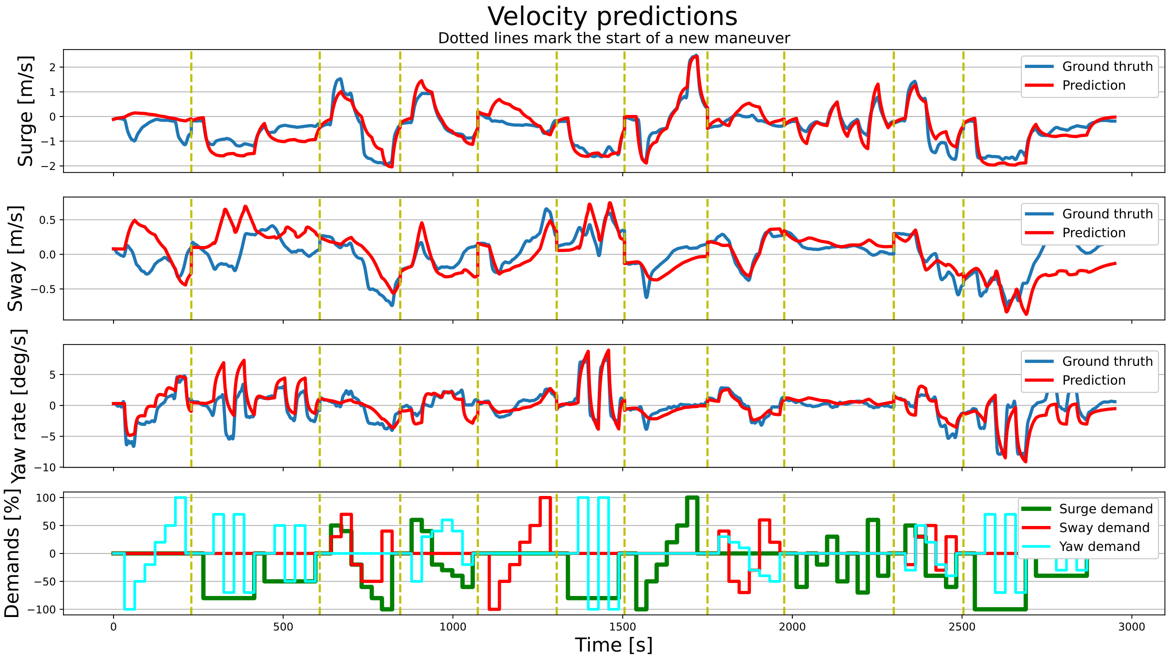


Figure 2.12: Predicted velocity on the test set for the submerged Stinger model.

### Vessel Model for Raised Stinger

The following describes the model that was used for simulating the USV with the Stinger submerged. It had the form eq. (2.1), with its separate components defined as:

$$\begin{aligned}
\mathbf{M} &= \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{11} & m_{23} \\ 0 & m_{32} & m_{33} \end{bmatrix} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{bmatrix} \\
\mathbf{C}(\boldsymbol{\nu}) &= \begin{bmatrix} 0 & 0 & -m_{22}v \\ 0 & 0 & m_{11}u \\ m_{22}v & -m_{11}u & 0 \end{bmatrix} \\
\mathbf{D}(\boldsymbol{\nu}) &= -\begin{bmatrix} X_u + X_{|u|u}|u| & 0 & 0 \\ 0 & Y_v + Y_{|v|v}|v| & 0 \\ 0 & 0 & N_r + N_{|r|r}|r| \end{bmatrix} \\
\boldsymbol{\tau} = \boldsymbol{f}(\boldsymbol{\tau}_d) &= \begin{bmatrix} X \\ Y \\ N \end{bmatrix} = \begin{bmatrix} k_0 x_d + k_1 |x_d|\, x_d \\ k_2 y_d + k_3 |y_d|\, y_d + k_4 n_d \\ k_5 n_d + k_6 |n_d|\, n_d + k_7 x_d + k_8 y_d \end{bmatrix} \\
x_d &= \begin{cases} k_9 x_d, & x_d \geq 0 \\ x_d, & x_d < 0 \end{cases}
\end{aligned} \tag{2.3}$$

where the only differences structure-wise between this model and the submerged Stinger model are that $\mathbf{M}$ now is diagonal, since the CO and CG are assumed to coincide without the Stinger in the water, giving $x_g = 0$, and without the Stinger in the water, we no longer need to split $k_7$. All the gains in $\boldsymbol{f}(\boldsymbol{\tau}_d)$ have the unit $[\mathrm{kg} \cdot \mathrm{m/s^2}]$, except $k_9$ which is unit-less. The parameter values are shown in table 2.3. The performance of the model is shown in fig. 2.13.

| Parameter | Value | Unit |
|---|---|---|
| $m$ | 7787 | kg |
| $I_z$ | 22367.8 | $\text{kg} \cdot \text{m}^2$ |
| $X_u$ | -198.7 | kg/s |
| $X_{\|u\|u}$ | -111.0 | kg/m |
| $Y_v$ | -1500.0 | kg/s |
| $Y_{\|v\|v}$ | -2100.0 | kg/m |
| $N_r$ | -1113.0 | $\text{kg} \cdot \text{m/s}$ |
| $N_{\|r\|r}$ | -12936.5 | $\text{kg} \cdot \text{m}$ |

| Parameter | Value |
|---|---|
| $k_0$ | 12.273 |
| $k_1$ | 0.023 |
| $k_2$ | 4.059 |
| $k_3$ | 0.032 |
| $k_4$ | -6.601 |
| $k_5$ | 2.881 |
| $k_6$ | 0.092 |
| $k_7$ | 1.075 |
| $k_8$ | 0.916 |
| $k_9$ | 1.536 |

Table 2.3: Model coefficients for the Raised Stinger model.

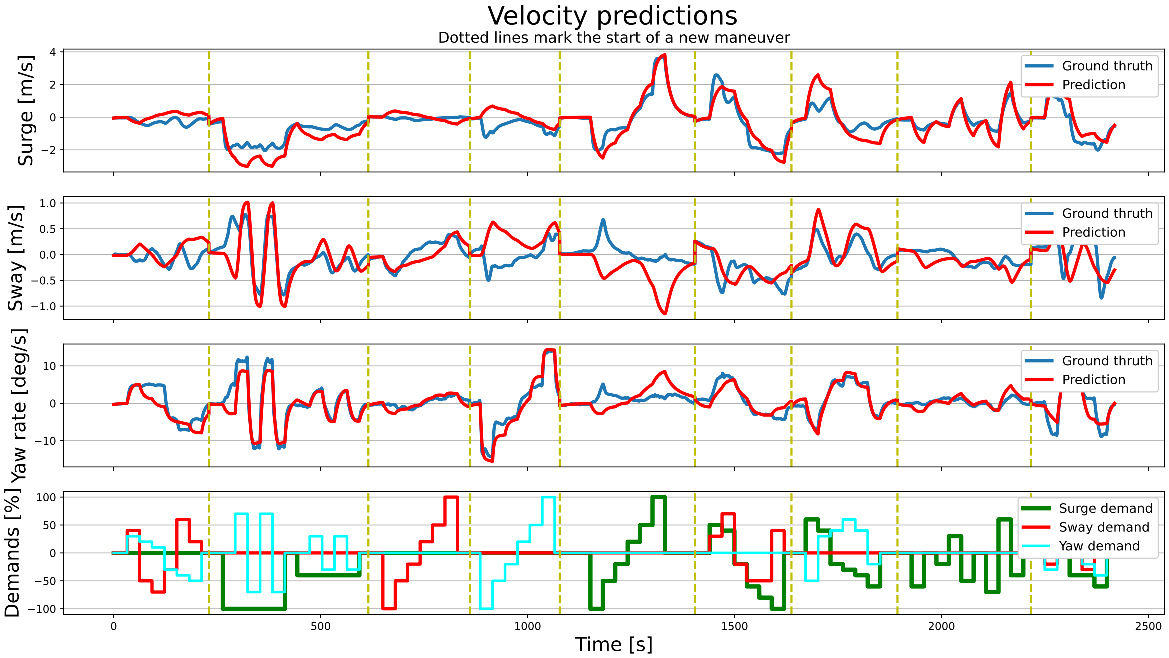


Figure 2.13: Predicted velocity on the test set for the raised Stinger model.

**Performance, Accuracy, and Validity of the Models**

Both models seem to be fairly accurate. They capture the dynamics in surge and yaw well with few large offsets, though they at times may miscalculate the (lack of) impact from the other demands. And the Stinger-induced asymmetry covered in section 2.3.1 gives yaw errors for the submerged Stinger model during some of the zig-zag movements, i.e. when the USV reverses with alternating yaw demands. The errors in sway, however, are larger and more frequent for both models. This is expected since the models are not designed for large movements in sway, but rather vessels with mostly forward surge speed and yaw rate, and little speed in sway.

The models, which are taken from [13], are displacement models. Displacement vessels move quite slowly through the water and have a large underwater profile, while their counterpart, planing vessels, ride almost on top of the water, reaching much greater speeds [21]. Whether a vessel is a displacement or planing vessel can be found by calculating the Froude number [22]. Frigg has been estimated to be a displacement vessel at velocities up to 4 m/s, a planing vessel at velocities above 9 m/s, and a semidisplacement vessel between the two. In the recovery task, we expect the USV to mostly operate at low velocities. However, the model is designed for vessels moving forwards, while we are also interested in capturing the dynamics of the USV when reversing. The model is symmetric in surge, in the YZ-plane, but the USV certainly is not. The hull of the USV is vastly different in the bow and in the stern, especially with the Stinger submerged, which inevitably will give different dynamics when reversing, compared to going forward. So while the model is trained on experimental data and seems to make fairly accurate predictions also when the USV is reversing, we must expect it to have some flaws and inaccuracies, as the structure of the model is not entirely valid when reversing. It would be desirable to have a model that captures the different dynamics of the USV for forward and reverse surge speed, as well as the change in dynamics when the Stinger is submerged, which gives a much more hydrodynamic complex structure. However, the author has not been able to find other work regarding this in the literature, meaning that basing the models on [13] was the best option available when developing the models.

## 2.5 Existing Control System

The last part of the preliminary work concerned the development of a proof-of-concept control system that could, under certain easy conditions, perform the recovery task in the simulator. In the preliminary work, this meant in the absence of performance-challenging elements such as measurement noise, time delay, and environmental disturbances. The control system was developed in Python, using ROS to communicate with the simulator. It is presented here since it will be used for testing in section 3 and development in section 4.

The control system has two main parts: A simple path planner that, based on the pose of the AUV, designs waypoints (WPs) for the USV to follow, and a set of controllers that enable the USV to reach them. We will here present the structure of the control system and the reasoning behind its design, the values of the controllers' gains, and the performance in the simulator.

### 2.5.1 Waypoints and Path Planning

The path of the USV was designed as shown in fig. 2.14, consisting of six waypoints. This path ensured that the USV kept a certain distance from the AUV to avoid affecting its pose when navigating to WP 5. From there, it would reverse into the AUV to try and recover it. Note that there are many possibilities for the path, and the chosen one is not necessarily the optimal one.

### 2.5.2 Path Following Controllers

The path following performed by following the waypoints was roughly divided into three stages:

1. Starting behind and far from the AUV, navigate the USV around the AUV (WP 1 to 4).
2. Position the USV accurately in front of the AUV (WP 5).
3. Slowly and accurately reverse the USV to have its stern touch the nose of the AUV (WP 6).

with different requirements for each stage, they also used different controllers, in a *gain-scheduling*-inspired approach.

### 2.5.3 Stage 1: Navigating Around the AUV

For the first stage, WP 1 to 4, we are mainly concerned with reaching the destination quite quickly, without affecting the pose of the AUV. For this stage, surge and sway are combined into a joint state, giving us a simplified 2-DOF system. Our controller consists of a heading autopilot and a throttle controller. Yaw demands are used to control the former, surge demands control the latter, while we

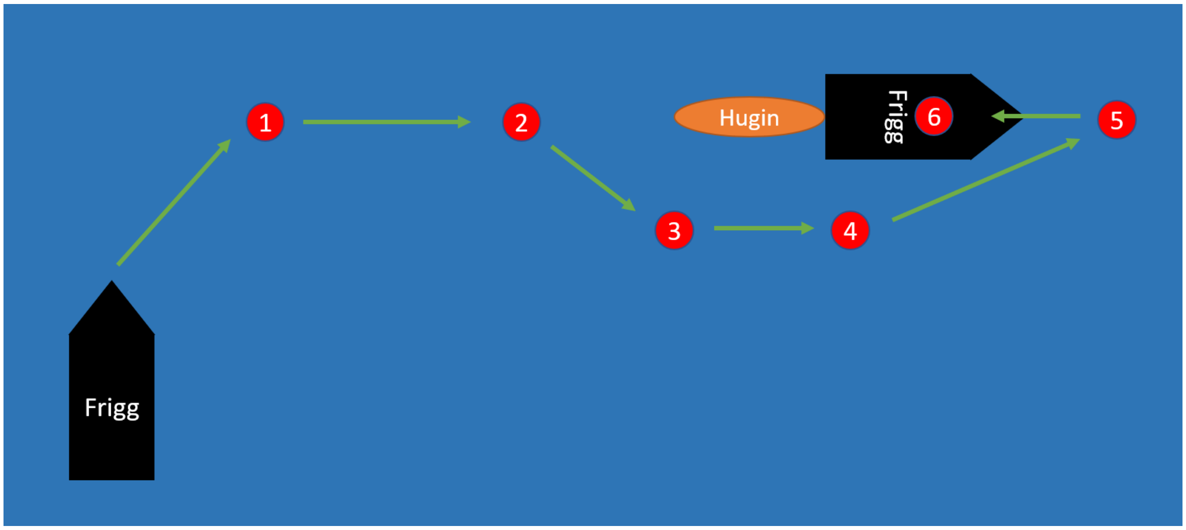


Figure 2.14: Visualization of the proposed waypoints for executing the recovery task.

use no sway demands. The heading reference is the angle from the current position to the current WP position. Our controller is defined as:

$$\begin{aligned}
\boldsymbol{\tau}_{ref} &= \begin{bmatrix} max(0, K_{p,u}||\mathbf{e}_p|| - K_{p,e_\psi}|e_\psi|) \\ 0 \\ K_{d,\psi}\dot{e}_\psi + K_{p,\psi}e_\psi \end{bmatrix} \\
\mathbf{e}_p &= \mathbf{p}_{wp} - \mathbf{p}_{usv} \\
\psi_{ref} &= arctan2(\frac{e_{p,y}}{e_{p,x}}) \\
e_\psi &= ssa(\psi_{ref} - \psi) \\
\dot{e}_\psi &= \dot{\psi}_{ref} - \dot{\psi}
\end{aligned} \tag{2.4}$$

where $\mathbf{e}_p$ is the positional error between the current WP and the USV, $\psi_{ref}$ denotes the heading reference, $e_\psi$ the heading error and $\dot{e}_\psi$ its derivative. As the heading reference varies very slowly, we here assumed $\dot{\psi}_{ref} \approx 0$ as the term could induce some unwanted oscillations on the controller when switching signs. The throttle controller uses the heading error as a damping term to make the control system initially use little throttle while reducing the heading error, and then use more throttle once the USV is headed towards the WP. Not including this damping term could for certain initial conditions yield a USV that would circle around the WP or spiral slowly towards it. The *max*-function ensures that the throttle is non-negative. *arctan2* is the *arctan*-function that returns the angle in the right quadrant, while the *ssa* (smallest signed angle) function maps the error $e_\psi$ between $[-\pi, \pi)$, such that the USV will take the smallest turn to reach the heading reference. The values for the gains are shown in table 2.4.

Note that the controller gives out a set of desired forces $\boldsymbol{\tau}_{ref}$, which needs to be converted to demands $\boldsymbol{\tau}_d$. This is done using a non-linear least squares solver [23] that minimizes the objective function

$$\boldsymbol{f}_{obj}(\boldsymbol{\tau}_d) = (\boldsymbol{\tau}_{ref} - \boldsymbol{f}(\boldsymbol{\tau}_d))^2 \tag{2.5}$$

| **Gain** | **Value** |
|---|---|
| $K_{p,u}$ | 100 |
| $K_{p,e_\psi}$ | 15000 |
| $K_{p,\psi}$ | 10000 |
| $K_{d,\psi}$ | 50000 |

Table 2.4: Gain values for the controller in stage 1.

with respect to $\boldsymbol{\tau}_d$.

When the distance between the USV and the waypoint is smaller than a chosen threshold, we move on to the next waypoint.

### 2.5.4 Stage 2: Positioning the USV Accurately in Front of the AUV

For WP 5, higher accuracy is needed than for the first 4 WPs. Since we have limited possibilities of inducing pure sway demands without also inducing too much water flow, as explained in section 2.1.2, the controller in stage 3 must be given an as easy task as possible. This means lining the USV up with the AUV, such that they have the same heading angle and close to zero offset in sway (in the BODY frame). To reach WP 5 we will utilize two controllers. The first one will steer the USV slowly and accurately towards the WP, and the other controller will fine-tune the position and line up the USV to have the same heading as the AUV, such that the USV can reverse towards it in a straight line. For the first controller, we use PD-controllers to control throttle and heading, defined as

$$
\begin{aligned}
\boldsymbol{\tau}_{ref} &= \begin{bmatrix} K_{d,u}\|\dot{\mathbf{e}}_p\| sgn(\Delta_{||\mathbf{e}_p||}) + K_{p,u}\|\mathbf{e}_p\| \\ 0 \\ K_{d,\psi}\dot{e}_\psi + K_{p,\psi}e_\psi \end{bmatrix} \\
\dot{\mathbf{e}}_p &= -\begin{bmatrix} u \\ v \end{bmatrix} \\
\Delta_{||\mathbf{e}_p||} &= ||\mathbf{e}_{p,k}|| - ||\mathbf{e}_{p,k-1}||
\end{aligned}
\tag{2.6}
$$

where the sign function $sgn(\Delta_{||\mathbf{e}_p||})$ compares the distance from the WP in the current and previous time step to decide if the USV is moving closer to or further away from the WP. The heading reference and error are defined as in eq. (2.4).

When sufficiently close to WP 5, we switch to the second controller, which is a 3-DOF controller. This means now controlling each of the states surge, sway, and yaw with their own PD-controller, on the form:

$$
\begin{aligned}
\boldsymbol{\tau}_{ref} &= \begin{bmatrix} -K_{d,u}u + K_{p,u}e_{u,body} \\ -K_{d,v}v + K_{p,v}e_{v,body} \\ K_{d,\psi}\dot{e}_\psi + K_{p,\psi}e_\psi \end{bmatrix} \\
e_\psi &= ssa(\psi_{auv} - \psi) \\
\dot{e}_\psi &= \dot{\psi}_{auv} - \dot{\psi} = -\dot{\psi}
\end{aligned}
\tag{2.7}
$$

where the position in the BODY frame is used for the error, and the heading of the AUV, which is assumed to be constant, is our heading reference. For WP 5, strict thresholds are used for both the position and heading error. When both of these are met, we move on to WP 6 and stage 3 of the path following. The controller gains are shown in table 2.5.

| Gain | Value |
|---|---|
| $K_{p,u}$ | 60 |
| $K_{d,u}$ | 800 |
| $K_{p,\psi}$ | 12000 |
| $K_{d,\psi}$ | 50000 |

| Gain | Value |
|---|---|
| $K_{p,u}$ | 80 |
| $K_{d,u}$ | 3000 |
| $K_{p,v}$ | 50 |
| $K_{d,v}$ | 30 |
| $K_{p,\psi}$ | 10000 |
| $K_{d,\psi}$ | 50000 |

Table 2.5: Gain values for the two controllers in stage 2.

### 2.5.5 Stage 3: Reaching the AUV

For the final stage, the USV is lined up, as accurately as possible, in front of the AUV. While the first two stages were considered the easy ones, the final stage is the hardest where high accuracy is required. To approach WP 6 (the AUV), the controller defined in eq. (2.6) is used, with the same gain values. As the USV approaches the AUV, we also switch to the submerged Stinger model to simulate the dynamics when the Stinger is submerged. Following the results of section 2.1.2, for WPs 3 to 6 the demands are saturated to [-60, 60] for surge, [-30, 30] for sway, and [-60, 60] for yaw to minimize the induced water flow. Note that the yaw demands are slightly outside of the minimum throttle use range of [-50, 50], which was chosen because the latter gave slower yaw dynamics than desired. For WP 1 and 2, the demands are unconstrained.

An overview of the thresholds required to move on from one WP to the next is displayed in table 2.6.

| Waypoint | Thresholds | |
|---|---|---|
| | Position Error [m] | Heading Error [deg] |
| 1 | 30 | - |
| 2 | 15 | - |
| 3 | 10 | - |
| 4 | 10 | - |
| 5 | 1.2 | 2.9 (0.05 rad) |
| 6 | - | - |

Table 2.6: Thresholds for moving on to the next waypoint. Note that only WP 5 has a requirement for the heading error.

### 2.5.6 Control System Performance

Figure 2.15 shows the path of the USV when performing the recovery task in the simulator. We see that the USV does not go through its first WPs, which is expected since the system changes to the next WP when close enough to the current one. While the control system is simple, it is within the accuracy threshold of ± 20 cm and thus manages to accomplish the recovery under the easy operating conditions.

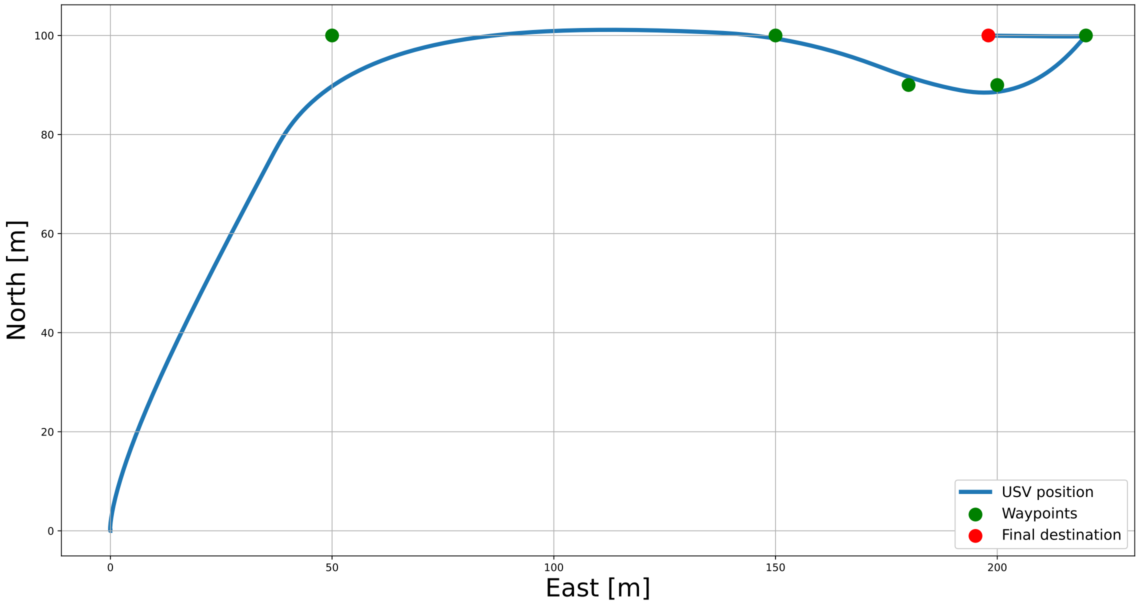


Figure 2.15: North-East path for the USV for simulation of the recovery task.

# 3 Improving the Simulation Environment

A simulator will always, to some extent, be different from its physical counterpart. This means that a controller that is optimal in the simulator is in general not optimal for the real system, a problem that grows with the difference between simulator and physical environment. This chapter focuses on reducing this gap between the existing simulator environment from the preliminary work and the physical environment. This will be done by adding *extensions*, which are elements that exist in the real system but were not present in the simulations in the preliminary work. The extensions covered in this section are:

- Delay
- Transient response
- Measurement noise
- Environmental disturbances

The purpose of the section is to model and implement these extensions to get a more realistic simulation environment, and develop measures to counteract them such that we get a robust control system that is better suited to perform the real-world recovery.

To test the extensions, we will use the control system presented in section 2.5. While not necessarily an ideal control system, or close to the final version, it accomplishes the recovery task without any "external challenges", and can be used to see the effect of the added challenges and control system functionality. Lastly, each of the added extensions should function as a stand-alone element. This means that each extension, both the element itself and the control system functionality for handling it, should be able to be turned on and off independently of the other extensions. Each of the extensions will be covered in its own subsection, which includes:

- A presentation of the problem
- The presence of the problem in the real system
- Modeling and implementing the problem in the simulator
- Investigating how the problem affects the control system
- Discussion and/or implementation of measures to counter the effects arising from the problem

## 3.1 Delay

Our system, like any real control system, is affected by time delay, meaning that desired demands are not instantly in effect. First, the control system must receive the latest measurements, calculate the desired thrust demands, and send them to the water jet interface. Then, the interface must receive and process the message, before lastly setting the desired values for the actuators, i.e. the water jets. Finally, it takes some time before the actuators reach their new desired values. Time delay in the system degrades the performance of the control system and can potentially destabilize it. Thus, it is necessary to investigate its presence in the real system, how it affects the performance of the control system, and implement measures to reduce its effect. The time delay is typically divided into two parts: The pure *delay*, the time it takes from a set of measurements are gathered to when the actuators start changing, and the *transient time*, the time from the actuators start changing to when they have reached their desired value. The delay is covered here, while the transient time is covered in section 3.2.

### 3.1.1 Presence in the Real System

First, we will examine how the real system, the USV and its components, are affected by delay. This is necessary in order to get realistic values when implementing delay in the simulator, and will help us implement measures that can account for delay when performing the physical tests. Note that the thrust demands are used as control inputs, but the values that actually affect the USV are the three different values of the water jets: RPM, nozzle, and bucket. Thus, we must inspect how these are affected by delay. From plot inspection, the differences between the port and starboard water jets were very small, so they will be modeled equally. For the plots in this subsection, we will look at the port water jet, and inspect the response to both small and large demand changes.

**RPM Delay**

To examine the delay on the RPM (throttle), we will see how it responds to surge demand changes (since the demands are sent simultaneously, it does not matter which of them, surge, sway, or yaw, we use). Figure 3.1 shows the response for a step from 0 to -100 in surge demand, where there is a 0.5 second delay, and the response for a step from 60 to 70 in surge demand, where there is a 0.4 second delay. The yellow lines mark the time when the demand is changed, and the time when the RPM starts changing.

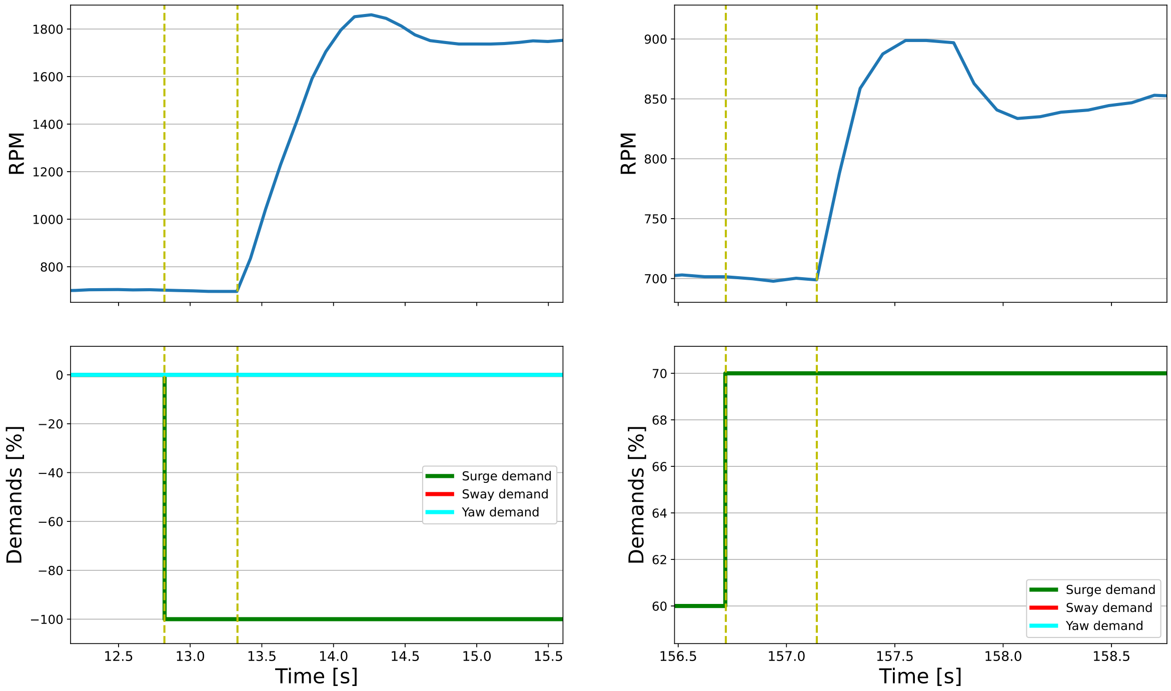


Figure 3.1: Time delay for the RPM for two different surge demand changes.

**Nozzle Delay**

To examine the delay on the Nozzle, we will see how it responds to yaw demand changes. Figure 3.2 shows the response for a step from 0 to -100 in yaw demand, where there is a 0.3 second delay, and the response for a step from 20 to 30 in yaw demand, where there is a 0.4 second delay.

**Bucket Delay**

To examine the time delay on the Bucket, we will see how it responds to surge demand changes.

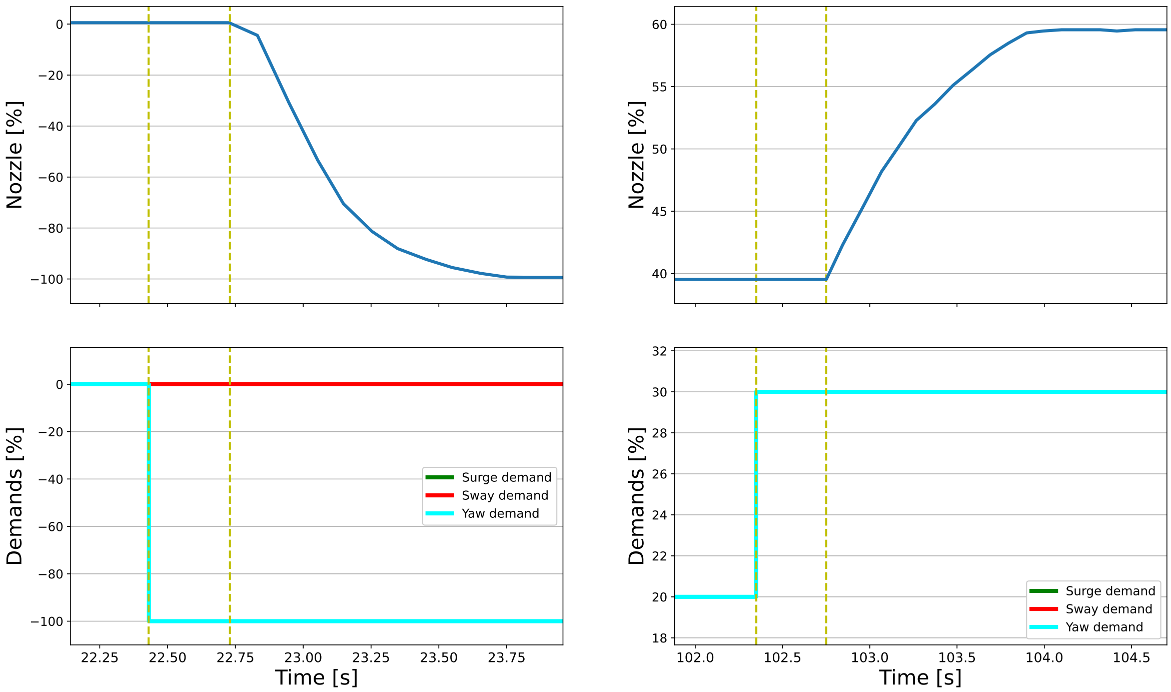


Figure 3.2: Time delay for the Nozzle for two different yaw demand changes.

Figure 3.3 shows the response for a step from 0 to -100 in surge demand, where there is a 0.4 second delay, and the response for a step from 60 to 70 in surge demand, where there is a 0.3 second delay.

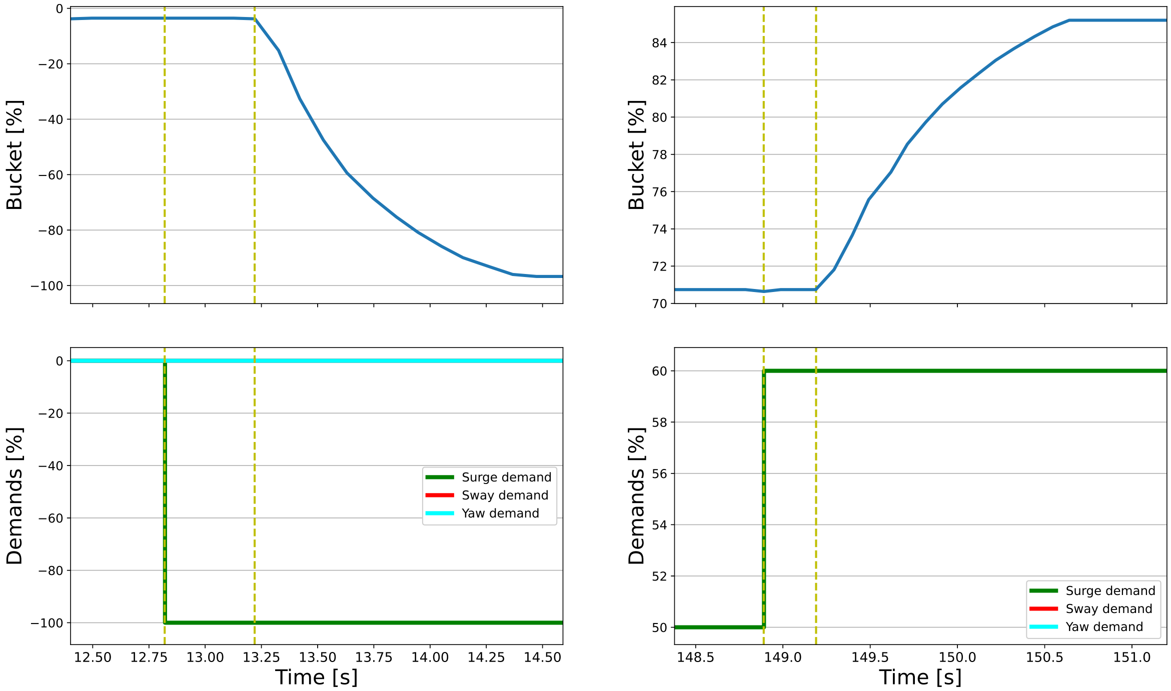


Figure 3.3: Time delay for the Bucket for two different surge demand changes.

### 3.1.2 Modeling the Delay

We saw in section 3.1.1 that the three water jet inputs had delays between 0.3 and 0.5 seconds. The differences do not seem to depend on the magnitude of the demand change, and could be due to coincidental variations. For simplicity, and due to the demands often combining the inputs, we will treat these as having the same delay. We choose the "middle ground" and get a delay of 0.4 seconds for the demands.

As mentioned, the delay found was the time between a demand is sent to when the actuators start changing. But the measurements must also be sent and received, and the control system must calculate the new control inputs. We assume each of these to use one time step each, meaning that the delay is expected to be an additional 0.2 seconds, 0.6 seconds in total. Note that this additional delay occurs naturally in the simulator as well, and thus already exists.

In the simulator, the delay is implemented by designing the control input variable as a queue. This queue has shape 3x(N+1), where N is the delay given in time steps. At each iteration, the calculated desired demands are appended to the end of the queue, and the first element in the queue is popped and used as the current demands (control inputs). This also enables us to easily change the delay, if desired.

### 3.1.3 Impact on the Control System Performance

Having designed and implemented the delay model, we can now test how it affects the performance of the control system, before introducing measures to counter it. In fig. 3.4, the performance with and without delay is compared. Though the USV still reaches each waypoint and is within the required error threshold of ± 20 centimeters when it makes contact with the AUV, with delay there is chattering on the yaw demands and oscillations for the heading errors. The controller has to react to the constantly delayed measurements, and as a result, it has a lot of rapid changes. With delay and relatively high gains in the damping terms in the original control system, the yaw demands are almost always at the minimum or maximum allowed values. Note that the AUV position and the final waypoint are not the same since the position of the USV is defined somewhere on the boat, whereas the endpoint of the Stinger will make contact with the AUV. Though the error is small enough for the USV to be able to accomplish the recovery, the delay gives several challenges and unwanted behavior that may be even more challenging in combination with other elements, inducing a need for measures to be taken to counteract the effects from the delay.

### 3.1.4 Measure: State predictor

While the problems arising from the delay are decreased accuracy, an oscillating heading response, rapidly changing demands, and increased wear and tear, the main cause that induces these is that our current states (measurements) are outdated when their corresponding desired control inputs are set. In other words, the control inputs in effect at time $t + dt$ were calculated using the states at time $t$, with $dt$ being the delay. One solution to the problem is to at time $t$ use the state values at time $t + dt$. This means predicting the future. A way to (try to) do this is to use our existing model to predict the coming states. The performance of this predictor will then depend on the accuracy of the model: If perfect, the model will 100 % correctly predict the states at time $t + dt$, but if it is inaccurate we will get a bad estimate that might even give worse values than we would have by simply using the measured values at time $t$. This predictor is very similar to the Smith Predictor [24], whose purpose is to control systems with a feedback delay.

In the simulations, the prediction is easy, as the predictor can be designed to work (almost) perfectly if we use the same model as the simulator does. But since the model will be imperfect compared to the real system, we must investigate how well the predictor performs for different model accuracies. This is done by adding adjustable inaccuracy gains in the Euler integration when predicting the next

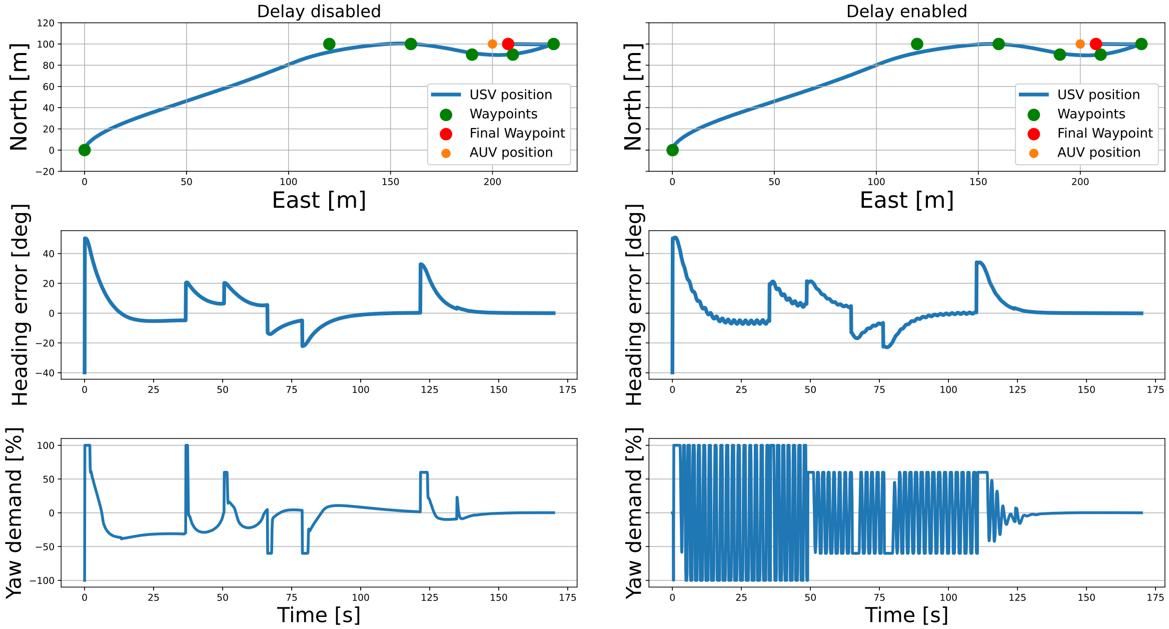


Figure 3.4: Control system performance with (right) and without (left) delay

states, the pose and velocity, which is for this purpose modified to:

$$
\begin{aligned}
\boldsymbol{p}_{k+1} &= \boldsymbol{p}_k + \Delta t \boldsymbol{R}(\psi_k)\boldsymbol{\nu}_k \\
\boldsymbol{\nu}_{k+1} &= \boldsymbol{\nu}_k + \Delta t \mathbf{M}^{-1}(\alpha_1 \boldsymbol{f}(\boldsymbol{\tau}_k) - \alpha_2(\mathbf{C}(\boldsymbol{\nu}_k)\boldsymbol{\nu}_k + \mathbf{D}\boldsymbol{\nu}_k))
\end{aligned}
\tag{3.1}
$$

where $\alpha_1$, $\alpha_2$ are the adjustable gains for the accuracy of the hydrodynamics and the control inputs of the USV. $\alpha_i = 1$ gives a perfect prediction from its corresponding component, $\alpha_i = 0$ disables the component, and other values for $\alpha_i$ sets the model component imperfection, which grows as $\alpha_i$ deviates from 1. $\Delta t$ represents the time step of the system, and in this interval the control inputs are assumed to be constant. With a frequency of 10 Hz (section 2.4), $\Delta t = 1/10 = 0.1s$. Thus, the prediction steps must be repeated several times, depending on the length of the delay. Note that the equation for updating the pose does not directly contain any inaccuracy gains, but will still be affected by the model inaccuracy as the update step depends on $\boldsymbol{\nu}_k$.

Figure 3.5 shows recovery simulations with the predictor using a perfect model, and for a slightly imperfect model with $\alpha = \begin{bmatrix}0.7, & 1.4\end{bmatrix}$. The latter values were chosen to mimic that the model is quite accurate, but not perfect. We see that the former gives almost the same results as with no delay, while the latter also yields a satisfactory response. The imperfect model, as the perfect one, counteracts the unwanted effects from the delay. The average use of yaw demand is much smaller than without a predictor, and there are seemingly no oscillations on the heading error. The predictor has also been tested with worse prediction accuracy parameters. As expected, the predictor performs worse as the parameters in $\alpha$ deviate from 1, but still, those predictors also counteracts the delay effects somewhat.

### 3.1.5 Discussion

The delay is an unwanted element that introduces unwanted effects to the system. With delay, the path of the USV is still quite accurate, but the use of yaw demands is much larger and suffers from chattering, as well as the heading and its error oscillating. And when adding the transient response,

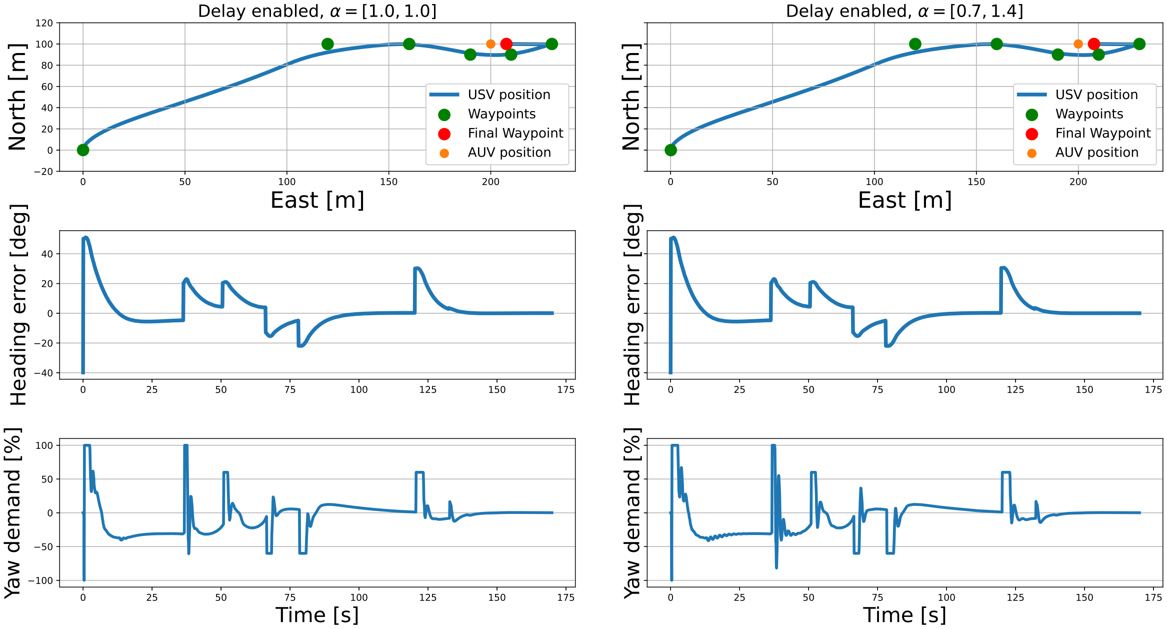


Figure 3.5: Recovery simulation with a predictor, using a perfect model (left) and one with a slightly imperfect model (right).

such that the controller cannot instantly react to sudden changes, we expect the path of the USV to be less accurate as well. While unwanted, the delay is present in the real system and cannot be directly removed. To counteract the effect of the delay, a predictor was implemented. This predictor reduced the heading oscillations and the large yaw demand use the delay induced, even with a slightly imperfect model.

It should be noted that the rapid changes in the yaw demands are somewhat due to large D-term gains in the controllers, and lowering them should also be considered in the coming control system development. A re-tuning of the controller could have been a just as effective measure. [25] argues that the benefits of a Smith Predictor are exaggerated and that you should forget about it and ”*instead use a well-tuned PI or PID controller*”. While the author's experience was that the predictor had a positive impact, this is something to keep in mind, and perhaps remove the predictor and re-tune the controllers, should we experience that its performance is not satisfactory. A drawback of the predictor is that its performance depends on the accuracy of the model, but a somewhat inaccurate model has also seemed to have a positive impact on the performance. Going forward, the slightly imperfect predictor with $\alpha = \begin{bmatrix} 0.7, & 1.4 \end{bmatrix}$ will be used in the simulations to mimic our model being imperfect, while we will use $\alpha = \begin{bmatrix} 1.0, & 1.0 \end{bmatrix}$ for the physical tests in section 5.

## 3.2 Transient Response

When the actuators have received a new reference value for them to follow, the new value is not reached instantly. Due to e.g. physical limitations and control design, the actuator values will have some transient response before reaching the new desired value. This means that even after the delay, a new set of control inputs will not affect the system instantly, but will gradually have more and more impact.

### 3.2.1 Presence in the Real System

As mentioned in section 3.1, the values that actually affect the USV are the three different values of the water jets, RPM, nozzle, and bucket. And they all work differently. Thus, to be able to model the transient response on the demands, we must learn both how the different demands use the water jet inputs, and the transient response of the water jet inputs. As in section 3.1, we will model the two water jets equally, and will look at the port water jet.

From the figures in section 3.1.1, we can see that all the water jet inputs have a transient response that resembles a first-order step response. But while the two responses are very similar for both the nozzle and the bucket, the two RPM transients have different durations.

Figure 3.6 shows how the surge demands use the water jet inputs. We see that the RPM is unchanged in the range [-60, 60], the nozzle is never changed, and the bucket is only changed in the range [-60, 70].

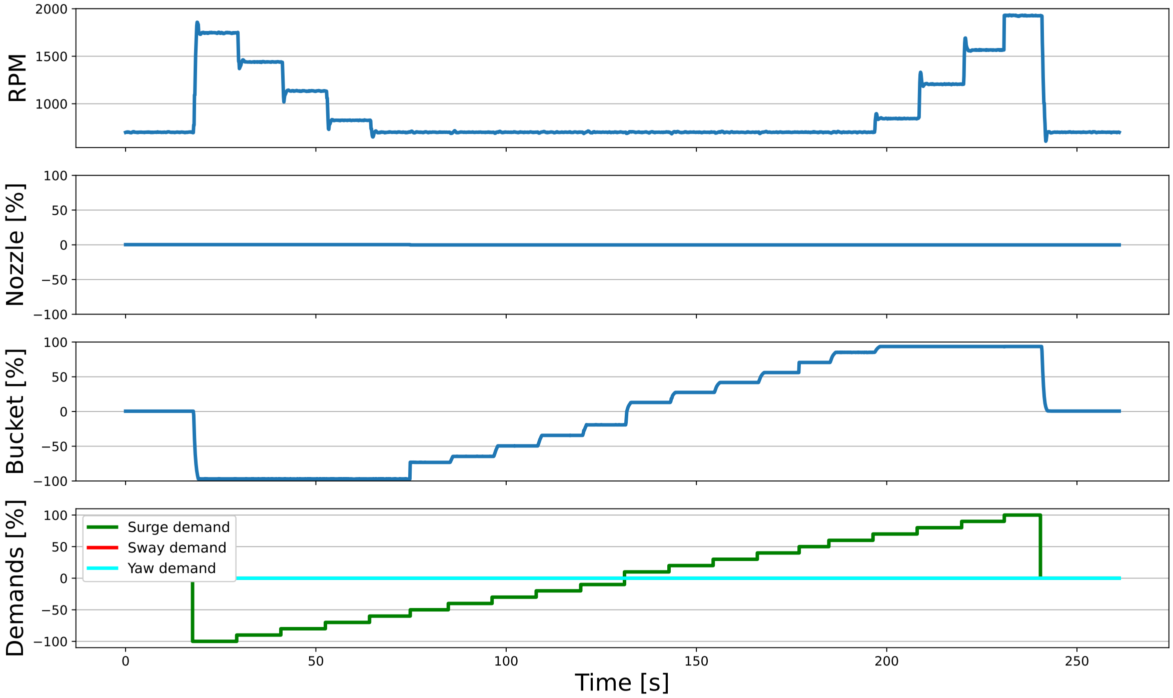


Figure 3.6: Use for water jet inputs for various surge demands.

Figure 3.7 shows how the sway demands use the water jet inputs. We see that the RPM is unchanged in the range [-30, 30], the nozzle is only changed in the range [-30, 40], and the bucket is only changed in the range [-30, 40].

Figure 3.8 shows how the yaw demands use the water jet inputs. We see that the RPM is unchanged in the range [-50, 50], the nozzle is only changed in the range [-50, 50], and the bucket is never changed. We do not have any data for yaw demands between (-100, -50] and [50, 100), but as argued in section 2.1.2, we expect the RPM to change everywhere outside of the range [-50, 50].

### 3.2.2 Modeling the Transient Response

The goal of this subsection is to model the transient response of the Thrust mode system. To do this, we need knowledge about the transient response of the water jet inputs and how they are used by

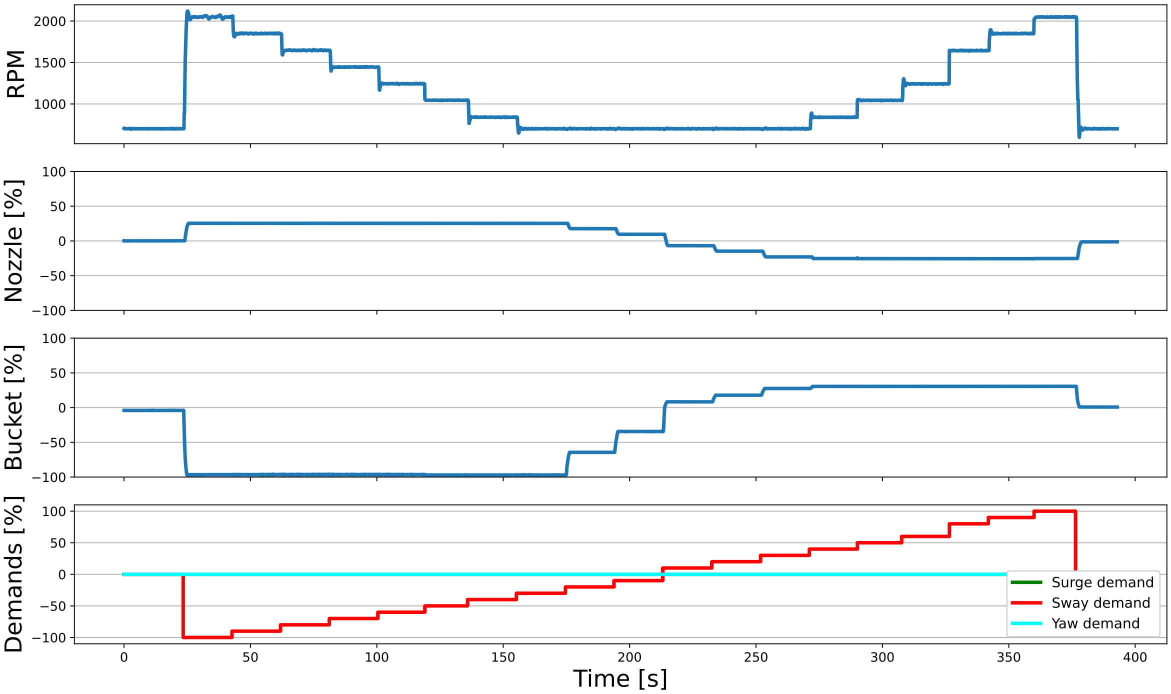


Figure 3.7: Use for water jet inputs for various sway demands.

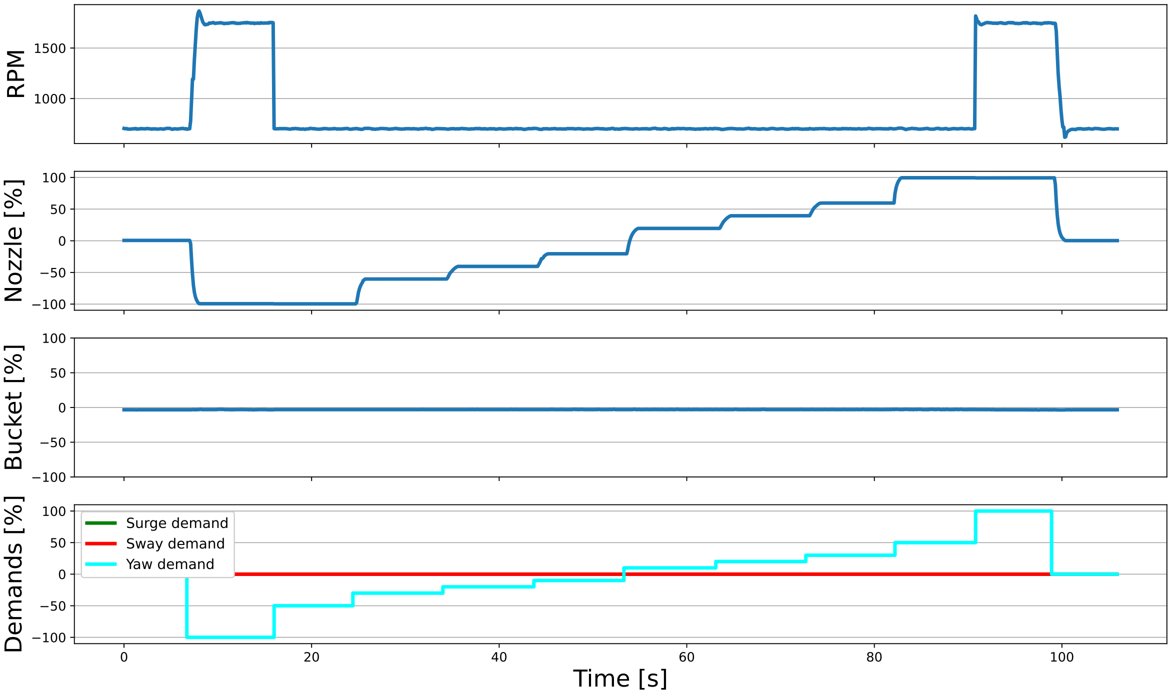


Figure 3.8: Use for water jet inputs for various yaw demands.

the thrust mode system. All of this is information that the manufacturer, *HamiltonJet*, for business purposes, is reluctant to share. Thus, to develop a model of the transient response, we will look at the raw data and use reverse-engineering to approximate a model that works well. Using the knowledge from section 3.1.1, we will try to model the transient response of the water jet inputs as a first-order step response, which is defined as

$$f(t) = K(1 - e^{-\frac{t-t_0}{T}}) + f_0, \quad t \leq t_0 \tag{3.2}$$

where the step K is the setpoint change (e.g. from 1000 to 1500 rpm), $T$ is the time constant, $t_0$ is the start time, and $f_0$ is the initial value. With this model, the actuators will, at first, change more rapidly as the setpoint increases. But due to physical limitations, there will be some finite limits to how fast they can change, which we will investigate by finding the time constant for different setpoint changes.

For the RPM, the time constant for setpoint changes of max 300 was found to be 0.15 seconds. For larger setpoint changes the time constant increased with the setpoint change, indicating that the RPM change had met its limitations and could not follow the same first-order response of the smaller changes. Inspecting the largest setpoint change in the dataset, going from 700 to 1925 rpm for 0 to 100 in surge demand, the rpm increased by ca. 175 in the first 0.1 s, with 0.1 s being the time step of our system. However, eq. (3.2) with T = 0.15 s and K = 1225 gives an increase of 584 rpm in the first 0.1 s. Thus, it is evident that the RPM change has saturated, with 175 rpm per 0.1 seconds being the largest possible change. For the bucket and nozzle, the time constant was found to be between 0.5 and 0.7 seconds for "any" reference change and did not seem to be saturated for larger demands. However, the largest available reference change was 100, meaning that the bucket and nozzle might be saturated, e.g for a -100 to 100 reference change where they would go from their minimum to maximum value (or opposite).

Next, we must model the active demands. By active, we mean the set of demands that affect the system during the transient response. Due to its complexity, some assumptions and simplifications will be made. The different demands might use several water jet inputs simultaneously, and they have different responses. We will use the slowest of the (changing) inputs when calculating the active demand. Also, we assume a one-to-one relation between input change and active demand change, meaning that if the slowest input change by x %, so does the active demand as well. Even when saturated, the RPM still has a quicker response than the nozzle and bucket. Thus, we can use the RPM response when only it is changing, and the bucket or nozzle response otherwise. The last simplification is to treat each demand as an individual, uncoupled component, meaning that e.g. the surge demand transient is calculated independently of the change in the sway and yaw demands, even though they are likely to affect each other in some way. The transient response algorithm for calculating the active demands is defined in algorithm 1.

To find the maximum allowed demand change, we convert the largest allowed RPM change and multiply by the demand change/total rpm change ratio, and get that the largest rpm-allowed change in active surge demand is

$$maxChange_{surge} = \frac{(100 - 0) * 175}{1225} = 14.3 \tag{3.3}$$

meaning that for each time step, the active surge demand cannot change by more than 14.3. We assume this to also be the case for sway and yaw, and that the *minChange* has the same magnitude, but with different sign.

**Algorithm 1** Calculate Active Thrust Demands

```
Input prevDemands, desiredDemands
  $step \leftarrow desiredDemands - prevDemands$
  $activeDemands \leftarrow \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$
  for $i = 0, i < 3$ do
      if only RPM changes then
          $demandChange \leftarrow step[i] * (1 - e^{\frac{-\Delta t}{T_{rpm}}})$
          $demandChange \leftarrow saturate(demandChange, minChange, maxChange)$
      else
          $demandChange \leftarrow step[i] * (1 - e^{\frac{-\Delta t}{T_k}})$        ▷ time constant of either nozzle or bucket
      end if
      $activeDemands[i] \leftarrow prevDemands[i] + demandChange$
  end for
  return activeDemands
```

### 3.2.3 Impact on the Control System Performance

The performance of the control system, with and without transient response, is shown in fig. 3.9. The yaw demands now follow a smoother trajectory that physically is more realistic. The addition of transient response alone does not seem to affect the performance much.

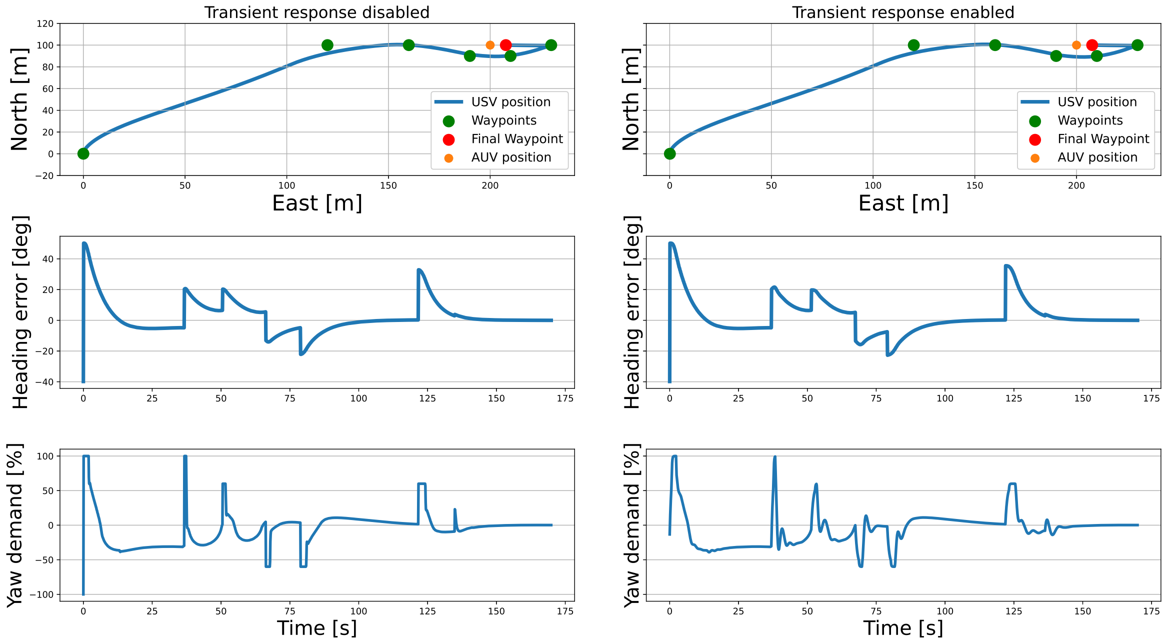


Figure 3.9: Control system performance with (right) and without (left) transient response.

However, when adding delay as well to the simulations, the effects of the transient response become evident. In fig. 3.10, the system has delay enabled and uses the slightly imperfect predictor from fig. 3.5, and is tested both with and without transient response. With the transient response, the control system struggles a lot more, and as a result, has oscillations in both the heading error and yaw demands. It still manages to reach all the waypoints, but the increased use of demands is undesirable, and in combination with other challenges the system may not be able to accomplish the recovery task.

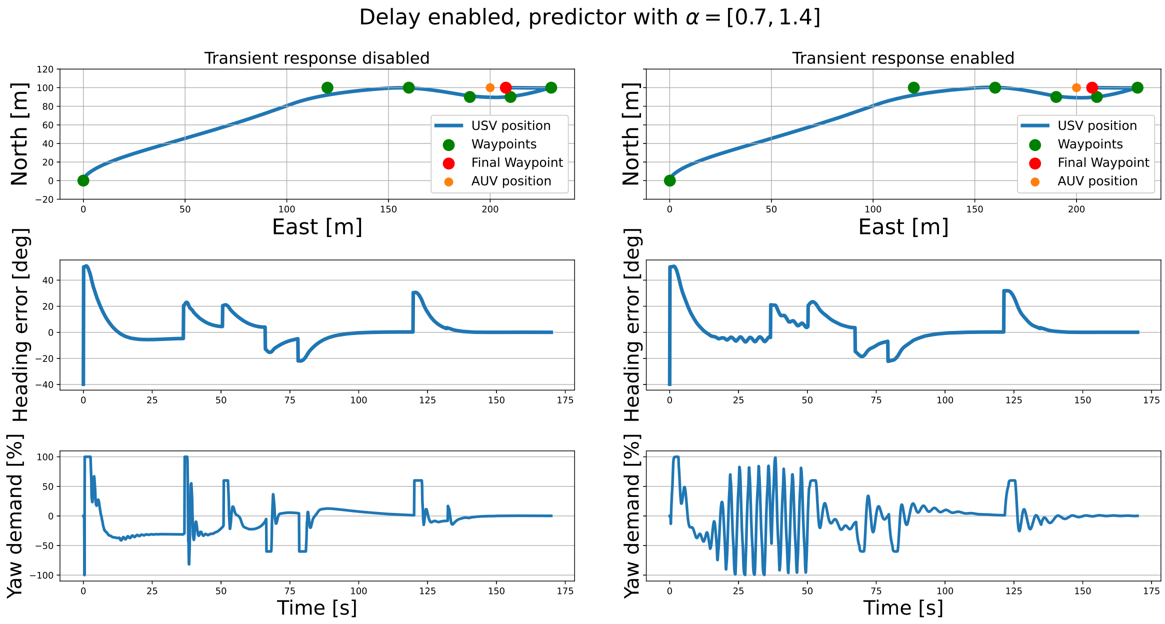


Figure 3.10: Control system performance with (right) and without (left) transient response, with delay enabled and a slightly imperfect predictor.

### 3.2.4 Measure: Re-tune controller gains

Due to the transient response only proving to be a problem in combination with delay, we will develop a measure that, in combination with the predictor, handles the problem occurring from time delay, i.e. delay *and* transient response.

From fig. 3.10 it seems that the controller overreacts. As in section 3.1.3, the delay causes a larger error to which the controller reacts harshly, but with the transient response, it cannot instantly set the desired (active) demands, causing oscillations in the heading error and yaw demand. To dampen these, we will re-tune the gains in the heading controllers to give a slower and more ”soft” response. We did not fine-tune the gains, but rather lowered the gains in all the heading controllers in section 2.5 to $K_{p,\psi} = 5000$, $K_{d,\psi} = 20000$ to test the general effect from smaller gains. Lowering the gains gave improved results. A comparison between the original and the re-tuned control system is shown in fig. 3.11, where the oscillations are almost non-existent for the re-tuned system.

### 3.2.5 Discussion

Like the delay, the transient response is an unwanted, but irremovable, element in the system. The model of the transient response is a simplified one, but one that hopefully is fairly accurate, though its accuracy is difficult to test. While the transient response alone did not notably affect the performance of the control system, it degraded the performance of the control system in combination with delay. The transient response staggers the ability to instantly react to errors, which again led to oscillations in both heading error and yaw demands when adding delay in the simulations.

The simple measure of re-tuning the heading controller, while also using the slightly imperfect predictor, seems to be effective. The oscillations were almost removed, and the accuracy in the recovery phase was still high. A drawback with the new gains is that the heading error is in general larger for the first waypoints. However, since the accuracy is not that important in this phase, this is deemed acceptable. It should be noted that the new gains are not optimal. They were found quickly by trial and error,

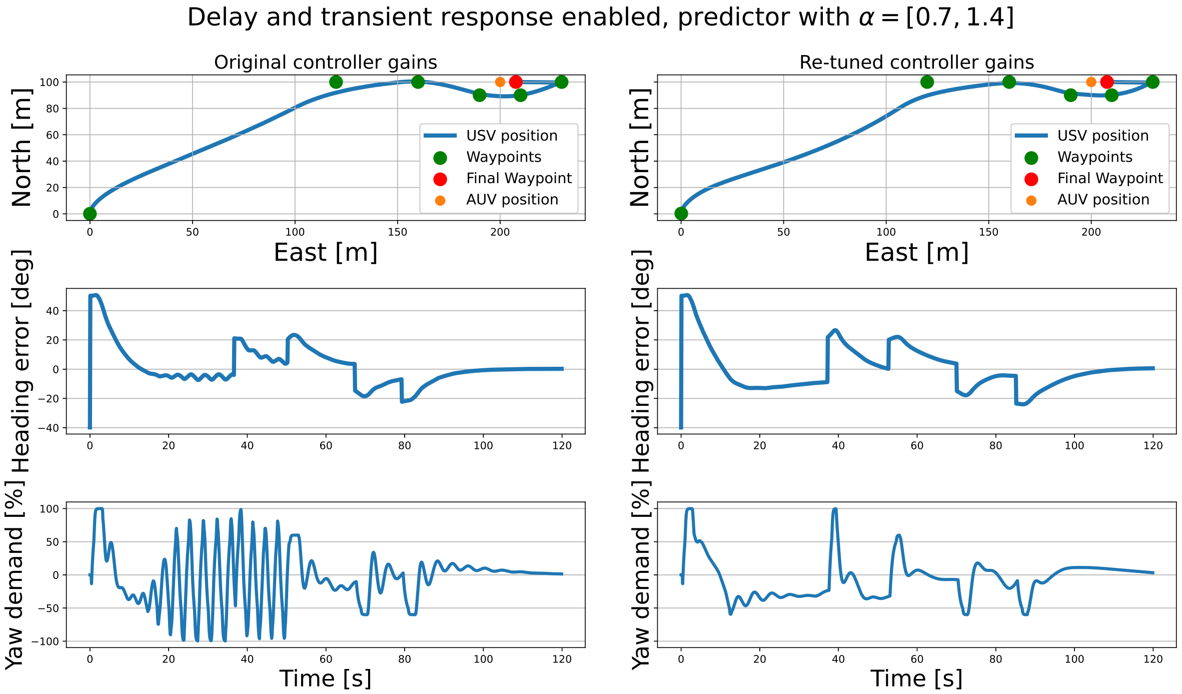


Figure 3.11: Control system performance with time delay enabled, with the old controller gains (left) and the re-tuned ones (right).

with the main goal being to show that a re-tuning could be effective. Also, the error and demands in surge are not covered here or in section 3.1 as they were not notably affected by neither the delay nor the transient response.

## 3.3 Measurement Noise

While noise was neglected in the simulations in the preliminary work, the sensors on the USV will inevitably be affected by noise to some extent. Too much noise may destabilize a controller, induce oscillations in the system, and increase wear and tear on the actuators. As a consequence of this, we must examine the presence of noise on the sensors, see how it affects the control system, and, if necessary, implement measures to prevent the unwanted noise-induced effects. The measurements we consider that may be affected by noise are for the pose and velocity, in all the three states. Using an existing model from FFI, section 3.3.2 is copied from [26].

### 3.3.1 Presence in the Real System

Two subsets of velocity measurements from the experimental data are shown in fig. 3.12. We see that the surge and sway measurements are not particularly affected by noise, with the noise on these having a magnitude that is no larger than 0.01 m/s. The sway and surge are rather affected by motion in roll and pitch. This motion can make the USV swing back and forth, causing some oscillations in surge and sway, with sway being affected the most. But unlike noise, these are real dynamics that are present in the system and can affect the performance of the control system. The yaw rate seems to be slightly more affected by noise, but its magnitude is not that large either.

The pose measurements for a subset of the experimental data are shown in fig. 3.13. Here it seems to be close to no noise at all. The sudden discontinuities in the heading are due to it being mapped

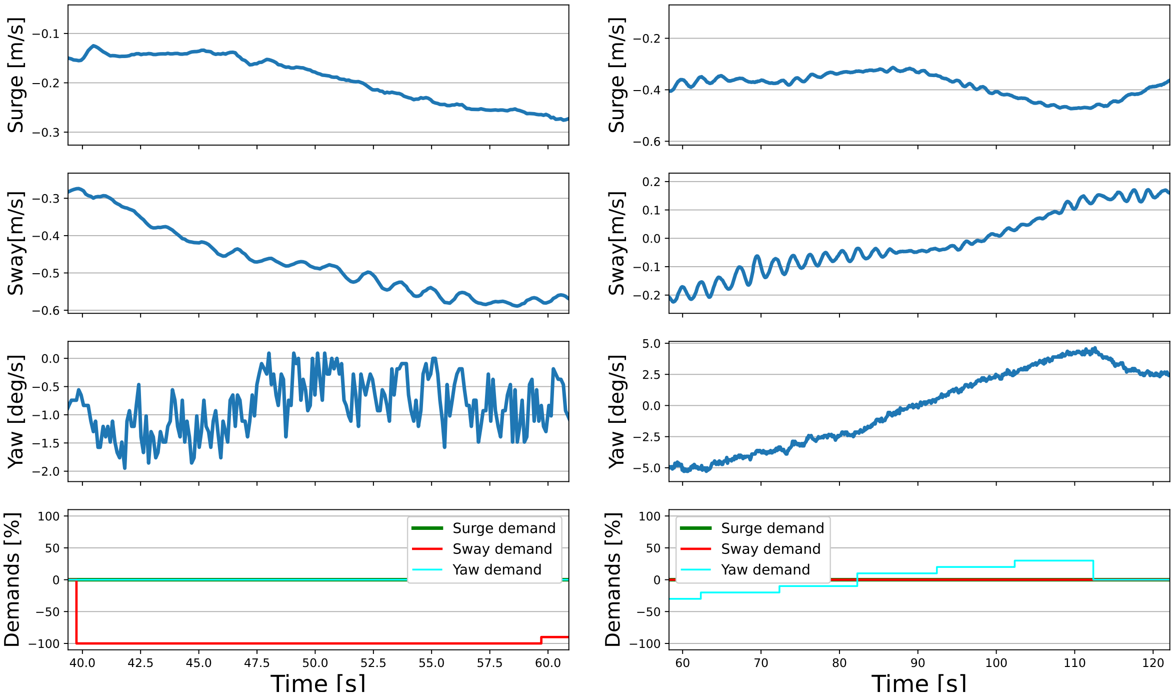


Figure 3.12: The presence of noise for two subsets of velocity measurements.

between [-180, 180) degrees.

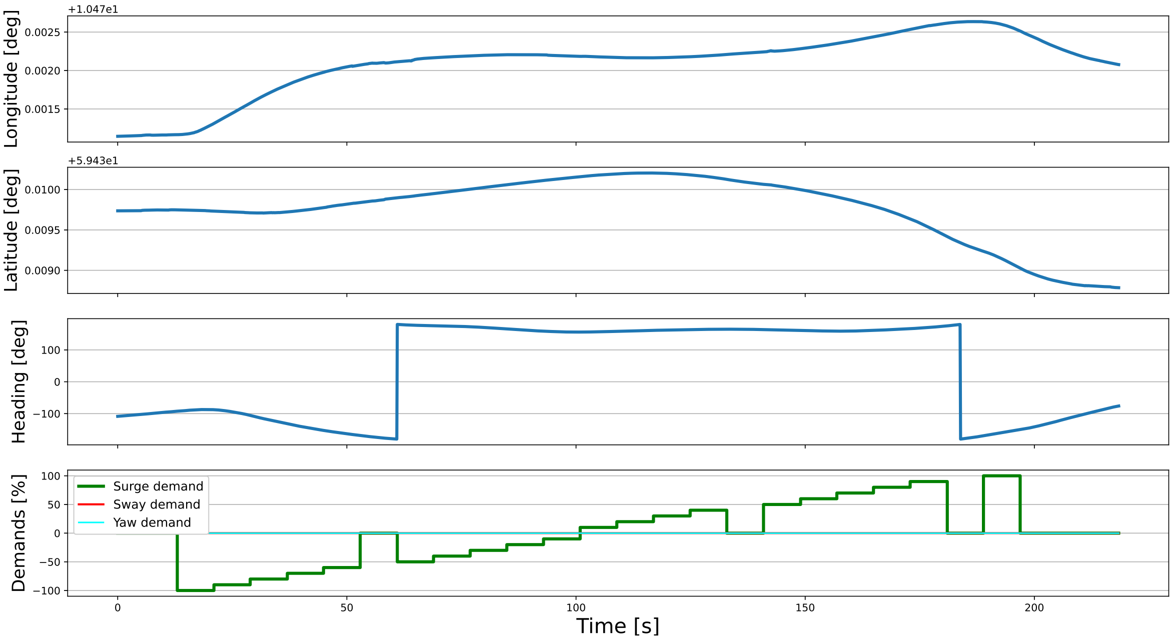


Figure 3.13: (Lack of) measurement noise on the pose.

The findings above agree well with the information given by FFI. Since the USV uses a high-quality inertial navigation system, the measured pose is expected to be very accurate. FFI could also inform that they have rarely cared about the noise in general, as it has never been much of a problem. They do, however, low-pass filter the velocity measurements in their speed controller.

### 3.3.2 Modeling the Measurement Noise

Despite the relatively small degree of noise, we will still implement and test the impact of noise in the simulator. Given the findings in section 3.3.1, we will only implement noise for the velocity measurements, and not the pose. FFI has an existing model for measurement noise in the simulator. In this model, noise is added as a combination of white noise and bias. This model is defined as:

$$\begin{aligned} b_i &= b_{i-1} + T_s \sigma_b w_k \\ n_i &= b_i + \sigma_w w_k \end{aligned} \tag{3.4}$$

where $b_i$ is the current bias, $T_s$ is the length of the time step, $\sigma_b$ and $\sigma_w$ are the standard deviations for the bias and white noise, and $w_k$ is the output from a Gaussian white noise process. We add noise to the velocity measurements of each of our three states, surge, sway, and yaw. Through experimentation, these coefficients were estimated such that the noise on the measurements in the simulator was similar to the noise in the real data. Since we in the experimental data have seen no signs of measurement bias, we assume the bias to be zero. The final values can be found in table 3.1, and the corresponding velocity measurements for the USV given no demands are shown in fig. 3.14.

| | $\sigma_b$ | $\sigma_k$ |
|---|---|---|
| **Surge** | 0 | 0.01 |
| **Sway** | 0 | 0.01 |
| **Yaw** | 0 | 0.002 |

Table 3.1: Noise model coefficients.

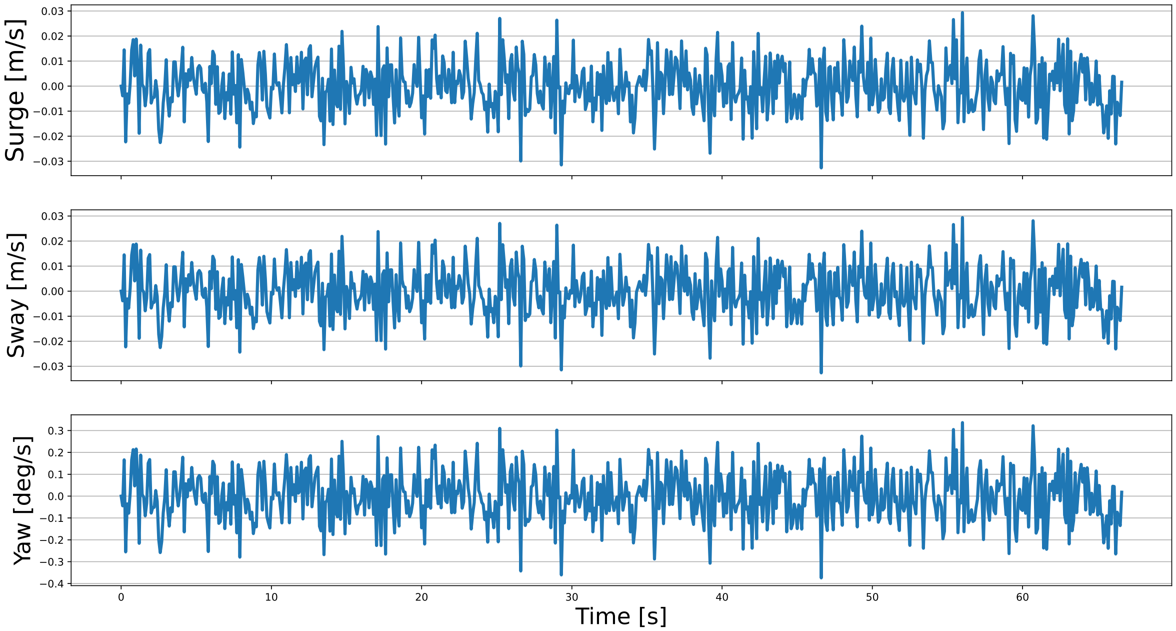


Figure 3.14: Velocity measurements in the simulator with the noise model enabled.

### 3.3.3 Impact on the Control System Performance

While not large, adding measurement noise has a notable impact on the control system. The noise directly affects the yaw demands, which now have loads of rapid changes, increasing the wear and tear on the actuators significantly. The control system is still accurate, but the use of yaw demand is not ideal and is expected to be a larger problem when combined with other challenges. The performance with and without noise is shown in fig. 3.15, where we have used the original controller gains.

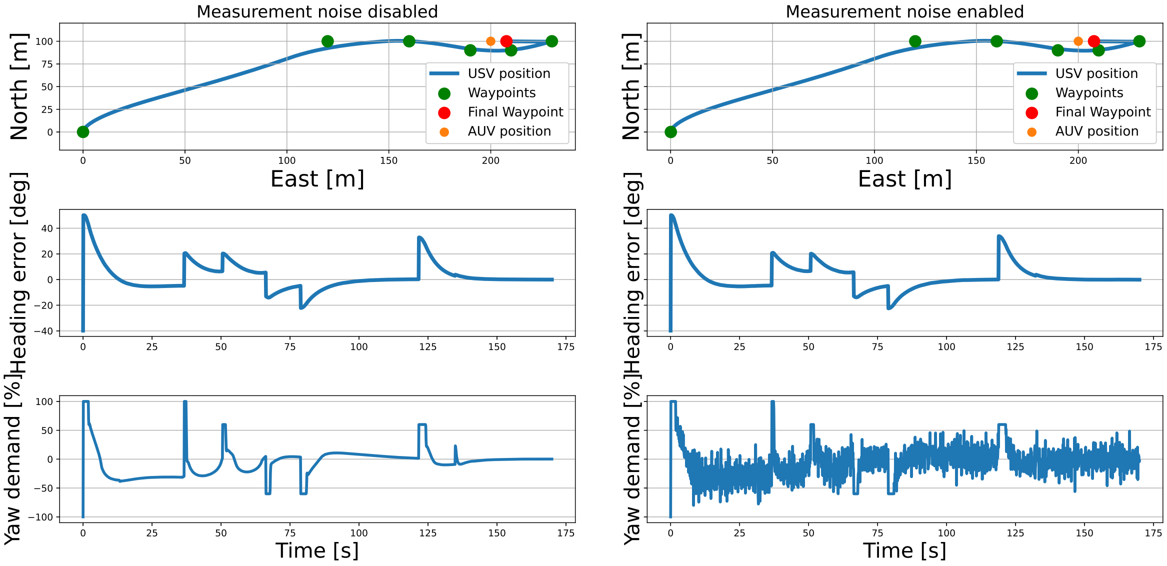


Figure 3.15: Control system performance with (right) and without (left) measurement noise.

### 3.3.4 Measure: Filtering the Measurements

A common procedure for removing noise is to low-pass filter the measurements. But the filtering also damps the signal somewhat and induces a delay (phase) on the measurements, which again gives the problems covered in section 3.1. The filter design decides the magnitude of time delay, but also how much the signal is damped and how well the noise is removed. This implies a trade-off when designing the filter, where we choose to minimize the damping and time delay, leading to the filter not removing all of the noise, but still a large part of it. While the surge and sway velocity measurements were assumed to be impacted by the same magnitude of noise, their average measured values are not the same. With the surge typically taking much larger values than sway, the surge is also relatively much less affected by noise. For surge, the filter-induced damping seemed to be a larger problem than the noise so we will not filter it, only sway and yaw. Figure 3.16 shows the original and filtered velocities, where we especially for the yaw rate see that it has a small delay and is slightly damped at its peaks.

As explained in section 3.2.4, the heading controller reacts quite harshly to heading errors. Changing to the new gains while still filtering the sway and yaw measurements suppressed the effects from the noise well. Figure 3.17 shows the performance with the old and re-tuned gains. We see that the low-pass filter alone significantly reduces the yaw demand oscillations compared to fig. 3.15. The new gains that slow down the heading controller further enhance this, but also yield slightly larger errors in the first 40 seconds.

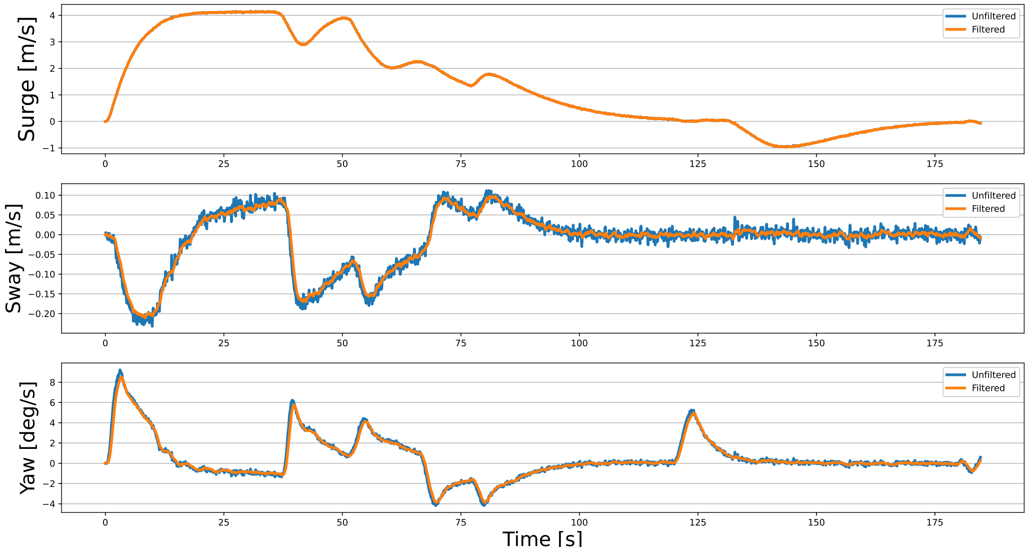


Figure 3.16: The filtered vs unfiltered measurements in the simulator.

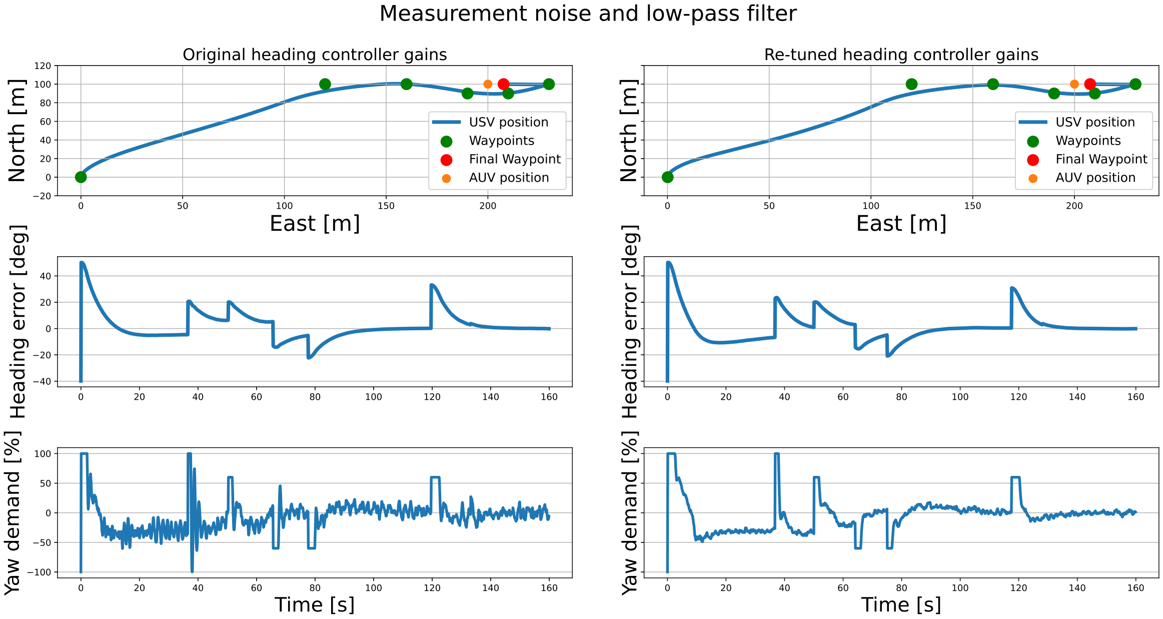


Figure 3.17: Control system performance with the original (right) and re-tuned (left) gains, when low-pass filtering the sway and yaw measurements.

### 3.3.5 Discussion

Measurement noise was almost non-existent for the pose measurements and quite small for the velocity measurements, but still had an impact on the control system. While introducing some unwanted effects, such as delay and signal damping, the low-pass filter seems to be an effective measure for handling the velocity measurement noise. The filter removes a large part of the noise, while the induced delay and signal damping are not too large. While improved, we still see that the yaw demands are affected by the noise. When changing to the heading controller gains from section 3.2.4, while still filtering the sway and yaw velocity measurements, the effects arising from measurement noise were further suppressed.

The extra delay from the filter could, like in section 3.1, be accounted for using a predictor. Another option is to remove the filter, as the change of gains alone also had a good effect on the performance, and an extra delay is in general unwanted. This depends on the actual magnitude of noise affecting the real system. The presence of a measurement bias, which we here have assumed to be zero, could also affect the system. A bias in the velocity measurements may lead to a constant error where the P and D terms pull equally in the opposite direction. This can typically be fixed by adding integral action to the controller.

## 3.4 Environmental Disturbances

By environmental disturbances, we mean forces and moments acting on our system that occur naturally and are outside of our control. At sea, the main environmental disturbances that act on the USV are wind, waves, and ocean currents. Using an existing model from FFI, the part of section 3.4.2 that concerns the modeling of wave disturbances is copied from [26].

### 3.4.1 Presence in the Real System

The environmental disturbances will always be present in the real system. Sometimes they can be small enough to neglect them, but at times they can also be large enough for the system to be greatly impacted by them. As stated in section 1.3, a requirement for the control system is that it should be able to handle some degree of environmental disturbances. Thus, we need to model and implement them in the simulator and test their impact and how well the control system handles them.

### 3.4.2 Modeling the Environmental Disturbances

**Wave disturbances**
For the wave disturbances, FFI has an existing wave model in the simulator. The waves are modeled as a second-order linear process with an integrator drift term, in each DOF, where each DOF is simplified to be uncorrelated. In the s-plane, the second-order process is given by

$$\begin{aligned} h(s) &= \frac{K_w s}{s^2 + 2\lambda\omega_n s + \omega_n^2} \\ y_{wave}(s) &= h(s) w_1 \end{aligned} \tag{3.5}$$

where $K_w$ is the amplitude of the waves, $\lambda$ and $\omega_n$ are the relative damping and natural frequency of the wave spectrum, and $w_1$ is a Gaussian white noise process. Wave drift forces are included by adding a slowly varying bias term

$$d(s) = \frac{1}{s} w_2 \tag{3.6}$$

where $w_2$ is a Gaussian white noise process. The forces and moments acting on the USV are then given by

$$\boldsymbol{\tau}^b_{wave} = \boldsymbol{R}^b_n \begin{bmatrix} y_{u,wave}(t) + d_u(t) \\ y_{v,wave}(t) + d_v(t) \\ y_{r,wave}(t) + d_r(t) \end{bmatrix} \tag{3.7}$$

where $y$(t) and $d$(t) are calculated using the bilinear transform from the s-plane domain. The coefficients of the wave model can be found in table 3.2, where the values are the default values defined by FFI in the simulator.

| | $K_w$ | $\lambda$ | $\omega_n$ |
|---|---|---|---|
| **Surge** | 1000 | 0.1 | 1.4 |
| **Sway** | 1000 | 0.1 | 1.4 |
| **Yaw** | 1000 | 0.1 | 1.4 |

Table 3.2: Wave model coefficients.

**Wind and ocean current disturbances**

For the wind and ocean currents, there is no existing model in the simulator, meaning that we must implement our own. Mathematical models of wind forces and moments are presented in [13]. However, these models require the wind coefficients for the vessel in question, which we do not have. Wind coefficients can be found through wind tunnel experiments, by using look-up tables such as [27], or approximated through experiments with the USV and its weather station and e.g. curve fitting techniques. But due to our limited access to the USV, and [27] mostly containing coefficients for much larger ships, these are not possible, and we will instead model both the wind and ocean currents similar to [28], where they are assumed to be a joint constant, irrotational force in the world frame. We design the disturbance as

$$\boldsymbol{\tau}_{dist} = \begin{bmatrix} X_c \\ Y_c \\ 0 \end{bmatrix} = \begin{bmatrix} Fcos(\alpha - \psi) \\ Fsin(\alpha - \psi) \\ 0 \end{bmatrix} \tag{3.8}$$

where F is the magnitude of the disturbance, $\alpha$ its direction, and $\psi$ the heading of the USV. Choosing F = 100 N gives a steady-state surge speed of 0.41 m/s when applied along the x-axis of the USV, and a steady-state sway speed of 0.06 m/s when applied along the y-axis of the USV. This was chosen to have an impact that was small enough to not dominate the behavior of the USV, but still large enough to have an impact. We will use this magnitude and an angle of 45 degrees relative to north, counter-clockwise, in the subsequent tests.

### 3.4.3 Impact on the Control System Performance

The performance of the control system with and without the environmental disturbances is shown in fig. 3.18. The path is very similar for the first WPs, so the figure focuses on the last WPs where the two simulations differ. When starting to reverse, the controller struggles to control the heading, which again affects the position, making the USV unable to come close enough to the AUV to accomplish the recovery task. We also see that the yaw demands no longer follow a (mostly) smooth trajectory, as it now has to constantly react to the disturbances acting on it, which is expected, and mostly stems from the wave forces.

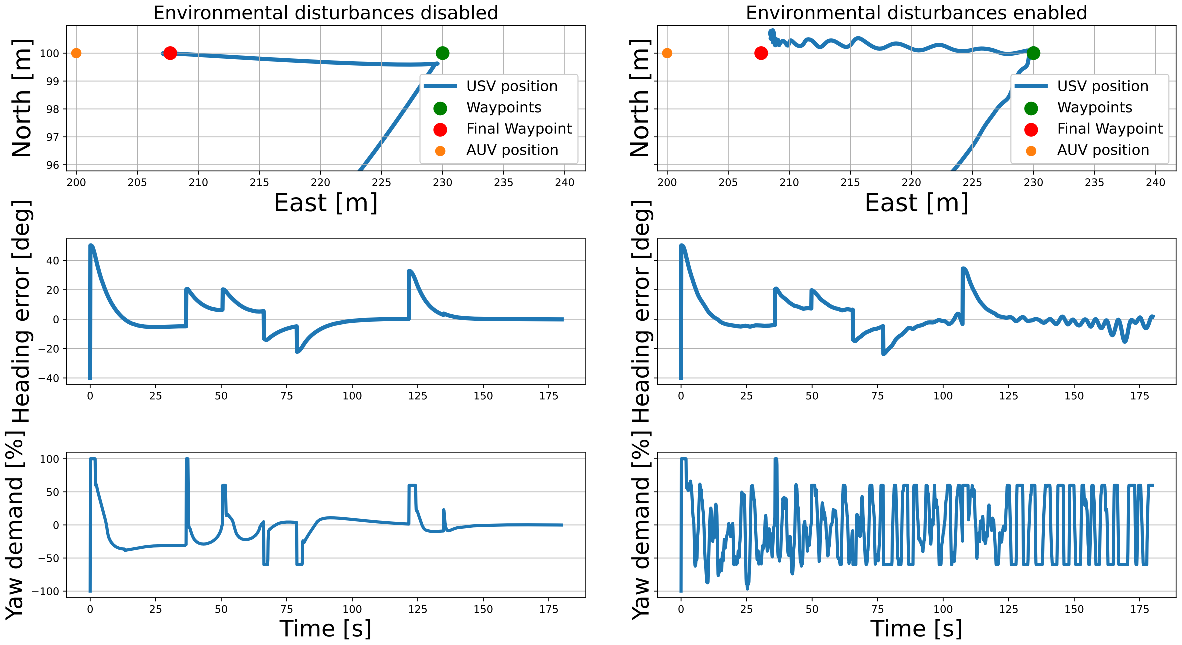


Figure 3.18: Control system performance for the last WPs, with (right) and without (left) environmental disturbances.

### 3.4.4 Measure: Revised strategy

The results from section 3.4.3 uncovered some vulnerabilities in the control system design and the USV dynamics that should be addressed. The chosen controller did not seem robust to environmental disturbances for the final WP. It struggled to control the heading, and as a result, the position was not as accurate as required in order to accomplish the recovery. However, the control system was still accurate when reaching WP 5 which also has strict accuracy requirements, which brings up another important topic: The (changing) yaw dynamics of the USV. While WP 5 and 6 have similar control and accuracy requirements, the yaw dynamics for the two cases are quite different. For WP 5 the USV has the Stinger still raised onto the USV while mainly using forward thrust, but for WP 6 the Stinger is submerged while the USV reverses towards the AUV. In the latter case, the yaw dynamics are far slower, degrading the controller's ability to quickly respond to errors. Another weakness of the control system is how the heading reference is calculated. The reference is found as the direction from the USV to the current WP, which works well when the WP is sufficiently far away. But as the USV gets very close to the WP the heading reference may change rapidly, especially when exposed to disturbances.

The environmental disturbances seem to be the biggest challenge so far, as they deteriorate the performance of the control system enough for it to fail to accomplish the recovery task. Thus, they may also require the most work in terms of measures to handle them. Possible measures for this is to include integral action, crab angle compensation, or a feed-forward term to account for the disturbances. However, it may be better to start by re-thinking the entire strategy of approaching the USV. This way we may reduce the impact of the environmental disturbances simply by solving the task in a smarter way, and then possibly add the suggested measures on top to further improve the robustness of the system. Since this is an extensive job it will not be covered here, but rather as a part of section 4.

### 3.4.5 Discussion

The environmental disturbances acting on the USV, wind, waves, and ocean currents, are forces and moments that will always be present to some extent in the real system. To investigate their impact, a simple model for the wind and currents combined, and an existing one for the waves, were used. These models, especially the one for wind and currents, are simplified models that are not necessarily representative of the true impact, but they do function well as external disturbances acting on the USV. There is also a lot of uncertainty regarding the magnitude of the disturbances. With the values in table 3.2, the wave model may affect surge and sway too little, and yaw too much. The wind and ocean currents model affects surge more than sway due to the hydrodynamic coefficients being larger for sway than surge. Thus, it probably resembles the impact from wind more than ocean currents, as the latter would for instance yield the same steady-state velocity for both states. But, if we assume that the AUV and the USV will both be affected almost equally by ocean currents, we are more interested in modeling the impact from wind forces anyway.

The addition of environmental disturbances had a large impact on the control system. The accuracy was good for the first part of the task, but the control system struggled a lot for the final waypoint, to the point where we did not manage to execute the recovery task. This seemed to be due to a combination of the yaw dynamics being more sensitive when reversing and having submerged the Stinger, weaknesses in calculating the heading reference, an imperfect recovery strategy, and the control system in general not being robust enough to the disturbances. Solutions to this will be further investigated in section 4.

## 3.5 Full Test

Having implemented the four extensions, we will now test the control system with all the extensions, and their corresponding measures, simultaneously. This means with delay, transient response, measurement noise, and environmental disturbances as extensions, and the predictor, low-pass filter, and re-tuned controller gains as measures.

The control system performance with all extensions and measures, compared to for none, is shown in fig. 3.19. Although it follows the first WPs fairly well, we see that the heading control is much worse. At WP 5, it is no longer accurate enough for it to move on to WP 6, as the wind makes the USV drift away from WP 5 while trying to align its heading with the AUV. The new gains seem to have suppressed the chattering issue quite well, but this may also just be because the error is rarely close enough to zero for the controller to start chatter. A re-tuning of the controllers could also have improved the performance of the control system.

## 3.6 Summary and Discussion

In this section, we have added several extensions to the simulation environment, in order to make it more similar to the physical system. We have added delay and transient response to the control inputs, wind, wave, and ocean currents as external disturbances acting on the USV, and noise to the velocity measurements. The modeling and implementation of these are not perfect, but they work and are good at testing the performance and robustness of the control system, as well as learning more about the system and the strengths and weaknesses of our strategy.

The measures we implemented seemed to be effective at suppressing the problems arising from the extensions, such that the performance deterioration was not too large. However, they were not enough alone to get the desired accuracy for the recovery phase. Thus, further improvements to the controllers, path planning, and general strategy are needed, which will be the focus of section 4.

Lastly, other extensions could be introduced to the system as well, given more time. Examples of these are to model Hugin to have a non-constant pose that is affected by e.g. collisions with the

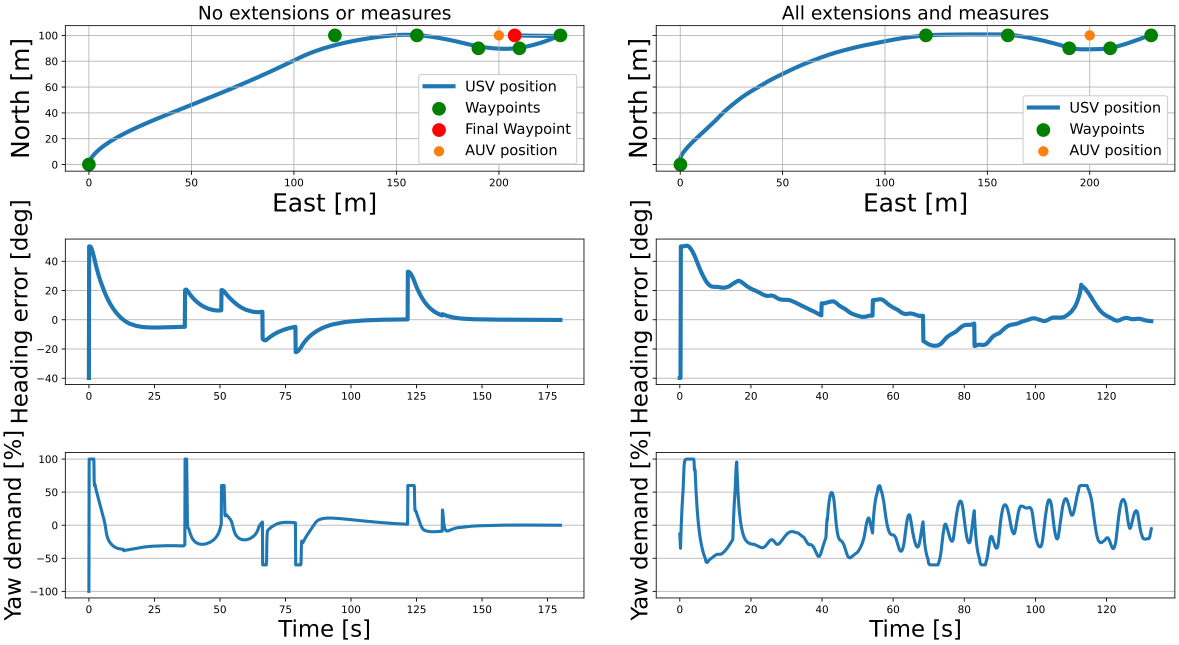


Figure 3.19: Control system performance with all extensions and measures enabled (right) and with none enabled (left).

USV or extending the model to also include how roll and pitch motion influence surge and sway. The ones that were implemented were chosen due to their relevance and fairly straightforward modeling and implementation, while the other ones were expected to be harder and more time-consuming to implement. Having a model of how the USV and the water-jets affect the pose of Hugin could be very useful, but could not be done in this thesis since we had no access to Hugin. Hugin would also be affected by the environmental disturbances. While the impact from waves and ocean currents is likely to be somewhat similar on both the USV and Hugin, the impact from wind is expected to be quite different for the two.

# 4 New Control Systems Development

Having implemented the added challenges in section 3 and seen how the USV and existing control system respond to them, we will now develop different control systems that can accomplish the recovery task. With the main focus and challenge being the reversing parts of the recovery, parts 2 and 3 in section 1.2.1, this will also be the primary goal when developing the new control systems. When testing their performances in the simulator we will use all the extensions from section 3, delay, transient response, measurement noise, and environmental disturbances, such that the simulator environment is closer to the physical system. We will refer to this as normal operating conditions. The pose of the AUV will be constant in the simulations, since we have not modeled its dynamics. Due to the limited time windows where physical tests could be carried out, and the iterative nature of control system development, it was not possible to address all challenges that arised during testing.

The section first reviews the main challenges that our control systems must address, before covering each of the developed control systems. Finally, their performances are compared and discussed.

## 4.1 Main Challenges

From our previous knowledge of the system, the findings from section 3, and the differences between simulations and the real world, there are some key challenges that the new control systems must overcome. These are:

- Water jet model inaccuracy
- Parameter tuning
- Robustness to disturbances
- Strategy for performing the actual recovery

### 4.1.1 Water Jet Model Inaccuracy

Recall the USV model, eq. (2.1), where the inertia matrix $\mathbf{M}$ takes the form

$$\mathbf{M} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix} \tag{4.1}$$

where m is the mass of the USV, $x_g$ is the distance from the center of origin to the center of gravity along the x-axis, and $I_z$ is the moment of inertia around the z-axis.

The acceleration of the USV is directly dependent on the inertia, or rather its inverse. This means that if you scale the inertia matrix by some constant $k$, you must also scale the other components by $k$ to keep the same acceleration. Thus, the values in the transfer function $\boldsymbol{f}(\boldsymbol{\tau}_d)$ that converts thrust demands to forces are linearly correlated to the values of $\mathbf{M}$, and if $\mathbf{M}$ is wrong, so will the values of $\boldsymbol{f}(\boldsymbol{\tau}_d)$ be. The value of $I_z$ is especially uncertain. Its value was estimated by looking at the angular acceleration when the yaw rate was close to zero, such that the control inputs were (almost) the only major force acting on the USV, and was estimated as:

$$I_z = \frac{\sum_i \tau_i}{\ddot{\psi}} \approx \frac{\boldsymbol{f}_\psi(\boldsymbol{\tau}_d)}{\ddot{\psi}} \tag{4.2}$$

where the torque was based on the forces from the water jets in x and y. This estimate is very uncertain, making the accuracy of $\boldsymbol{f}(\boldsymbol{\tau}_d)$ in yaw uncertain as well.

Our controllers will operate in the force domain. This means that the desired forces must be converted to the actual control inputs, the demands, which is done using (the inverse of) $\boldsymbol{f}(\boldsymbol{\tau}_d)$. But if the model is inaccurate, the resulting performance of the control system will also degrade as it will not be able to find the demands that yield the desired dynamics from the USV. Bad estimates in $\mathbf{M}$ may give model inaccuracies, and any model will inevitably be imperfect to some extent. Thus, the control systems should be able to handle some degree of inaccuracy for the water jet model.

### 4.1.2 Parameter Tuning

The controllers in our system will contain a number of tunable parameters that can be adjusted to optimize the system performance. While good estimates for these values can be obtained by tuning in the simulator, it is not certain that these values are optimal for the actual system, and may need to be re-tuned in the physical tests. Since the physical test time is very limited, we cannot use much, if any, of this time to tune parameters, and should limit the number of tunable parameters in the control system, as well as the need for re-tuning. This may also imply a trade-off between simplicity/robustness and the performance of the system. This problem is somewhat related to the previous, as we e.g. expect well-tuned parameters to be able to account for some model inaccuracy.

### 4.1.3 Robustness to Disturbances

The largest problems occurring from section 3 were the ones due to the presence of environmental disturbances. The effects from the disturbances made the USV unable to reach WP 6, the AUV, as accurately as required, and thus failed to accomplish the recovery. The robustness of the control system must be increased such that it can handle these disturbances.

### 4.1.4 Recovery Strategy

In the preliminary work and section 3, we have mainly considered the general dynamics and movements of the USV. By this, we mean movement forwards and backwards while changing the heading, mostly by having the USV follow different waypoints. For the last part of the recovery, however, when connecting to the AUV, a much more specific movement is required. To attach the Stinger to the hook of the AUV, it must be approached with the correct angle and force, and perhaps have a bit of luck to get the timing correctly. This gives a need to design a clever strategy for how we approach the AUV to maximize the chances of accomplishing the recovery.

## 4.2 Control System 1: Upgrading the Existing Control System

The control system from the previous work, while simple, had a relatively solid performance in most of the simulations in section 3. Here, we will use this control system, the measures and lessons from section 3, and implement some further upgrades to, in little time, get a control system that hopefully is capable of accomplishing the recovery task.

Since it is likely that the USV often will fail to connect to the AUV, the control system must be designed to repeatedly attempt to connect to the AUV, readjust its position if it fails, and retry to connect. We will design this as a state machine, as shown in fig. 4.1. We start from an initial position and navigate to a rendezvous (RV) point corresponding to WP 5 in fig. 2.14. From there, we can repeatedly attempt the recovery until successful. In fig. 4.1, state 1 corresponds to the initial phase, state 2 to the recovery phase, state 3 to the phase after a failed recovery, and state 4 to a successful recovery.

### 4.2.1 Path Planning and Guidance

The path planning in section 2.5 works fine, but under the conditions that we start behind the AUV, and that the pose of the AUV is constant. Now, we want to improve it to be more generalized and

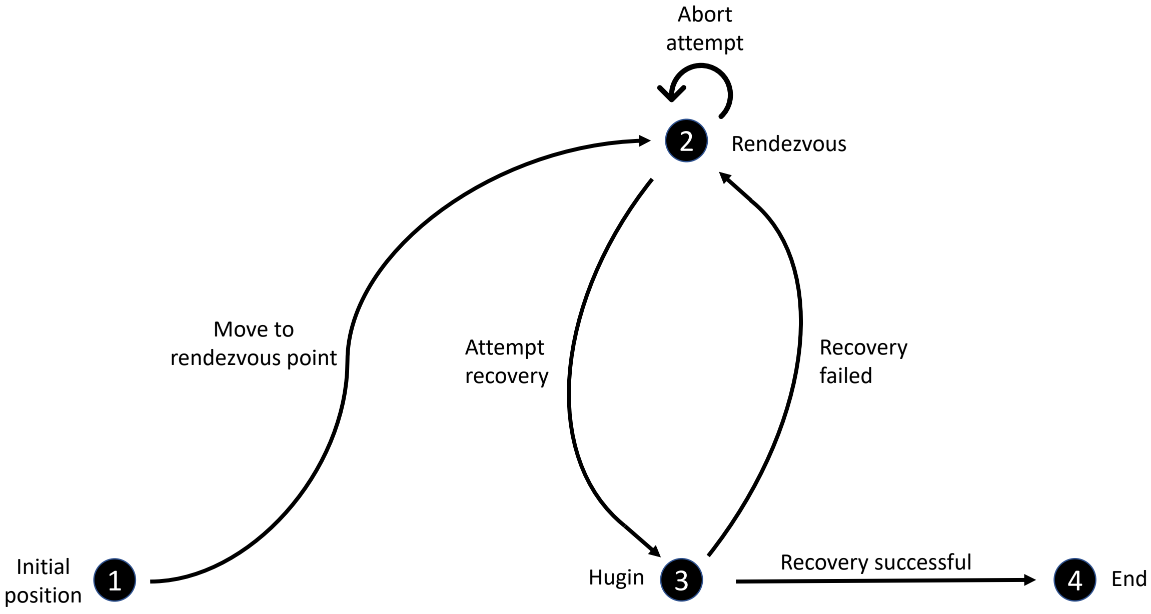


Figure 4.1: The general strategy for accomplishing the recovery task.

efficient. We will assume that the USV can initially be positioned in any direction relative to the AUV. With the assumption from section 1.3 of open waters, the only obstacle that must be considered is the AUV itself. To generalize, we will plan a path that depends on the current phase and the pose of the USV relative to the AUV, both of which will be continuously updated.

In the initial phase, we design the path such that we get a short path to the RV point while ensuring that we keep a large enough distance to the AUV. The design is shown in fig. 4.2. If the USV is positioned in front of the AUV, in the green area, we go straight to WP 3 (the RV point). If the USV is in the yellow area, we go to WP 2 first. If the USV is in the red corridor behind the AUV, we go to WP 1 before the two other WPs to ensure that we do not collide with the AUV. For the two latter cases, we pass the AUV on the closest side. Note that these WPs are not the same as in section 2.5. Going forward, the RV point and the AUV will be the main waypoints of interest. This path planner does not handle corner cases e.g. related to the pose of the USV and is not necessarily optimal for all initial positions close to the AUV. But this will rarely be relevant as we typically will start at a distance from it in the yellow or green areas, and follow a relatively straight path towards the RV point.

Previously, the heading references have been set following a pure pursuit guidance law, i.e. having the heading reference points straight towards the waypoint. For phases 1 and 3, the initial phase and when returning to the RV point after a failed recovery, we will use the pure pursuit guidance. But, for the recovery phase, we also require the USV to be at a certain angle, namely in the same direction as the AUV. To ensure this, we will at the beginning of the recovery phase use Line-of-Sight (LOS) guidance, using the definition from [13]:

$$\psi_{ref} = \pi_p - \arctan(\frac{y_e^p}{\Delta}) \tag{4.3}$$

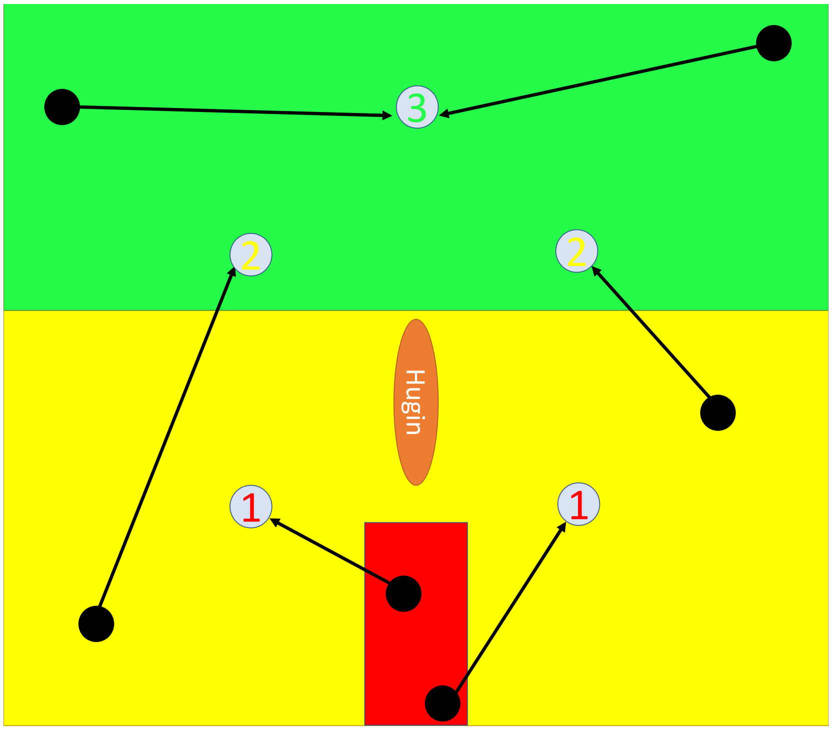


Figure 4.2: The generation of waypoints for the initial state, depending on the position of the USV (Frigg) relative to the AUV (Hugin).

where $\pi_p$ is the angle between north and the line drawn by the RV point and the AUV, $y_e^p$ is the cross-track error, and the lookahead distance $\Delta > 0$ is a design parameter. But, since we also require the USV to physically touch the AUV, we will switch back to pure pursuit when sufficiently close to the AUV, to ensure that the USV and the Stinger point directly towards the AUV when making contact.

### 4.2.2 Control

**The Initial Phase**

In section 3, the controllers have little issues when the USV is moving forward, so we will re-use those controllers from the original control system, with the gains from section 3, for reaching the RV point. This means using the controller from eq. (2.4) at first, before switching to the one in eq. (2.6) to accurately reach the RV point.

Having reached the RV point, we will, as before, align the heading of the USV with the AUV while staying in or moving towards the RV point, before proceeding to the recovery phase. This alignment phase is still considered as part of the initial phase as it is part of step 1 in section 1.2.1. For the alignment we use the controller from eq. (2.7). The controllers are chosen since they have few parameters, and are quick and intuitive to tune.

**Recovery Phase**

The recovery attempts, or more specifically when reversing, were the times where the control system in section 3 seemed to be the most fragile. To increase the robustness, we will redesign the controllers for this phase. For surge, we now use a PI speed controller, as opposed to the previous PD (position) controller. For the heading, we hope that the addition of LOS guidance will increase robustness, and still use a PD-controller. This gives controllers on the form

$$
\begin{aligned}
\boldsymbol{\tau}_{ref} &= \begin{bmatrix} K_{p,u}e_u + K_{i,u}e_{u,int} \\ 0 \\ K_{p,\psi}e_\psi + K_{d,\psi}\dot{e}_\psi \end{bmatrix} \\
e_u &= u_{ref} - u, \quad u_{ref} = -1\,m/s \\
\dot{e}_{u,int} &= e_u \\
e_\psi &= \psi_{ref} - \psi \\
\dot{e}_\psi &= \dot{\psi}_{ref} - \dot{\psi}
\end{aligned}
\tag{4.4}
$$

where $\psi_{ref}$ is given by eq. (4.3) (LOS) when far away from the AUV, and as in eq. (2.4) (pure pursuit) when close to the AUV. To avoid windup, the integrator term $e_{u,int}$ is saturated and is reset to zero when we exit the recovery phase. The controller gains for the recovery phase are given in table 4.1.

| **Gain** | **Value** |
|---|---|
| $K_{p,u}$ | 2000 |
| $K_{i,u}$ | 100 |
| $K_{p,\psi}$ | 5000 |
| $K_{d,\psi}$ | 20000 |

Table 4.1: Gain values for the recovery phase for the first control system.

The PI speed controller is chosen since the USV is more stable and less affected by disturbances when keeping some speed, and because we require some force to connect to the AUV. We do not include a D-term as differentiating noise may negatively affect the controller. Also, we mainly require the USV to keep *some* speed, it is not important that it is exactly equal to the desired speed.

**Failed Recovery Attempt**
After a failed recovery, the same controllers as for the initial phase are used to reach the RV point, before re-attempting the recovery.

### 4.2.3 Simulator Performance

The performance of the control system is shown in fig. 4.3. We see that the environmental disturbances affect the USV, which fails to perfectly follow the line spanned by the AUV and the RV point. Still, the error seems to be small enough for the USV to be able to connect to the AUV, meaning that this could have resulted in a successful recovery. Note that the values of the axes in the right plot make it seem like the USV approaches the AUV at a steep angle, which actually is less than 5 degrees different from the AUV. Note that the oscillations on the path to the right are due to waves in the simulations, which have a high frequency and are hard for the controller to cancel out.

### 4.2.4 Discussion

The control system, which is based on the one from section 3, has now a more generalized path planner and uses LOS guidance in the final part of the task. This has made the system more robust and increased the accuracy when approaching the AUV. The control system is quite simple and has few parameters that need tuning. But, not all the challenges presented in section 4.1 are addressed. We see that the environmental disturbances affect the path of the USV. Upgrading to integral LOS is a possibility to address this. We also have not taken any measures to handle modeling errors, which may be a problem for the physical tests. And the controller gains are far from guaranteed to be optimal

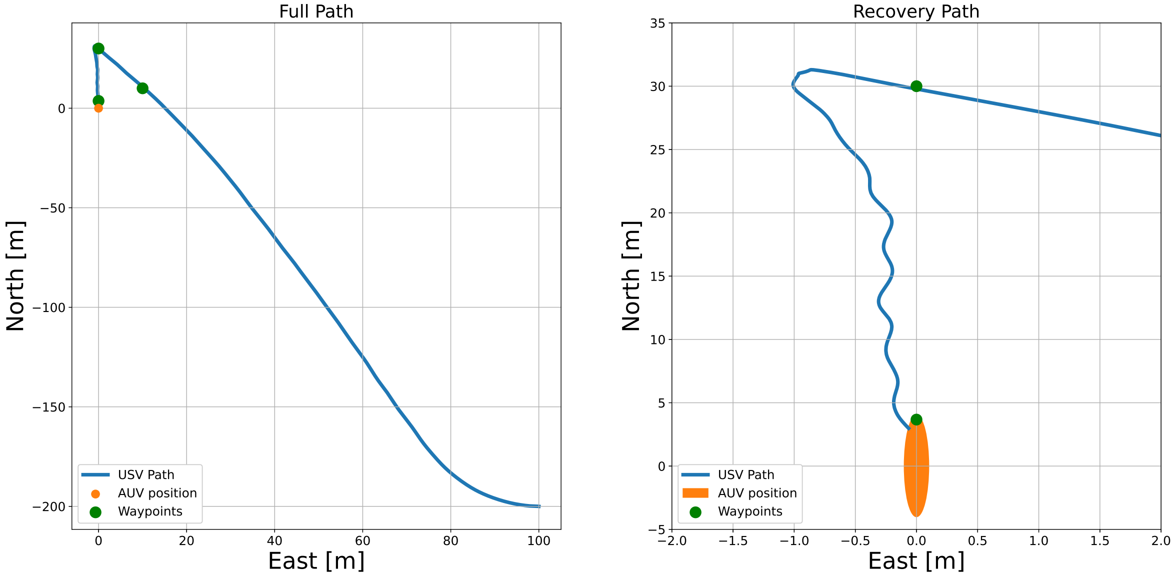


Figure 4.3: The USV path for the first control system.

for the physical system. While not perfect, we still expect the control system to perform quite well, at least under somewhat easy operating conditions. And the physical tests will likely show us more about the strengths and weaknesses of the system.

The parameters of the recovery phase controller were tuned using a trial-and-error approach. We tuned the heading controller to resemble the response of a critically damped second-order system, i.e. reaching its reference as quickly as possible without overshooting. For the PI surge controller, we tuned it to reach its reference fairly quickly while accepting some overshoot, since we do not require it to be exactly accurate, some error is acceptable.

## 4.3 Control System 2: Sliding Mode Control for Recovery

The upgraded control system in section 4.2 does not explicitly handle two of the main challenges from section 4.1, Water jet model inaccuracy and robustness to disturbances. In this section, we will implement a controller that addresses these. We will use the same general strategy as in section 4.2 for solving the task. Since the main accuracy requirement lies with the heading, this is the state the new controller will be applied to.

### 4.3.1 Path Planning and Guidance

Our path planning will be the same as in section 4.2, including LOS guidance when approaching the AUV.

### 4.3.2 Control

**The Initial Phase**

Since we still have very loose accuracy requirements and assume the USV to be far easier to control when going forwards than backwards, we will again re-use the controllers from section 3, solving the initial phase exactly as in section 4.2.

**Recovery Phase: Sliding Mode Control**

Sliding mode control (SMC) is a robust nonlinear control method. A major benefit of this method is its ability to handle structured uncertainties and unmodeled dynamics [13], meaning that it can handle model errors and environmental disturbances, the two main challenges section 4.2 did not address. We will use this method to control the heading for the recovery attempts.

To get the control law, we first rewrite the model from eq. (2.1) to the form

$$\begin{aligned} \ddot{\boldsymbol{x}} &= \boldsymbol{f}(\dot{\boldsymbol{x}}, \boldsymbol{x}) + \boldsymbol{b}\boldsymbol{u} + \boldsymbol{d} \\ \ddot{\boldsymbol{x}} &= \dot{\boldsymbol{\nu}} \\ \boldsymbol{f}(\dot{\boldsymbol{x}}, \boldsymbol{x}) &= -\boldsymbol{M}^{-1}(\mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})) \\ \boldsymbol{b} &= \boldsymbol{M}^{-1}\boldsymbol{B} \end{aligned} \tag{4.5}$$

where $\boldsymbol{B}\boldsymbol{u}$ is a linearized, uncoupled version of $\boldsymbol{f}(\boldsymbol{\tau}_d)$, and $\boldsymbol{d}$ is a joint disturbance term consisting of both internal and external disturbances. We then define the sliding variable

$$\begin{aligned} \sigma &= K_p\tilde{e} + K_i \int_0^t \tilde{e}\, dx + K_d\dot{\tilde{e}} \\ \tilde{e} &= \psi_d - \psi \end{aligned} \tag{4.6}$$

where $\psi_d$ is the desired heading. Then, consider the Lyapunov function

$$\begin{aligned} V &= \frac{1}{2}\sigma^2 \\ \dot{V} &= \sigma\dot{\sigma} \\ &= \sigma(K_p\dot{\tilde{e}} + K_i\tilde{e} + K_d\ddot{\tilde{e}}), \quad \ddot{\tilde{e}} = -\ddot{\psi} \\ &= \sigma(K_p\dot{\tilde{e}} + K_i\tilde{e} - K_d(f_\psi + b_\psi u_\psi + d_\psi)) \end{aligned} \tag{4.7}$$

where $b_\psi$ is the inverse of the moment of inertia $I_z$. Choosing the heading control law

$$u = \frac{1}{K_d b_\psi}(K_p\dot{\tilde{e}} + K_i\tilde{e} - K_d f_\psi + K_\sigma sgn(\sigma)) \tag{4.8}$$

gives

$$\begin{aligned} \dot{V} &= -K_\sigma|\sigma| - K_d d_\psi \sigma \\ &< 0 \quad \forall\sigma \neq 0, \quad K_\sigma > K_d d_{max} \end{aligned} \tag{4.9}$$

which ensures that the equilibrium point $\sigma = 0$ is globally asymptotically stable. Note that for the sliding mode controller we are required to know the maximum disturbance acting on our system.

For large values of $K_\sigma$, the term $K_\sigma sgn(\sigma)$ can lead to chattering. To avoid this we will, as suggested in [13], replace $sgn(\sigma)$ with $tanh(\sigma/\phi)$, where $\phi > 0$ decides the slope of $tanh(\sigma)$ when close to zero. Substituting this into eq. (4.8) gives the final control law for the heading. The values of the controller gains can be found in table 4.2.

The SMC controller was tuned using the same approach and reasoning as in section 4.2.4. During the tuning, the integral term was found to do more harm than good, and was disabled by setting $K_{i,\psi} = 0$.

| Gain | Value |
|---|---|
| $K_{p,u}$ | 2000 |
| $K_{i,u}$ | 100 |
| $K_{p,\psi}$ | 2 |
| $K_{i,\psi}$ | 0 |
| $K_{d,\psi}$ | 8 |
| $K_{\sigma}$ | 3 |
| $\phi$ | 0.5 |

Table 4.2: Gain values for the recovery phase for the second control system.

**Failed Recovery Attempt**

After a failed recovery, the same controllers as for the initial phase are used to reach the RV point, before re-attempting the recovery

### 4.3.3 Simulator Performance

The performance of the control system is shown in fig. 4.4. We see that the path of the USV with the SMC is very similar to the previous control system, still being accurate and able to connect to the AUV, as well as having some wave-induced oscillations on the path. However, we still have the problem of the USV drifting to the side due to wind, indicating that the SMC alone does not handle the impact of environmental disturbances.

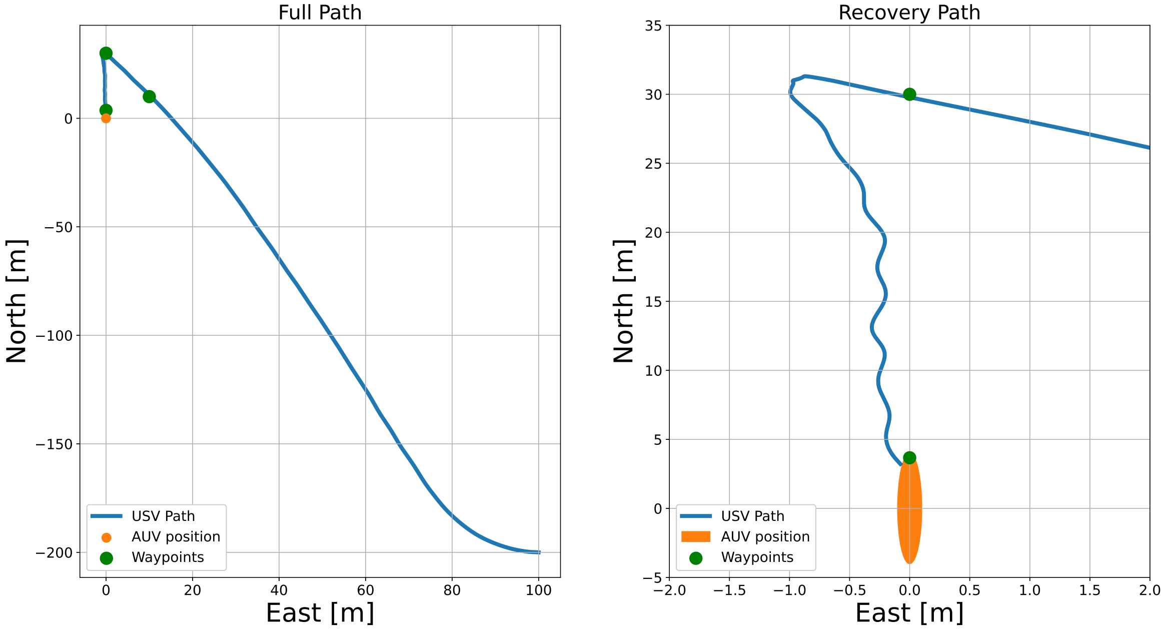


Figure 4.4: The USV path for the second control system.

### 4.3.4 Discussion

The new control system, in particular the sliding mode controller, does not really seem to fix all the problems it was intended to. We believe the main reason for this to be that the environmental disturbances do not only affect the yaw, the state the SMC controls, but also surge and sway, the latter being especially important. Even if the heading controller perfectly followed its reference, we would still have a force acting on the USV sideways, pulling it away from the line it is tasked to follow. Measures must be made in order to account for this, with an extension of the guidance law to Integral Line-of-Sight (ILOS) a possible option.

Moreover, the SMC should make the system robust toward modeling inaccuracy, but this has not been properly tested and may be uncovered for the physical tests. A weakness of the SMC is the number of tunable parameters. The heading PD-controller in eq. (4.4) had only two tunable parameters, whereas the SMC has 4 without the integral term. The parameters in the SMC are also less intuitive to tune than the ones in the PD.

Lastly, the available time limited the development of this control system. The first physical test was scheduled earlier than expected, forcing a change of focus to testing the control system, removing bugs, and integrating the code with the FFI systems. Given more time, we could have looked at implementing e.g. ILOS and an adaptation law to estimate the water jet model.

## 4.4 Control Systems Comparison

To conclude this section, we will compare the performance of the control systems in the simulator. Since only the final part of the control systems differ, this is naturally also the part we will compare.

### 4.4.1 Performance Comparison

From fig. 4.5 and fig. 4.6 we see that the performances of the two control systems are very similar. The paths are almost identical, and so is the heading error. The most significant difference is that the SMC uses slightly less yaw demands.

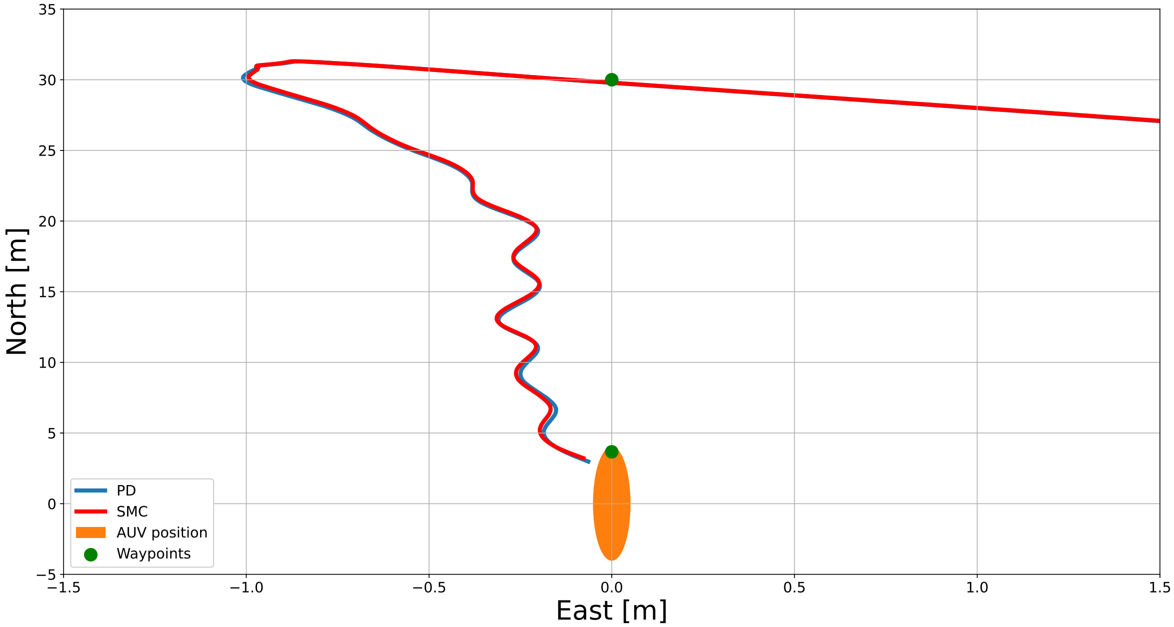


Figure 4.5: USV path comparison for the two control systems, during the recovery phase.

### 4.4.2 Discussion

The main takeaway from fig. 4.5 and fig. 4.6 is that the performances of the two control systems are very similar. In the recovery phase, they follow a very similar path and both finish at approximately

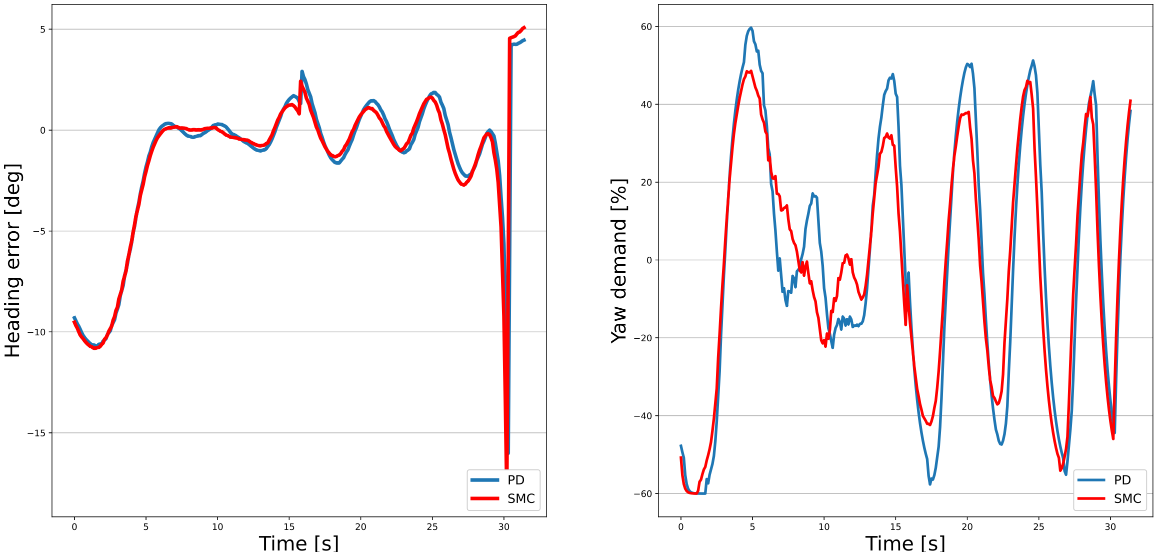


Figure 4.6: A comparison of the heading errors and corresponding yaw demands for the two control systems, during the recovery phase.

the same point, which looks close enough to be able to accomplish the recovery. This also indicates that the heading controller is not the most important factor for the accuracy of the control system, but that other measures, such as integral action for the references, may be just as important.

While similar in performance, the two control systems have different strengths and weaknesses related to the challenges presented in section 4.1. These will probably be more evident in the physical tests. Due to the limited available time, we could not test the control systems more thoroughly before the physical tests, so the two control systems will serve as a solid starting point for these. Then, we can evaluate their performance, strengths, and weaknesses, before choosing one of them and develop it further.

# 5 Physical Testing

This section concerns the physical tests, and improvements to the control system based on the results. Section 5.1 covers the first set of physical tests where we test the two control systems and evaluate their performance, and find and remove errors. Based on the results, we choose one of the control systems, and make adjustments to further improve it in section 5.2. Then, we in section 5.3 conduct a final set of physical tests to test the performance of the final control system.

## 5.1 Test 1: Testing the Developed Control Systems

The first tests of the control systems were carried out on March 31st, in the bay at Karljohansvern, Horten, Norway. For the tests, there were calm waters with little wind, though there could at times be some influence from waves and ocean currents. For these tests, we did not have access to Hugin. This meant that the tests would not have the goal of performing the actual recovery, but rather test the general performance and accuracy of the control systems on open waters. Without the physical component, Hugin was instead simulated as a fixed point on the ocean surface.

### 5.1.1 Execution of the Tests

The control systems were the same as in the simulations, only with some small differences in the communication. The data was sent and received on different ROS topics, and the position was measured in latitude and longitude as opposed to the simulator which gives the position in meters according to a local coordinate system, while the heading now had the opposite sign due to the difference between ENU and NED coordinate systems in the simulator and physical system, respectively. To give the controllers the same inputs as in the simulator, we created a local coordinate system with the pose of the nose of the AUV as Origo. Measurements then subtracted the AUV position and were multiplied with coefficients converting differences in latitude and longitude to meters.

To test the control systems, we ran several test runs with both systems from arbitrary initial positions, such that we could see that the systems were robust enough to reach the RV point regardless of the initial point, as well as learn whether or not the initial position mattered for the results.

The communication with the Stinger did not work properly. This meant that for the Stinger to be submerged, an operator had to manually press a button to start the process. This was done at the end of the alignment phase. This did not impact the results much and is a problem that is expected to be easy to fix.

### 5.1.2 Results

For the initial phase, both control systems seemed to work well and give a good response. They reached the RV point regardless of the starting position, in a similar manner as in the simulations. This was in line with the expectations. For each control system, we ran two full tests that were to mimic a real recovery, the only difference being that we did not have the physical AUV.

**Control System 1**
The results for the first control system are shown in fig. 5.1 and fig. 5.2. We see that the accuracy is significantly reduced compared to the simulations. Most significant is the heading control, where we see that the USV is much worse at following the desired line during the recovery phase. Also, we can see that the GPS measurements are not entirely perfect, as they at times seem to jump back and forth. Note that with the nose of the simulated AUV as Origo, the position of the AUV in the plots is slightly different from the simulations in section 4.

Figure 5.3 shows a comparison of the control system performance for the first physical test run and in the simulator, with similar initial pose and velocity. For the simulation, we see a small error which

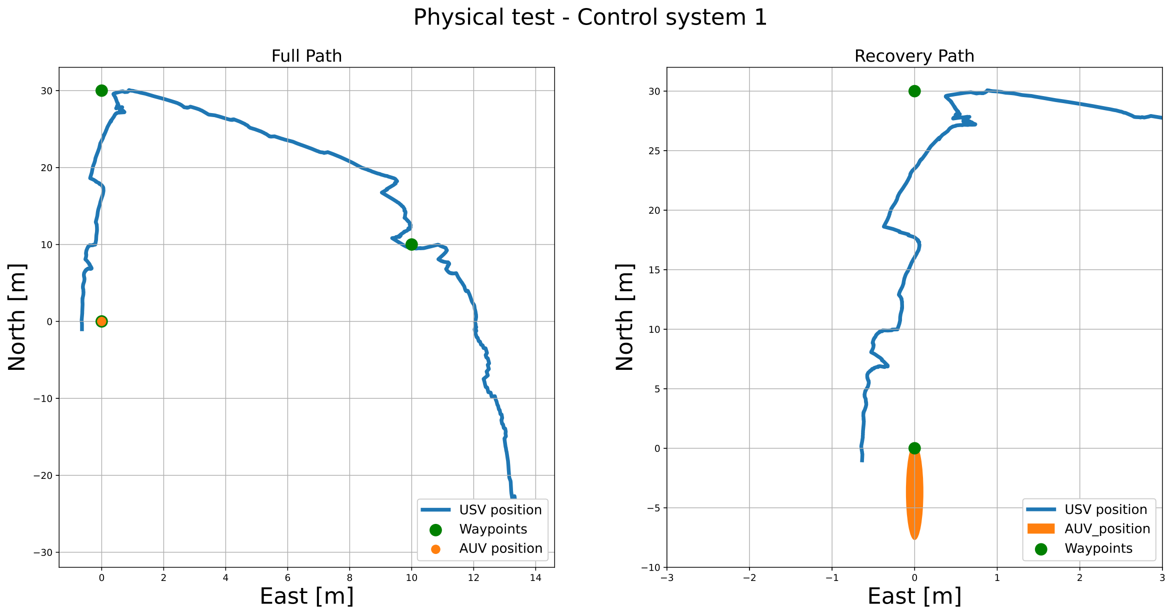


Figure 5.1: The first full test run for control system 1.

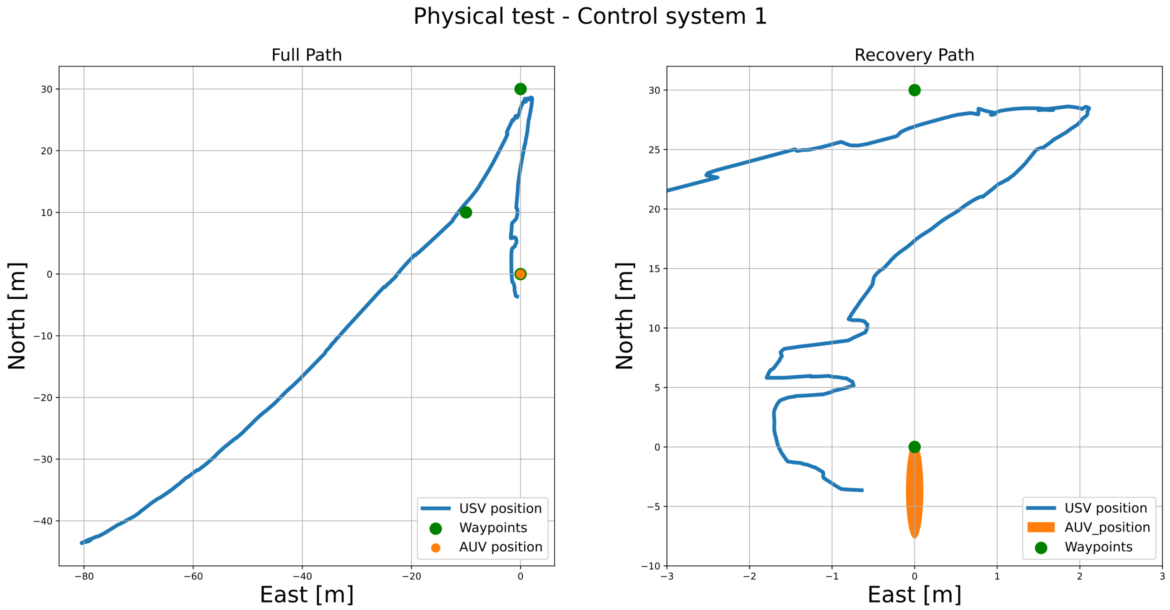


Figure 5.2: The second full test run for control system 1.

is mainly caused by the wind and ocean current disturbance, while the physical one has a far larger error, and generally struggles to follow its heading reference. Figure 5.4 shows the heading error and corresponding yaw demands during the first part of the recovery phase. The controller fails to correct the negative heading error, and at times even applies positive demands.

### Control System 2

The results for the second control system are shown in fig. 5.5 and fig. 5.6. We see that, as for control

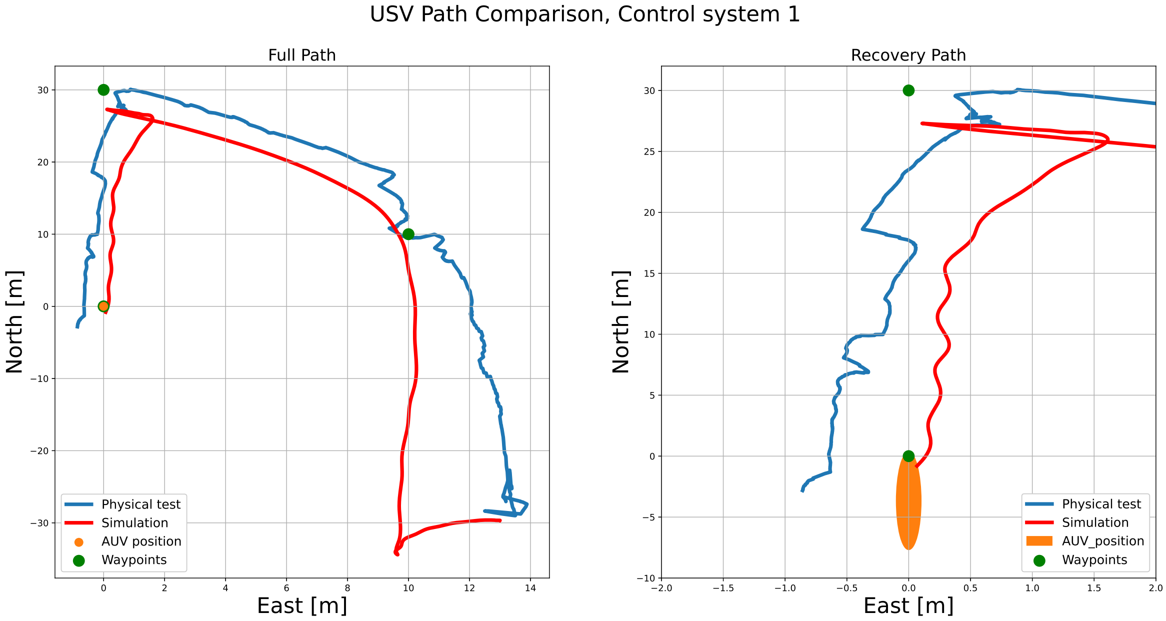


Figure 5.3: Physical test run results versus simulation results for similar initial conditions, for control system 1.

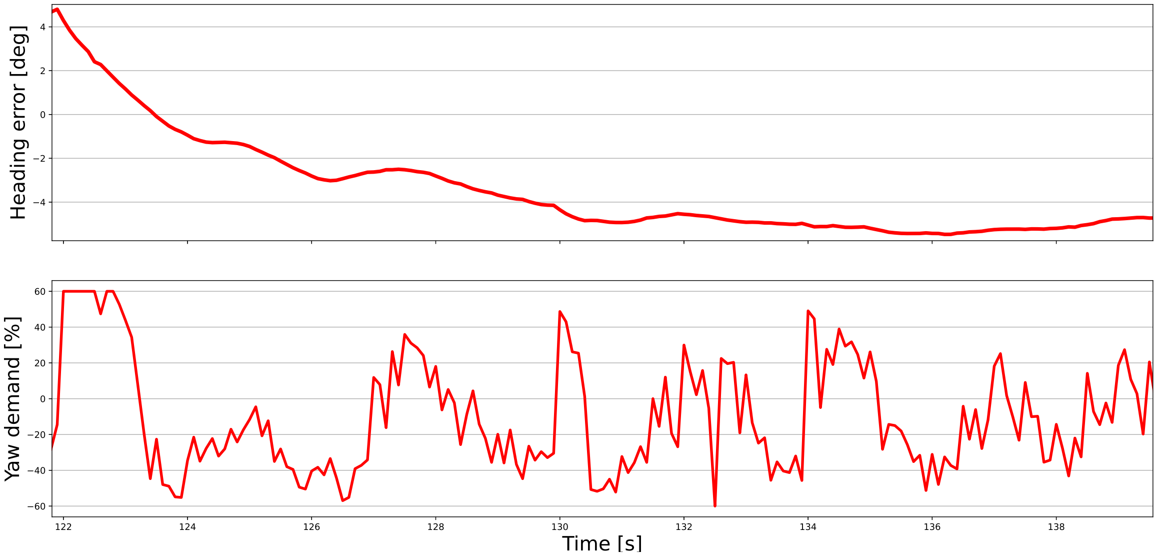


Figure 5.4: The heading errors and corresponding yaw demands for control system 1, for the first part of the recovery phase.

system 1, the controllers struggle to follow its heading references during the recovery phase. But now, the errors are even larger, and the system is far from being able to connect to the (simulated) AUV.

The performance of control system 2 in the simulator and for the first physical test run is shown in fig. 5.7. We see that the error is far larger in the physical test. During its alignment phase in the RV point, the phase where the USV attempts to stay in the RV point while aligning its heading with the

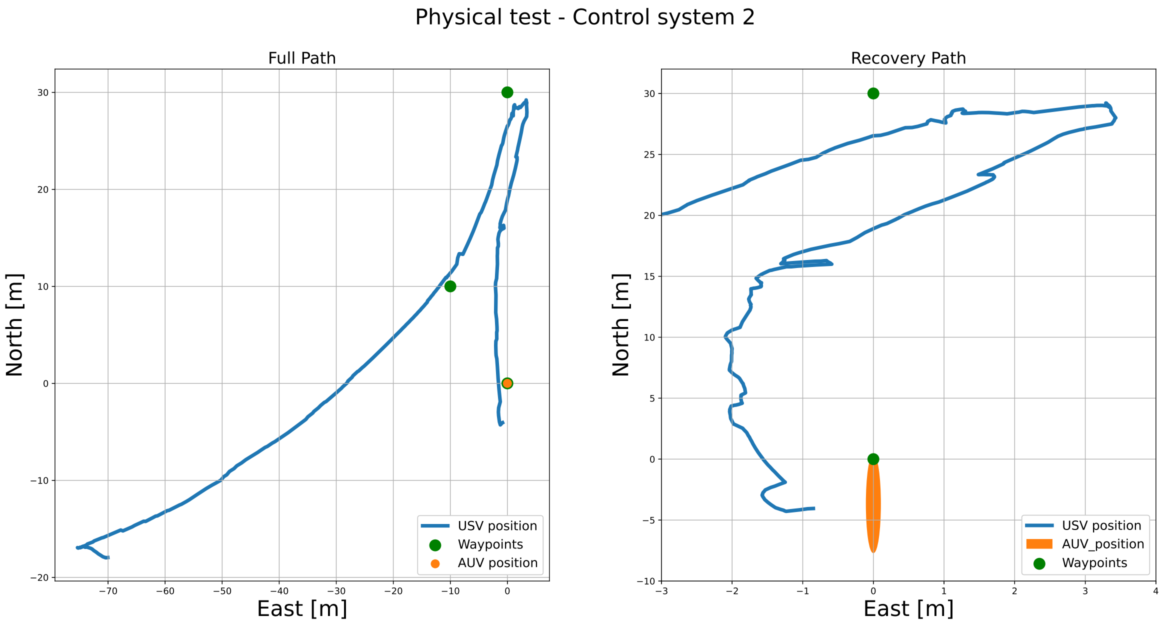


Figure 5.5: The first full test run for control system 2.

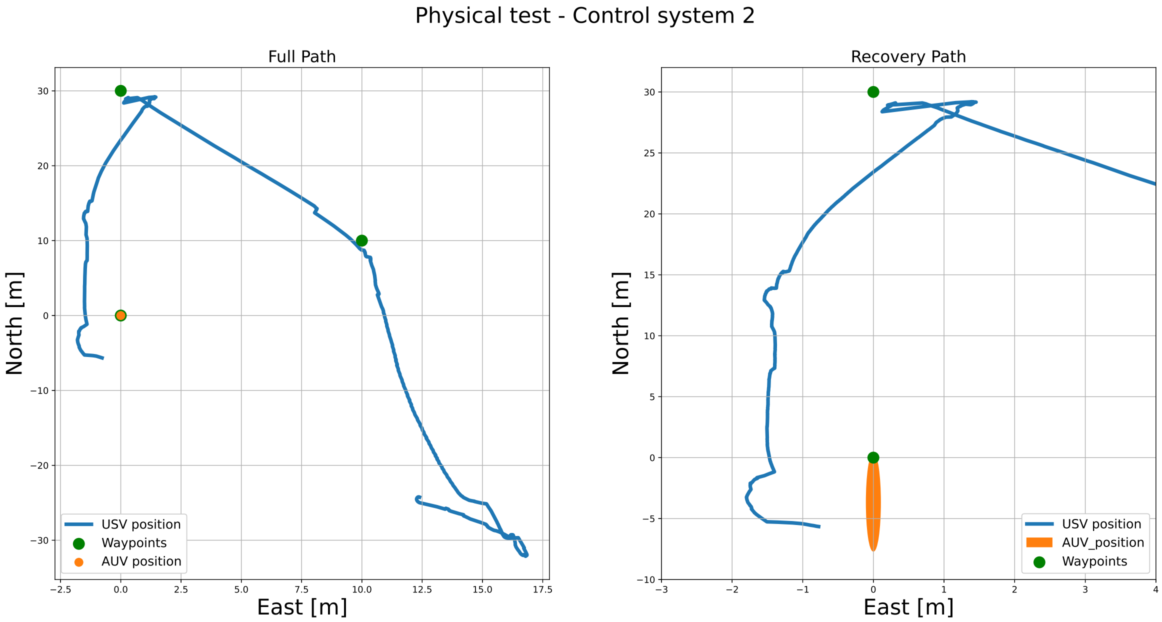


Figure 5.6: The second full test run for control system 2.

AUV, we see that the USV drifts more than 3 meters to the side. The heading error and corresponding yaw demands during the first part of the recovery phase are shown in fig. 5.8. With the heading error being larger than in fig. 5.4, the demands are almost always negative, but the controller still fails to decrease the error.

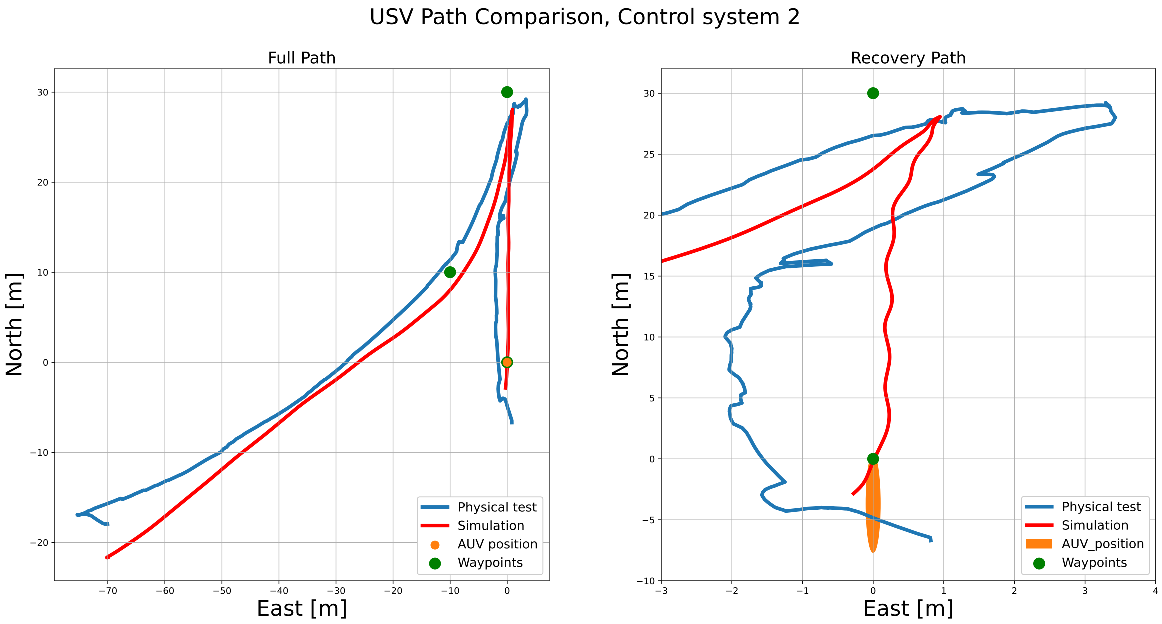


Figure 5.7: Physical test run results versus simulation results for similar initial conditions, for control system 2.

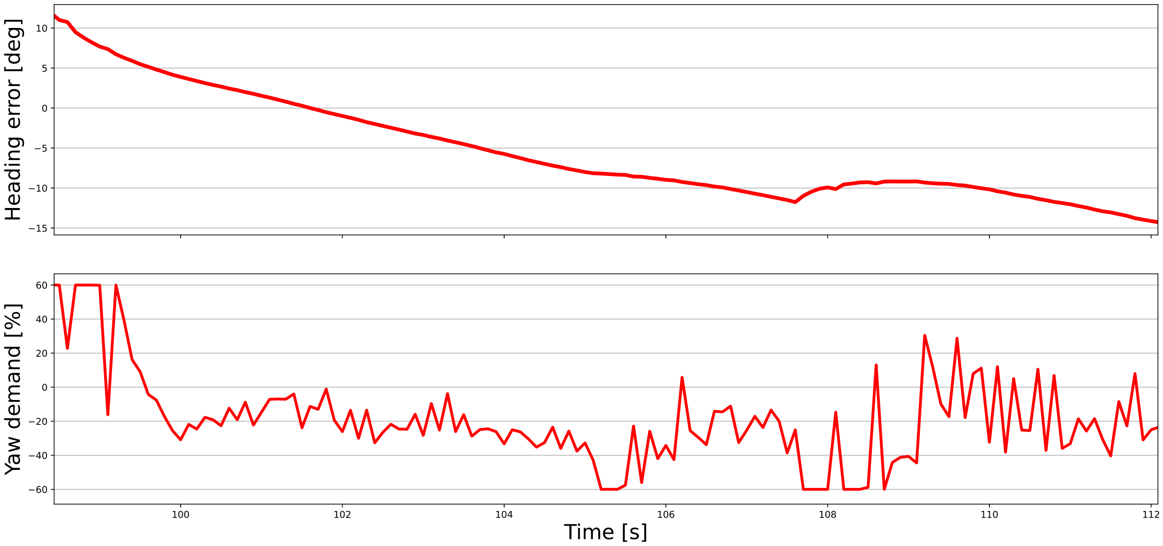


Figure 5.8: The heading errors and corresponding yaw demands for control system 2, for the first part of the recovery phase.

**Uncovered Errors**

When analyzing the results of the first physical tests, two major performance-degrading errors were discovered. The first was an error in the simulator code, which caused an old, existing model of the USV without the Stinger to be active, instead of the new ones with the Stinger. While the control inputs dynamics were correct, the hydrodynamics, mass, and moment of inertia were not. This means that the control systems had been developed and tuned on the wrong USV dynamics. The dynamics of the USV without Stinger and with the Stinger raised are not too different, but the dynamics with the Stinger submerged in the water are significantly different. This explains a lot of the results and observed deviations from the simulations.

Secondly, we found an error in the code regarding the conversion between NED and ENU during the recovery phase. Recall that in the simulator the heading was measured in ENU, while the physical USV measures it in NED, giving the angles opposite orientations. In the recovery phase, the derivative of the heading error is defined as:

$$\dot{e}_\psi = \dot{\psi}_{ref} - \dot{\psi} \tag{5.1}$$

But while the term $\dot{\psi}$ had the correct orientation, the term $\dot{\psi}_{ref}$ still followed the sign of the ENU frame. The consequence of this was a term in the controller that would constantly pull it away from the origin. This is probably the cause of the unexpected values of the yaw demands. The demands should all be negative, but has some positive spikes which likely occur due to the controller trying to resist the changes in the heading reference because of the sign error.

### 5.1.3 Discussion

The assumption that the first part, where the USV is going forwards, was the easy part, and that the recovery stage was the difficult one, seems to hold well. Despite being tuned on the wrong model, and in general spending little time on it during the development, the initial stage of reaching the rendezvous point was successful every time, though the USV drifted to the side a bit more than we would like during its alignment phase. A slower approach and/or more damping could improve this.

For the recovery phase, we will evaluate the results in relation to the main challenges, which, as stated in section 4.1, were expected to be:

- Water jet model inaccuracy
- Parameter tuning
- Robustness to disturbances
- Strategy for performing the actual recovery

The water jet model inaccuracy, or rather model inaccuracy in general, seems to be the most significant problem, which is mainly caused by the simulator error. Having been developed and tuned on a model with a far too small moment of inertia, the controller expected the dynamics to be much faster than what was the case. This also meant that the lookahead distance $\Delta$ was too small. The too small $\Delta$ made the USV approach the desired line too fast, giving an overshoot path-wise as the USV failed to adjust its heading quickly enough when close to the line.

We expected the controllers to be imperfectly tuned, as the simulator inevitably would be different from the physical system, which was now very different due to the simulator error. But tuning takes time, which we did not have for the first physical test. This must be kept in mind when improving the system further as we for the second tests will have even less time available, and must have a system that does not require tuning outside of the simulator. The SMC, with more and less intuitive parameters to tune, seems to be especially vulnerable to poor parameter tuning.

Since there were few environmental disturbances present we gained little knowledge about this during the tests.

The strategy was OK, but should be changed somewhat for improved results. The lowering of the Stinger contributed to a large error when starting the recovery phase, and the slower yaw dynamics made the heading controller struggle to follow its references. A solution to this is to put the RV point further from Hugin, giving the USV more time to approach the desired line, and increase the lookahead distance to account for the slow yaw dynamics. Also, the FFI employees noted that the USV should have a higher speed when approaching Hugin, as this has been the case for the manual recoveries. They also suggested that we could use more thrust. The manual recoveries had been successful using some force at times and should be okay to do when reversing as the water will flow away from Hugin. This means that we could loosen the bounds on the demands.

The SMC, which is more complex than the PD controller, was in theory expected to give the best results, but the opposite was the case during the tests. It seems that the non-optimal tuning of the SMC parameters may be the main reason for the poor results, and that the SMC is more sensitive to bad parameter values than the PD controller. The SMC can be viewed as a saturated PID-controller, giving minimum or maximum throttle for large errors, and a smoother throttle in between, depending on the slope of the $tanh(\cdot)$ function. But if the parameters are wrong, the controller will not work well. In our case, the slope of the $tanh(\cdot)$ function may have been too wide, or the gain $K_\sigma$ was not large enough to cancel out all the external and internal unmodeled dynamics.

### 5.1.4 Conclusion

From the first physical tests, we saw relatively good results, especially given the errors later uncovered. The controllers and path planners worked as expected, controlling the USV in a desirable manner. Unfortunately, an error in the simulator code and a sign error for $\dot{\psi}_{ref}$ caused a lower accuracy than we would have expected otherwise. We have identified some key issues that will be addressed to improve the performance. With the first control system performing the best, this is the one we will keep and re-develop, while discarding the other one with the SMC that performed worse and seemed more fragile in terms of parameter tuning.

The uncovered errors from section 5.1.2 heavily affected the results and were a large part of the reason why the accuracy was not good enough to successfully perform the recovery. But, part of the intention behind the first physical tests was to uncover and remove errors. And if we manage to improve the control system *and* now have found and removed all errors, the first physical tests have served their purpose. We also note that the GPS measurements of the position, which have been believed to be impeccable and almost unaffected by noise, seemed to have jumps during the tests, which can impact the performance of the control system and may need to be accounted for.

## 5.2 Further Development

Using the results and knowledge from the first physical tests, we will now implement measures to improve the control system. As stated in section 5.1.4, control system 1 from section 4.2 is the one we have chosen to further develop.

First, we will revise the strategy for the recovery phase. To give the USV more time to approach the desired line, the RV point is now placed 60 meters in front of the AUV, as opposed to the previous 30 meters. Following the feedback from FFI, for the recovery phase, the bounds on the use of surge and yaw demands are increased from [-60, 60] to [-80, 80] for both to allow more power use, and the surge reference speed is increased to -1.5 m/s. To account for the slower yaw dynamics of the USV when the Stinger is submerged, the lookahead distance $\Delta$ is increased from 5 to 10. We now introduce integral action into the LOS reference, using the non-linear ILOS in [29]:

$$\begin{aligned} \psi_{ref} &= \pi_p - \arctan(K_p y_e^p + K_i y_{int}^p) \\ \dot{y}_{int}^p &= \frac{\Delta y_e^p}{\Delta^2 + (y_e^p + \kappa y_{int})^2} \\ K_p &= \frac{1}{\Delta}, \quad K_i = \kappa K_p \end{aligned} \tag{5.2}$$

where $\kappa > 0$ is a design parameter. In our system, we use $\kappa = 0.05$. The non-linear ILOS is chosen over the traditional ILOS since it is more robust against wind-up in the integrator term.

Finally, due to the simulator implementation error that caused the control system to be simulated and tuned using the wrong model, we also needed to re-tune some of the controller gains. This meant mainly increasing some of the gains, especially for the recovery controllers. The controller gains for the final control system can be found in table 5.1. Note that we have added a damping term in surge in table 5.1a, and integral terms for surge and yaw in table 5.1b, in an attempt to reduce errors and increase robustness towards disturbances when reaching the RV point. The gains were tuned using the same approach and reasoning as in section 4.2.4.

| Gain | Value |
|---|---|
| $K_{p,u}$ | 100 |
| $K_{d,u}$ | 500 |
| $K_{p,e_\psi}$ | 10000 |
| $K_{p,\psi}$ | 10000 |
| $K_{d,\psi}$ | 60000 |

(a) Controller gains for the start of the initial phase

| Gain | Value |
|---|---|
| $K_{p,u}$ | 60 |
| $K_{i,u}$ | 10 |
| $K_{d,u}$ | 2000 |
| $K_{p,\psi}$ | 8000 |
| $K_{i,\psi}$ | 5000 |
| $K_{d,\psi}$ | 50000 |

(b) Controller gains for reaching the RV point

| Gain | Value |
|---|---|
| $K_{p,u}$ | 80 |
| $K_{d,u}$ | 5000 |
| $K_{p,v}$ | 50 |
| $K_{d,v}$ | 30 |
| $K_{p,\psi}$ | 5000 |
| $K_{d,\psi}$ | 30000 |

(c) Controller gains for the alignment phase

| Gain | Value |
|---|---|
| $K_{p,u}$ | 2000 |
| $K_{i,u}$ | 100 |
| $K_{p,\psi}$ | 12000 |
| $K_{d,\psi}$ | 60000 |

(d) Controller gains for the recovery phase

Table 5.1: Controller gains for the final control system.

### 5.2.1 Simulation Results

A comparison between the improved control system and its predecessor is shown in fig. 5.9, under normal operating conditions, meaning with the extensions from section 3. We see that the improved control system seems better tuned in the initial phase, as its path is more straight and accurate. In the recovery phase, we see that both control systems are fairly accurate, both seemingly accomplishing the recovery. However, we see that the improved system seems better at following the desired line, with the old one being more affected by the wind and current force that points towards the northeast. This indicates that the ILOS algorithm has enhanced the performance of the control system.

The differences between the two control systems are more evident when tested under more extreme conditions. In fig. 5.10, we have increased the magnitude of the wind from 100 N to 200 N and placed the RV point slightly to the side, to give harder initial conditions when starting the recovery phase. We see that the improved control system still is accurate, while the old one is unable to both follow the line and recover the AUV.

### 5.2.2 Discussion

The changes made in this subsection seem to have significantly improved the control system. The revised recovery strategy is better suited to the (slow) dynamics of the USV, the implementation of ILOS seems to make the system more robust towards the environmental disturbances, and the re-tuning

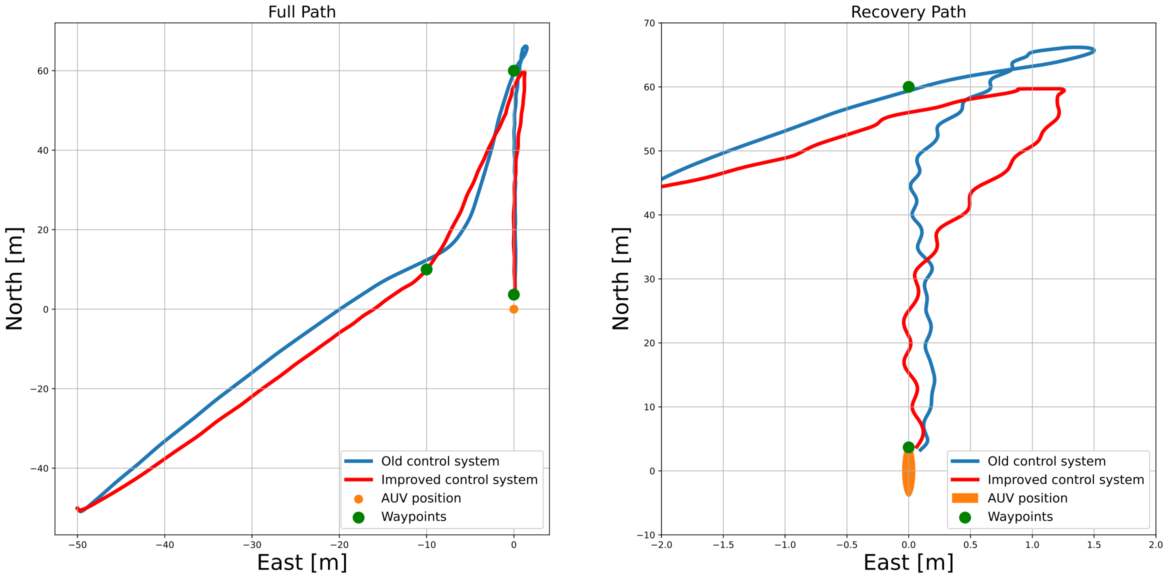


Figure 5.9: The old control system versus the improved one.

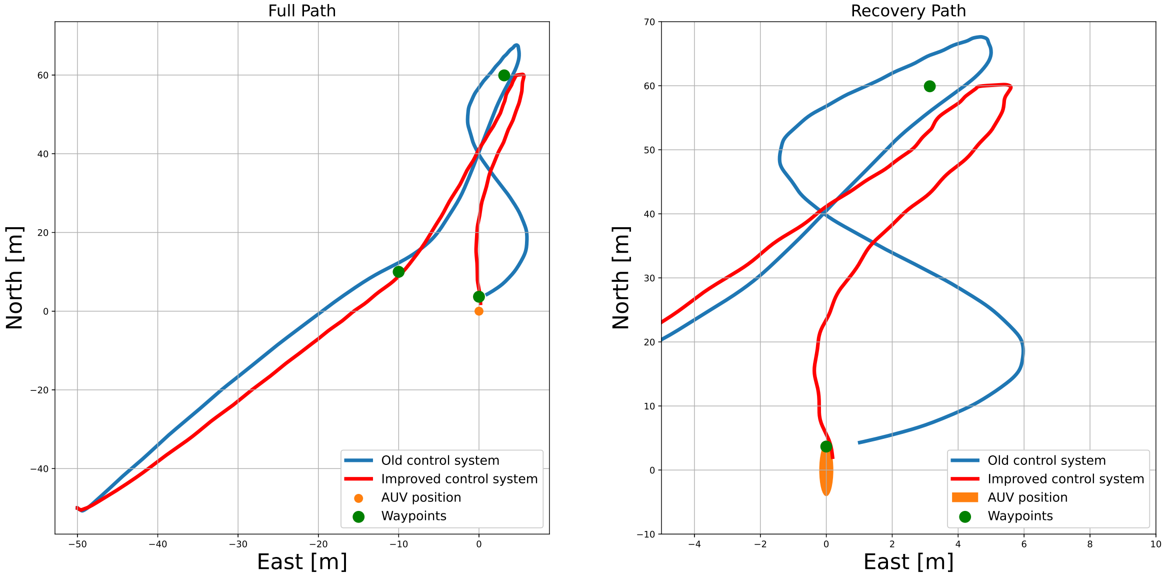


Figure 5.10: The old control system versus the improved one, under hard operating conditions.

of the controllers improved the overall accuracy. Note that the latter is true under the assumption that the new models are more accurate than the previous, which is fairly certain at least for the recovery phase. Thus, the control system has been improved, and is hopefully now able to perform the recovery task in the physical world.

While the control system now in general seems more robust, it still has room for improvement. One potential flaw regards reaching the RV point. If the distance threshold for reaching the RV point is too small, external disturbances may push the USV away and make it unable to reach the RV point and continue to the recovery phase. If the point is close to and directly to the side of the USV, it may end up encircling it while keeping a constant distance from it. This has not been a problem in the simulations or in the first physical tests, but should still be kept in mind, and perhaps addressed anyway. A potential solution is discussed in section 6.3.

## 5.3 Test 2: Testing the Final Control System

The final physical tests were conducted on May 11th and 13th. As the first, these were also conducted in the bay at Karljohansvern, Horten, Norway. We did not have access to the physical AUV for these tests either, but again simulated it as a fixed point on the ocean surface. This meant that the tests were very similar to the first ones, the upgraded control system being the only major difference, though the weather conditions may have been somewhat different. Since tests were conducted on two different days, we also had the opportunity to make some adjustments to the control system in between.

### 5.3.1 Execution of the Tests

The tests were executed as in section 5.1, though the author was not present. The control system was tested from different starting positions. The Stinger communication did not work for these tests either, but was instead manually set to be lowered by an operator, as in the first tests. This was done as the control system entered the recovery phase.

### 5.3.2 Results

**May 11th**

The results from the best test run on May 11th are shown in fig. 5.11, fig. 5.12, and fig. 5.13. We see that the USV struggles to reach the RV point, as it uses a long time to come close enough to it to start the alignment phase. And during the alignment phase, we see that the USV drifts away from the RV point, such that it starts the recovery phase 3 meters away from the desired line, as opposed to under 1 meter at the start of the alignment phase. For the recovery phase, the positional accuracy is high. While overshooting a bit, the sideways error is always below 1 meter, and it seems that this could have resulted in a successful recovery. While the heading control is quite accurate in the recovery phase, we see that the controller is extremely aggressive and chatters between its minimum and maximum value throughout the entire phase, also leading to oscillations in the heading and heading error. As for the surge, we see that the USV reaches the desired speed of -1.5 m/s fairly quickly and stayed around this value for the rest of the recovery phase. The surge speed controller worked well for all test runs, and will not be explicitly shown for other test runs. The plots of the full path are not cropped in this subsection in order to make it easier to see the heading of the USV when it is close to the AUV. The red dots in fig. 5.11 mark the points on the path where the control system switched to a new phase.

The results from the worst test run are shown in fig. 5.14 and fig. 5.15. Again, we see that the control system struggles with reaching the RV point, and drifts even further away from it during the alignment phase. The USV then has to spend most of the recovery phase trying to reduce the cross-track error, which it fails to achieve completely. At the end of the recovery phase, it seems that the damping term is too high, resulting in the USV for a short spell moving away from the desired line before the control system switches to PP guidance.

**Adjustments**

With a free day between the tests, we had the opportunity to analyze and evaluate the results and make some adjustments to the control system to try to improve it. With the control system struggling to reach the RV point due to imperfect heading control, we reverted the gains of the two first heading controllers back to the ones from the first physical tests, with slightly higher P-terms to try to penalize heading error harder. With the FFI operators reporting the USV to be very slow during the initial phase, the P-term in the surge controller was increased, and the D-term removed. For the controller for reaching the RV point, the integral term was removed as we suspect windup in this term could be

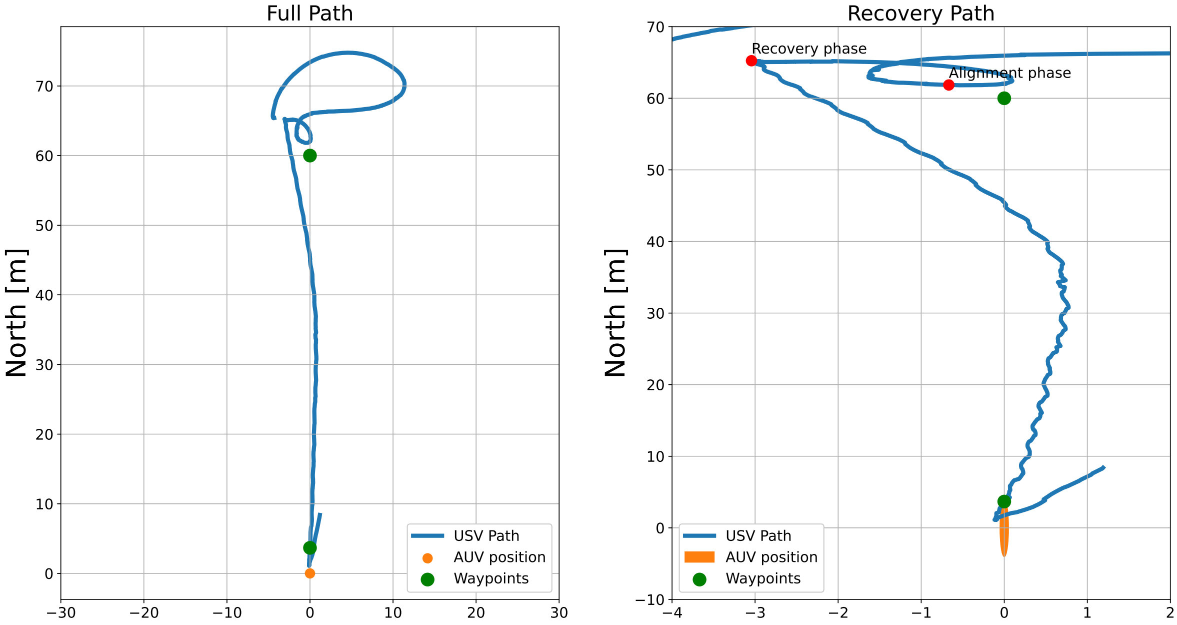


Figure 5.11: The path of the USV for the best test run.

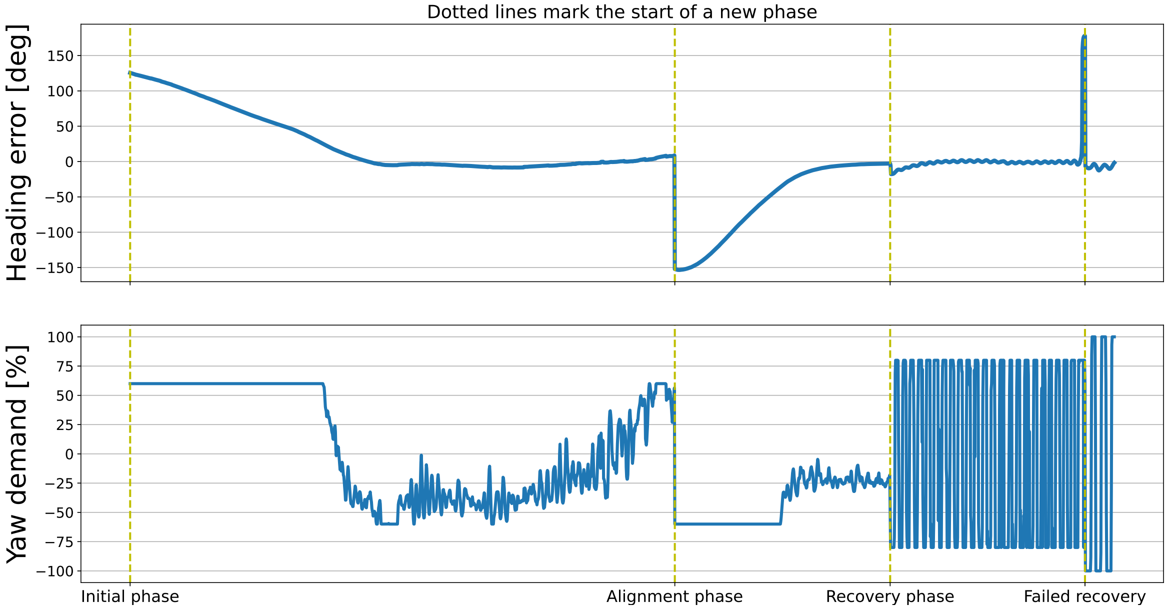


Figure 5.12: The heading errors and yaw demands for the best test run.

a cause of the USV struggling the reach the RV point. For the alignment phase, the controller was re-tuned to be far more aggressive, to ensure that the recovery phase started close to the RV point with a small cross-track error. The bounds on the demands were also increased to [-80, 80] for all three demands, allowing the controller to use more throttle. The gains for the recovery controller were decreased a lot, to attempt to remove the constant oscillations and chattering on the heading and yaw demands. The final gains for the control system can be found in table 5.2.

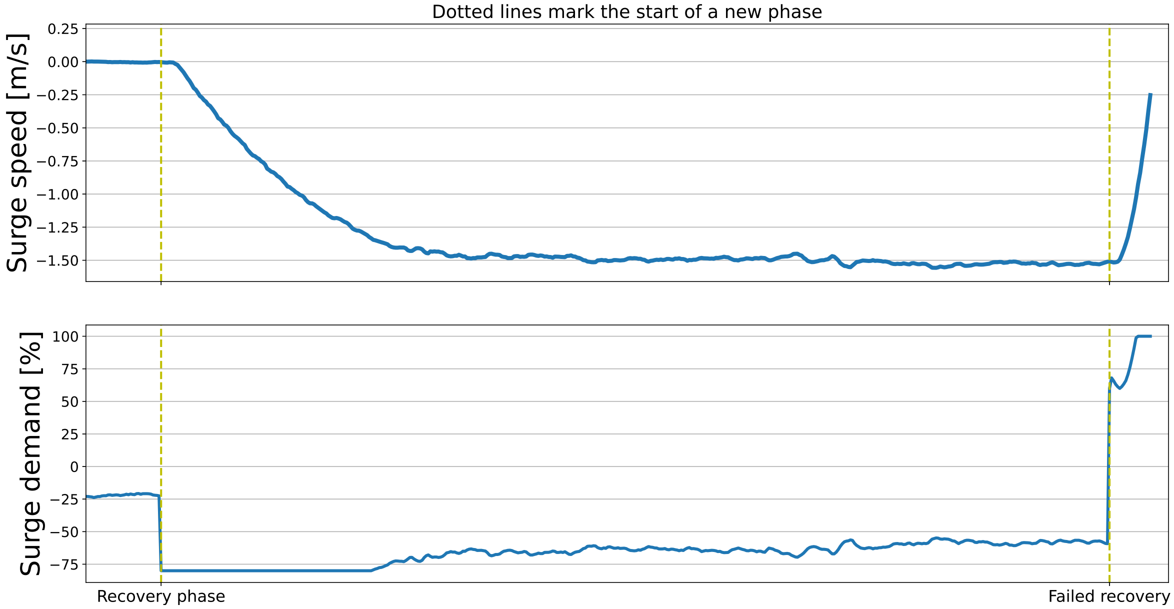


Figure 5.13: The surge speed and demands for the best test run.

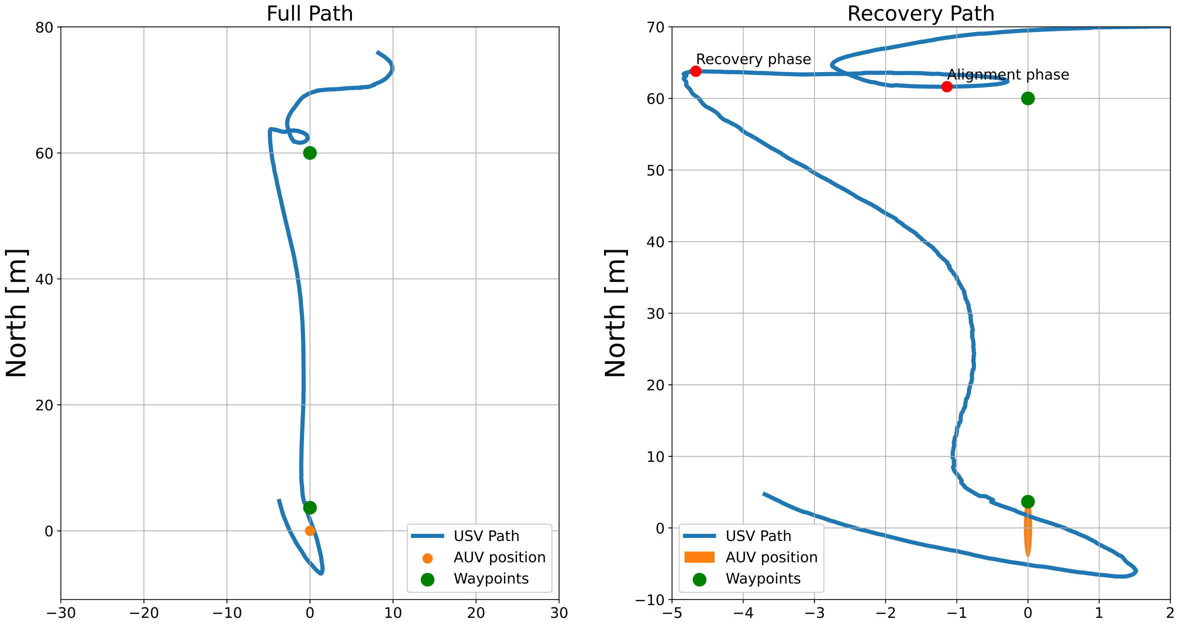


Figure 5.14: The path of the USV for the worst test run.

### May 13th

The results from the final test run, performed on May 13th, are shown in fig. 5.16 and fig. 5.17. We see that the USV still struggles to reach the RV point immediately, again approaching it in a snake-like trajectory. The alignment phase seems to have improved, as the sideways error now is significantly reduced when starting the recovery phase. For the recovery phase, we see that the chattering on the yaw demands and the heading oscillations are almost gone, but with the drawback that the controller does not follow its reference as well as in fig. 5.11. Though the sideways error is almost 2 meters at

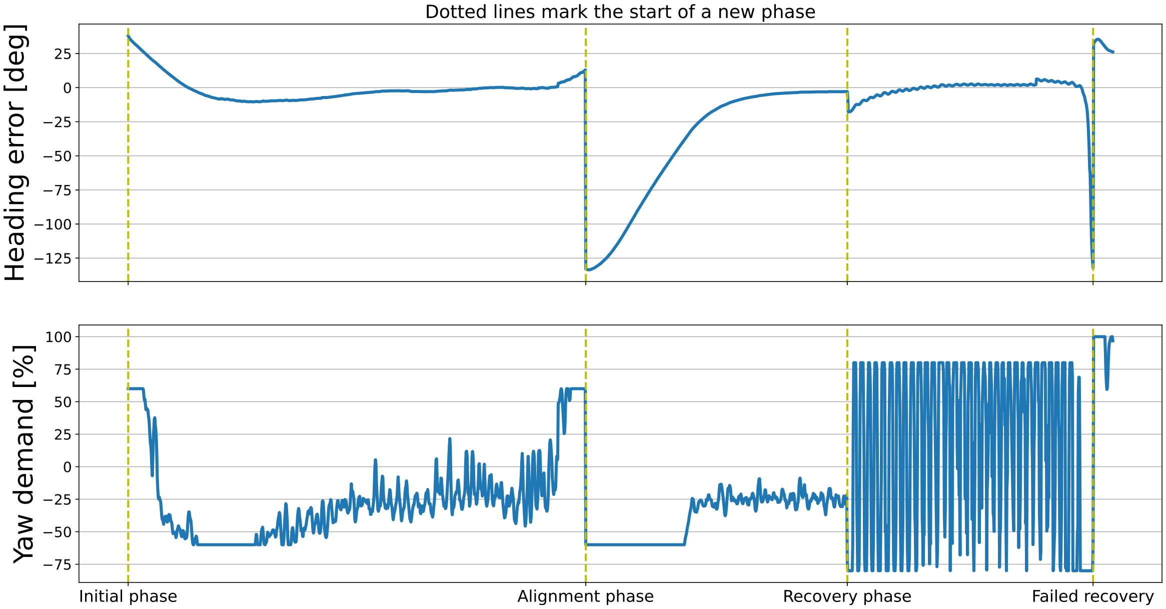


Figure 5.15: The heading errors and yaw demands for the worst test run.

| Gain | Value |
|---|---|
| $K_{p,u}$ | 200 |
| $K_{d,u}$ | 0 |
| $K_{p,e_\psi}$ | 5000 |
| $K_{p,\psi}$ | 7000 |
| $K_{d,\psi}$ | 20000 |

(a) Controller gains for the start of the initial phase

| Gain | Value |
|---|---|
| $K_{p,u}$ | 60 |
| $K_{i,u}$ | 10 |
| $K_{d,u}$ | 2000 |
| $K_{p,\psi}$ | 7000 |
| $K_{i,\psi}$ | 0 |
| $K_{d,\psi}$ | 20000 |

(b) Controller gains for reaching the RV point

| Gain | Value |
|---|---|
| $K_{p,u}$ | 300 |
| $K_{d,u}$ | 1000 |
| $K_{p,v}$ | 300 |
| $K_{d,v}$ | 1000 |
| $K_{p,\psi}$ | 5000 |
| $K_{d,\psi}$ | 30000 |

(c) Controller gains for the alignment phase

| Gain | Value |
|---|---|
| $K_{p,u}$ | 2000 |
| $K_{i,u}$ | 100 |
| $K_{p,\psi}$ | 6000 |
| $K_{d,\psi}$ | 30000 |

(d) Controller gains for the recovery phase

Table 5.2: The final gains for the control system.

most during the recovery, the USV reduces this when switching to PP guidance, and this test run *may* have resulted in a successful recovery.

### 5.3.3 Discussion

Though the final test run may have resulted in a successful recovery, there is still room for improvement for the control system. The USV, for all test runs of the second physical tests, struggled to reach the RV point, which was not a problem for the first physical tests. This may be because the controllers and their gains were slightly different for the two tests, but it may also be caused by external factors. Figure 5.18 shows the velocity for the final test run. We see that in both the initial phase and the recovery phase, the speed in sway is often non-zero, despite using no sway demands. Though this may come from the other demands, especially yaw demands, or from the Stinger-induced asymmetry, it may also be caused by ocean currents. And with the waypoints being constant, the current will only affect the position of the USV, whereas with the physical AUV we would expect a similar impact from the current on all components. As the USV approaches the RV point in the initial phase, the surge speed is approaching zero, making movement in sway more impactful. With the heading being almost

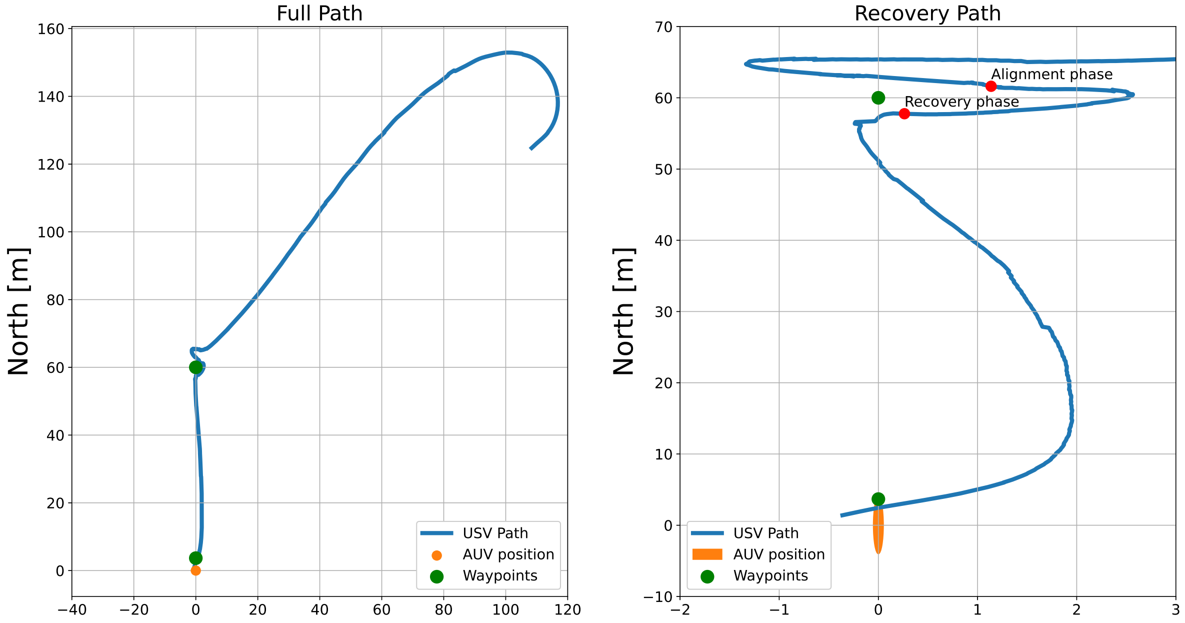


Figure 5.16: The path of the USV for the final test run.

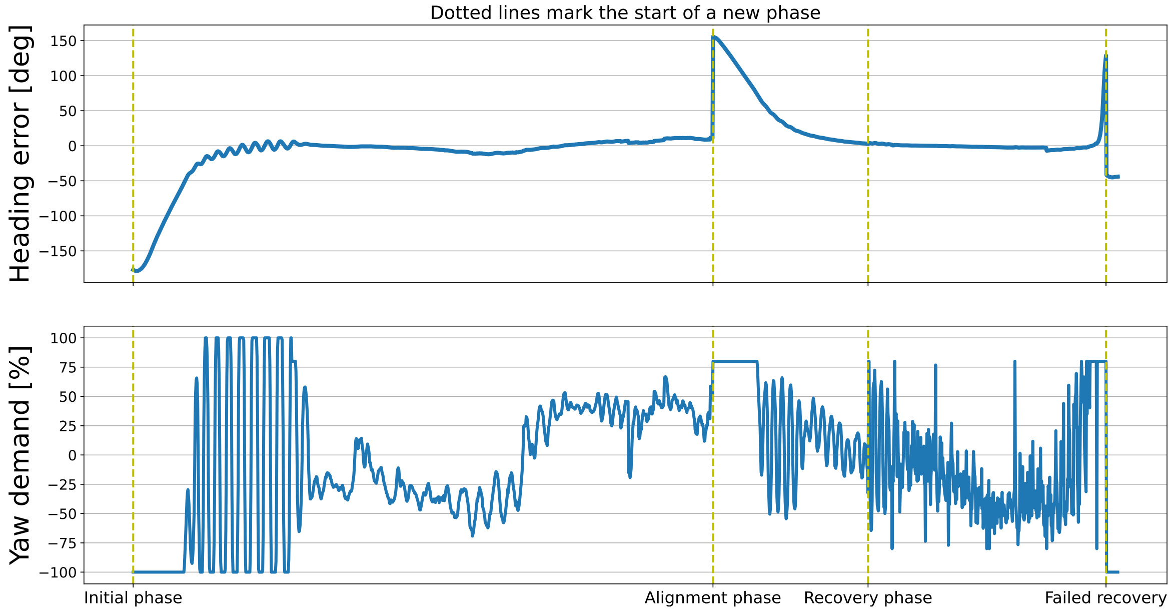


Figure 5.17: The heading error and yaw demands for the final test run.

constant, motion stemming from an ocean current is a possible cause of the sudden curve in fig. 5.16 close to the RV point. For the first physical tests, the speed in sway was rarely above 0.1 m/s in sway, indicating that there may have been less currents for those tests.

The control during the alignment phase seemed to be improved, with the sideways error significantly reduced during the phase. However, the error along the x-axis was just as large after the alignment, only with opposite sign. We do expect that this can be further improved by fine-tuning the controller and letting the USV stay a bit longer in the alignment phase. The re-tuned recovery controller now

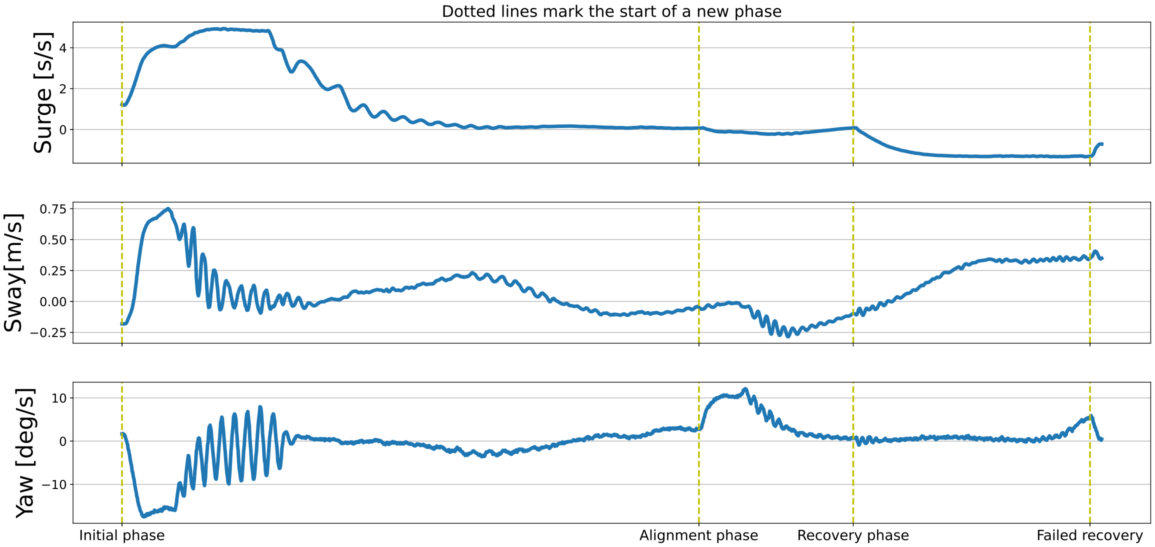


Figure 5.18: The velocity of the USV for the final test run.

had less chattering and oscillations, but on the other hand, worse trajectory tracking as the demand use was decreased. A further fine-tuning of the controller could see a middle ground between the two. Another possibility is to discard the ILOS guidance and instead use PP guidance during the entire recovery phase, as the latter produces slowly-varying, almost constant heading references that are easier for the USV-Stinger system to follow. This does however assume the alignment phase to take the USV very close to the RV point, and that there are few disturbances pulling the USV away from the AUV-RV line. This would also remove two parameters, $\kappa$ and $\Delta$, which require tuning. While quite robust against it, the integral action in ILOS will inevitably produce *some* windup, which may negatively affect the performance of the control system as we most likely do not have time to unwind the integral term during the recovery phase. And the lookahead $\Delta$ must be tuned according to the yaw dynamics of the USV, which we have only been able to do in the simulator.

# 6 Conclusion, Discussion and Further Work

## 6.1 Conclusion

This thesis covers the development of a control system aiming to accomplish the recovery of an AUV using a USV, which has been tested both in a simulator and on the physical system. To make the existing simulator environment more similar to its physical counterpart, extensions such as time delay, measurement noise, and environmental disturbances have been modeled and implemented. Using the existing knowledge of the system and experience from the impact of the simulator extensions, two slightly different control systems were developed. These were tested on the physical system before the best performing one was further developed into a final control system. Lastly, the final control system was tested in a final physical test to evaluate its performance and ability to accomplish the recovery task, although without access to the AUV, whose pose was instead represented by a simulated fixed point for the physical tests.

The simulator extensions were very useful. Some of them, especially the delay and measurement noise, seem to closely resemble the behavior they attempt to model, reducing the difference between the simulator and reality. The environmental disturbances are more uncertain in terms of the modeling and parameter values, but are convenient for testing the control system's robustness against external disturbances.

The two control systems from section 4 could have benefitted from more development time, but served their purpose fairly well. They had a good performance in the simulator, covered somewhat different control methods for the recovery phase, and were expected to perform reasonably well in the physical tests.

In the first physical tests, the control systems were tested to evaluate their performance, with the most simple control system performing the best. The tests also uncovered some errors critical to the performance of the system, and showed flaws in the recovery strategy. The further development of the chosen control system fixed the errors, improved the strategy, and added some further upgrades. These enabled the control system to, in the simulator, better withstand the impact from environmental disturbances, produce heading references better suited to the slow yaw dynamics of the USV during the recovery phase, and follow its references more closely. The final physical tests showed improvements in terms of strategy and general performance, but the control system now struggled more with the initial phase of the recovery task. The final test run showed that the final control system can be improved, but may also be capable of accomplishing the recovery task.

In general, the author considers the goal of the thesis as partly achieved: Using the enhanced simulation environment and physical tests, a control system has been developed, which seems capable of accomplishing the recovery task under the conditions and requirements in section 1.3. Since we have not tested with the AUV, we cannot conclude whether or not the control system can accomplish the recovery, but from the physical tests, it looks promising.

## 6.2 Discussion

The results from section 5.3 indicate that the control system may be capable of accomplishing the recovery task. However, the physical tests were not executed exactly as an actual recovery would be, the most significant difference being that we did not have the AUV for the tests. Thus, we have only shown that the control system is capable of navigating accurately towards a fixed point, at a straight angle with little sideways error, and at the desired velocity. A large difference for a test with the AUV is that its pose would not be constant. The requirement of minimizing water jet-AUV interaction has also not been properly tested. For the final development in section 5.2, the control system was allowed to use more throttle than before, increasing the risk of induced water flow affecting the pose of the AUV.

Section 1.3 states that the control system should be able to handle some degree of environmental disturbances. While we have simulated with these, their presence was small in the physical tests. The simulations indicate that the system handles the impact from a constant wind force well, but we did not have much wind in the physical tests, and we have not tested the impact of wind gusts. For the second physical tests, ocean currents may have been a challenge, and potentially the reason behind the problems with reaching the RV point. And while we speculate in section 5.3.3 that this impact may have been smaller with the physical AUV if all components were affected similarly, an ocean current affecting the USV alone is still something the control system should be able to handle.

Section 4.1 listed four main challenges the control system had to address. Of these, *water jet model inaccuracy* and *parameter tuning*, which are somewhat related, have not been completely handled. There are some uncertainties regarding the model that converts demands to forces and moments, especially for the yaw state. This could be overcome with good tuning of the controllers, but in this thesis, we have not been able to tune them against the physical system. The control system with the SMC was discarded mainly for this reason, but the final control system is not immune to model inaccuracies or bad parameter values either. This was evident for the second physical tests where several controllers were not optimal. Having mainly been able to tune using the simulator, we were bound to have controllers with sub-optimal gains. Tuning, and in general developing a control system, only in a simulator has its limitations, and we would expect increased access to testing on the physical system to enhance the control system a lot.

One of the main hypotheses from the preliminary work was that the strategy, guidance, and path planning were more important than using the most advanced controllers. This still seems to be likely, though there are certainly many controllers we have not tried that could give good results. The final control system solely uses different PID controllers, some without the I or D-term, and they in general seem to serve their purpose, apart from some tuning issues. Revising our strategy, and adding integral action to the guidance law (in LOS), were powerful measures that significantly increased the performance.

For the final physical tests, the USV struggled to reach the RV point, often encircling it or approaching it in a snake-looking trajectory. The initial phase, whose aim is to reach this point, was assumed to be far easier than the recovery phase, leading to the focus mainly being on the latter phase. The author may have underestimated this phase. It is also worth mentioning that the two are related, and a strong alignment typically makes the recovery phase easier. Thus, more focus should be put on the initial phase. The slightly flawed strategy of reaching the RV point, as mentioned in section 5.2.2, could be a wise starting point for improving this part of the control system.

A central part of a PID controller, that has often been neglected in this thesis, is the integral term. The integral term removes stationary offsets, and could for many of our challenges seem like a good solution. The main reason for avoiding the integral term is the challenge of tuning and windup. The term adds another tunable parameter and creates a need for an anti-windup scheme. We would expect to not be able to remove the windup completely, and with relatively short distances between waypoints, the controllers would rarely have time to unwind, and thus the integral term may do more harm than good. But, the integral action is a potential improvement that could be explored.

There are still some uncertainties that require more testing in order to fully conclude, especially the ones related to the AUV. The control system has room for improvement, with better tuned controllers, a more robust strategy for the initial phase, and perhaps a further revised recovery strategy possible points for improvement.

## 6.3 Further Work

Following the discussion, there is still room for improvement, giving a foundation for possible further work. This concerns the simulation environment, the control system, and integration with other systems related to the maritime mine countermeasures project.

The simulation environment has in this thesis been extended, but can be improved and extended further. The models of the USV are fairly accurate, but especially the submerged Stinger model can be improved. The model is based on a displacement model designed for forward-moving vessels with a conventional hull, while the USV reverses during the recovery phase, and the Stinger in the water creates a much more complex object. This weakens the validity and accuracy of the model. It is probable that one can design a model that captures the dynamics of the USV and submerged Stinger better than the one in this thesis. One could also include more extensions to the simulator. A dynamic model of the AUV would be very useful. Such a model could cover how environmental disturbances and the water jets affect its pose, as well as collisions with the USV. Another possible extension is to model how the dynamics in roll and sway affect the three original states.

As discussed in section 5.2.2, the strategy of reaching the RV point is slightly flawed, as external disturbances may unable the USV to reach it the way the controller eq. (2.6) was designed. One could redesign the controller, but it is perhaps simpler to redesign the RV point instead. Since the main purpose of the RV point is to align the USV with the AUV at some distance in front of it, we could instead set the USV to reach the line formed by the RV point and the AUV. This way the USV may start the alignment phase a bit more or less than the 60 meters in front of the AUV, which does not matter much as long as the USV is positioned directly in front of it. With this approach, it is far less likely that the USV fails to reach the RV point (or RV line).

With the AUV not physically present in the tests, we naturally did not have any AUV-USV collisions either. But such collisions, which are believed to change the heading of the AUV a lot, are unwanted and must be avoided as much as possible. This could be done by adding an abortion scheme that aborts a recovery attempt mid-way if it is believed to fail, e.g. if the error is too large or the AUV's pose has changed. The USV can then instead return to the rendezvous point before re-attempting the recovery.

To accomplish the task fully autonomously, a way to detect a successful recovery is also needed. For a human, it is easy to see that the AUV has been attached to the Stinger, but the control system cannot see this as easily. A possible solution is to, when the AUV is suspected to be attached, drive slowly forwards, and check if the distance between the AUV and the USV is almost constant for some time. If that is the case, the Stinger and AUV are raised onto the USV, and the recovery is deemed successful. Another small task needed to complete full autonomy is communication between the control system and the Stinger. Since this did not work during the tests, the Stinger had to be submerged manually by an operator pressing a button, but is believed to be easy to fix.

Since the endpoint of the Stinger is the one that is to make contact with the AUV, we in principle want to control this point. In this thesis, we have controlled the CO of the USV, while controlling the heading to ensure that the Stinger makes contact with the AUV. [30] presents control methods for marine vehicles using the *hand position* approach, where one controls the motion of a point along the x-axis of the vehicle, just like our Stinger point behind the USV. Using this approach, one can control the surge and sway motion of the point using inputs in surge and yaw. This could be very relevant for our task. However, [30] assumes the surge speed to be positive, while we reverse during the recovery phase, which is the main reason why the hand position approach was not considered in this thesis.

Lastly, this thesis covers a subtask of a larger project, namely the maritime mine countermeasures project. Thus, the control system most likely requires integration with other FFI systems. Some of this was done during the physical tests, but closer integration with other modules may also be necessary. And while this thesis has had quite loose constraints and easy assumptions, such as no external obstacles, the final system might need to adhere to far stricter requirements.

# References

[1] Teknisk Ukeblad, “Forsvarets siste bestilling er en milepæl i overgangen til mer autonom krigføring,” https://www.tu.no/artikler/forsvarets-siste-bestilling-er-en-milepael-i-overgangen-til-mer-autonom-krigforing/378760, 2017, accessed 06-12-2021.

[2] V. Saraogi, “How will autonomy shape the UK shipping industry?” https://www.ship-technology.com/features/how-will-autonomy-shape-the-uk-shipping-industry/, 2019, accessed 20-09-2021.

[3] MI News Network, “Everything You Ever Wanted to Know About Autonomous Underwater Vehicle (AUV),” https://www.marineinsight.com/types-of-ships/everything-you-ever-wanted-to-know-about-autonomous-underwater-vehicle-auv/, 2016, accessed 20-09-2021.

[4] B. Coxworth, “Underwater robot designed to keep tabs on fish farms,” https://newatlas.com/marine/underwater-robot-cagereporter-fish-farms/, 2020, accessed 20-09-2021.

[5] U. Skoglund, “Førerløse ferger kan erstatte gangbruer,” https://gemini.no/2018/06/forerlose-ferger-kan-erstatte-gangbruer/, 2018, accessed 20-09-2021.

[6] Kongsberg, “Autonomous Underwater Vehicle, Hugin,” https://www.kongsberg.com/no/maritime/products/marine-robotics/autonomous-underwater-vehicles/AUV-hugin/, accessed 16-12-2021.

[7] Henriksen, “The Stinger - A Versatile AUV Launch and Recovery System,” https://hhenriksen.com/launch-and-recovery-auv-stinger/, accessed 20-09-2021.

[8] E. Sarda and M. Dhanak, “Unmanned recovery of an AUV from a surface platform manuscript, oceans ’13 MTS/IEEE San Diego,” in *2013 OCEANS - San Diego*, 2013, pp. 1–6.

[9] E. I. Sarda and M. R. Dhanak, “A USV-Based Automated Launch and Recovery System for AUVs,” *IEEE Journal of Oceanic Engineering*, vol. 42, no. 1, pp. 37–55, 2017.

[10] A. A. Proctor, Y. Zarayskaya, E. Bazhenova, M. Sumiyoshi, R. Wigley, J. Roperez, K. Zwolak, S. Sattiabaruth, H. Sade, N. Tinmouth, B. Simpson, and S. M. Kristoffersen, “Unlocking the Power of Combined Autonomous Operations with Underwater and Surface Vehicles: Success with a Deep-Water Survey AUV and USV Mothership,” in *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*, 2018, pp. 1–8.

[11] M. Breivik and J.-E. Loberg, “A Virtual Target-Based Underway Docking Procedure for Unmanned Surface Vehicles,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 13 630–13 635, 2011, 18th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016458143

[12] FFI, “Autonom minerydding,” https://www.ffi.no/forskning/prosjekter/autonom-minerydding, accessed 03-05-2022.

[13] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control.* John Wiley & Sons, Ltd, 2021.

[14] M. M. Hammad, A. K. Elshenawy, and M. E. Singaby, “The notation of SNAME (1950) for marine vessels.” https://plos.figshare.com/articles/dataset/The_notation_of_SNAME_1950_for_marine_vessels_/5179675/1, Jul 2017, accessed 28-09-2021.

[15] DMS, “Waterjets: When to Use, Pros and Cons,” https://dmsonline.us/waterjets-when-to-use-pros-and-cons/, 2019, accessed 24-09-2021.

[16] Boatswain's Locker, "HJ Series Waterjets," https://www.boatswainslocker.com/?page=customer&file=customer/boloin/customerpages/Commercial/WaterJets/HamiltonJet.htm, accessed 27-05-2022.

[17] Hydro International, "This Is How an Unmanned Launch and Recovery System Operates," https://www.hydro-international.com/content/news/this-is-how-an-unmanned-launch-and-recovery-system-operates, accessed 23-05-2022.

[18] ROS, "ROS," https://www.ros.org/, accessed 07-12-2021.

[19] B.-O. Eriksen and M. Breivik, *Modeling, Identification and Control of High-Speed ASVs: Theory and Experiments.* Springer International Publishing, 06 2017, pp. 407–431.

[20] IBM, "Overfitting," https://www.ibm.com/cloud/learn/overfitting, accessed 08-12-2021.

[21] Aceboater, "Displacement & Planing vessels," https://aceboater.com/usa/en/hull-designs-and-uses, 2016, accessed 28-09-2021.

[22] A. O. Akan, *Open Channel Hydraulics.* Elsevier Ltd, 2006.

[23] SciPy, "scipy.optimize.least_squares," https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html, accessed 30-09-2021.

[24] J. M. Smith, "Closer control of loops with dead time," in *Chemical Engineering Progress*, vol. 53, no. 5, 1957, pp. 217–219.

[25] Chriss Grimholt Sigurd Skogestad, "Should we forget the Smith Predictor?" *IFAC-PapersOnLine*, vol. 51, no. 4, pp. 769–774, 2018, 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control PID 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896318305007

[26] FFI, "USV for future maritime mine counter measures," FFI report [exempt from public].

[27] W. Blendermann, "Parameter identification of wind loads on ships," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 51, no. 3, pp. 339–351, 1994. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0167610594900671

[28] Thomas Stenersen, "Guidance System for Autonomous Surface Vehicles," https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2352498?locale-attribute=no, accessed 03-03-2022.

[29] Borhaug, Even and Pavlov, A. and Pettersen, Kristin Y., "Integral LOS control for path following of underactuated marine surface vessels in the presence of constant ocean currents," in *2008 47th IEEE Conference on Decision and Control*, 2008, pp. 4984–4991.

[30] Paliotta, Claudio, "Control of Under-actuated Marine Vehicles," https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2456310, Sep 2017, accessed 28-04-2022.

# A Modeling the USV

The models of the USV were developed using the approach that was developed in the preliminary work. Here we will go through the approach. The approach consists of the following:

1. Gather experimental data from the USV that captures as much of its dynamics as possible
2. Choose a model design, both structure and parameters
3. Use a non-linear solver to adjust the parameters of the chosen model, such that it predicts the states in the next time step as accurately as possible

For the model, its inputs were the current velocity in surge, sway, and yaw, the current thrust demands, and the length of the time-step. The goal of the model is to accurately predict the velocity in *surge*, *sway*, and *yaw* in the next time step.

## A.1 Experimental Data

The experimental data was gathered in the bay of Karljohansvern in Horten, Norway, the same place the physical tests were conducted. The data was gathered by having an operator on the USV set various thrust demands in all states, to get a broad variety of the dynamics of the USV. The data was gathered both with the Stinger raised and submerged, setting the same set of demands for both. We gathered data two times, such that one data set served as a training set, and the other as a test set.

A demand is set by sending a message on the form $\begin{bmatrix} X_{demand} \\ Y_{demand} \\ N_{demand} \end{bmatrix}$, with the desired demand for each state. Every once in a while we also sent a zero demand in order to get to 0 velocity. This was done to capture the entire transient response for some demands. All demands were also held for enough time that the USV reaches and for some time stays in steady state. Thus, we captured both the steady-state and the transient response in the data. A plot from one of the datasets is shown in fig. A.1.

## A.2 Model Design

In the preliminary work, the models had the form:

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \boldsymbol{\sigma}(\boldsymbol{\nu}) = \boldsymbol{f}(\boldsymbol{\tau}_d) \tag{A.1}$$

where $\mathbf{M}$ is the inertia matrix, $\boldsymbol{\sigma}(\boldsymbol{\nu})$ is a damping term, similar to the one in [19], $\boldsymbol{f}(\boldsymbol{\tau}_d)$ is a transfer function from demands to forces/moments, and $\boldsymbol{\tau}_d = \begin{bmatrix} x_d & y_d & n_d \end{bmatrix}^T$ denotes the demands for surge, sway and yaw. The purpose of $\boldsymbol{\sigma}(\boldsymbol{\nu})$ was to have a joint term that approximates the dynamics of $\mathbf{C}(\boldsymbol{\nu})$ and $\mathbf{D}(\boldsymbol{\nu})$ in eq. (2.1).

For this thesis, we developed revised models on the form eq. (2.1) as the model eq. (A.1) gave at times unrealistic behavior, especially under the influence of environmental disturbances.

## A.3 Training the Model

Given the current velocity and control inputs (thrust demands), we want the model to predict the velocity in the next time step. Using Euler's method to solve the differential model equation, the prediction is then computed as

$$\hat{\boldsymbol{\nu}}_{k+1} = \boldsymbol{\nu}_k + \delta_t \mathbf{M}^{-1}(\boldsymbol{f}(\boldsymbol{\tau}_d) - \boldsymbol{\sigma}(\boldsymbol{\nu})) \tag{A.2}$$

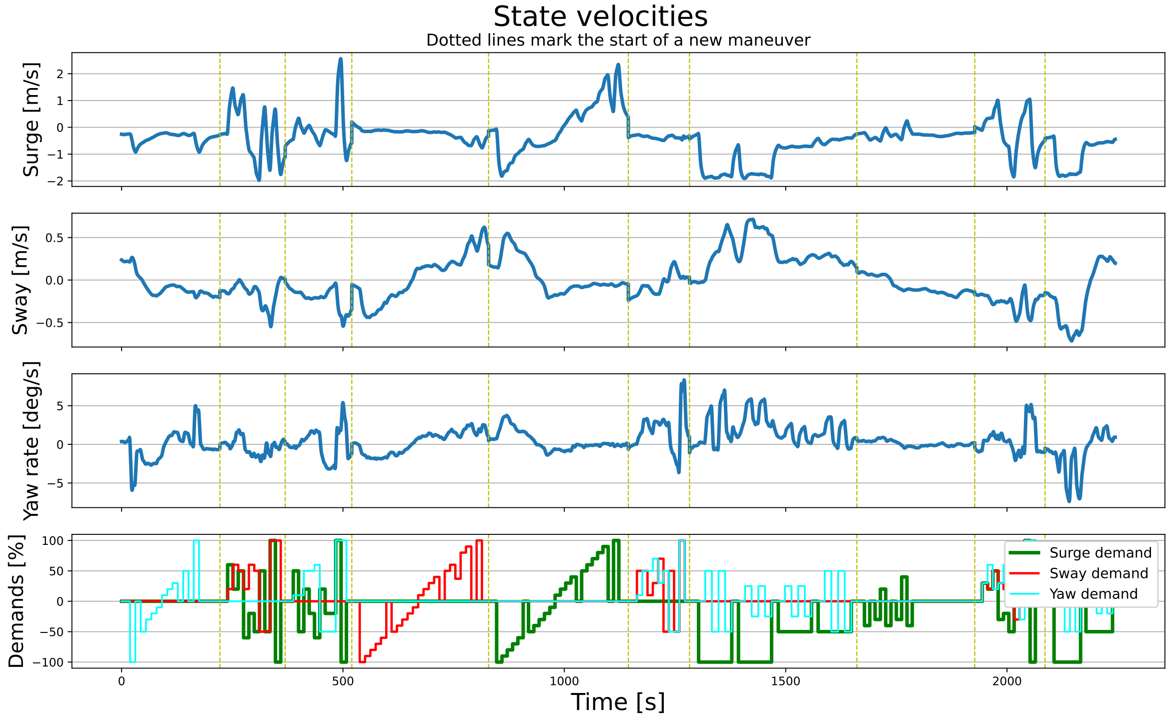


Figure A.1: The velocity and corresponding demands for the USV for the training set, with the Stinger submerged.

where $\hat{\boldsymbol{\nu}}_{k+1}$ denotes the predicted velocity, $k$ the current time step, $\delta_t$ the length of the time step, and we here have used eq. (A.1) as the model equation.

After designing the structure of the model, we need to train it by tuning its parameters. This is done using an optimization-based approach, where we use a non-linear least squares (NLLS) solver to approximate all tunable parameters at once. We let the model make predictions for each time step in the data set. The errors, that is the differences between predictions and ground truth, give the objective function that the NNLS solver is to minimize, and are given by

$$\begin{aligned} \mathbf{f}(\boldsymbol{\nu}) &= \begin{bmatrix} \mathbf{e}_0^T & \mathbf{e}_1^T & \dots & \mathbf{e}_{N-1}^T \end{bmatrix} \\ \mathbf{e}_k &= \hat{\boldsymbol{\nu}}_k - \boldsymbol{\nu}_k \end{aligned} \tag{A.3}$$

where $\hat{\boldsymbol{\nu}}_k$ and $\boldsymbol{\nu}_k$ are the predictions and ground truth for time step $k$, $\mathbf{e}_k$ is the corresponding error, and $\mathbf{f}(\boldsymbol{\nu})$ is the objective function the solver is to minimize. The solver will optimize the parameters in order to minimize the objective function until some convergence criteria is met, before returning the final values of the parameters.

NTNU

Norwegian University of
Science and Technology