Characterization and Control of a Saab Seaeye Thruster

by

M. Amos Buchanan
B.A., Reed College, 2002

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science

in the Department of Mechanical Engineering

Characterization and Control of a Saab Seaeye Thruster

by

M. Amos Buchanan
B.A., Reed College, 2002

Supervisory Committee

Dr. Bradley Buckham, Co-Supervisor
(Department of Mechanical Engineering)

Dr. Colin Bradley, Co-Supervisor
(Department of Mechanical Enigneering)

Supervisory Committee
Dr. Bradley Buckham (Department of Mechanical Engineering)
Co-Supervisor
Dr. Colin Bradley (Department of Mechanical Engineering)
Co-Supervisor

ABSTRACT

The use of Remotely Operated Vehicles (ROVs) in exploring and building infrastructure in the ocean is expanding. ROVs are performing tasks underwater that would be difficult or impossible to do with human divers. These vehicles are being used in increasingly complicated and demanding environments that require improvements in the methods for controlling these vehicles. Currently, research into semi-autonomous control is being conducted to aide ROV pilots in compensating for environmental disturbances and unknown dynamics. To effectively implement semi-autonomous control, precise thrust forces must be elicited from the thrusters.

This work discusses a low-level thruster controller that can be used as part of a semi-autonomous guidance, navigation and control system for a ROV. A thruster dynamics model describing the thrust force of a propeller-type underwater thruster was derived and implemented for the thruster on the Saab Seaeye Falcon ROV. The thruster dynamics model described is a quadratic equation that uses the propeller velocity to determine thrust force. This model includes a mechanism for compensation against the external motion of the thruster, such as occurs when the ROV moves through the water.

Several experiments were performed to fully characterize the quadratic thruster dynamics model and test its ability to accurately predict thrust force based on a known ambient water velocity and propeller angular velocity. The drag force was calculated and removed from the force measurements to get the thrust force used in the model. The model coefficients were determined and then the resulting model was tested against experimental data to determine the efficacy of the model in the lab environment and compare it to a widely used linear thruster dynamics model. The results showed the quadratic model improved upon the linear model, and the quadratic model was valid over a larger range of ambient water velocities.

The quadratic model was then inverted to provide a thruster control algorithm that determines the propeller angular velocity necessary to produce a desired thrust force. This algorithm was used to design a low-level thruster controller. This controller was designed to be used on an existing vehicle where thrust force feedback is not available and difficult or expensive to add. This allows it to be used in a wider range of applications than controllers that rely on such feedback to operate. The controller was implemented using a PID control loop to drive the angular velocity of the propeller to the desired rate. An iso-parametric mapping, which transforms the linear PID output to the non-linear thruster input, was added to provide a faster response time for the controller over the entire range of the propeller velocity. The performance of this low-level thruster controller was demonstrated in the test environment. The low-level thruster controller followed a desired thrust force under a range of ambient water velocities.

The thruster characterization and low-level thruster controller was designed to be used on an existing ROV. The motivation behind this work is to build a controller that may be imple-

mented for use by a high-level vehicle controller. The low-level thruster controller presented here does not depend on sensors or equipment that is largely unavailable on vehicles without costly retrofits, and the experimental characterization does not require intimate knowledge of the inner workings of the thruster. This makes it easy to implement and generalize to a variety of thrusters. The results of this work show a low-level thruster controller than can be used in a control schema for existing ROVs.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I've had a good deal of support in the course of this research, and I would not have been able to finish without the aide of several people in my life.

I'd like to thank my advisers, Brad Buckham and Colin Bradley. Brad spent many hours with me, pushing me to think critically about every choice I made and guiding my understanding of the analytical tools necessary to finish my research. Colin's guidance kept me focused and grounded, necessary to keep this research from ballooning out of control. They provided me with both the funding and knowledge I needed to realize my work from beginning to end. They also provided space and equipment I needed to conduct my research.

I'd like to thank Alison Proctor, who took me under her wing and guided me throughout my time here. With her help I was made the transition into mechanical engineering. I gained both knowledge and skills under her tutelage, and without her the research presented here would not have been possible.

I'd like to thank the people at the Ocean Technology Laboratory: Jeff Kennedy, Darryl Gamroth, Emmett Gamroth and Lori Muck. They gave me the tools I needed to construct a working apparatus. Jeff Kennedy imparted much of the knowledge and expertise that went into the design of my test apparatus. Emmett Gamroth and Darryl Gamroth provided assistance in the technical aspects of LabView. Lori Muck provided administrative support, keeping the wheels turning forward where it's not always immediately visible.

I'd like to thank the people who supported me in the fluids lab: Peter Oshkai and Mostafa Rahimpour. They gave me time on their flume tank, which allowed me to run my experiment. Without their training and equipment I would not have been able to finish.

I'd like to thank my parents for their support in my academic endeavours. Without their assistance I would not have made it through the last few years of school.

I'd like to thank all of my roller derby family, especially the crew of officials all over Vancouver Island and the islands nearby. They provided me with an outlet for my frustrations, and the confidence to keep learning and growing. They accepted me into their community, and gave me a home in a new place. I'd like to thank all of my blues dance family. They provided me with an escape and welcomed me into this new home I made for myself in this country.

And finally, I'd like to thank my friends that have supported me emotionally through many years, up to and including my return to school. Kathryn Bray and Candy Tan have spent many hours getting me through all the difficult times in my life, and without them I wouldn't have made it this far.

Chapter 1

Introduction

The ambitions of ocean scientists and engineers have grown rapidly in regards to the deep-sea environment. Using underwater remotely controlled vehicles have extended human presence to great depths for sustained periods of time. Environmental researchers install long-term data collection pods on the ocean floor to study tectonic shifting, cellular biology, ocean wildlife and ocean chemistry. Wave and tidal alternative energy platforms require innovative construction techniques and long-term maintenance. Human divers are capable of some of these tasks, but they are limited in depth and safety is a concern in high-flow environments or when working near operating machinery. These applications are driving the development of underwater vehicle technology, and these vehicles have become instrumental in the installation and maintenance of these structures. Underwater vehicles could also contribute to data collection as mobile sensor platforms, and to deliver instruments to the water column at specified times. They are emerging as a great boon to research efforts, but their use is still limited in scope.

Unmanned underwater vehicles are classified as either Remotely Operated Vehicles (ROVs) or Autonomous Underwater Vehicles (AUVs). The classification is delineated by the degree of independence from a human controller. AUVs do not have a physical connection to a human operator and include internal autonomous control systems that follow a preprogrammed mis-

sion. ROVs are connected to a Surface Control Unit (SCU) via a tether and a human operator controls the vehicle directly in real-time. The operator typically relies on video for feedback on vehicle position and this makes navigation very difficult in murky water. Traditionally, every aspect of the vehicle is controlled manually; for large ROVs this requires a team of operators working together.

The boundary that delineates AUV and ROV technologies is becoming fuzzier. Current research in ROV automation is being implemented which allows operators to focus on high-level control of the vehicle. On a traditional ROV, an operator must stabilize the vehicle by manipulating the thrusters directly. With only limited visual feedback, a pilot's response to a disturbance is often late. The situation is complicated by the irregular nature of the disturbances. For example, the tether exerts unknown forces on the ROV as it moves around; these forces increase in magnitude as the vehicle moves further away from the operator.

Installation of a sensor suite and a vehicle controller could mitigate the ROV operator's burden. Automating and abstracting low-level tasks allows more complex missions to be done using fewer operators. Using sensors mounted on the vehicle, the effects of currents and the tether may be compensated for immediately, rather than after the pilot visually detects the problem. A vehicle controller could use this information to correct the vehicle's motion without the operator's input. Operators could then focus on high-level tasks, such as mission planning, navigation and tool manipulation, while the autonomous controller keeps the ROV stable. However, such control requires accurate knowledge of the ROVs thruster dynamics; corrections to an ROV's motion requires the ability to deliver a desired thrust from each individual ROV thruster.

1.1 Remotely Operated Underwater Vehicles

All ROVs have similar characteristics:

Figure 1.1 – Approximate size difference between the XLX 150HP work class ROV (left) and the Seaeye Falcon observation class ROV (right).

- They are tethered to the surface and controlled by a human operator.
- They rely on multiple propeller-type thrusters to provide manoeuvrability.
- They often have an open-frame construction.
- They are equipped with underwater cameras to allow a human operator to observe the environment.

ROVs can be separated into two broad classes: work and observation. Work class ROVs are generally larger and are designed to carry heavy payloads and perform heavy underwater lifting. They have hydraulic manipulators for performing sub-sea construction, maintenance of pipelines and drilling support. Their operating depth often goes up to 3,000 m, allowing for deep water work that is impossible for human divers to achieve. Observation class ROVs are much smaller in size. They are equipped with a camera and occasionally have a small gripper arm. They can perform some small maintenance tasks, but are more useful for inspection operations. They are also limited to much shallower depths, often less than 300 m. Table 1.1 shows a comparison of an example observation and a working class ROV. The relative sizes of these two ROVs are shown in Figure 1.1.

Table 1.1 – Difference in specifications of two sample ROVs of different classes: the Forum Energy Technologies (FET) XLX 150HP and the Saab Seaeye Falcon.

FET XLX 150HP [17]

Saab Seaeye Falcon [11]

Class	Work	Observation
Width (m)	1.9	0.6
Length (m)	3.6	1.0
Height (m)	2.1	0.5
Weight (kg)	4750	60
Payload (kg)	300	14
Depth (m)	3,000	300

ROVs manoeuvre using multiple propeller-type thrusters working in conjunction. Figure 1.2 shows the thruster placement diagram for the Seaeye Falcon. The thrusters are placed in such a way that they can provide surge, heave and sway forces, a yaw moment, or a combination of all these effects. Manoeuvring the vehicle requires a controller to convert the desired direction from these effects into specific thrusts for each of the five thrusters. The high-level controller chooses a path based on these constraints, then a thruster allocation algorithm must determine the thrust required from each thruster. Once this allocation has been determined, each thruster is given a desired thrust.

A low-level controller on each thruster is needed to determine an input command for the thruster that will deliver the desired thrust, as shown in Figure 1.3. Different control algorithms choose to handle this low-level controller in different manners. Some high-level autonomous vehicle controllers use a closed loop low-level thruster control based on force measurements [5, 18]. However, force feedback is not always an option on a vehicle. Other high-level vehicle

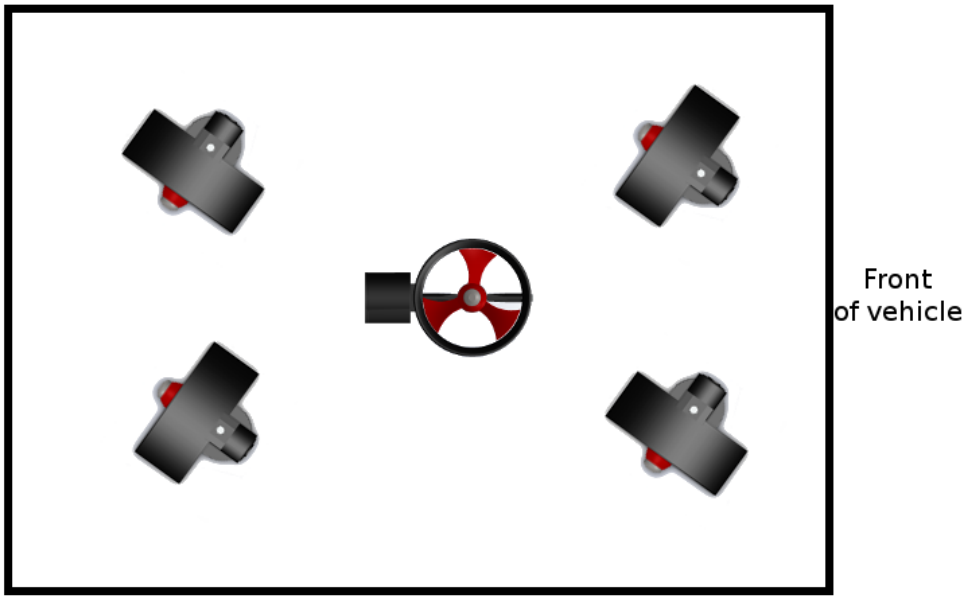

Figure 1.2 – Placement of the thrusters around the Seaeye Falcon.

controllers use a simple linear thrust model and apply a much more rigorous hydrodynamic characterization of the vehicle as a whole [2]. The focus in this approach is on the vehicle guidance and navigation rather than accurate low-level thruster control: inaccuracies in the thrust output are assumed to be compensated for in the high-level vehicle controller. However, a low-level thruster controller would be able to handle several issues that are particular to the thruster: deadband, nonlinearities in the thruster input and thruster saturation. Compensating these effects at the thruster should yield more accurate control of the ROV.

1.2 Review of Thruster Dynamic Models and Experiments

A thruster dynamics model is a model that uses known, derived or measured physical parameters to predict a thruster's response to a given control input. The physical characteristics of

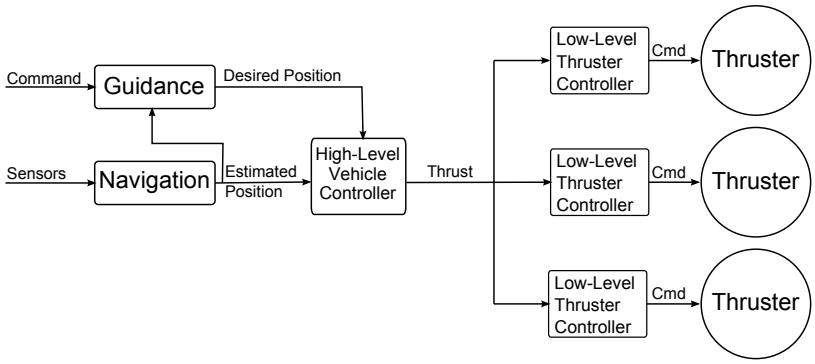

Figure 1.3 – The location of a low-level thruster controller in an autonomous control schema.

the thruster can be determined empirically through experimentation, but it is difficult to capture the full range of operating conditions the thruster may experience as it moves through the water. A model may use a any number of parameters, some examples of which are shown in Figure 1.4. Thruster dynamics models vary in their scope and applicability. Some of the models use detailed mechanical information about the thruster including motor inertia and armature windings. Other models will abstract those details into a semi-empirical model, which may abstract some of the detailed workings of the thruster itself into lumped parameter coefficients. Inverting the model allows the required thruster command to be determined for a desired thrust. The thruster dynamics model that is chosen may then be used to build a low-level thruster controller by performing this inversion. This section summarizes the experiments used by other researchers to determine the physical characteristics of thrusters that are used to create and verify thruster dynamics models. The next section reviews controllers that have been created using different thruster dynamics models. Details on different thruster dynamics models, including how the thruster dynamics models used in this work, have been built and may be applied are given in Chapter 2.

The experiments from other researchers tend to fall into one of two approaches:

- Physical identification experiments may be performed to empirically determine any missing values required for the model.

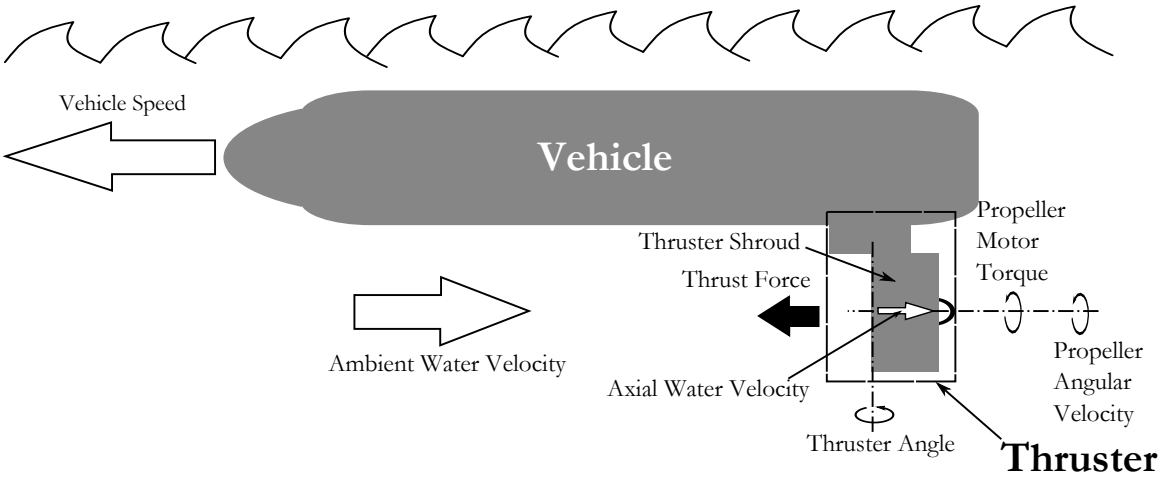

Figure 1.4 – An example of some of the physical parameters that may be used in a thruster dynamics model.

- The thruster may be modelled in a Finite Element Analysis (FEA) software package, which gives a detailed view of thruster parameters that are difficult to measure empirically, such as stress and deformation.

The experimentation in this thesis uses the former approach. The latter route is commonly used with intimate knowledge of the inner workings of the thruster, which is not the case for this work. However, it is instructive to study both approaches.

Cooke [4], and later Yoerger et al.[19], built a model based on the mechanical characteristics of a thruster and validated their model design with a custom test stand. They controlled the thruster's input voltage using their model, and measured the fluid velocity behind the thruster, the propeller angular velocity and the thrust. They validated their model using a straight bollard pull test. However, their test stand interfered with the flow of water through the thruster, though the degree this influenced the results is unknown. The testing method may cause inaccuracies in the model when the thruster is moved from the test stand to application on the ROV. When designing the test apparatus used in this work[1], minimizing interference with water flow was taken into consideration.

McLean [12] created a thruster dynamics model and verified it using experimentation. His

[1]See Chapter 3

model considered many of the properties of the thruster and environment such as: input voltage, current, friction, propeller torque, fluid momentum in the shroud and energy conservation. McLean's experiment used a thruster built to their specifications, which provided a great deal of knowledge of the physical parameters. McLean tested the thruster design with a software simulation, then built a prototype thruster to verify the simulations. Cody [3] expanded on McLean's work to add fluid acceleration and momentum changes over time. Cody then tested these additions to verify their applicability to the model.

Healey et al.[8] modelled the fluid moving through the thruster shroud and interacting with the propeller using airfoil theory. They looked at the drag and lift coefficients of the propeller as it interacted with the fluid to determine thrust. The disadvantage of this theory is that it relies on a detailed model of the fluid moving through the thruster shroud in order to apply the drag and lift coefficients. In their experiments, Healey et al. isolated the thruster as much as possible from any external effects, placing the test thruster into a small tunnel outside the influence of the test stand. This allowed them to tightly control the thrust for different inputs.

Bachmayer et al.[1] expanded on Healey's model to add the effects of rotational fluid velocity and inertia on the thruster response and to more accurately represent the lift and drag characteristics at each blade of the thruster. They used specialized sensors to experimentally determine the fluid flow characteristics through the propeller. Bachmayer et al. compared their expanded model to Healey's and showed the updated model was an improvement over Healey's model through experimentation. The drawback of their method was the precise measurements of fluid flow that were required to fully characterize the thruster.

Fossen and Blanke [7] also used experimental results to validate their model. They built a test thruster apparatus that put as little of the test stand into the water as possible. Their method seemed to give plenty of space for the thruster to rotate, which was necessary for their work. The experiments done by Kim and Chung [9], who built upon Fossen and Blanke's model, agree with Fossen and Blanke's data and results.

Muljowidodo et al.[13] designed and built a thruster for their ROV, the SHRIMP ROV-ITB. They based their design on the design requirements of their ROV. Their main consideration was with the thrust of the vehicle, both forward and backward. They used FEA software and statistical techniques to build and study their design. They built a scale model of their thruster and performed a bollard pull test to verify their design. Their testing focused on supporting the design generated in software rather than fully characterizing the thruster, making it difficult to expand their thruster model for general use.

Shea et al.[16] designed and tested a propeller and thruster custom designed for the Marport SQX-500. Their main testing goals were to show the performance and propulsive efficiency of the thruster. They used FEA software to test their design, as with Muljowidodo et al. The thruster motor was built based on appropriate RPM and torque values for their design. Once the motor was built, they performed a bollard pull test to verify their FEA results.

Shea et al. performed a water velocity test in a flume tank, but their test turned out to be unable to accurately verify their thruster dynamics model. The flume tank's cross-sectional area was too small for the propeller. Vortexes were created around the propeller blades, creating significant edge effects that affected their performance. The water speed varied significantly based on the depth of the tank. The water speed itself did not reach speeds consummate with real-world vehicle speeds. The problems they encountered in their tests were considered to help determine what problems may occur in the experimental setup of this work, and what errors to expect. Possible sources of error are discussed in Chapter 3.

1.3 Application of Models to Thruster Controllers

There are several thruster dynamics models to choose from and each of them depends on different levels of knowledge about the thruster and vehicle as a whole. A low-level thruster controller that produces a desired thrust using only commonly available sensors on an ROV is important for use in the field. This section gives an overview of how thruster dynamics models

fit into controllers used in a guidance and control system for a whole vehicle.

Cooke [4] analyzed thruster dynamics and created a high-level controller based on these dynamics. He built a sliding mode controller with lead compensation to follow a desired thrust profile. This closed-loop controller used a load cell to provide feedback about the output thrust. He used his results on a simulated vehicle to show that incorporating the thruster dynamics in the high-level control of an underwater vehicle significantly improved the vehicle's ability to follow a changing desired velocity. The limitation of his controller was that he used a load cell as part of the feedback loop; load cells are generally not available on a vehicle and it is difficult to retrofit a vehicle to add these sensors.

Leonessa and Poirrier [10] used a similar thrust-based closed loop low-level thruster controller. They used this low-level thruster control loop in a higher-level vehicle guidance system which tracks vehicle velocity. By including the vehicle dynamics and these measurements in addition to the thruster force, they were able to build a high-level controller that allowed a vehicle to track a desired velocity. The difference between this work and Cooke's controller is the separation of the low-level thruster control from the higher-level controller for following the vehicle velocity. However, this approach suffers from the same limitation as the previous one, in that it relies on a force measurement in the thruster control loop.

Fossen and Blanke [7] used a thruster dynamics model that did not depend on measuring the thrust force. They proposed a method that estimates thrust based on motor dynamics, measuring the propeller angular velocity and propeller torque. They built a low-level thruster controller that makes use of these measurements to follow a specified thrust force, which can be translated into an input command for the thruster. The benefit of this method is that it relies on values that are more likely to be obtainable on an existing vehicle. To further improve their low-level thruster controller, they added an observer for the fluid velocity in the vicinity of the propeller, also known as the axial water velocity. This observer is also based on the propeller angular velocity and propeller torque values. They showed through simulation that their low-

level thruster controller was able to follow a desired thrust using the observer even when the ambient flow conditions were changing. There are two difficulties with this approach. The first is the dependence on the propeller torque. A sensor to measure the propeller torque is not common on vehicles in use, so this may not be available. The second is that the observer is based on a linear approximation using a subset of the experimental data that corresponds with a specific operating regime. This is further expanded upon in Section 2.2.

Omerdic and Roberts [14] started from the same linear thruster model as Fossen and Blanke, but used simplifications to estimate the torque using the value for thrust force. They used this simplification to build a high-level controller for multiple thrusters working in conjunction. Their thruster model gets around the problem of requiring a propeller torque sensor in the controller. They tested their high-level controller in simulation and showed that it was able to adapt to changing environments. However, their equation for propeller torque as a function of thrust force neglects the effects of ambient water velocity. This ambient water velocity can have a significant effect on the dynamics of the thruster. Ideally, a low-level thruster controller can take into account both the thruster dynamics and the ambient fluid dynamics to create an accurate low-level thruster controller using commonly available sensors. This is the problem considered in this research.

1.4 Research Motivation and Objectives

Progress has been made in building and applying dynamic models for use in low-level thruster controllers, but many of them are difficult to apply in the field. Most ROVs do not provide measured thrust feedback from the individual thrusters, and an accurate profile of the thruster output is not available to the controller. This work seeks to develop and implement a thruster dynamics model on a small inspection class ROV: the Saab Seaeye Falcon. The thruster dynamics model will be used to control the thruster on this vehicle to provide a desired thrust. The goal is to find a model of the thruster that can be used to estimate the thrust using mea-

surements that are commonly available on thrusters for underwater vehicles.

A widely used model for thrust force is the linear model from Fossen and Blanke [7]. Kim and Chung [9] extended this model over a larger regime, as described in Section 1.2. The inputs required for this extended model are easily obtained without intimate knowledge of the thruster's construction. A controller based on this extended model that only uses the sensors available on the Seaeye Falcon can improve thruster control. An accurate open-loop controller for the Falcon would aide in accurate control of the vehicle by the guidance and navigation systems shown in Figure 1.3.

The objectives of this thesis are:

- Build two thruster models of the Seaeye thruster: a linear model based on Fossen and Blanke's work [7] and a quadratic model based on Kim and Chung's work [9]. These models are compared for efficacy and range of effective usage.
- Show how a thruster dynamics model may be inverted to create a practical low-level thruster controller for the thruster without extensive modifications to the vehicle.
- Test the low-level thruster controller to determine whether the controller is viable in practise.

1.5 Thesis Overview

This work reviews strategies for characterizing a thruster. An experiment to develop a thruster dynamics model for a thruster is designed and implemented. Using this model, a low-level controller is implemented. This controller is then tested on the thruster to provide a practical analysis.

The candidate thruster dynamics model is developed in Chapter 2. This model is based on physical models of the thruster and the water passing through the thruster shroud. The latter is difficult to measure, even in a controlled testing scenario. The thruster dynamics model

described in Chapter 2 uses a measurable ambient water velocity to estimate this value.

Having established the thruster dynamics model framework, Chapter 3 describes the experiments performed to characterize the Saab Seaeye SI-MCT01 thruster. A sensor frame was built for this experiment to determine the model's coefficients. The design of the equipment and the experimental methodology is also given in Chapter 3, including the sensor requirements and calibration.

The experimental results from the tests are presented in Chapter 4. This includes a description of the results from the measurements and how they are used. Analysis of the thrust profiles under a variety of ambient water speeds is described. Some previously undocumented characteristics of the thruster are also discussed in Chapter 4.

The process of exploiting the thruster dynamics model inside a thruster controller is described in Chapter 5. In the context of thrust control, the model is used to calculate a desired propeller angular velocity as a function of desired thrust and ambient water speed. A PID controller with an iso-parametric map is used to command the desired propeller velocity from the thruster.

The steady-state response of the resulting controller is presented in Chapter 6. This chapter compares several different control methods and shows the results. The steady-state thrust is measured and tested against the actual thrust to determine the efficacy of the controller. The working controller is characterized and the error in the desired output is described.

A summary of the major findings and contributions of the research is given in Chapter 7. This chapter concludes by considering future work using the results presented here.

Chapter 2

Thruster Dynamics Modelling

This chapter outlines the semi-empirical model used to predict the thrust force developed by the thruster. The primary hydrodynamic phenomena that occurs at the thruster are introduced and related to the dynamics of the ROV.

Section 2.1 provides the basic physical principles from which the model is derived, including the relationship between the ambient water velocity, the propeller angular velocity and the thrust force generated by the thruster. Section 2.2 translates the thruster hydrodynamics into linear equations relating the ambient water velocity, propeller angular velocity and the thrust force; and describes the dimensionless coefficients that are derived from empirical results. In Section 2.3, the linear model is expanded to a quadratic model derived from a physical approximation in the fluid dynamics. This quadratic model is further extended in Section 2.4 to account for different water flow states in the vicinity of the propeller.

2.1 Physical Model of Thrust Force

Thruster dynamics models start with the mechanics of a generic motor attached to a propeller. These models typically abstract the low-level mechanics of the thruster motor to a level that is more convenient to measure. This abstraction is used as the basis for a control schema. The

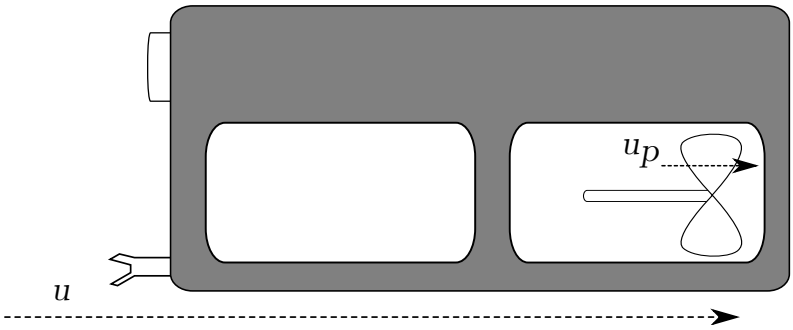

Figure 2.1 – The difference between the axial water velocity, u_p, ambient water velocity, u. The ambient water velocity can be caused by vehicle speed, water currents or discharge from other thrusters. All of these phenomenon are combined into ambient water velocity for the purpose of this work.

hydrodynamic properties of the fluids acting on the propeller are then added into the models.

McLean's work [12] provides a detailed model of a thruster motor using basic circuit analysis of a brushless motor. By considering the internal workings of the motor, an equation for the torque on the propeller shaft delivered by the motor as a function of the input voltage is created. This torque model is a function of the following factors:

- input voltage,
- friction inside the motor,
- shaft and propeller momentum, and
- hydrodynamic resistance against the spinning propeller.

McLean's thruster dynamics model gives an accurate representation of the thrust once it reaches a steady-state, and an accurate representation of the motor dynamics. However, in practise some of the model coefficients are difficult to measure. This model also requires detailed knowledge of the construction of the thruster motor. This knowledge is not always readily available.

For the hydrodynamic resistance, McLean's model uses momentum analysis to build an equation for the thrust force. The momentum of the fluid along the axis of the thruster is calculated as it is pushed through the thruster shroud, with the assumption that it is only the

thruster that causes the fluid to accelerate. The axial water velocity, u_p, is difficult to measure directly. This value depends on intrinsic factors such as the ambient water velocity, vehicle speed, thruster orientation and external factors related to the configuration of the open frame ROV. McLean's work does not consider these secondary effects, focusing only on the dynamics between the thruster propeller and the fluid.

Research building on McLean's work focused on calculating the thrust by estimating u_p using readily available values. Healey's work [8] applied the concepts of lift and drag from aerodynamics research to each blade of the spinning propeller. Equations are built for thrust force and propeller motor torque. Healey's work supposes certain knowledge of the motor characteristics that can be determined experimentally. He proposes a two-state thruster model that uses the motor angular velocity and the axial water velocity to drive all thrust calculations.

The work done by Yoerger et al.[19] takes a different approach than McLean [12]. Instead of calculating the inner workings of the thruster itself, only the momentum and inertia of the fluid moving through the vicinity of the propeller is considered. The torque of the thruster motor is measured directly and is used in the model. Yoerger et al. created a two-state model that used thrust force and propeller angular velocity as state variables.

The work by Fossen et al.[7] expanded Yeorger's model to create a three-state model, adding thruster motor torque as one of the states. Fossen et al.'s work assumed water to be an incompressible disk of fluid that is pushed by a propeller. This disk has an area, A_p, and a mass, m_f. The flow of water in the vicinity of the propeller is the axial water velocity, u_p, as defined by McLean. The work done by Fossen et al. used the momentum of this disk of fluid with density ρ to determine the force on the thruster. Part of the momentum calculation included the ambient water velocity, u. Ambient water velocity is the component of velocity of the water outside the immediate vicinity of the thruster, as shown in Figure 2.1. Fossen's

final equation for thrust force, T, is shown in equation (2.1).

$$T = k\rho A_p |u_p|(u_p - u) + f_l u_p + m_f \dot{u}_p \tag{2.1}$$

There are three parts in this equation:

- $k\rho A_p |u_p|(u_p - u)$, comes from the steady-state flow of the fluid through the propeller. This term is taken from the work done by McLean [12]. The value of k is a constant that Fossen et al. determined empirically.
- $f_l u_p$, corresponds to the linear skin friction of the fluid across the propeller, f_l.
- $m_f \dot{u}_p$, is the rate of change of the axial water velocity caused by the thruster.

One of the significant disadvantages of the model proposed by Healey is that it requires knowledge of the torque applied to the propeller shaft by the motor. This can be a difficult quantity to measure and is often not a standard sensor on the thruster. In addition, motor torque is an intermediate variable that must be translated to thrust force using hydrodynamic knowledge of the thruster's blades, the physical quantity that the controller is charged with regulating. The work done by Kim et al.[9] proposes an alternative approach that focuses on the ambient water velocity and the propeller angular velocity. This approach abstracts the mechanical model using measurements commonly available on-board a thruster to estimate the thrust force.

Equation (2.1) has terms for the linear friction across the propellers and the acceleration of the axial water velocity. Kim's work assumes a quasi steady-state condition. In normal operation the thruster is commanded at a slow enough rate that the propeller reaches a steady-state before the next command arrives. For this reason, it is assumed that the axial water acceleration is zero, i.e. $\dot{u}_p = 0$. In this quasi steady-state, the axial water velocity is fully developed and the linear friction is not separable from the thrust force. The final thrust force equation is given in equation (2.2).

$$T = k\rho A_p |u_p| (u_p - u) \tag{2.2}$$

The next section uses the hydrodynamics of the water moving through the propeller to build a model.

2.2 Thruster Dynamics and Linear Model of Thrust Force

The above-mentioned mechanical and fluid dynamics may be combined to produce a single set of state equations. This work focuses on a model of the thrust force that depends only on the fluid dynamics of the propeller. This is done for the following reasons:

- It is more practical from a controls perspective, the desired thruster control value is often based on the desired thrust force.
- Thrust force is easier to measure in a test environment than propeller torque.
- The thruster under consideration does not have a direct method of determining voltage applied to the thruster nor a direct relationship between input and torque.

Given these stipulations, it makes sense to build a controller based on the fluid dynamics. This work shows that this construction is valid. Equation (2.2) shows the steady-state thrust equation built from the fluid dynamics, but requires knowledge of the axial water velocity, u_p, which is difficult to measure. The following discussion describes how the model is constructed and how the issue of the unknown axial water velocity is circumvented.

Non-dimensional analysis is a widely used technique in dynamic modelling. In a non-dimensional approach, a coefficient is determined which can be used to relate thrust force to ambient water velocity, which can be adapted to thrusters with different physical characteristics. In the case of thruster dynamics, a non-dimensional thrust coefficient, K_T, is defined which relates the thrust force to the propeller angular velocity.

$$K_T(J_0) \triangleq \frac{T}{\rho D^4 \Omega |\Omega|} \tag{2.3}$$

The value of the thrust coefficient can be determined experimentally by measuring thrust force. The thrust coefficient K_T varies as a function of the advance ratio, J_0. The advance ratio wraps both the ambient water velocity and the propeller angular velocity into a single non-dimensional value that changes with the conditions of the propeller. It is a ratio of the ambient fluid velocity to the propeller angular velocity, defined in equation (2.4).

$$J_0 \triangleq \frac{u}{D\Omega}. \tag{2.4}$$

Once a model has been created that relates K_T to J_0, equation (2.3) can be rearranged to predict the thrust force for a given propeller angular velocity, as shown in equation (2.5).

$$T = K_T(J_0) \rho D^4 \Omega |\Omega| \tag{2.5}$$

The sign of the advance ratio gives some insight into the characteristics of the axial water velocity for any given values of u and Ω:

- In the case of $J_0 < 0$ and $\text{sgn}(u) = \text{sgn}(\Omega)$, the propeller is pushing water in the same direction as the ambient water velocity.
- In the case of $J_0 < 0$ and $\text{sgn}(u) \neq \text{sgn}(\Omega)$, the propeller is fighting the ambient water velocity.

The consequences of these cases are explored further in Section 2.4.

Plotting K_T vs J_0 allows a comparison of the performance of different thrusters. The work done by Fossen [6] proposed an empirically derived linear relationship between advance

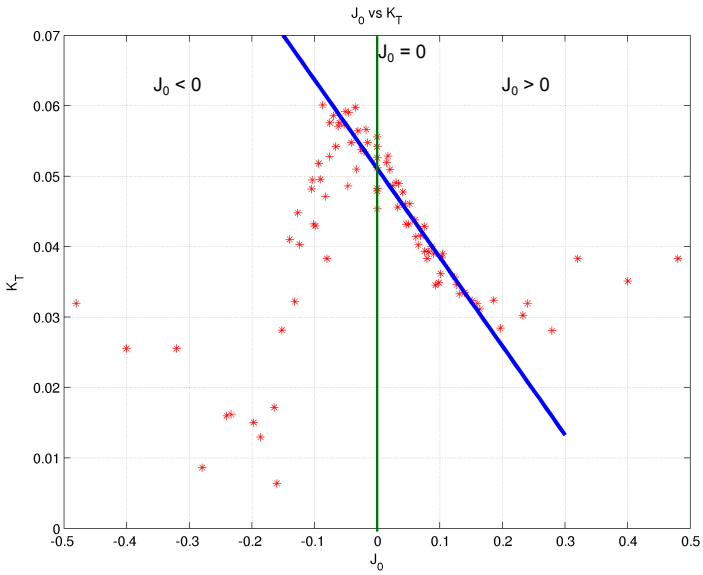

Figure 2.2 – A linear equation for the thrust coefficient, $K_T(J_0)$, that focuses on the case where $J_0 > 0$, as discussed by Fossen [6].

ratio, J_0, and the thrust coefficient, K_T, shown in equation (2.6).

$$K_T(J_0)=\alpha_1 J_0 + \alpha_2 \tag{2.6}$$

The constants α_1 and α_2 are determined using a linear approximation of the experimental results where $J_0 > 0$, shown in Figure 2.2. In this regime, ambient water is flowing in the same direction as the thruster is propelling the fluid, as previously stated. The linear form of equation (2.6) is a reasonable approximation for that regime, but it does not capture the thrust coefficient variation outside that regime. To address these limitations, Kim and Chung [9] proposed a different semi-empirical model, shown in the next section.

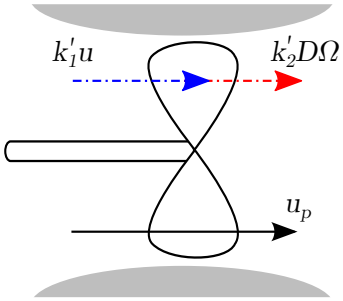

Figure 2.3 – The physical interpretation of the definition for axial water velocity, u_p [9].

2.3 Quadratic Model of Thrust Force

To address the limitations of the linear model, Kim and Chung [9] use a quadratic model. To justify this change, they first define a model for axial flow velocity as a linear combination of the ambient water velocity, u, propeller angular velocity, Ω and the diameter of the propeller, D, as shown in equation (2.7). The physical motivation for this equation is shown in Figure 2.3.

$$u_p \triangleq k_1' u + k_2' D\Omega \tag{2.7}$$

The coefficients k_1' and k_2' are determined experimentally. This definition is based on a linear combination of the ambient water velocity and the action of the propeller on the water. This is not the only way u_p could be defined, but it is sufficient for the steady-state equations, as shown by Kim [9]. If this definition for u_p is substituted into equation (2.2), a thrust equation for the propeller that depends only on ambient water velocity and propeller angular velocity is generated, as shown in equation (2.8).

$$T = k\rho A_p |k_1' u + k_2' D\Omega| (k_1' u + k_2' D\Omega - u) \tag{2.8}$$

This equation includes the water density, ρ, and the frontal area of the propeller, A_p. The frontal area may be written in terms of the diameter of the propeller, D, as shown in equation (2.9).

$$A_p = \frac{\pi}{4} D^2 \tag{2.9}$$

Equation (2.10) uses the substitution from equation (2.9) and rewrites equation (2.8) as a quadratic.

$$\begin{aligned} T &= \frac{\pi}{4}\rho D^2 (k_1 u^2 + k_2 u D\Omega + k_3 D^2 \Omega|\Omega|) \\ &= \frac{\pi}{4}\rho D^4 \Omega|\Omega| (k_1 \left(\frac{u}{D\Omega}\right)^2 + k_2 \frac{u}{D\Omega} + k_3) \\ &= \frac{\pi}{4}\rho D^4 \Omega|\Omega| (k_1 J_0^2 + k_2 J_0 + k_3) \end{aligned} \tag{2.10}$$

In equation (2.10), k_1, k_2 and k_3 are experimentally determined coefficients. This equation has been arranged in a form similar to equation (2.5), and suggests the quadratic model for $K_T(J_0)$, shown in equation (2.11), will be a better representation than the linear model from equation (2.6).

$$K_T(J_0) = \frac{\pi}{4} \left(k_1 J_0^2 + k_2 J_0 + k_3\right) \tag{2.11}$$

This research uses a slightly modified form, shown in equation (2.12), which rolls the $\frac{\pi}{4}$ into the coefficients k_1, k_2 and k_3.[1]

$$K_T(J_0) = k_1 J_0^2 + k_2 J_0 + k_3 \tag{2.12}$$

[1]This is for the case where the angle of attack is 0°. In the case of non-zero angle of attack, Kim has an expanded version of $K_T(J_0)$ that isn't produced here.

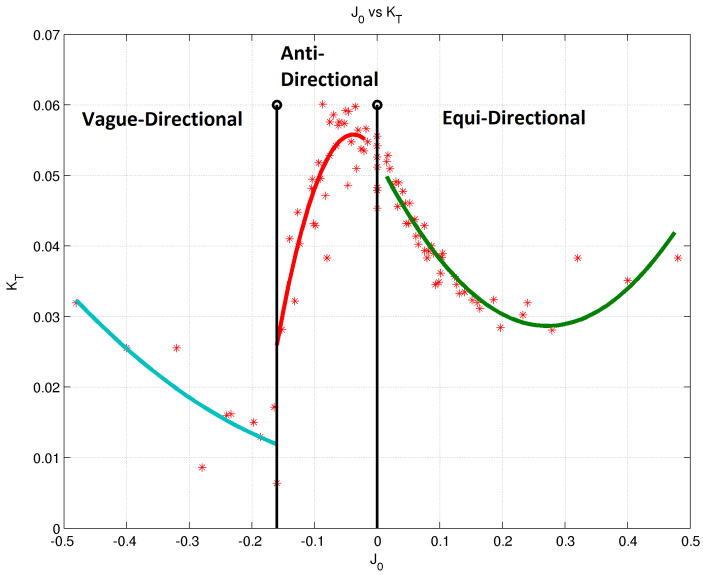

Figure 2.4 – Quadratic model for thrust coefficient, $K_T(J_0)$, divided into three states, as per experimental work done by Kim et al. [9].

As with the linear model, the values of k_1, k_2 and k_3 are determined empirically through experimentation.

Equation (2.12) fits the data much better than previous models. However, the experimental results, shown in Figure 2.4, have two discontinuities. To handle this, the operating envelope is divided into three states, and a thrust coefficient is defined for each region. This is justified by the difference in physics for u_p in these three states, which are summarized in the next section.

2.4 Three-State Extension of Quadratic Model

Using the experimental data shown in Figure 2.4 as a guide, Kim separates the $K_T(J_0)$ equation into three states: equi-directional, anti-directional and vague-directional. These states are shown in Figure 2.5 and can be related back to the definition of J_0 in equation (2.4).

equi-directional state: The ambient water velocity u is moving in the same direction as the axial water velocity, or $\text{sign}(u) = \text{sign}(u_p)$. In this case, $J_0 > 0$.

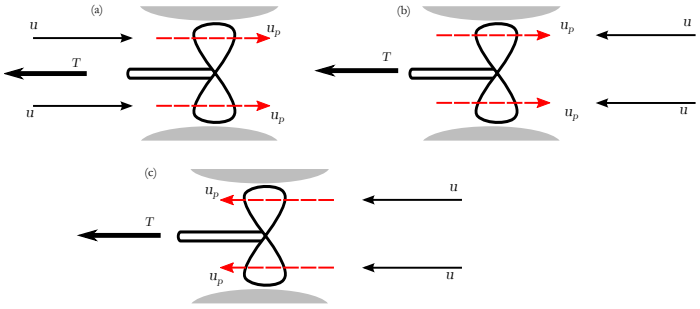

Figure 2.5 – Three flow states: (a) equi-directional state; (b) anti-directional state; (c) vague-directional state [9]

anti-directional state: The ambient water velocity is counter to the thrust force of the propeller, but the propeller is strong enough to overpower it. The axial water velocity is fully developed, with $\text{sign}(u) \neq \text{sign}(u_p)$. This is where the advance ratio $J_0 < 0$, but higher than a specific value of J_0, which is described below.

vague-directional state: The thrust force generated by the propeller is insufficient to counter the ambient water velocity. This is where the advance ratio $J_0 < 0$, and also less than the values of J_0 that comprise the anti-directional state. The axial water velocity, u_p, is not fully developed and is in an unknown state.

The equi-directional and anti-directional states are separated by $J_0 = 0$, and Kim introduces the Critical Advance Ratio (CAR), J_0^*, to separate anti-directional and vague-directional states. Using the definition of the advance ratio, J_0, from equation (2.4) and the axial water velocity, u_p, from equation (2.7), the three states may be differentiated as follows:

Equi-directional state:

$$J_0 > 0 \tag{2.13}$$

$$u_p = k_1' u + k_2' D\Omega > 0 \tag{2.14}$$

Anti-directional state:

$$J_0^* < J_0 < 0 \tag{2.15}$$

$$u_p = k_1' u + k_2' D\Omega > 0 \tag{2.16}$$

Vague-directional state:

$$J_0 <= J_0^* \tag{2.17}$$

$$u_p = k_1' u + k_2' D\Omega < 0 \tag{2.18}$$

There are also two special cases to consider. The first is the case of $J_0 = 0$, where the vehicle velocity $u = 0$. For this case, $K_T(J0) = k_1$. The other is where $\Omega = 0$, where $J_0 \to \infty$.

In Figure 2.4, the CAR, J_0^*, is the minima in $K_T(J_0)$ for $J_0 < 0$. This corresponds to the limiting thrust force as the ambient water velocity increases or the propeller angular velocity decreases. In the anti-directional state, where $J_0^* < J_0 < 0$, the thrust force is strong enough to completely over-power the ambient water velocity. When the advance ratio is than J_0^*, the thrust force generated drops off as the thruster cannot overpower the ambient water velocity, which corresponds to the vague-directional state.

The value of J_0^* may be determined by setting $u_p = 0$. Putting this into equation (2.7), a relation for Ω may be found.

$$\Omega = -\frac{k_1' u}{k_2' D} \tag{2.19}$$

Combining equations (2.19) and (2.4), the value of J_0^* may be found:

$$J_0^* = -\frac{k_2'}{k_1'} \tag{2.20}$$

The values of k_i' in equation (2.20) relate back to the experimentally determined values of k_i from equation (2.7). Using the derivation from equation (2.8) to equation (2.10), this relationship is shown in equations (2.21) and (2.22).

$$k_1' = \frac{1 \pm \sqrt{1 + 4k_1}}{2} \tag{2.21}$$

$$k_2' = \pm\sqrt{k_3} \tag{2.22}$$

This method is used in Chapter 4 to fit the experimental data to the model presented here.

Chapter 3

Thruster Testing Setup and Plan

This chapter outlines the experiment performed to characterize the thruster. The experimental data gathered provides an accurate representation of the thrust force characteristics of the thruster independent of the vehicle. The thruster under test is the Saab Seaeye SI-MCT01, shown in Figure 3.1, which is used on the ROV Saab Seaeye Falcon. This data is used to identify a thruster dynamics model based on the ambient water velocity and the propeller angular speed. A summary of this process is given in Section 3.1.

First, a sensor frame was built to collect the data for this experiment, shown in Figure 3.2. The sensor secures the thruster and measures the forces exerted by the thruster, and the propeller angular speed. Detailed descriptions of the equipment are given in Section 3.2.

Several tests were performed to validate the data and determine how well the experimental data fits the model. These tests were designed to calibrate the thruster's internal measurements and verify that the model accurately determines thrust force. The full test plan is described in Section 3.3. The results from the testing described in this chapter are shown in Chapter 4, and used to create controller described in Chapter 5.

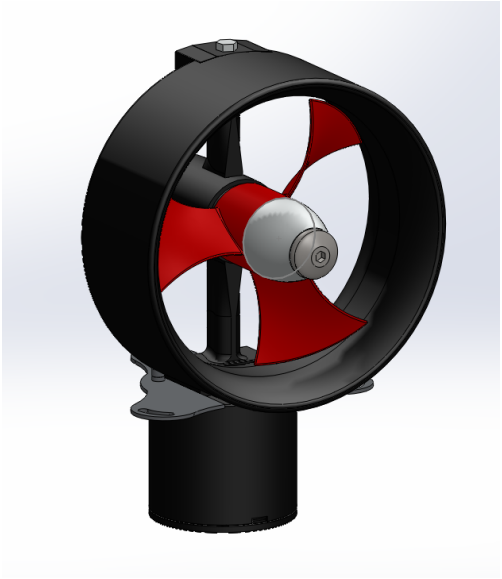

Figure 3.1 – The SI-MCT01 thruster for the Seaeye Falcon.

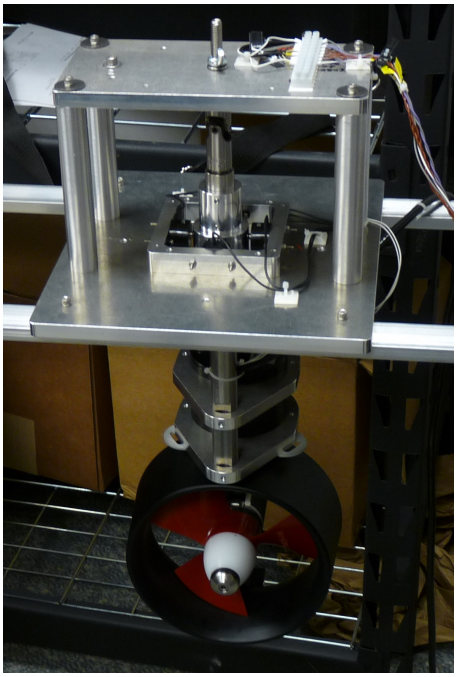

Figure 3.2 – The full sensor frame with thruster.

3.1 Overview of Experiment

This section provides a high-level overview of the experiment. This includes what physical parameters are required and what data is collected to determine those parameters.

The thruster dynamics model uses three equations to predict thrust force. These are equations (2.12), (2.4) and (2.5), which are reproduced below.

$$K_T(J_0) = k_1 J_0^2 + k_2 J_0 + k_3 \tag{3.1}$$

$$J_0 = \frac{u}{D\Omega} \tag{3.2}$$

$$T = K_T(J_0)\rho D^4 \Omega|\Omega| \tag{3.3}$$

The experiment was designed to collect the physical parameters necessary to determine the k_i coefficients for equation (3.1). As described in Section 2.4, there are three sets of coefficients: one set for each of the equi-directional, anti-directional and vague-directional states. The physical quantities required to calculate these coefficients are summarized in Table 3.1. These quantities include:

Thrust Force: This is the total thrust force, T, that the thruster generates. This is used to determine the thrust coefficients for equations (3.3) and (3.1).

Propeller Angular Velocity: The propeller angular velocity, Ω, is used to calculate the advance ratio, as well as estimate the value of the thrust coefficient.

Ambient Water Velocity: The ambient water velocity, u. This value is used directly in the model in equation (3.2).

The steps to calculate an estimated thrust using these values are:

- Calculate the advance ratio, J_0, from equation (3.2),
- Use this advance ratio to determine the state of the system: equi-directional ($J_0 > 0$), anti-directional ($J_0^* > J_0 > 0$) or vague-directional ($J_0 < J_0^*$),

Table 3.1 – List of physical quantities required by the model.

Variable	Units	Description
T	N	Thrust Force
Ω	rad/s	Propeller Angular Velocity
u	m/s	Ambient Water Velocity

- Calculate the thrust coefficient, $K_T(J_0)$, using the advance ratio and the correct coefficients for the state with equation (3.1).
- Estimate the thrust force using the thrust coefficient with equation (3.3).

The sensor frame is shown above in Figure 3.2. This sensor frame contains the sensors and thruster connections required for the experiment, as well as securing the thruster. In addition to the sensor frame, there is a real-time PXI, a control computer and power supplies as part of the setup. This setup uses both a free-standing tank, and a flume tank that can create an ambient water velocity.

The physical quantities listed above were calculated from the measurements taken using the sensor frame. These measurements are summarized in Table 3.2, and described in detail below.

Thruster Reported Speed: The thruster reports a speed, ω, that is proportional to the propeller angular velocity. Thruster output is described in Section 3.2.1.

Thruster Speed Scaling Factor: The thruster reported speed is related to the propeller angular velocity by a scaling factor, which is denoted by γ. The relationship between thruster reported speed, thruster speed scaling factor and the propeller angular velocity is detailed in Section 3.2.1.

Thruster Current: This is the electric current drawn by the thruster, denoted by I. This is measured directly by the thrusters. Thruster output is described in Section 3.2.1.

Table 3.2 – List of measurements required by the design.

Variable	Units	Description	Notes
ω	—	Thruster Reported Speed	Section 3.2.1
γ	—	Thruster Speed Scaling Factor	Section 3.2.1
I	A	Thruster Current	Section 3.2.1
(T_x, T_y)	N	Thrust Coordinates	Section 3.2.2
Ω_{rd}	rad/s	Independent Propeller Angular Velocity	Section 3.2.3
ζ	%	Thruster Input Command	Section 3.2.1

Thrust Coordinates: The thrust force was measured along two axes, T_x and T_y. This was used to determine total thrust force. The sensors used for these measurements are described in Section 3.2.2.

Independent Propeller Angular Velocity: This is an independent measure of the propeller angular velocity, Ω_{rd}. This value is used to determine the thruster speed scaling factor, γ. This measurement is discussed in Section 3.2.3.

Thruster Input Command: This is the control command for the thruster, ζ. The thruster input command is a percentage input varying between -100 and 100. Thruster commands are described in Section 3.2.1.

3.2 Equipment Description and Calibration

This section describes the equipment used in the experiment. The details of how it is used to gather the measurements described in the previous section are specified. In addition, the calibration and setup required by the equipment is described.

The sensor frame was designed in SolidWorks® and custom machined for this experiment. This sensor frame attaches to the thruster and holds all the sensors, as shown in the block diagram in Figure 3.3. It holds the thruster below the water, while providing some water protection to the electronics above. Underwater drag was minimized as part of the design.

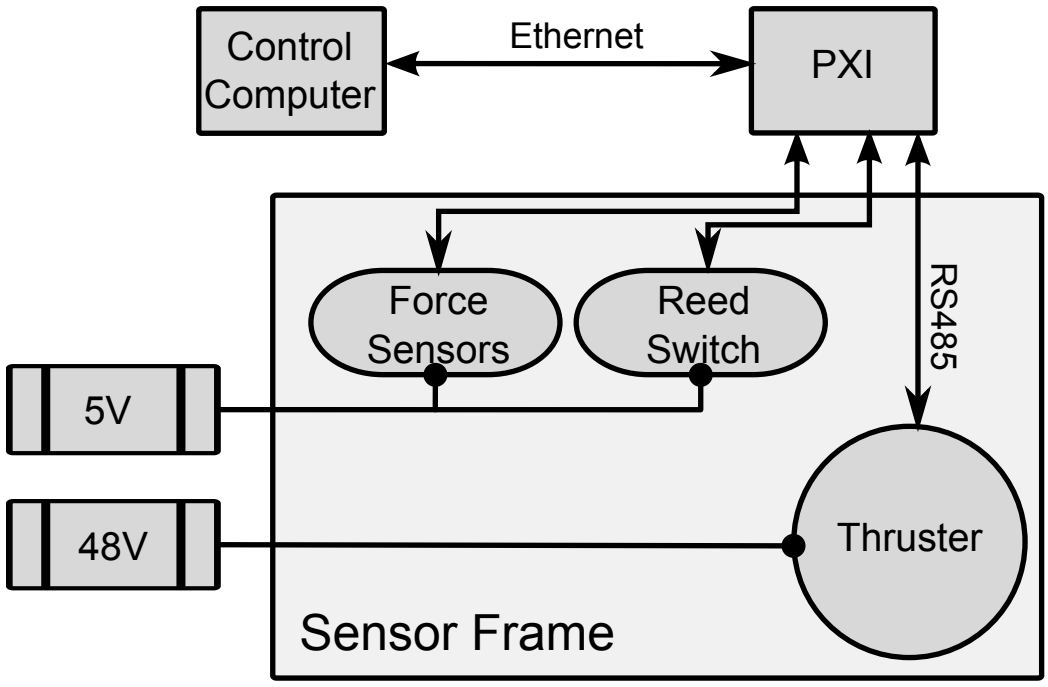

Figure 3.3 – Box diagram of the setup.

The frame was designed to maximize rigidity and to minimize vibrations while the thruster is running. More discussion on drag, vibrations and how they are compensated for is provided in Chapter 4.

The experiment uses two water tanks, a free-standing tank of water and a flume tank. The free-standing tank was used for propeller angular velocity tests. The flume tank provides a channel of water where an ambient water velocity may be specified. Descriptions of the tanks and how they were used are presented in Sections 3.2.5 and 3.2.6.

Sensors to measure the force applied by the thruster and moving water were installed in the sensor frame. Four sensors were added to get data along both the x-axis (along the direction of the water flow) and y-axis (perpendicular to the direction of water flow). The force sensors are attached to the sensor frame as shown in Figure 3.4. The force sensors and their calibration are detailed in Section 3.2.2. The sensor data was recorded using the National Instruments®real-time PXI running LabView®. All of the data was time-stamped to ensure it was properly synchronized for analysis. The complete list of equipment used is shown in Table 3.3.

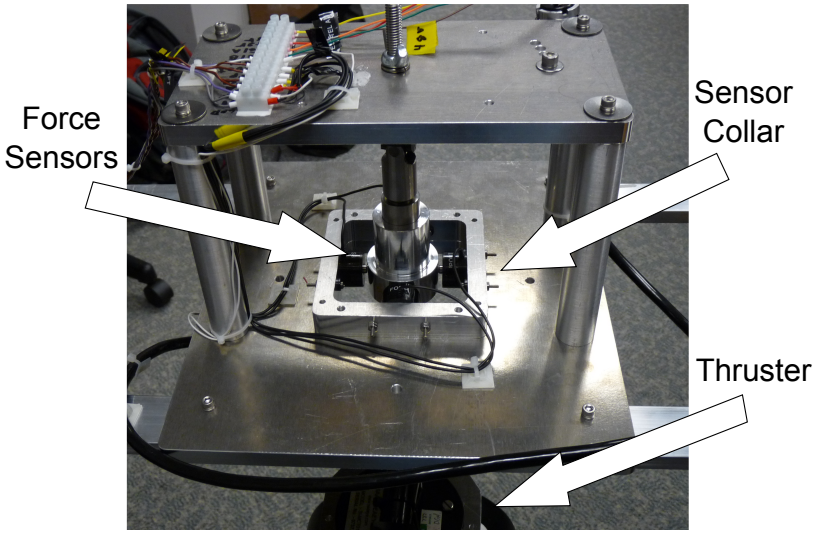

Figure 3.4 – Collar and sensor frame for the thruster.

Table 3.3 – Summary of equipment used in the experiment. Setup diagram is shown in Figure 3.3.

Item	Description	Notes
Sensor Frame	Frame that holds the thruster and the sensors.	Figure 3.2
Seaeye Thruster	Thruster under test in this experiment.	Section 3.2.1
Force Sensors	Four sensors to measure the thrust force.	Section 3.2.2
Reed Switch	Used for independent measure of propeller angular velocity.	Section 3.2.3
PXIe-1071	Real-time computer from National Instruments®(NI).	
PXI RS485	Serial communications board from NI.	In PXIe-1071
PXI-6254	Data Acquisition (DAQ) board from NI.	In PXIe-1071
Control Computer	Used to control the thruster and collect data from the sensors.	
48V Power	Thruster power source.	
5V Power	Force sensor/reed switch power source.	
Flowmeter	Used to measure the ambient water velocity while testing in the flume tank.	Section 3.2.4
Flume Tank	Water channel that can create an ambient water velocity.	Section 3.2.5
Free-Standing Tank	Water tank used for propeller angular velocity testing.	Section 3.2.6

3.2.1 Command of Thruster

The thruster is shown above in Figure 3.1. The thruster communicates through a RS485 serial connection to a National Instruments PXI real-time computer. The PXI communication software sends the thruster input command, ζ, and requests the thruster reported speed, ω, and motor current, I. There is a measurable lag between sending a command and receiving a response. This lag is measured and discussed in Chapter 4.

The thruster input command is a value in the range $[-100, 100]$, where -100% corresponds to maximum positive thrust and 100% is the maximum negative thrust.[1] There is an internal controller on the thruster which converts this into thrust power, the details of which are undocumented. The experiment will test how the thruster's controller responds to step inputs.

The thruster reported speed, ω, is also queried. This reported speed is not a physical value, but is a scalar proportional to the propeller angular velocity, and is defined by equation (3.4).

$$\Omega = \frac{2\pi}{60}\gamma\omega \tag{3.4}$$

The thruster speed scaling factor, γ, must be determined experimentally. To determine the value for γ, an independent measure of the propeller angular velocity was completed using a reed switch, which is further discussed in Section 3.2.3.

3.2.2 Force Sensors and Calibration

The thrust force is measured using four load cells, two for each of the x- and y- axes. These are placed on the frame as shown in Figure 3.4 above. These load cells must be rated to measure the maximum thrust force, the drag force of the water against the thruster and any forces resulting from the torque created by the sensor frame itself. These forces are illustrated in Figure 3.5.

[1] This convention is motivated by the thruster's orientation on the ROV.

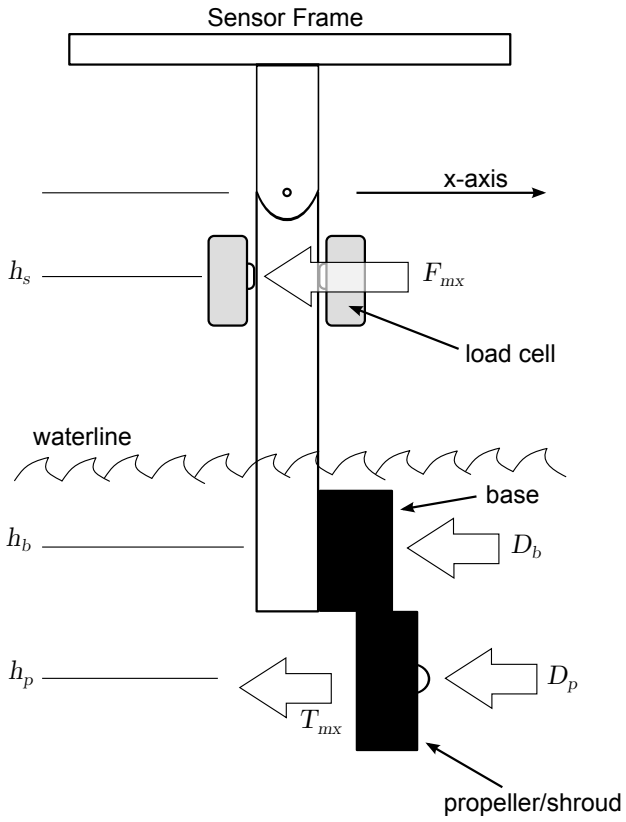

Figure 3.5 – Force diagram of the sensor frame along the x-axis. The load cell must be able to measure the sum of the maximum expected drag and thrust forces. The forces create a torque in the arm of the sensor frame, which must be accounted for.

The maximum thrust force expected is based on the thruster's documentation and shown in Table 3.4. A drag profile of the thruster was calculated by splitting the thruster into two sections and determining the drag of each separately. These sections are the propeller and shroud assembly, and the thruster base, as shown in Figure 3.5. The general equation for the drag, D, of each section is given in equation (3.5).

$$D=\frac{1}{2}C_d\rho u^2 A_F \tag{3.5}$$

In this equation, A_F is the frontal area of the section and C_d is the drag coefficient. The drag

coefficient is dependent on the shape and size of the section. The values calculated here are upper bound estimates. In place of the ambient water velocity, u, the maximum vehicle speed is used, which is given in the documentation as 1.2m/s. The moment arm of the sensor frame must also be taken into account when calculating the maximum expected force on the load cell. This moment arm is shown in Figure 3.5. The full calculation is given in equation (3.6).

$$F_{mx}=\frac{h_b}{h_s}D_b+\frac{h_p}{h_s}D_p+\frac{h_p}{h_s}T_{mx} \tag{3.6}$$

Since the x-axis is the primary axis for the thruster, the maximum thrust force along the x-axis is the maximum thrust force of the thruster. This is the thrust force value used for T_{mx}. These values are presented in Table 3.4.

The sensors chosen for the x-axis direction are FC23 compression load cells made by Measurement Specialties®. These have a maximum rating of 8800 N, which is sufficient as per the maximum limits in Table 3.4, with a resolution of 0.27 N. This load cell was chosen because it has the smallest resolution while still able to measure the largest expected force.

For the y-axis sensors, a similar calculation is made. The ambient water speed is only along the x-axis direction, so this is not considered in the maximum force calculation. The maximum angle, θ_m, for the y-axis direction is 0.349 rads (20°), as designed in the sensor frame. Taking this into account, the maximum thrust force expected in the y-axis direction is given by equation (3.7).

$$T_{my}=T_{mx}\sin(\theta_m) \tag{3.7}$$

The maximum force expected on the load cell after the moment arm is taken into consideration is given in equation (3.8).

$$F_{my}=\frac{h_p}{h_s}T_{my} \tag{3.8}$$

Table 3.4 – Values for the forces on the sensors.

Variable	Value	Units	Description
C_{db}	1	-	Drag coefficient for the base section
A_b	0.0072	m^2	Frontal area of the base section
C_{dp}	1.28	-	Drag coefficient for the propeller section
A_p	0.0547	m^2	Frontal area of propeller section
ρ	1025	$\frac{kg}{m^3}$	Density of water
u	1.5	$\frac{m}{s}$	Maximum ambient water velocity
D_b	8.30	N	Maximum drag force of base section
D_p	80.79	N	Maximum drag force of propeller section
D	89.09	N	Total drag force on thruster
h_b	322.6	mm	Height from axis to centre of base
h_p	482.6	mm	Height from axis to centre of propeller
h_s	114.3	mm	Height from axis to centre of force sensors
T_{mx}	127.49	N	Maximum thrust force exerted by thruster
T_{my}	43.60	N	Maximum expected y-axis thrust force exerted by thruster
F_{mx}	902.83	N	Maximum force expected along x-axis on load cell
F_{my}	184.11	N	Maximum force expected along y-axis on load cell

Table 3.5 – Slope results for each sensor.

Location	Value	Units	Notes
X-Axis Positive	0.090	N/V	FC23 sensor
X-Axis Negative	0.092	N/V	FC23 sensor
Y-Axis Positive	0.018	N/V	FC22 sensor
Y-Axis Negative	0.018	N/V	FC22 sensor

The sensors chosen for the y-axis direction were FC22 compression load cells by Measurement Specialties®. These have a maximum rating of 440 N. This is sufficient in the y-axis direction, as per the values in Table 3.4. The resolution for this sensor is 0.01 N.

The sensors were calibrated before use. They output a voltage linearly proportional to the applied force. The slope of each was determined by measuring the output to a series of known weights starting from 0 N. A line was fit to the data using a least-squares approximation. The y-intercept was calculated at the time of data collection, as detailed in Chapter 4. The resulting graphs for calculating the slopes are given in Figure 3.6, and the values are given in Table 3.5.

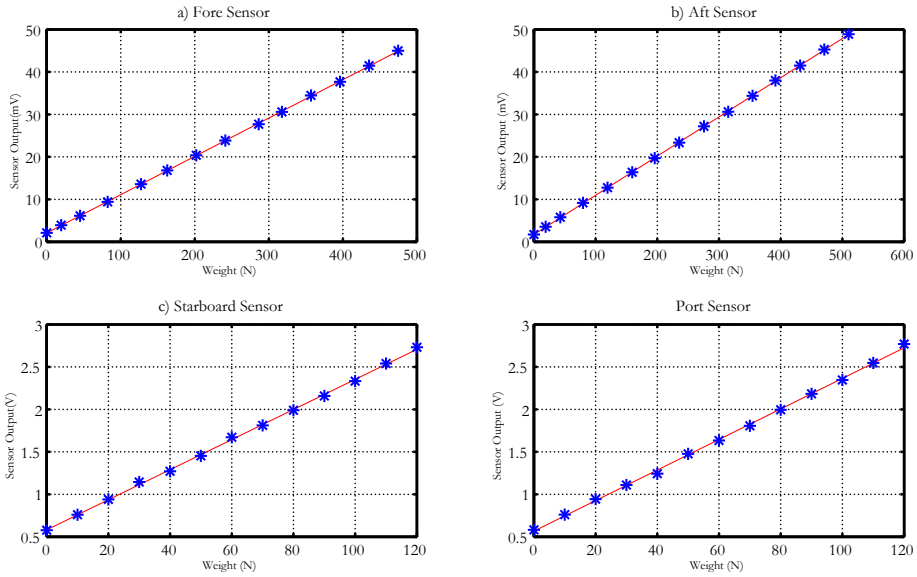

Figure 3.6 – Results of the force sensor calibration for each of the four sensors.

3.2.3 Propeller Angular Velocity Measurements

As described in Section 3.2.1, the thruster reports a value that is proportional to the propeller angular velocity: the thruster reported speed. The thruster speed scaling factor, γ, is introduced to transform the thruster reported speed into the propeller angular velocity, as shown in equation (3.4). An independent measurement of the propeller angular velocity was required to determine the value of the thruster speed scaling factor.

Independent measurement of the propeller angular velocity was done using a reed switch and three magnets. A reed switch consists of a pair of ferrous metal reeds that come into contact when a magnetic field is applied. Rare earth magnets were glued in three places around the thruster, roughly equidistant from each other, as shown in Figure 3.7. They were placed behind the blades to minimize water flow impact. They're 2.60 mm high by 6.35 mm in diameter and were assumed not to impact axial water velocity, u_p, in a measurable way. The reed switch detects the magnets as a simple on/off gate. The voltage is measured via the PXI DAQ expansion card. The reed switch and magnets are attached to the thruster as are shown

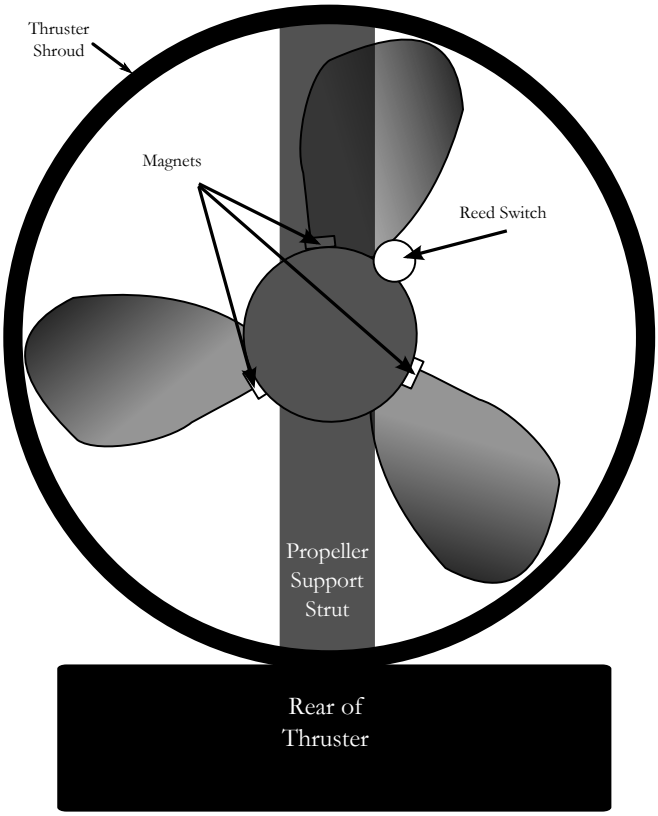

Figure 3.7 – Diagram showing placement of the magnets and reed switch, from the rear of the thruster.

in Figure 3.8.

A possible source of error is interference from the thruster itself. The thruster collar is magnetic and the motor itself is brushless and uses electromagnets to spin. Out-of-water testing was done to ensure the thruster operation did not interfere with the reed switch detection of the magnets.

Sample output from the reed switch is shown in Figure 3.9. There are two methods of determining the propeller angular velocity from this data:

- Measure the time between two spikes from the same magnet. This measures the time of a single revolution of the propeller and gives an average propeller angular velocity over a single revolution. This measurement is more accurate for the steady-state condition and is not dependent on the direction of the propeller. It is less accurate while the propeller

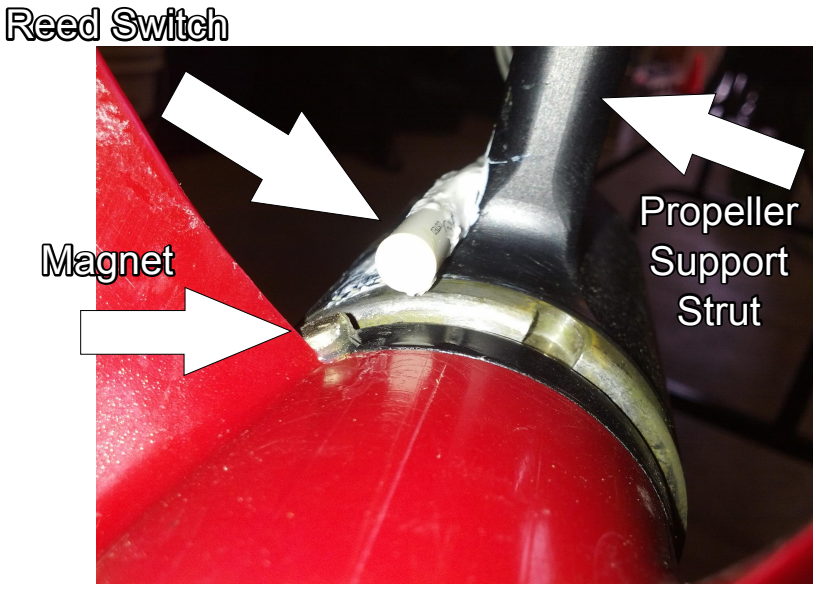

Figure 3.8 – The magnets and reed switch attached to the thruster, from the front of the thruster.

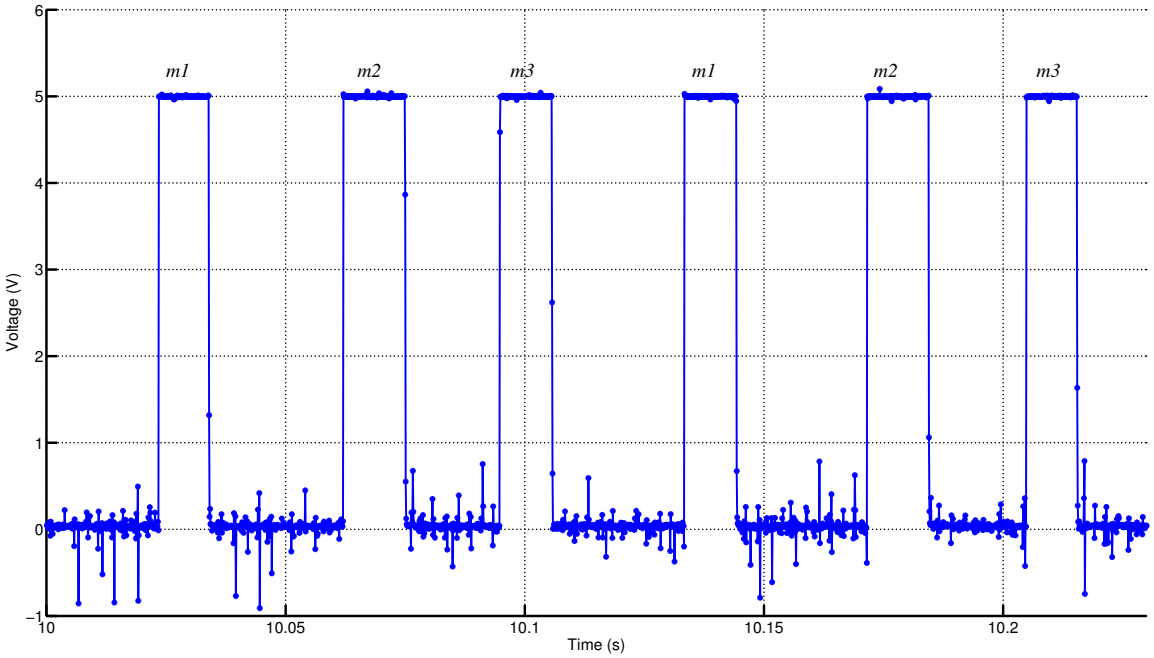

Figure 3.9 – Example of output from the reed switch. The plateaus correspond to the three magnets: m_1, m_2 and m_3.

Figure 3.10 – The flowmeter used to measure ambient water velocity.

velocity is changing.

- The second method is to measure the arc length of each magnet. These values are not equal, as can be seen in Figure 3.9. They may be measured while the propeller is in a steady-state velocity. The length of time it takes the propeller to pass through the arc of influence from a magnet gives a measure of the propeller angular velocity for the distance of that arc. This may also be done for the non-magnet sections, giving a total of 6 measurements per revolution instead of just one. This method is more accurate when measuring the propeller angular acceleration, but these measurements are also noisier and subject to significantly more error.

For the purpose of this work, the first method is chosen. The measurements were done while the thruster is in a steady-state condition, and the first method provides a more accurate value for the thrust in this case.

3.2.4 Ambient Water Velocity Measurement with Flowmeter

A flowmeter, shown in Figure 3.10, was used to measure the steady-state speed of the water flow in the flume tank. It has a minimum measurable flow speed of 0.1 m/s. The flowme-

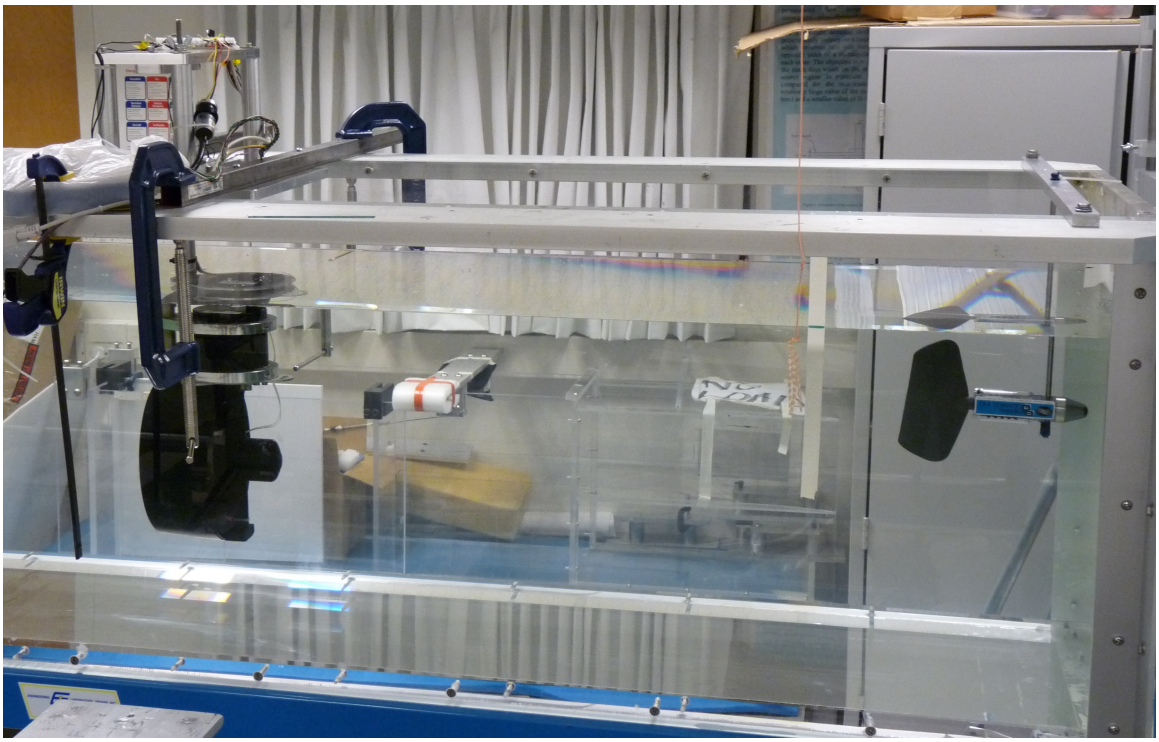

Figure 3.11 – The flume tank.

ter works by incrementing a counter as the rotor rotates in the water channel. The equation for finding the ambient water velocity given the counts, cnt, and time, t, is provided by the documentation and reproduced in equation (3.9).

$$u = \frac{5,756,000}{999,999}\frac{cnt}{t} \tag{3.9}$$

When the water channel in the flume tank is at steady-state, the flowmeter is put into the water and allowed to spin for 5 minutes. The number of revolutions is recorded and used in equation (3.9). The accuracy of the measurement increases with the time spent measuring. Five minutes is long enough to reduce measurement error to $err < 1\%$ for all measurements.

3.2.5 Flume Tank

The flume tank consists of a motor that drives water at a steady state down an open water channel, shown in Figure 3.11. It was used to set the ambient water velocity, u, for testing. The flume tank has discrete speeds that can be set. The flowmeter described in Section 3.2.4 was used to measure the speed at each step. The thruster was tested first in the forward direction, then rotated 180° and tested in the reverse direction.

The tank is 2.5 m long. The cross-sectional area of the flume tank is 482 mm wide by 432 mm deep. This gives a minimum of 126 mm clearance between the sides and the thruster. This should be sufficient to discount edge effects for the propeller interacting with the walls. In addition to this, the shroud around the propeller helps prevent these edge effects as well.

There is a water velocity gradient dependent on the proximity to the side walls. This will cause a slight moment at the thruster's attachment point. The sidewall effects are assumed to be symmetric and opposite along the y-axis, negating their contribution. Another gradient is due to depth; the water velocity increases as depth decreases. When considering drag, this puts the centre of the drag point higher than that calculated in Section 3.2.2, which has an effect on the force on the x-axis sensors. There is no way to measure this velocity gradient directly. This has two effects:

- It increases the maximum force on the force sensors. However, this effect will not cause the maximum force to rise above the maximum sensitivity force measurement of the x-axis sensors.
- This may affect the drag measurements. However, the force due to drag is removed in the final analysis, so it is ignored.

The drag force of the thruster is discussed in further detail in Chapter 4 below.

The maximum ambient water velocity used was 1.2 m/s. This is the maximum speed of the vehicle according to the documentation. In practise, the vehicle is unlikely to reach this speed.

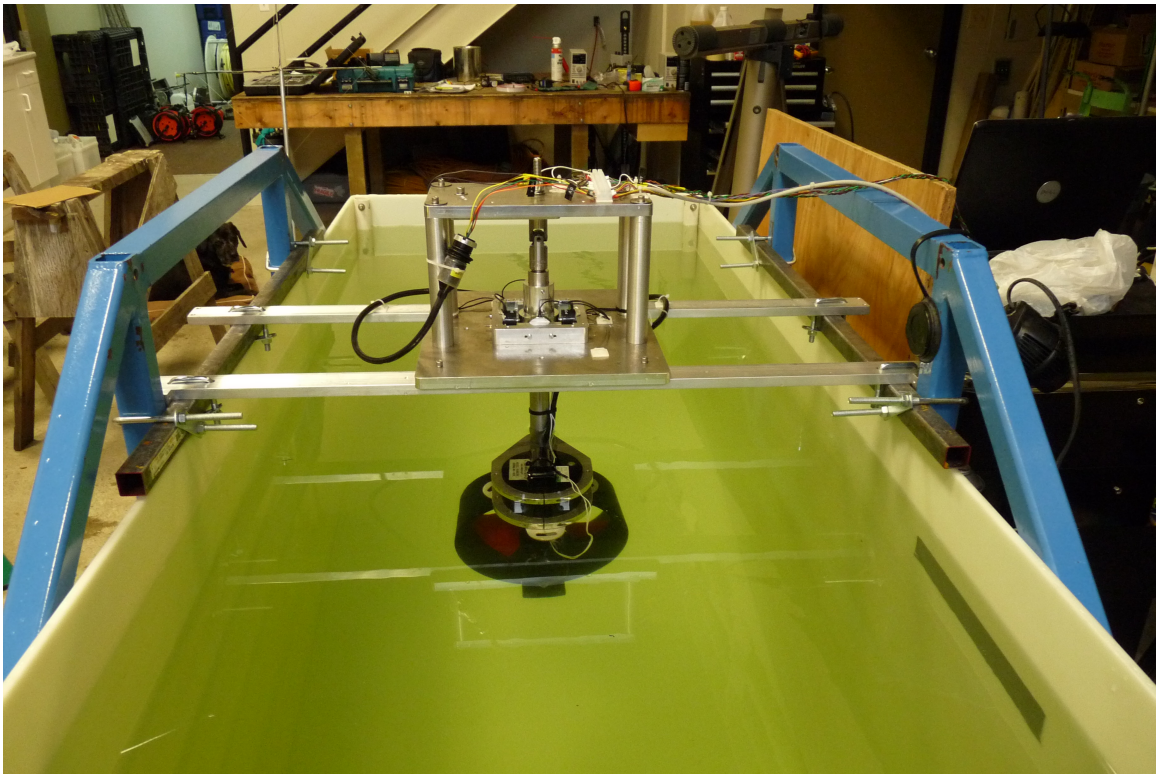

Figure 3.12 – The free-standing tank used in propeller angular velocity tests.

As the ambient water speed increases in the water channel, there is an increased likelihood of near-surface effects in the propeller at high propeller velocities. Pulling air through the propeller prevents the thruster from imparting full thrust. This caused spikes in the thrust force data at high ambient water velocities. Near-surface effects on the analysis is further discussed in Chapter 4 below.

3.2.6 Free-Standing Tank

A free-standing tank was used for some of the tests, shown in Figure 3.12. The tank is a plastic bin with dimensions 0.8 m x 1.8 m x 1.0 m. The sensor frame is attached to the metal frame of the tank. This tank was used primarily for propeller testing; all force measurements were taken with the flume tank described in Section 3.2.5. However, tests involving the reed switch, thruster motor speed, deadband calculations and the steady-state propeller angular velocity

were done in the free-standing tank.

3.3 Test Plan

This section outlines how the equipment described in Section 3.2 was used to generate the data needed for the model.

Thruster reported speed scaling factor: As discussed in Section 3.2.1, the scaling factor for the thruster reported speed, ω, must be scaled to determine the propeller angular velocity, Ω. This scaling factor must be determined experimentally. This was done through the independent measurement of propeller velocity, Ω_{rd}. This test was performed in the free-standing tank. The thruster was run at steady-state for several different thruster input commands ranging from $[-100, 100]$. The scaling factor was verified to be the the same for the representative propeller angular velocities. The full results and analysis are provided in Section 4.1.

Propeller angular velocity and ambient water velocity: The propeller angular velocity, Ω, is dependent on two variables: The thruster input command, ζ, and the ambient water velocity, u. In the literature, the contribution by the ambient water velocity was negligible and ignored. However, this experiment confirmed that for the Seaeye thruster there is a strong dependency on the ambient water velocity that cannot be discounted. The thruster was tested at 10% increments across a range of ambient water velocities. This created a full map of the dependency. The results of this test are shown in Section 4.2.

Deadband calculation: The range of the deadband in the thruster was measured. There are two different types of deadband: static deadband and dynamic deadband. Static frictional forces resist against moving the propeller until a certain thruster input command is reached. The static deadband is the range of inputs between starting the propeller from a stopped position to rotating. Once the propeller is rotating, the smaller kinetic

friction and the angular momentum keep the propeller rotating for smaller values of thruster input command. This is the dynamic deadband, which is smaller than the static deadband. Both of these deadbands were measured. The static deadband was measured by incrementing the thruster input command ζ in 1% increments from 0% until the propeller started moving. This was done in both the positive and negative ζ directions. The dynamic deadband was tested in the opposite direction. A thruster input command outside the static deadband was given, causing the propeller to rotate. Then the thruster input command was reduced until the propeller stopped spinning. This was done for both positive and negative thruster input commands. The results and a discussion of how the deadband affects controller design is given in Section 4.3.

Thrust force tests: The model for the controller calculates the thrust force as a function of the propeller angular velocity and the ambient water velocity. The steady-state thrust force for a given propeller velocity and ambient water velocity is measured. These values were used to determine the coefficients for $K_T(J_0)$, from equation (2.12). This data was also used to determine the Critical Advance Ratio (CAR), J_0^*. The measurements were done in the flume tank at a set of water speeds representative of the ROV's expected range. The thruster inputs used were across the range of $[-100, 100]$ at 10% increments. This test comprised the bulk of the data required for the model.

The results from these tests and the application of the model to the thruster are provided in Chapter 4.

Chapter 4

Results and Analysis of Thruster Tests

This chapter presents techniques used to post-process the data gathered in the experiments, and the main results of the experiments. These findings are used to calculate the coefficients required for the linear model from equation (2.6), and for the quadratic model from equation (2.12). This chapter is divided by the steps used to derive each result. Section 4.1 describes calculating the propeller angular velocity from the reed switch, which is used to scale the thruster reported speed. The relationship between thruster input command and the propeller velocity under different ambient water velocities is shown in Section 4.2. The measurements for the thruster's deadband are shown in Section 4.3.

Section 4.4 provides a discussion on the choice of a thrust datum, including an outline of the multiple ways to interpret the thrust. Force sensor bias and drag of the thruster is calculated in Section 4.5. A discussion on near-surface effects and their implications is given in Section 4.6. The results of calculating the coefficients for $K_T(J_0)$ in each of the linear and quadratic models is given in Section 4.7. These results are then used in determining and comparing the thrust force across the two different models in Section 4.8. Finally, the characteristics of the thruster's internal controller are outlined in Section 4.9.

4.1 Propeller Velocity Scaling Value

As discussed in Section 3.2, the thruster reported speed is proportional to the propeller angular velocity as shown in equation (3.4). Since this measure is intrinsic to the thruster, it is desirable to use this in the controller for the propeller velocity. To translate between the thruster reported speed and the propeller angular velocity the thruster speed scaling factor, γ, was measured. It was also verified that the thruster speed scaling factor is constant across the range of thruster input command. Once this was verified, the thruster speed scaling factor itself was calculated.

The propeller angular velocity is measured independently using the reed switch. The output from the reed switch measurements is converted to this independent propeller angular velocity, and is denoted by Ω_{rd} to distinguish it from the value of the the propeller angular velocity calculated from the thruster reported speed. To determine this factor from the independent propeller angular velocity reported by the reed switch, Ω_{rd}, the relation in equation (4.1) is used.

$$\gamma = \frac{60\Omega_{rd}}{2\pi\omega} \tag{4.1}$$

Equation (4.1) is applied for every instance of a (Ω_{rd}, ω) pair and a regression analysis is used to get a best-fit γ value.

The output from the reed switch is a square wave, as shown in Figure 4.1. The propeller angular velocity from the reed switch is determined by measuring the time between two detections of the same magnet. The distance between these two points is one revolution, which can be translated to the independent propeller angular velocity measurement as shown in equation (4.2). The time-stamp, ts, for each data point is given by equation (4.3). The time-stamp is used to synchronize the propeller angular speed measurement with the thruster reported speed. The initial time, t_o, corresponds to the first measurement where the output voltage is above 4.5V, and the final time, t_f, is the first measurement where the output voltage is below 3.5V.

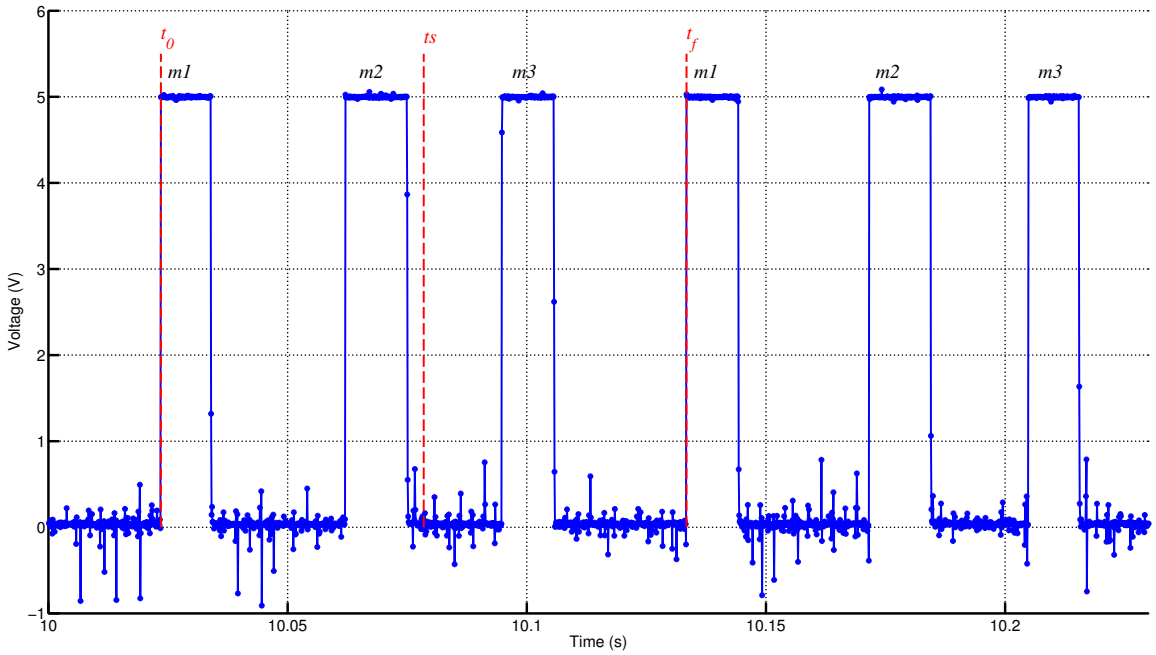

Figure 4.1 – Output of the reed switch, which is a square wave. Each plateau corresponds to a single magnet closing the reed switch, as shown above. The length of time between two measurements of magnet 1, $m1$, are shown in this figure by t_0 and t_f. The timestamp for this magnet is shown by ts.

This process is done for each magnet in turn, which provides a resolution of 3 data points per revolution.

$$\Omega_{rd} = \frac{360\pi}{2\pi(t_f - t_0)} \tag{4.2}$$

$$ts = t_0 + \frac{t_f - t_0}{2} \tag{4.3}$$

Using this method, a measurement for the independent propeller angular velocity, Ω_{rd}, can be compared to the thruster reported speed, ω. An example of the difference between the independent propeller angular velocity and the thruster reported speed is shown in Figure 4.2. To determine the scaling factor, propeller angular velocity data from the free-standing tank tests was used. Steady-state data was taken from six thruster input commands: -100%, -50%,

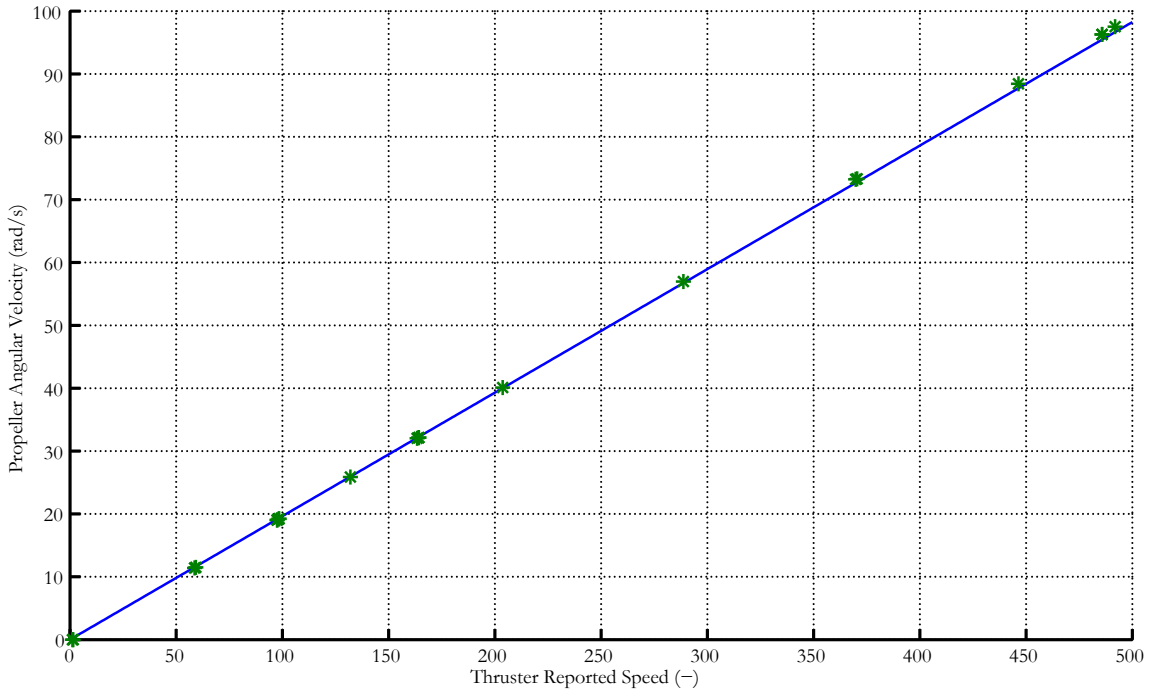

Figure 4.2 – The thruster reported speed, ω, compared to the independent measure of propeller angular velocity, Ω_{rd}, by absolute values. The line indicates the best fit using the Thruster Speed Scaling Factor, γ.

-30%, 30%, 50% and 100%. The ratio of measured RPM to thruster reported speed was determined for each. The mean of these ratios was used to determine the correct thruster speed scaling factor. This data is shown in Table 4.1.

Based on this data, the scaling factor does not change appreciably with different input values of ζ, and the mean value was used for all inputs, as shown in Table 4.1. This scale factor is used in calculations involving propeller angular velocity for the rest of the results, as well as building the controller as described in Chapter 5.

4.2 Thruster Input Command and Propeller Velocity

The thrusters used in experiments by Fossen and Blanke, and Kim et al. had a direct relationship between thruster input and propeller angular velocity, with negligible dependence on the ambient water velocity. This property seemed to be true of most thrusters described in the

Table 4.1 – Thruster speed scaling factor, γ for different thruster input commands, ζ.

ζ	γ	Std Dev
−100%	0.1983	0.95e-3
− 50%	0.1961	0.66e-3
− 30%	0.1953	1.03e-3
30%	0.1953	1.17e-3
50%	0.1958	1.20e-3
100%	0.1982	1.27e-3
Mean Value	0.1965	0.0133

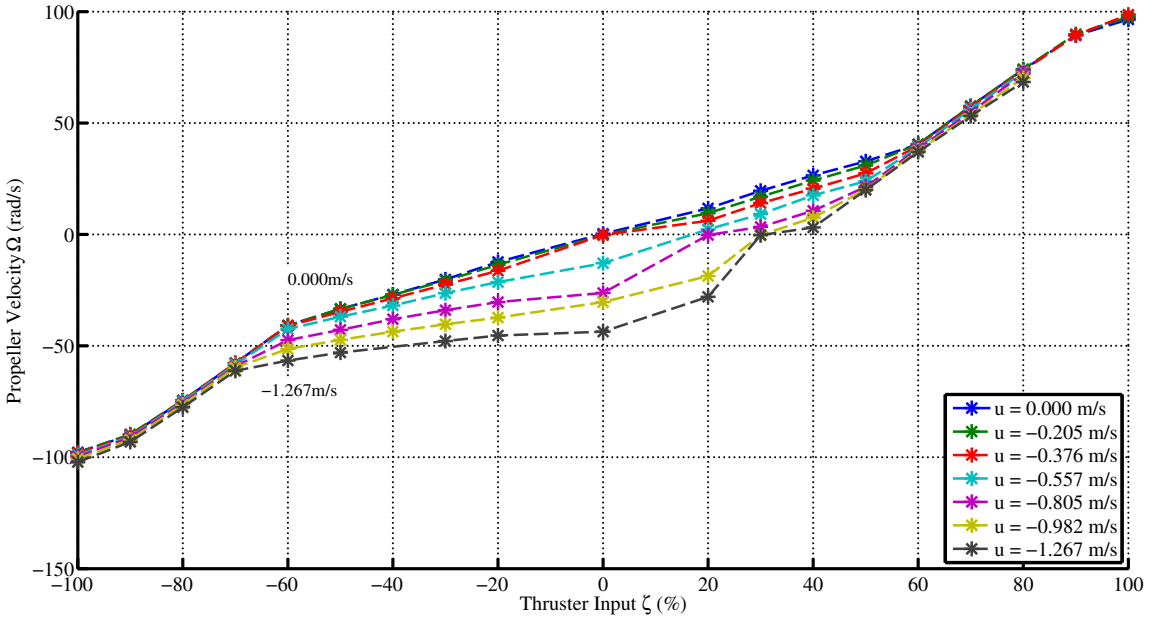

Figure 4.3 – Relationship between thruster input command and propeller angular velocity for different water speeds.

literature.

For the Seaeye falcon thruster used in this work, testing showed the propeller angular velocity to be a function of both thruster input command and the ambient water velocity, as shown in Figure 4.3. There is a similar relationship between input command and thruster current, shown in Figure 4.4.

Because the relationship between thruster input command and propeller angular velocity

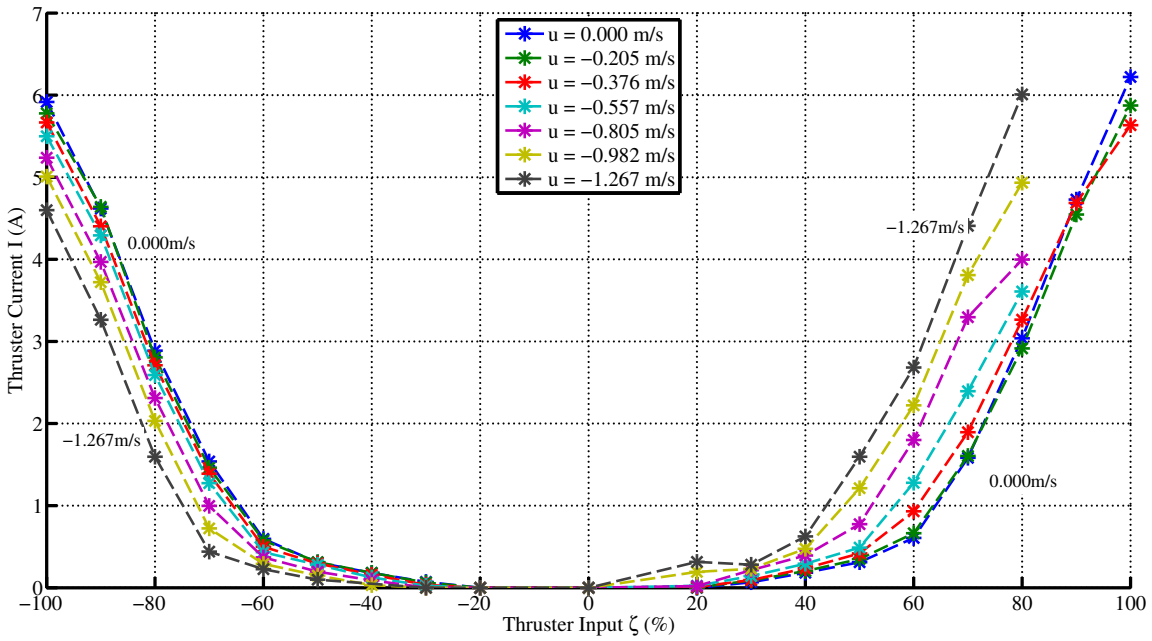

Figure 4.4 – Relationship between thruster input command and thruster current for different water speeds.

also depends on ambient water velocity, it is not clear how the linear and quadratic models can be directly applied to build a controller, which is the subject of Chapter 5. One consequence of the interdependence is the case where the thruster input command, ζ, and the propeller angular velocity, Ω, have opposite signs. This can be seen in Figure 4.3 where the ambient water velocity, u, is at -1.267 m/s. When the thruster input command is positive and below a certain threshold, there is a negative propeller angular velocity. Once the threshold is reached, about 30%, the internal controller is able to overpower the ambient water velocity and drive the propeller in the positive direction. There may be other consequences to this behaviour as well, particularly in how the force is determined from the load cell data. A full discussion on the force determination is given in Section 4.4.

Table 4.2 – Values for the deadband measurements.

Input Direction	Deadband
Static Positive	15%
Static Negative	−20%
Dynamic Positive	7%
Dynamic Negative	− 7%

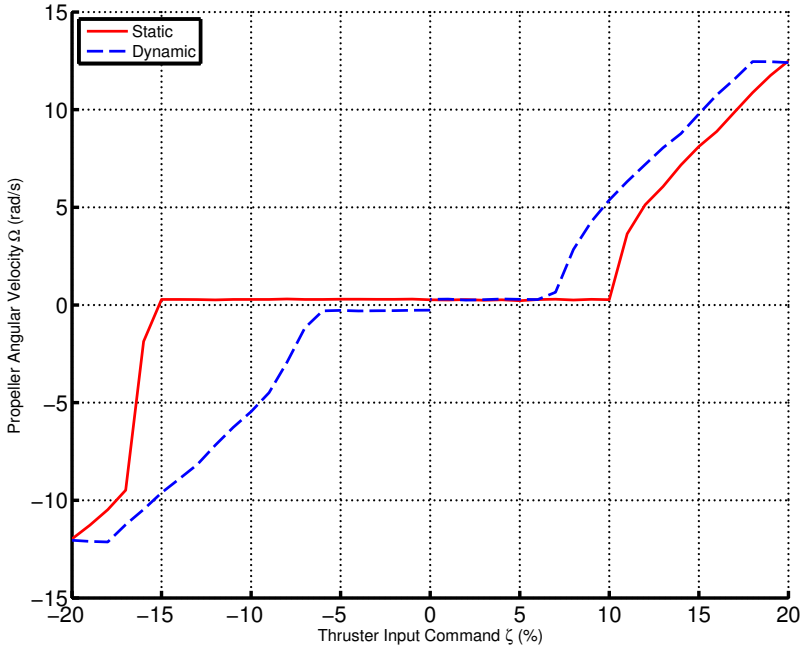

Figure 4.5 – Hysteresis curve of the thruster deadband, shown for the case where the ambient water velocity is zero.

4.3 Thruster Deadband Measurements

The thruster deadband is the set of thruster input commands near zero where the propeller is not engaged. This is due to mechanical friction inside the propeller as well as forces acting on the propeller itself. The deadband is determined in the free-standing tank. The deadband is different for positive and negative thruster input commands, which is expected. In addition, there is a difference between the 'static' deadband (propeller starting from rest) and 'dynamic' deadband (propeller spinning). The difference between these two values is shown in the hysteresis curve in Figure 4.5. The values are summarized in Table 4.2.

As discussed in the previous section, the propeller speed, Ω, and the thruster input, ζ, do not directly correlate — there is a dependency on the ambient water speed, u. The hysteresis curve shown in Figure 4.5 is the case of no ambient water velocity. At higher ambient water velocities, the propeller angular velocity will not be zero while the thruster input command is within the deadband. This is another consequence of the behaviour described in the previous section: the behaviour that the propeller angular velocity having a dependence on the ambient water velocity. This behaviour can change the way the force is measured when the ambient water velocity is non-zero, which is described in the next section.

4.4 Choice of Thrust Force Datum

Since propeller angular velocity is dependent on both the thruster input command and the ambient water velocity, the thrust force datum is subjective. The choice of datum has implications in the determination of best fit thrust coefficient values in all three regimes of operation. In the original works that describe these two models, the force datum was defined to exist for a thruster input[1] and propeller angular velocity of zero, irregardless of ambient water velocity. This choice is unavailable for the thruster under test as the propeller angular velocity is not zero when the thruster input command is zero at higher ambient water velocities as shown later in Section 4.2. Two options for the force datum in this work were considered:

- One option of datum is $T = 0\text{N}$ at $u = 0\text{m/s}$ and $\zeta = 0\%$ (this set of values also corresponds to $\Omega = 0\text{rad/s}$). At non-zero water speeds, $\zeta = 0\%$ may result in $\Omega \neq 0\text{rad/s}$, which produces a negative thrust. This choice of datum means that the absolute force recorded by the load cells on the sensor frame are attributed to thrust. The problem with this method is that this thrust measurement includes the hydrodynamic drag of the thruster assembly — these mechanics are likely to change when the thruster is on the

[1] In this case, 'thruster input' is the input cited works used, which is often a direct control of the thruster voltage.

vehicle.

- Another option is to set $T = 0\text{N}$ at $\Omega = 0\text{rad/s}$, regardless of the thruster input command or ambient water velocity. This will give a different thrust datum for each ambient water velocity. The advantage of this option is that it removes the drag forces on the propeller caused by fluid moving through the shroud. The drawback of this method is that not all the data runs have a thrust measurement at $\Omega = 0\text{rad/s}$. This can be overcome by interpolating the thrust at $\Omega = 0\text{rad/s}$ using the nearest measured values.

Of the two options for the thrust datum, the latter is preferable. In a real-world operating environment, the drag forces will be different from what is measured in the flume tank. Removing the drag forces in the measurements, as is done in the latter option, allows the thrust force recorded and the associated thrust coefficients to be used outside the lab environment. In addition, the latter option more closely matches the thrust datum used implicitly when building the linear and quadratic models that are used in this work. To use this datum, the drag on the propeller must be calculated and removed from the force measurements. This is described in the next section.

4.5 Drag Force Calculation

The choice of thrust datum from the previous section requires separating the thrust force from the total force measured by the sensors. The total force measured, F_t, is a function of the thrust force, T, and the drag force, F_d, as shown in equation (4.4).

$$F_t = T + F_d \tag{4.4}$$

To determine the thrust force, the drag force, F_d, caused by a non-zero ambient water velocity must be calculated and removed from the total force measured. The drag force of the

thruster was calculated when the propeller angular velocity was zero. This corresponds to zero thrust force, as per the datum defined in Section 4.4. The force is measured under steady-state conditions, thus only static drag has an effect on the force measured. The general equation for the drag in an underwater system as a function of ambient water velocity, u, is given in equation (4.5).

$$F_d = \frac{1}{2}\rho C_Q u^2 + \frac{1}{2}\rho C_L u \tag{4.5}$$

In this equation, C_Q is the coefficient for quadratic drag, C_L is the coefficient for linear drag, F_d is the drag in Newtons when $\Omega = 0\text{rad/s}$ and ρ is the fluid density. The coefficients C_Q and C_L are system dependent and constant across all ambient water speeds. These two coefficients must be experimentally determined. For the purposes of this work, the constants may be combined with these two coefficients to create a drag model given as a quadratic function of the ambient water velocity. This is shown in equation (4.6).

$$F_d = C_2 u^2 + C_1 u \tag{4.6}$$

In the case of zero ambient water velocity, there is no drag force. Also, with zero ambient water velocity and zero propeller angular velocity there is no total force. However, due to bias in the sensor a force may still be measured. In order to ensure there is zero force when there is zero propeller angular velocity, the bias, b, is considered within the drag calculation as shown in equation (4.7).

$$F_d = C_2 u^2 + C_1 u + b \tag{4.7}$$

Unlike the coefficients C_2 and C_1, this bias term is not static over time. This bias may change

in random directions by up to 6N over the course of two days, as shown in Figure 4.6. The maximum force recorded is 116 N, which would mean the change in bias would cause up to a 5% error over time. When the ambient water velocity is zero, the bias can be measured directly using equation (4.7). There are three methods of estimating the sensor bias that were considered:

- The bias may be calculated at $u = 0\text{m/s}$, then this value is used for all runs up until to the next bias measurement at $u = 0\text{m/s}$. This is an improvement, but there are several hours between bias measurements.
- The bias may also be estimated by using a linear interpolation between the previous bias measurement and next at any given ambient water speed. This method assumes the bias changes linearly between any two measurements.
- The bias may be estimated for each water speed when $\Omega = 0\text{rad/s}$ as part of the drag calculation.

The last method of estimating the bias is consistent with the choice for the thrust datum in Section 4.4. This last method was chosen for this experiment, as motivated by the drag equation shown in equation (4.7).

To calculate the drag force, the coefficients C_1 and C_2 from equation (4.7) were determined for the system. The drag force calculations were made using the following two assumptions:

- The sensor bias does not change appreciably during the course of a single data collection run at a test ambient water velocity.
- The sensor bias does not change appreciably from the measurement made at zero ambient water velocity through the drag measurements at the first two non-zero ambient water velocities.

With these two assumptions, the coefficients can be determined:

- The bias was calculated at zero ambient water velocity.

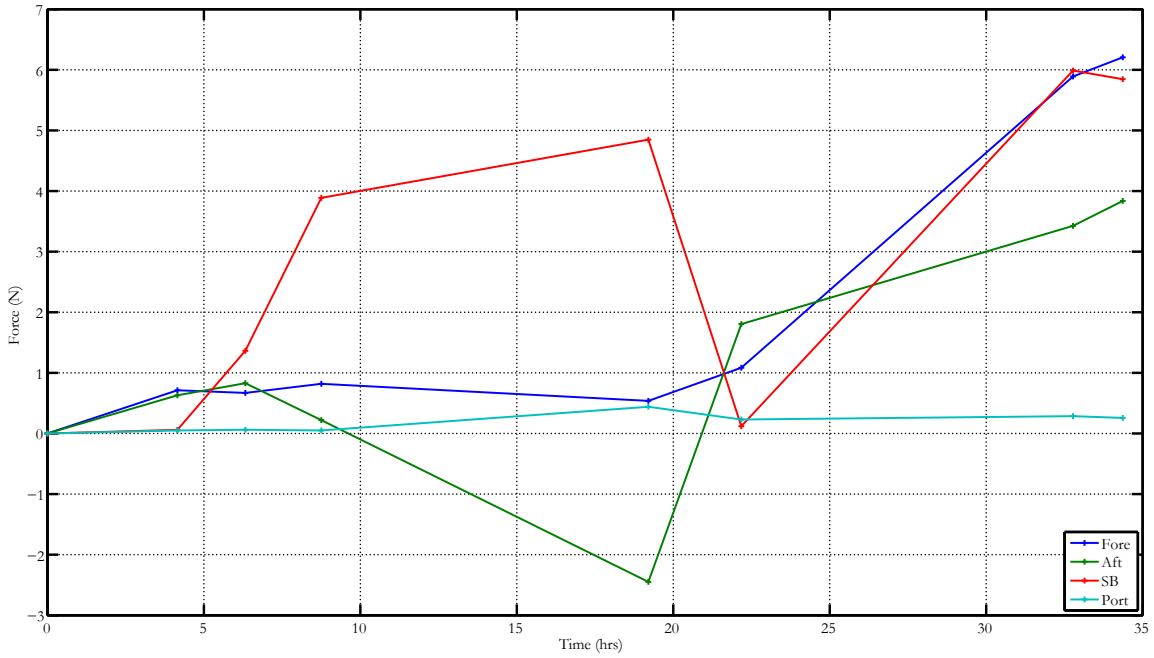

Figure 4.6 – Bias drift of force sensors for $u = 0\text{m/s}$ and $\Omega = 0\text{rad/s}$ over the course of 35 hours. As can be seen, the bias is random over time; future values of bias cannot be extrapolated from existing data.

- Using this bias for the first and second non-zero water velocities, the two values for the drag, F_d, were measured at $\Omega = 0\text{rad/s}$.
- Using the two drag measurements at two non-zero ambient water velocities, the coefficients C_2 and C_1 were determined algebraically using equation (4.7).

The measurements of the coefficients are shown in Table 4.3.

Once the drag coefficients have been calculated, the bias may be calculated for any ambient water velocity at zero propeller angular velocity using equation (4.8).

$$b = F_d - C_1 u^2 - C_2 u \tag{4.8}$$

The drag force and sensor bias may then be calculated to determine the thrust force, shown in

Table 4.3 – Results of the C_1 and C_2 measurements from the drag calculation in equation (4.7). The forward direction is the ambient water velocity moving against the front of the thruster; the reverse direction is the ambient water velocity moving against the back of the thruster. The construction of the thruster is meant to minimize drag in the forward direction, which is reflected by the values being much smaller in this direction.

Direction	C_1	C_2
Forward	-6.37×10^{-2}	-2.66×10^{-3}
Reverse	20.2	1.78

equation (4.9).

$$T = F_t - C_2 u^2 - C_1 u - b \tag{4.9}$$

The bias is recalculated for each set of trials at a give ambient water speed. The drag force and bias calculated this way across the range of ambient water speeds is shown in Figure 4.7.

4.6 Near-Surface Effects In The Force Measurements

The proximity of the propeller to the surface of the flume tank caused some problems in data collection. At higher propeller angular velocities, the propeller pushed enough water through to reach the surface of the tank, which caused the propeller to pull in air instead of water. This is shown in Figure 4.8. This causes the measured thrust to drop unexpectedly, which makes determining the steady-state values difficult. An example of the force measurement is shown in Figure 4.9.

On the vehicle, the thrusters are rarely subject to this effect. The only exception is the vertical thruster when near the surface, but that condition is generally avoided during vehicle operation. To mitigate this problem in the analysis, steady-state data was taken in places where this effect did not occur. Data that has no stable areas of force was discarded.

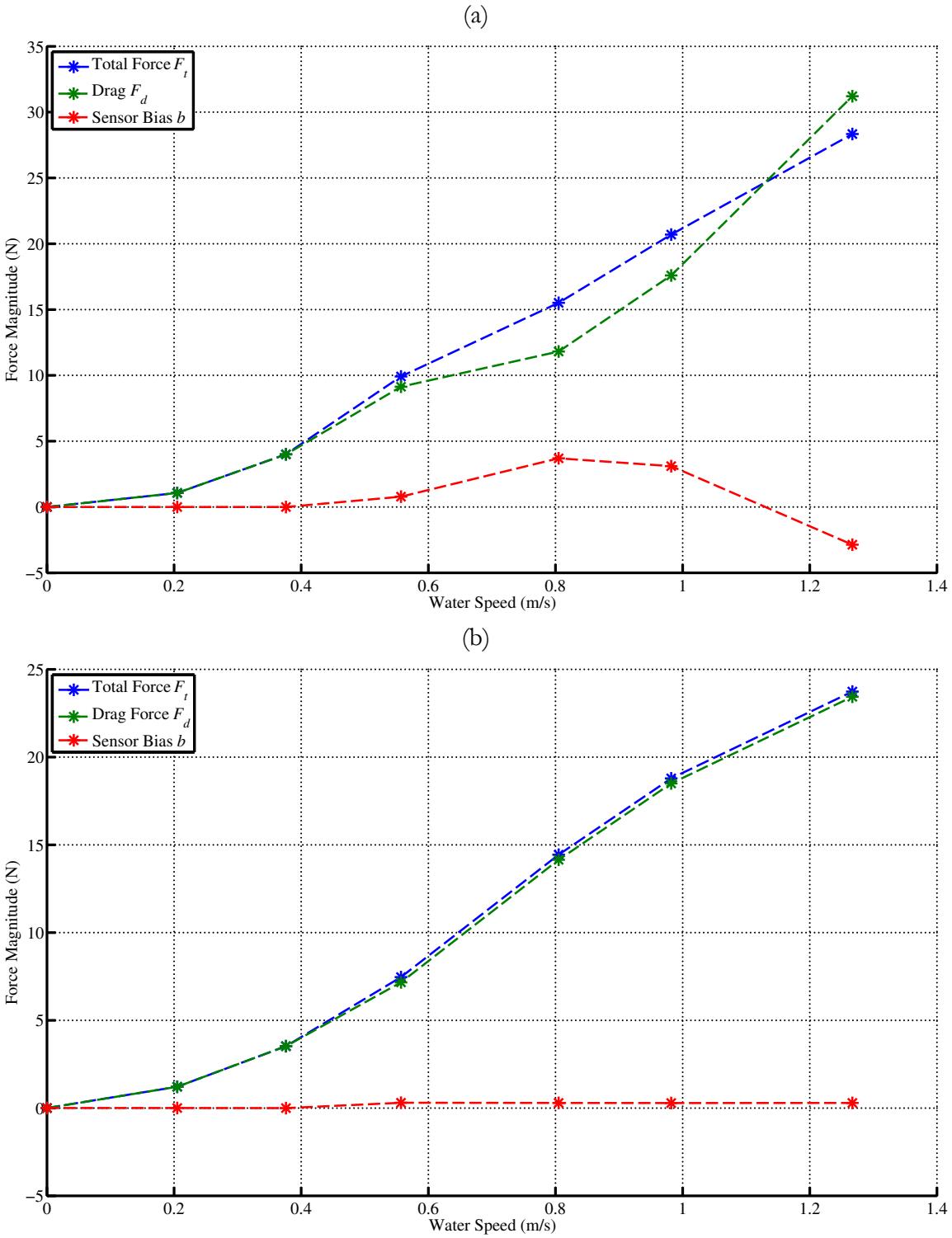

Figure 4.7 – The estimated drag force for $\Omega = 0$ rad/s at different water speeds. These graphs show the full calculated drag force, the estimated sensor bias and the estimated drag force for (a) the front of the thruster and (b) the rear of the thruster.

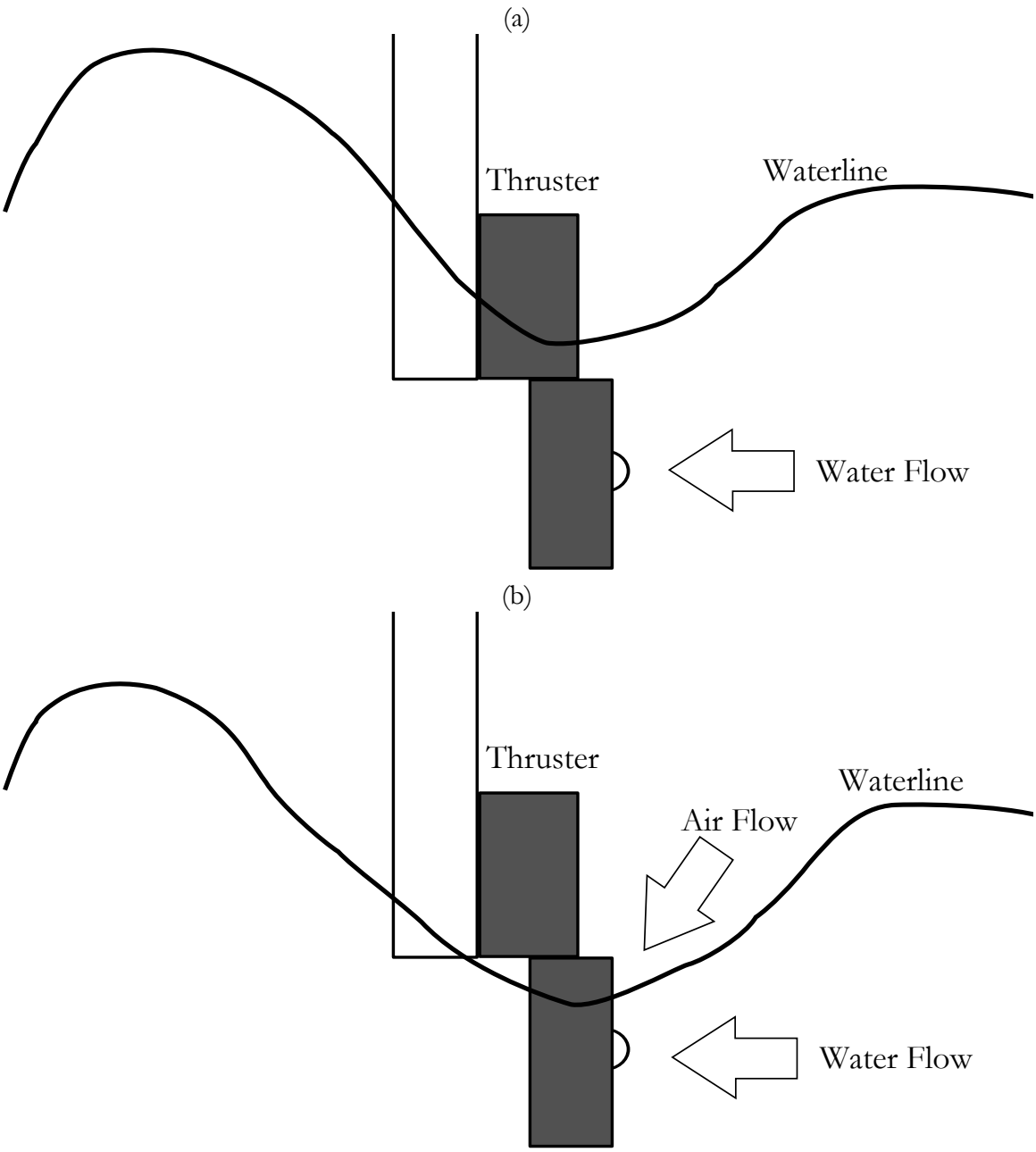

Figure 4.8 – (a) The propeller is completely beneath the surface during testing, which gives a steady force measurement. (b) At higher propeller angular velocities, air is sucked in through the propeller, causing unstable force measurements.

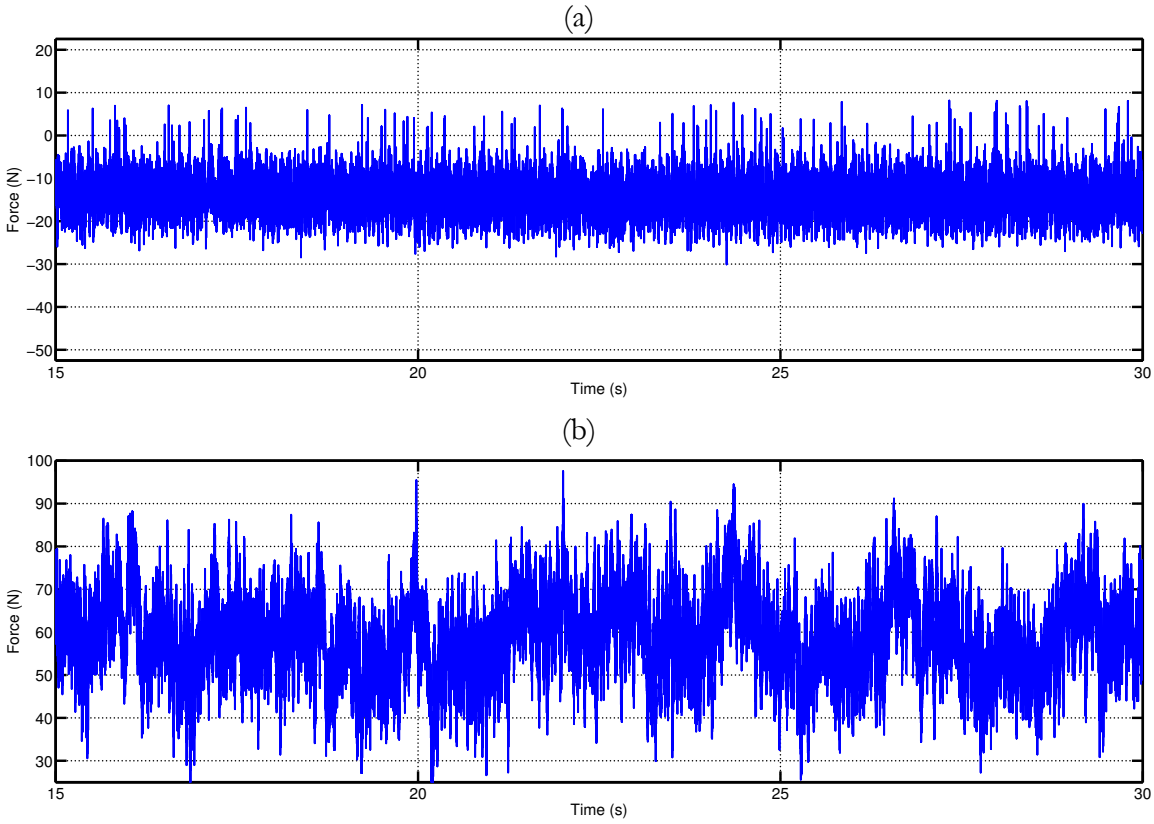

Figure 4.9 – Comparison of measured force (a) with only water sucked through the propeller and (b) with both water and air sucked into the propeller. Both graphs show 15s of data with a 75N spread.

This is a source of error that transfers through to the controller when operating at higher input commands and higher ambient water velocities. For the thruster model and tests, the problems with the data cannot be avoided without different equipment. On the vehicle there are several other sources of error that cannot be measured but must be overcome as part of the controller as well. It is assumed that this error can be ignored in the data analysis, and accommodated for as part of the error in the final controller on the vehicle.

4.7 Calculation of Thrust Coefficient

The previous sections interpreted the data such that it can be fit into the thruster dynamics model. This section uses the data to generate the thrust coefficient, $K_T(J_0)$. Equation (2.12)

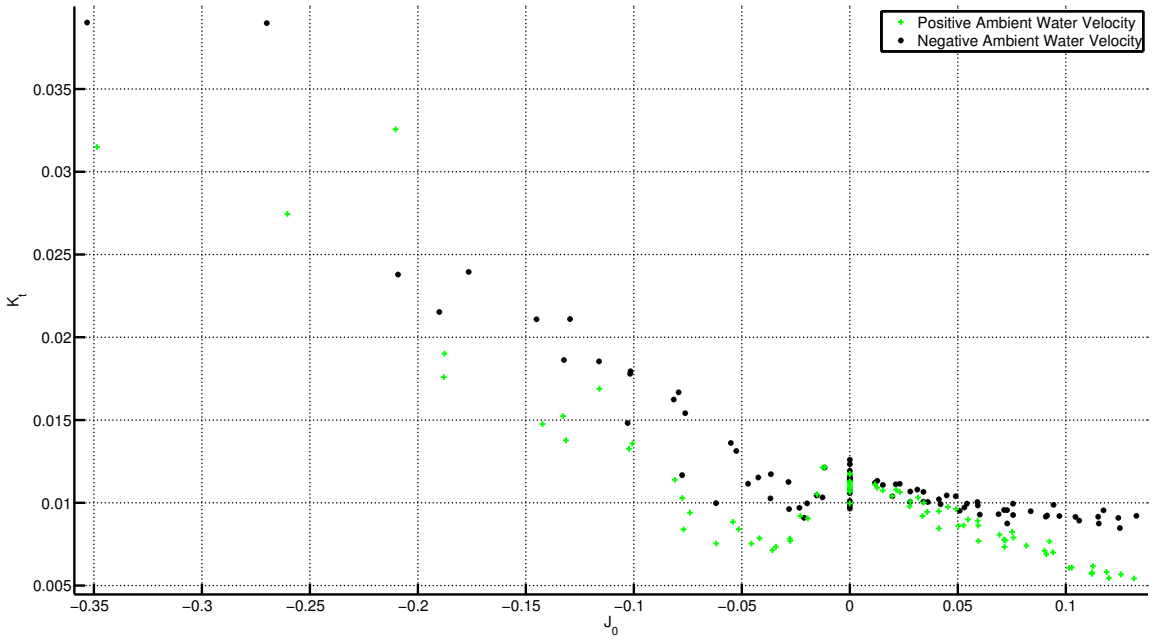

Figure 4.10 – The difference between positive and negative ambient water velocities on the thrust coefficient, where the propeller angular velocity $\Omega \geq 3$rad/s. Negative ambient water velocity corresponds to the forward direction, positive ambient water velocity to the reverse direction.

is a piece-wise function for each of the three states, so a separate set of coefficients must be determined for each state. In addition, the thruster is not front/back symmetric. Figure 4.10 shows this asymmetry, where the curve in the thrust forward direction is different than in the thrust reverse direction. Each direction (forward and reverse) has a different set of coefficients. This results in six sets of coefficients for $K_T(J_0)$, as shown in equation (4.10).

$$K_T(J_0) = \begin{cases} k_{e1}J_0^2 + k_{e2}J_0 + k_{e3} & if J_0 > 0 \\ k_{a1}J_0^2 + k_{a2}J_0 + k_{a3} & if 0 > J_0 > J_0^* \\ k_{v1}J_0^2 + k_{v2}J_0 + k_{v3} & if J_0^* > J_0 \end{cases} \tag{4.10}$$

The results of the three states can be seen in Figures 4.11 and 4.12. The resulting equation is discontinuous at $J_0 = 0$ and $J_0 = J_0^*$, as is seen in the results.

The value of J_0^* was determined iteratively based on equation (2.20). This value is at the in-

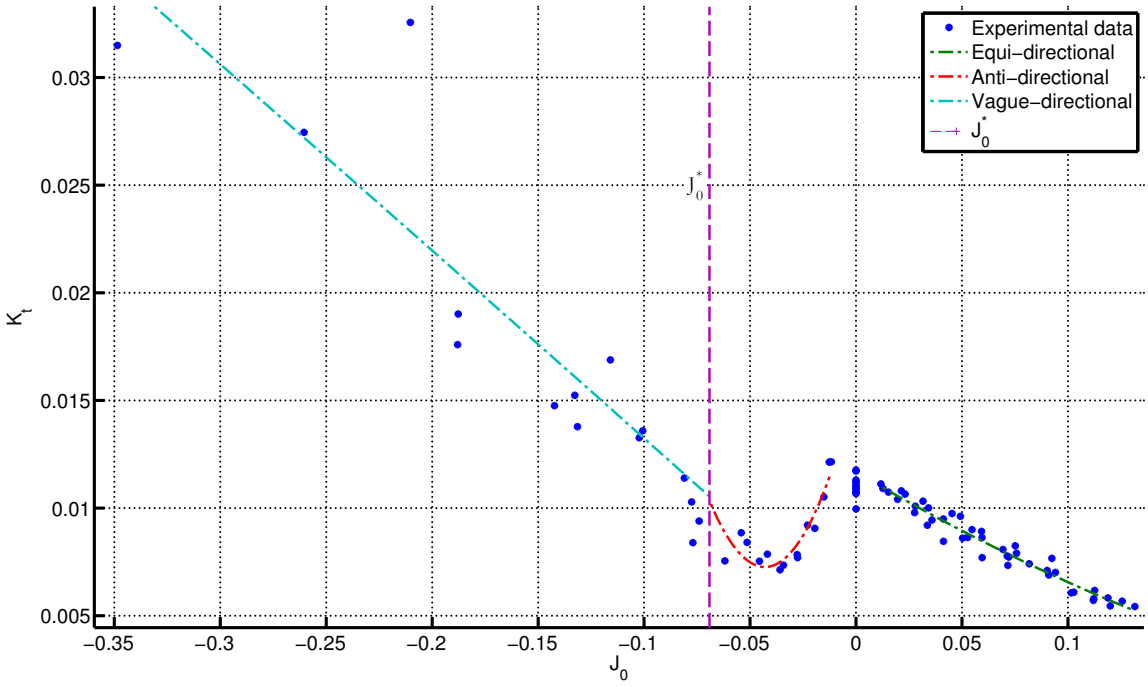

Figure 4.11 – Graph showing K_T vs. J_0 where thrust $u > 0$, experimental results.

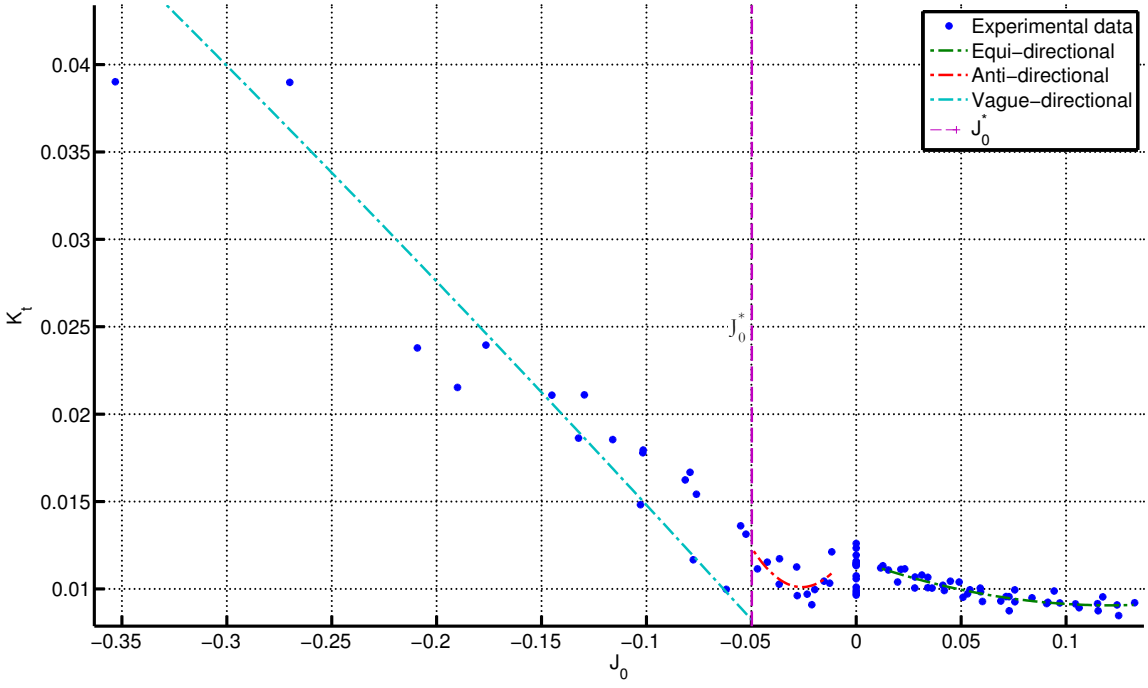

Figure 4.12 – Graph of K_T vs. J_0 where thrust $u < 0$, experimental results.

tersection of two different quadratics (corresponding to the anti- and vague-directional states), and it affected the equations in both states. The algorithm used here depends on a temporary value of J_0^*, which is denoted by J_{0a}^*, and changing that value at each iteration until the value of J_0^* is found. The steps involved in finding J_0^* are outlined here:

1. Find the local minima of K_T where $J_0 < 0$ and set J_{0a}^* to be this point.
2. Use the values of $K_T(J_0)$ where $J_{0a}^* < J_0 < 0$ to fit a curve for the anti-directional state.
3. Use the coefficients to determine a new value for J_{0a} using equations (2.21), (2.22) and (2.20).
4. Repeat until a value of J_0^* is found.

The value of J_0^* is somewhat dependent on the initial choice of J_{0a}^*. An improved process may be found to determine this value, however the value found through this method is borne out from the experimental results. Once the value for J_0^* was set and the anti-directional coefficients for $K_T(J_0)$ found, the vague-directional coefficients were calculated.

A consequence of this method is that there is a discontinuity in in $K_T(J_0)$ at $J_0 = J_0^*$. A possibility of future work would be to require this point to be C^0 or C^1 continuous and determine if this improves the results. Due to the unknown nature of the axial water velocity, u_p, it is not known whether this would improve the controller in a significant manner. This is further discussed in Chapter 7.

The case where there is no ambient water velocity is treated separately. In this case $J_0 = 0$, and there is no change based on thruster direction. In this case $K_T(J_0)$ is given by equation (4.11).

$$K_T(0)=k_1 \tag{4.11}$$

Thrust in this case is based entirely on angular propeller velocity. There are some options on how to handle this case:

- Attach the $J_0 = 0$ point into the domain of either the positive of negative ambient water velocity direction. This approach does not treat zero ambient water velocity as a special case. A choice must be made as to the equation for which direction will be used (positive or negative ambient water velocity) from Section 4.7. Also, the choice of whether to use the equi-directional or anti-directional coefficients for this direction must be made, since $J_0 = 0$ is the separation point between these two states.
- Treat $J_0 = 0$ as its own case, with its own coefficient. This approach takes all the values taken at zero ambient water velocity for every run uses the mean value to determine the coefficient. The drawback of this option is that there is now one more equation that must be created in the controller.

In this work, the latter approach is used and a separate coefficient is created for the case of $J_0 = 0$.

4.8 Comparison of Thrust Model and Actual Force

This section covers the results of calculating the steady-state force. The thrust coefficients are calculated for both the linear and quadratic models for $K_T(J_0)$. The error between the actual and calculated values are given and compared between the linear and quadratic equations. Using the linear model given in Section 2.2, the thrust coefficients can be calculated for the equi-directional state. The resulting model and a comparison to the data is shown in Figures 4.13 and 4.14. The results from the quadratic model from Section 2.3 are given in Figures 4.15 and 4.16. The complete list of the thrust model coefficients are given in Table 4.4 for both the linear and quadratic models.

A comparison of the quadratic and linear thrust coefficient models is shown in Figures 4.18

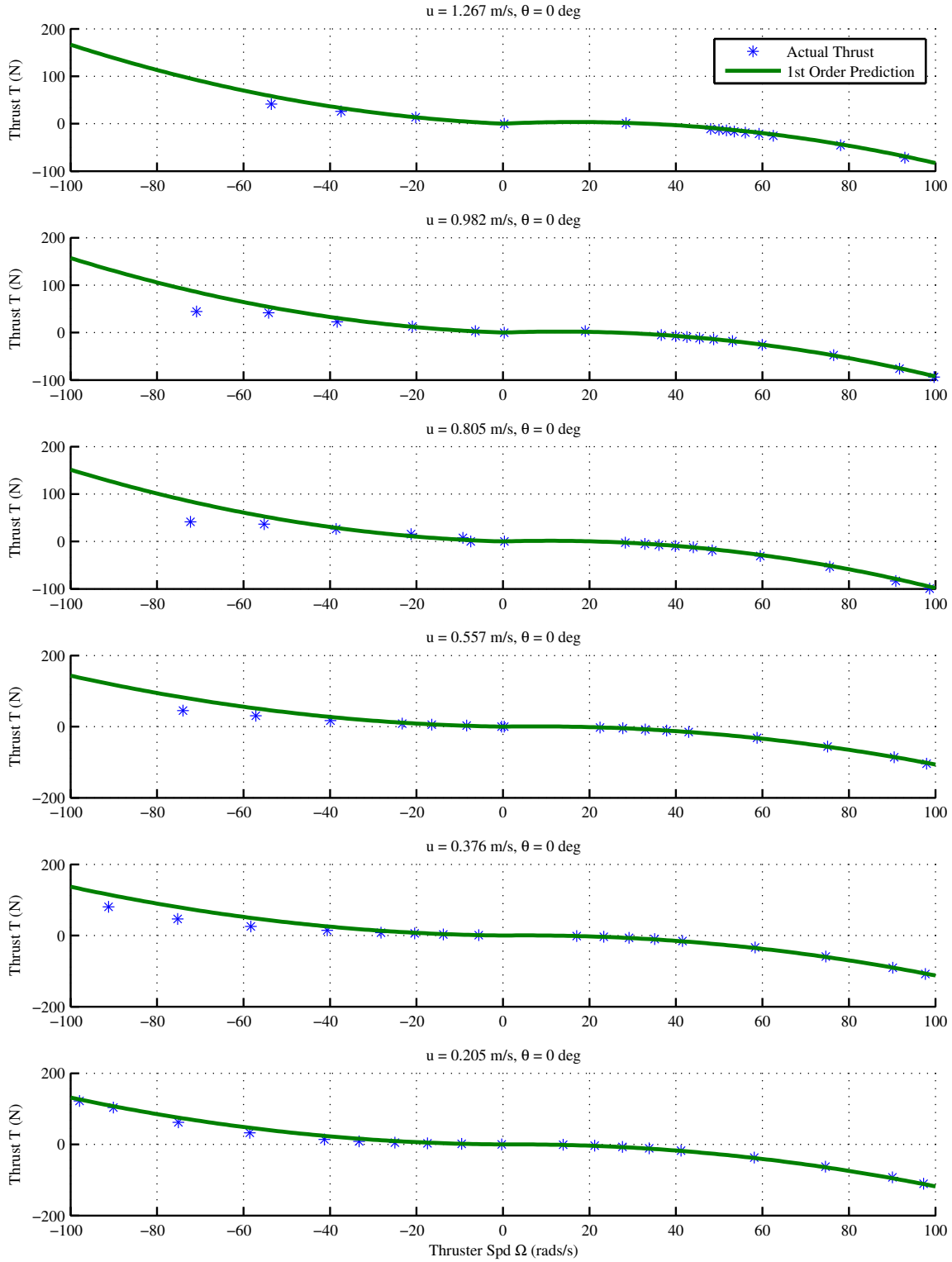

Figure 4.13 – Results of the linear thrust model, $u > 0$.

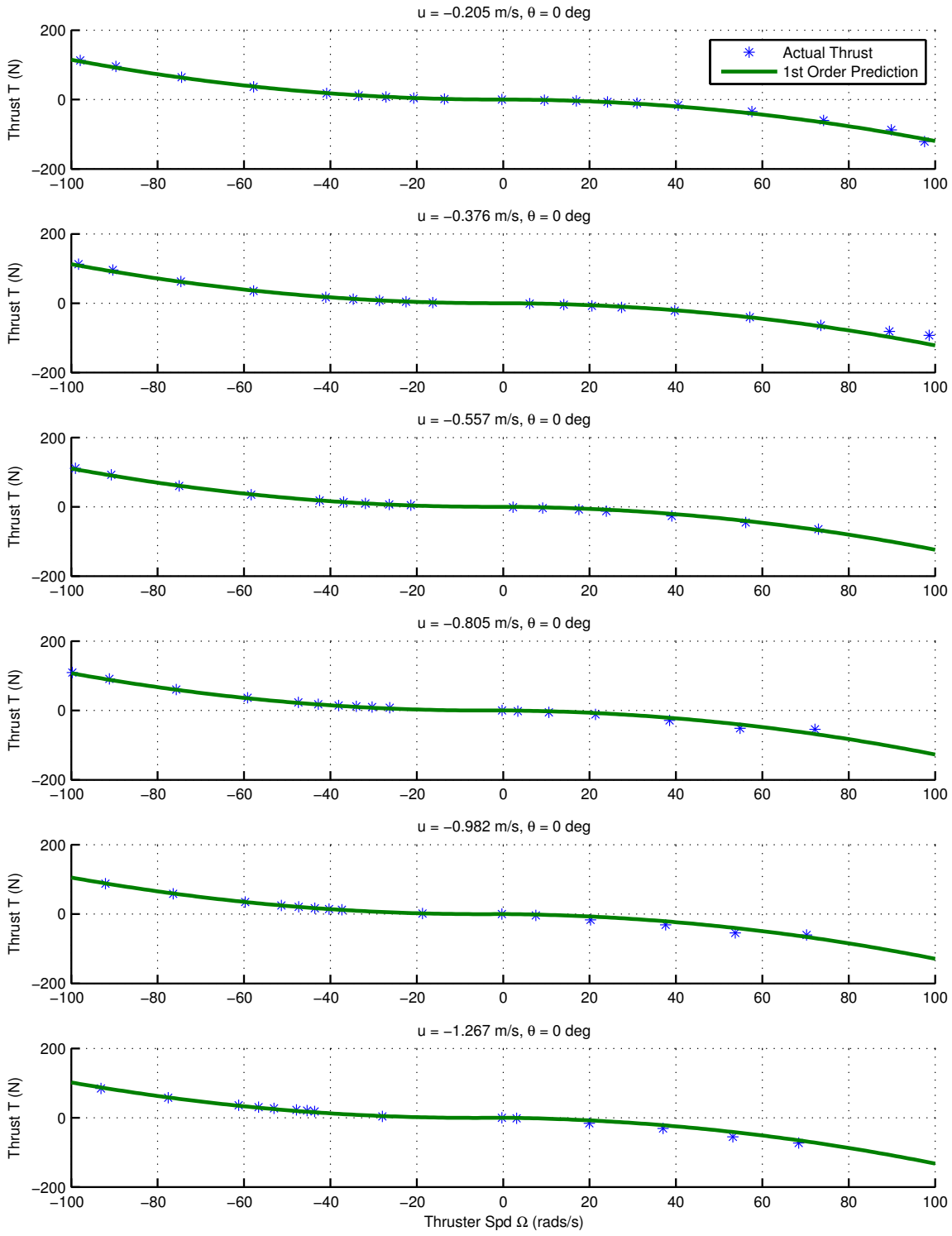

Figure 4.14 – Results of the linear thrust model, $u < 0$.

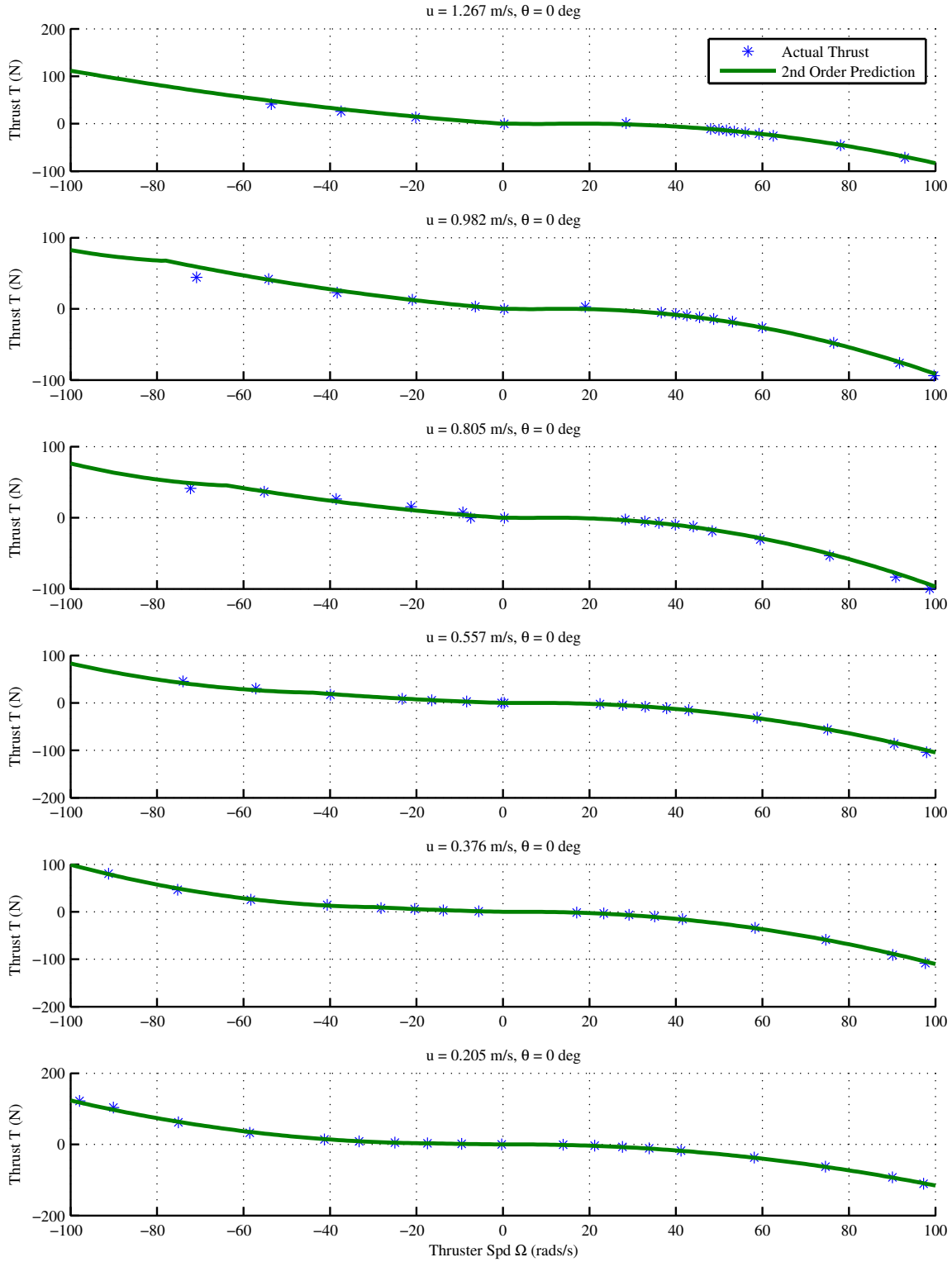

Figure 4.15 – Results of the quadratic thrust model, $u > 0$.

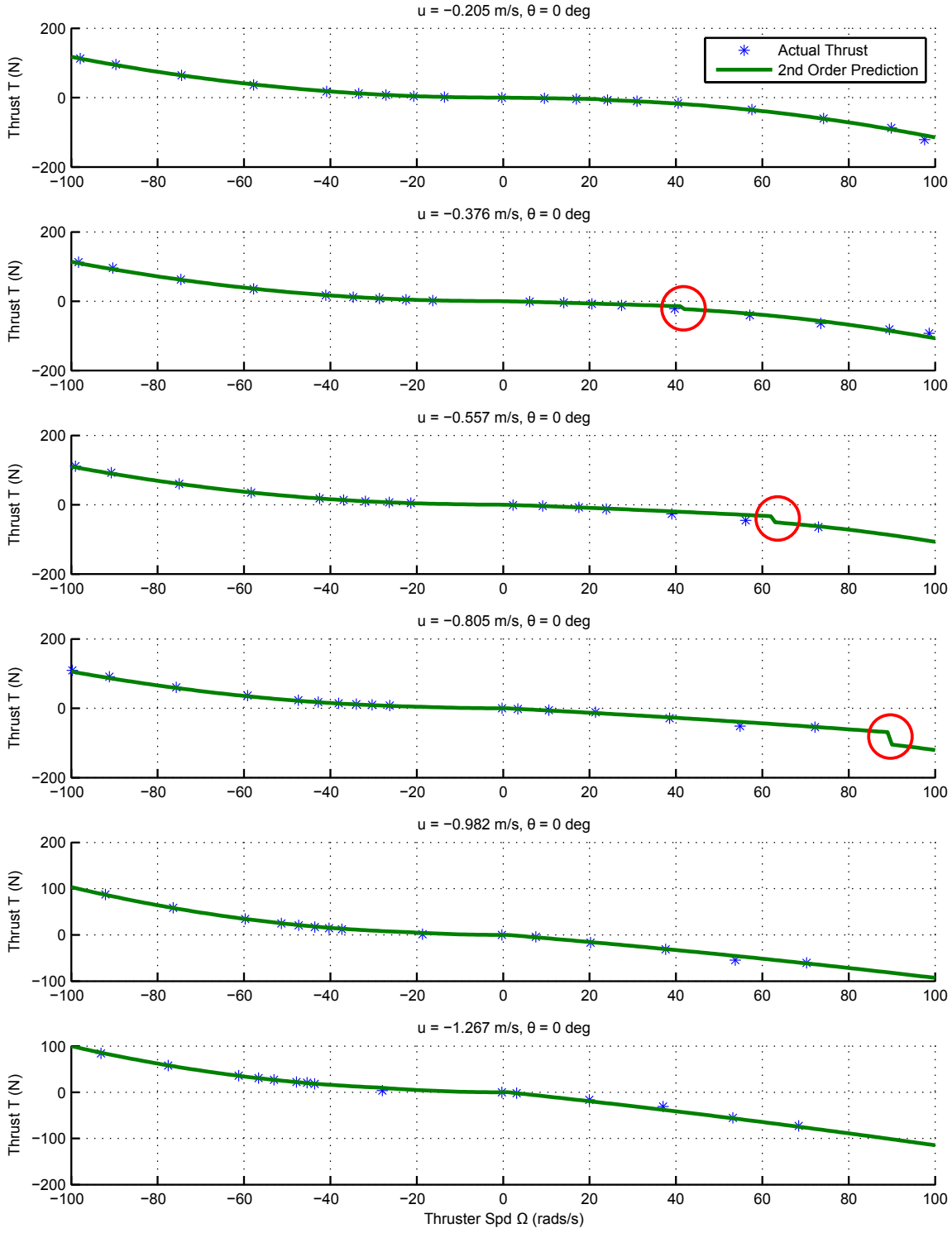

Figure 4.16 – Results of the quadratic thrust model, $u < 0$. There are pronounced discontinuities in these graphs, as indicated. These discontinuities correspond to the discontinuity at J_0^*, shown in the $K_T(J_0)$ graph in Figure 4.12. The vague-directional state is to the left of the discontinuity, and the anti-directional state is to the right.

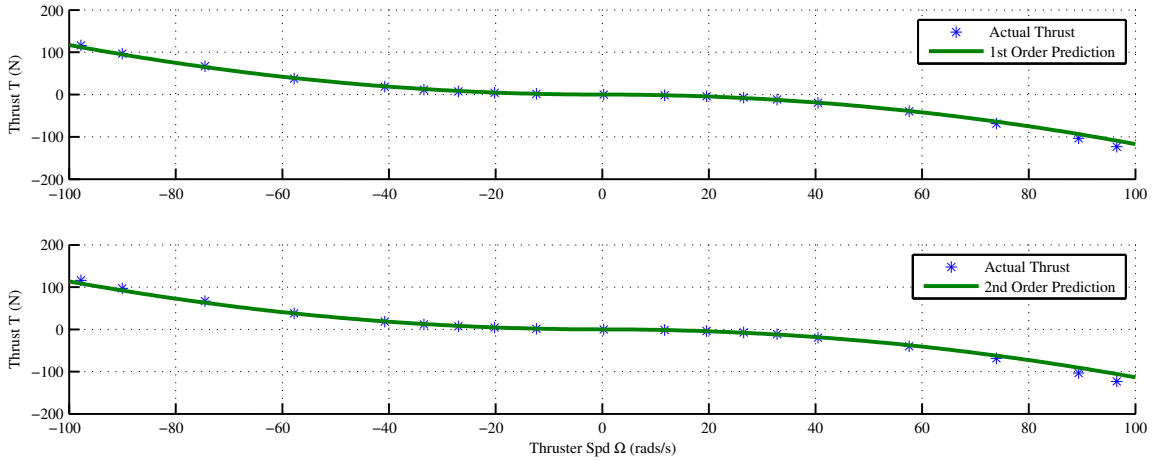

Figure 4.17 – Comparison between both linear and quadratic results for no ambient water velocity, $u = 0$.

and 4.19. These graphs show the difference in thrust errors for each ambient water speed tested. As seen in the graphs, the linear thrust coefficient tends to do well for lower propeller speeds and ambient water velocities, especially in the equi-directional state. As the ambient water speed increases, the error in the linear approximation for the thrust coefficient also increases. This is particularly noticeable in the anti-directional state, where $\text{sign}(u) \neq \text{sign}(\Omega)$ and the magnitude of the propeller angular velocity is high. As the ambient water velocity increases, the error from the linear thrust model rises much higher than the quadratic thrust model. These differences can be seen in Table 4.5, which separates the error by state. At the higher water speeds, data for the anti-directional state becomes unreliable due to near-surface effects, as discussed in Section 4.6.

When there is no ambient water velocity, $u = 0$, the quadratic thrust model has more error than the linear model. This may be due to the wide distribution of $K_T(0)$ values, as seen in Figure 4.10. It is unclear if this is a problem with the underlying model or the data collection. Given that once the propeller is spinning it is likely the vehicle carrying the thruster will be moving through the water, thereby increasing the ambient water velocity, this is expected to be an uncommon use case. As the ambient water velocity increases, the quadratic model error

Table 4.4 – Values for the thrust coefficient, $K_T(J_0)$, for each state.

	State	Coefficient	Value
Linear	$u > 0$	α_{1p}	-0.0565
		α_{2p}	0.0119
	$u < 0$	α_{1n}	-0.0207
		α_{2n}	0.0112
Quadratic	Equi-Direction, $u > 0$	k_{1p}	0.0681
		k_{2p}	-0.0579
		k_{3p}	0.0117
	Anti-Direction, $u > 0$	k_{1p}	4.5069
		k_{2p}	0.3877
		k_{3p}	0.0156
	Vague-Direction, $u > 0$	k_{1p}	-0.0033
		k_{2p}	-0.0882
		k_{3p}	0.0045
	Equi-Direction, $u < 0$	k_{1n}	0.1771
		k_{2n}	-0.0429
		k_{3n}	0.0117
	Anti-Direction, $u < 0$	k_{1n}	3.9231
		k_{2n}	0.2037
		k_{3n}	0.0127
	Vague-Direction, $u < 0$	k_{1n}	-0.0216
		k_{2n}	-0.1343
		k_{3n}	0.0016
	$u = 0$	k_{1z}	0.0108

is lower than the linear model.

If we constrain the values of the ambient water speed to $u \in [-0.8, 0.8]$, which is the realistic top speed of the vehicle, the quadratic model improves the average calculated thrust by as much as 20.5 N in the anti-directional state and 5.5 N in the equi-directional state. This is a modest gain in the equi-directional state (5% assuming a top thrust of 100 N), it is a large improvement in the anti-directional state (20%).

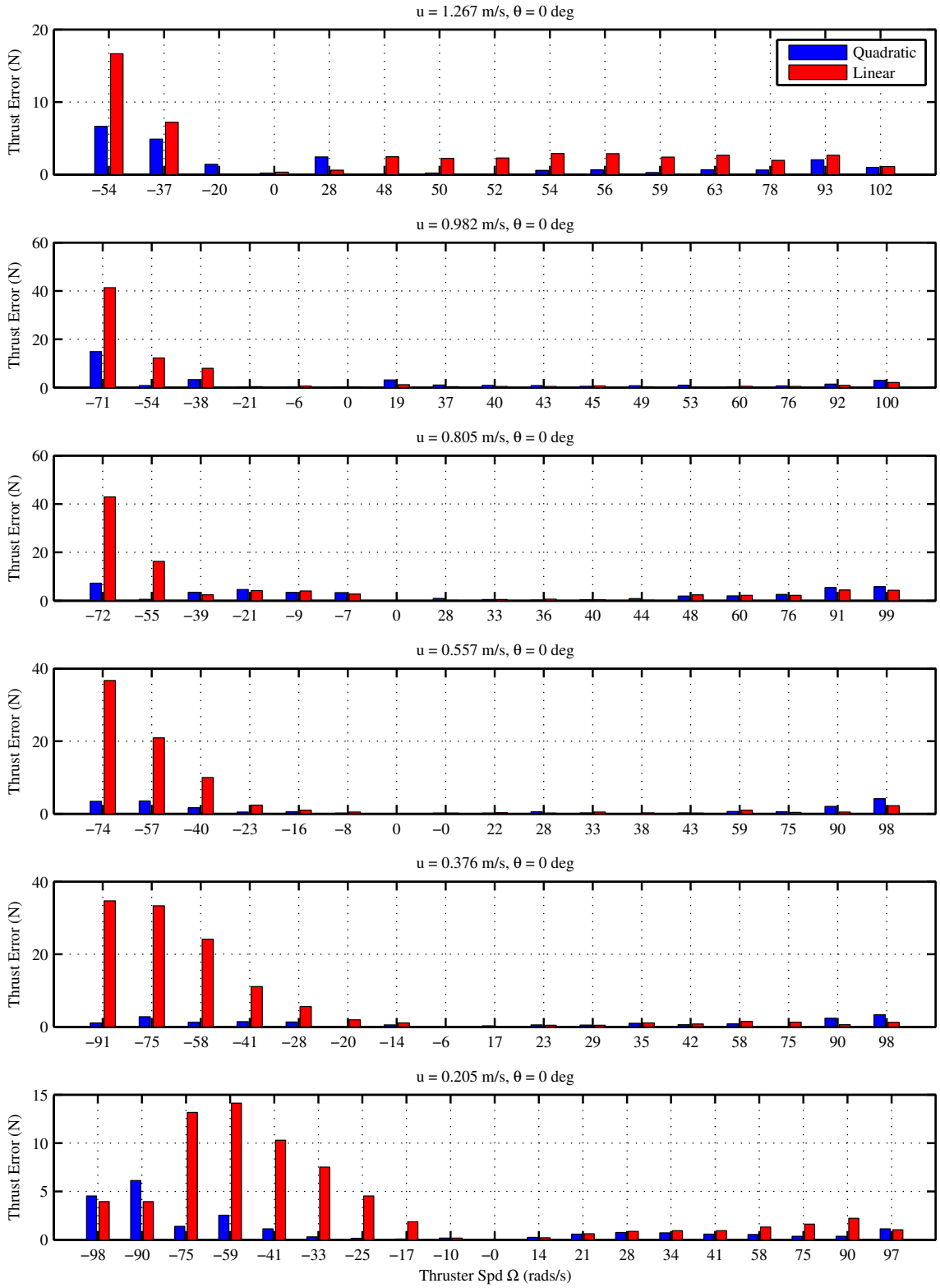

Figure 4.18 – Comparison of error between linear and quadratic results for positive ambient water velocities, $u > 0$.

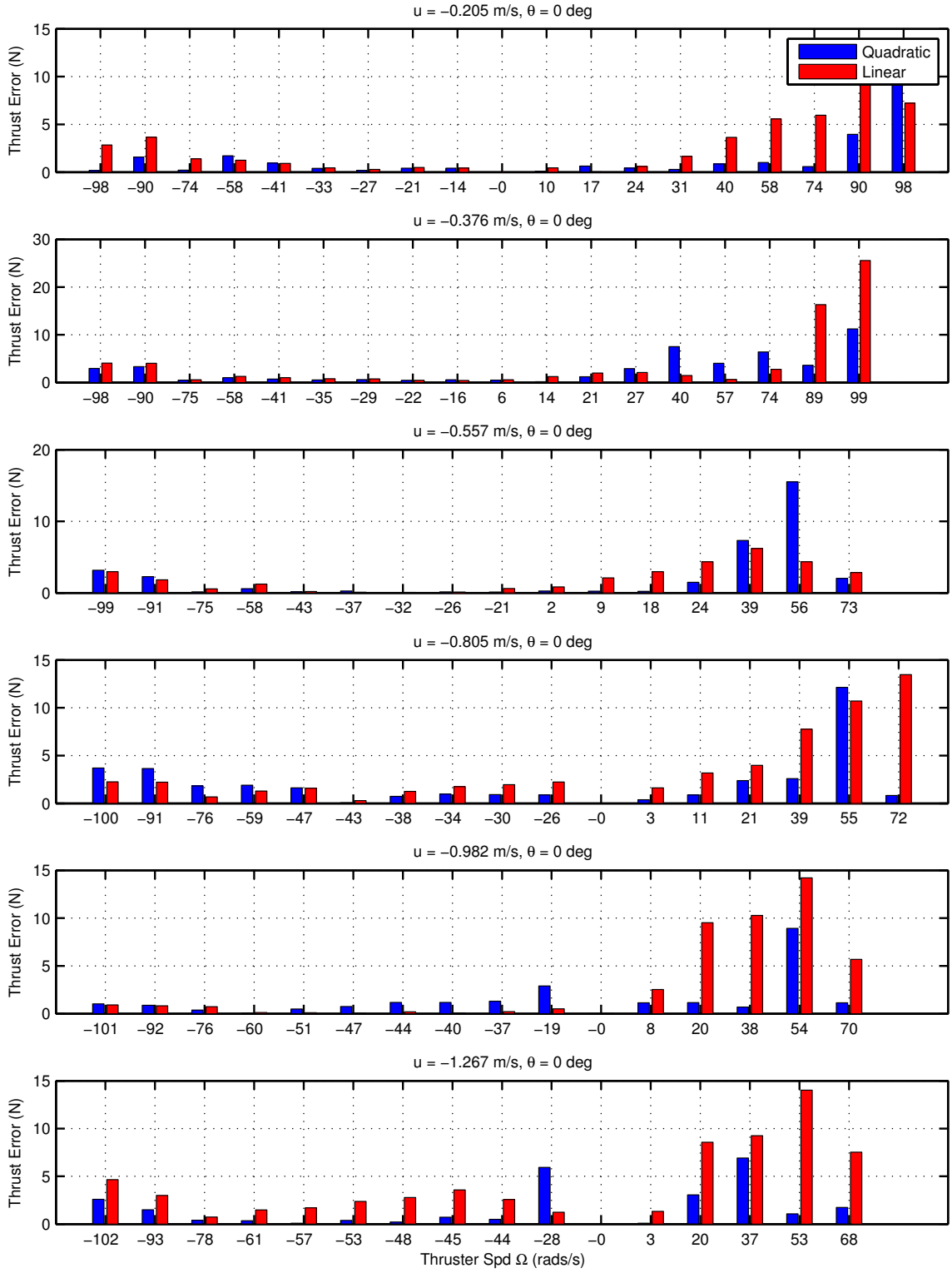

Figure 4.19 – Comparison of error between linear and quadratic results for negative ambient water velocities, $u < 0$.

Table 4.5 – Summary of absolute errors (in N) for each ambient water speed. Due to poor data, mostly due to near-surface effects as discussed in Section 4.6, there is no reliable data for the anti-directional state at higher water speeds.

Water Speed	State	Linear Err	Quadratic Err	Difference
1.267	Equi-Dir	10.047	0.738	9.309
	Anti-Dir	—	—	—
	Vague-Dir	3.221	4.320	−1.100
0.982	Equi-Dir	7.571	1.172	6.399
	Anti-Dir	—	—	—
	Vague-Dir	6.408	3.879	2.529
0.805	Equi-Dir	4.382	1.880	2.503
	Anti-Dir	—	—	—
	Vague-Dir	8.640	3.780	4.860
0.557	Equi-Dir	2.980	0.863	2.117
	Anti-Dir	23.955	3.407	20.548
	Vague-Dir	3.234	1.090	2.144
0.376	Equi-Dir	2.023	1.063	0.960
	Anti-Dir	20.452	1.706	18.746
	Vague-Dir	2.020	0.676	1.344
0.205	Equi-Dir	1.133	0.591	0.542
	Anti-Dir	6.255	2.315	3.940
	Vague-Dir	0.398	0.092	0.306
0.000	Zero	2.446	3.238	−0.792
−0.205	Equi-Dir	0.430	0.606	−0.176
	Anti-Dir	8.540	2.760	5.780
	Vague-Dir	0.465	0.373	0.092
−0.376	Equi-Dir	1.833	1.193	0.640
	Anti-Dir	22.599	6.328	16.271
	Vague-Dir	0.716	2.470	−1.754
−0.557	Equi-Dir	3.889	0.759	3.129
	Anti-Dir	15.379	2.026	13.353
	Vague-Dir	1.405	4.174	−2.768
−0.805	Equi-Dir	7.248	1.477	5.770
	Anti-Dir	—	—	—
	Vague-Dir	5.488	3.196	2.292
−0.982	Equi-Dir	7.646	0.926	6.720
	Anti-Dir	—	—	—
	Vague-Dir	6.468	2.614	3.855
−1.267	Equi-Dir	12.167	1.147	11.020
	Anti-Dir	—	—	—
	Vague-Dir	4.270	2.571	1.699

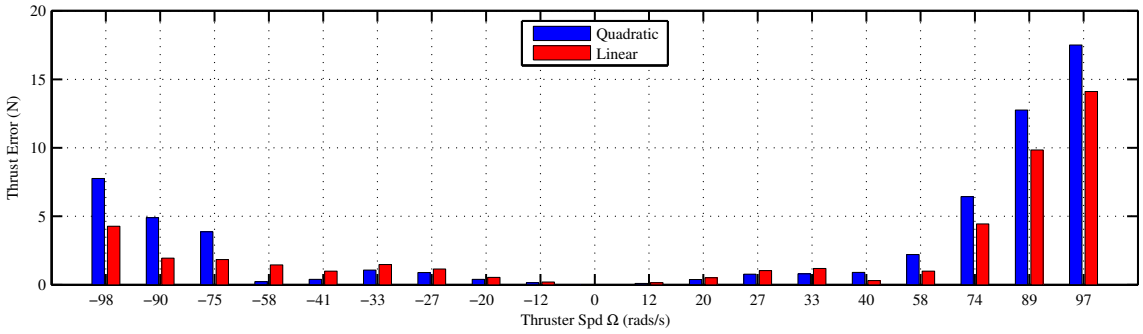

Figure 4.20 – Comparison between linear and quadratic results for no ambient water velocity, $u = 0$.

4.9 Thruster Command Response Times

Three commands are sent to the thruster over RS-485 on each command loop. After each command is sent, a reply is received from the thruster either acknowledging the command given (in the case of thruster input command) or returning a value (for thruster reported speed and thruster current queries). There is a measurable time delay between sending each command and receiving the response.

An unexpected result of the thruster tests was the variable latency in the thruster commands; the latency changes based on the command given and on the ambient water speed, shown in Figure 4.21. The response times can vary significantly. In addition, the response time increases at higher thruster input values.

This chapter has reviewed the results of the testing and fit these results to the linear and quadratic models from Chapter 2. The fit of both of these models were compared against the actual data. The next chapter uses the results of this chapter to build a low-level thruster controller that is able to compensate for the ambient water velocity.

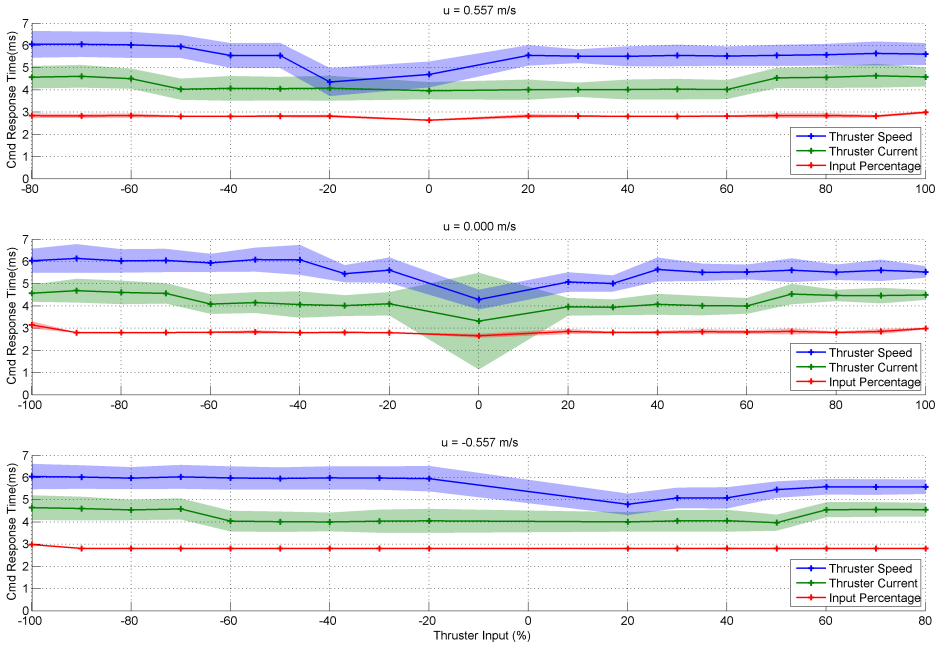

Figure 4.21 – Thruster response times for the three commands: thruster input command ζ, get current I and get thruster reported speed ω. Shown are the response times for $u = 0\text{m/s}$ and $u = -1.267\text{m/s}$. The shaded region indicates one standard deviation from the mean values shown.

Chapter 5

Design of a Thruster Controller

This chapter presents the thruster controller that was built based on the experimental results. The quadratic model given in Chapter 2 shows how to calculate an estimated thrust force based on the ambient water velocity, propeller angular velocity and known functions for a thrust coefficient. The results of Chapter 4 provided the thrust coefficient values, and the model can now be used in practise to estimate thrust force for the Saab Seaeye thruster.

In practise, the desired thrust force is provided by a thruster allocation algorithm in the high-level vehicle controller. The quadratic thrust model developed in Chapter 2 was inverted and used in a thruster controller that compensates for ambient water velocity to estimate a desired propeller angular velocity. The process of determining the desired propeller angular velocity is discussed in detail in Section 5.1.

The desired propeller angular velocity is then translated into a thruster input command. The propeller angular velocity is a function of both the thruster input command and the ambient water velocity, as shown in Figure 4.3. The ambient water velocity must be taken into account when calculating the thruster input command. The details of the calculation for the thruster input command is discussed in Section 5.2.

The controller takes the desired thrust force along with the ambient water velocity, and

estimates a desired propeller angular velocity needed to provide that thrust force. It then commands the thruster using the actual propeller angular velocity as feedback. Testing of the controller is discussed in Section 5.3. The controller tests were completed using same experimental equipment and setup described in Chapter 3. The results of the controller testing are provided in Chapter 6.

5.1 Determining the Desired Propeller Velocity

The quadratic force model estimates a thrust force using the ambient water velocity, u, and propeller angular velocity, Ω. To find the propeller angular velocity, equation (2.5) was inverted to give a desired propeller angular velocity, Ω_d, as a function of desired thrust force, T_d. This section outlines the procedure for performing this inversion and calculating Ω_d. The thruster controller presented here assumes the ambient water velocity is provided.

In a typical application, the desired thrust force, T_d, is known, rather than the desired propeller angular velocity. The thrust force is related to $\Omega|\Omega|$ in equation (2.5). For the first stage of the inversion, this is simplified to Ω^2. The impact of the absolute value shall be discussed later in this section. Combining this equation with equation (2.12) gives a quadratic relation for the desired propeller angular velocity, Ω_d, as shown in equation (5.2).

$$0 = \left(\frac{k_1 u^2}{D^2} - \frac{T_d}{\rho D^4}\right)\left(\frac{1}{\Omega_d}\right)^2 + \frac{k_2 u}{D}\frac{1}{\Omega_d} + k_3 \tag{5.1}$$

$$= \left(\frac{k_1 u^2}{D^2} - \frac{T_d}{\rho D^4}\right) + \frac{k_2 u}{D}\Omega_d + k_3\Omega_d^2 \tag{5.2}$$

The process for determining Ω_d is:

- Use ambient water velocity and the sign of the desired thrust force to determine sign of J_0.

- Determine the state of the system from the sign, as shown in Figure 2.5.
  - If $J_0 > 0$, the state is equi-directional.
  - If $J_0 < 0$, the state is either anti-directional or vague-directional. Further analysis is required to determine the state, which is described below.
- Use the state to determine the set of k_i coefficients to use from Table 4.4, on page 4.4.
- Find the non-imaginary root of equation (5.2).

The coefficients k_i in equation (5.2) are dependent on which of the three states the system is in. Determining the state of the system is not a straight-forward process. The simplest state is if the desired thrust force is zero. If so, then by the choice of force datum stated in Section 4.4 the desired propeller angular velocity is also zero. If the desired thrust force is not zero, the sign of the ambient water velocity is used to determine the state. The sign indicates which direction the water is being pushed through the thruster and which $K_T(J_0)$ curve to use, from Section 4.7. If the ambient water velocity is 0, then $J_0 = 0$ and the coefficient for this case can be used. If the ambient water velocity is not 0, then the sign of J_0 can be determined with the signs of ambient water velocity and the desired thrust force.

If the thrust force is positive, the propeller is driving the axial water velocity into the negative direction. In the equi-directional and anti-directional states, the axial water velocity is fully developed regardless of the ambient water velocity. In the vague-directional state, the axial water velocity is not fully developed and is unknown. The reverse is true for the case where thrust force is negative: the propeller drives the axial water velocity in the positive direction. The equi-directional state is the case where the axial water velocity is the same as the ambient water velocity, as seen in Figure 2.5. Thus, if $\text{sign}(T_d) \neq \text{sign}(u)$ then the advance ratio is in the equi-directional state.

If $\text{sign}(T_d) = \text{sign}(u)$, the desired thrust is fighting the ambient water velocity. The desired thrust may be able to overpower the ambient water velocity, which causes the axial water velocity to be fully developed. This is the anti-directional state. Otherwise, the axial water

velocity is not fully developed; this is the vague-directional state. These states are differentiated by the CAR, J_0^*, as described in Section 2.4. Since the value of J_0 is unknown, a method of determining which of these two states the system is in must be found.

A method that uses the first term of equation (5.1) is used to different between these to states. If the absolute value from equation (2.5) is restored over the desired propeller angular velocity, this term has the form in equation (5.3).

$$\frac{k_1 u^2}{D^2 \Omega_d |\Omega_d|} - \frac{T_d}{\rho D^4 \Omega_d |\Omega_d|} \tag{5.3}$$

This term has embedded into it the values for $K_T(J_0)$ from equation (2.3) and J_0 from equation (2.4), as shown in equation (5.4).

$$\frac{k_1 u}{D|\Omega_d|} J_0 - K_T(J_0) \tag{5.4}$$

In equation (5.4), the sign of Ω_d is preserved through the value for J_0, which is not reflected in equation (5.1). Equation (5.4) can be unwrapped to determine the sign of the desired propeller angular velocity. Equation (2.5) shows that the thrust force and propeller angular velocity have different signs when $J_0 < 0$. Using this sign preservation, the transition between the anti- and vague-directional states is determined to be the point where equation (5.4) is zero, as shown in equation (5.5).

$$\frac{k_1 u}{D|\Omega_d|} J_0 - K_T(J_0) = 0 \tag{5.5}$$

Using this as the transition point a condition for the anti-directional state is created, shown in equation (5.6). If this relation holds true, the system is in the anti-directional state. The

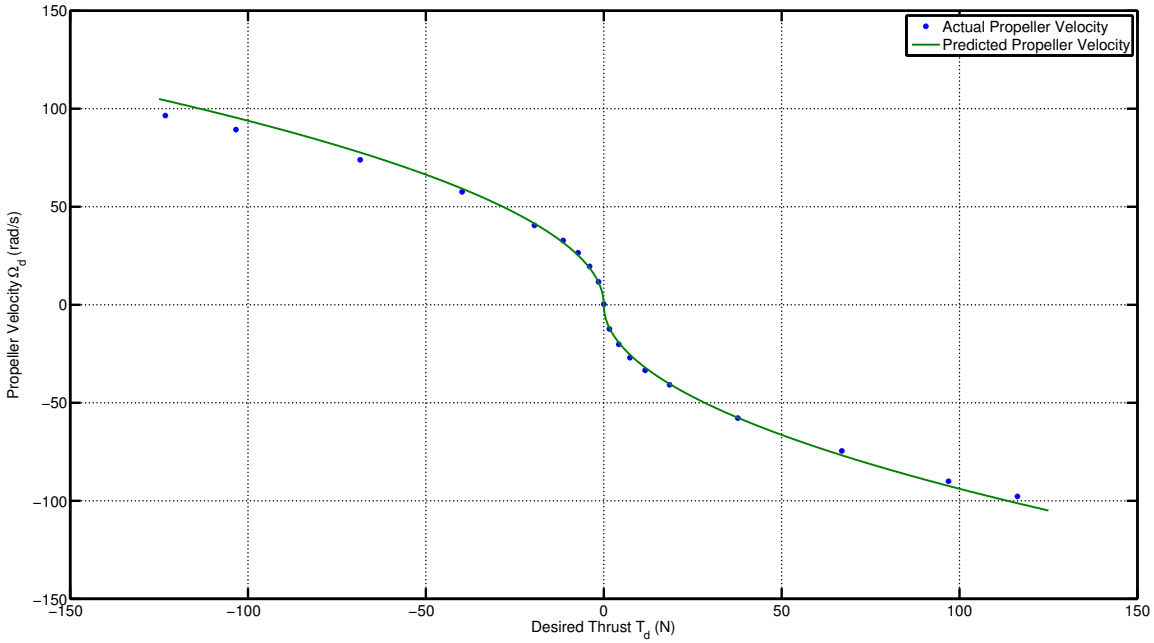

Figure 5.1 – Predicted Ω_d compared with the experimental data, for $u = 0$.

absolute value around T_d is because the sign of the ambient water velocity, u, is lost squaring.

$$\frac{k_1 u^2}{D^2} < \frac{|T_d|}{\rho D^4} \tag{5.6}$$

If the relation in equation (5.6) does not hold true, the thrust force given is in the vague-directional state. Once the state, and thus the set of k_i coefficients, to use is determined, the roots are calculated using the quadratic formula. Figures 5.1, 5.2, 5.3 and 5.4 show the output of the predicted Ω_d using the above process against the experimental data already collected. The calculated output in these graphs include discontinuities where $J_0 = 0$ and $J_0 = J_0^*$. This is from the discontinuities in the piecewise function of the thrust coefficient, $K_T(J_0)$.

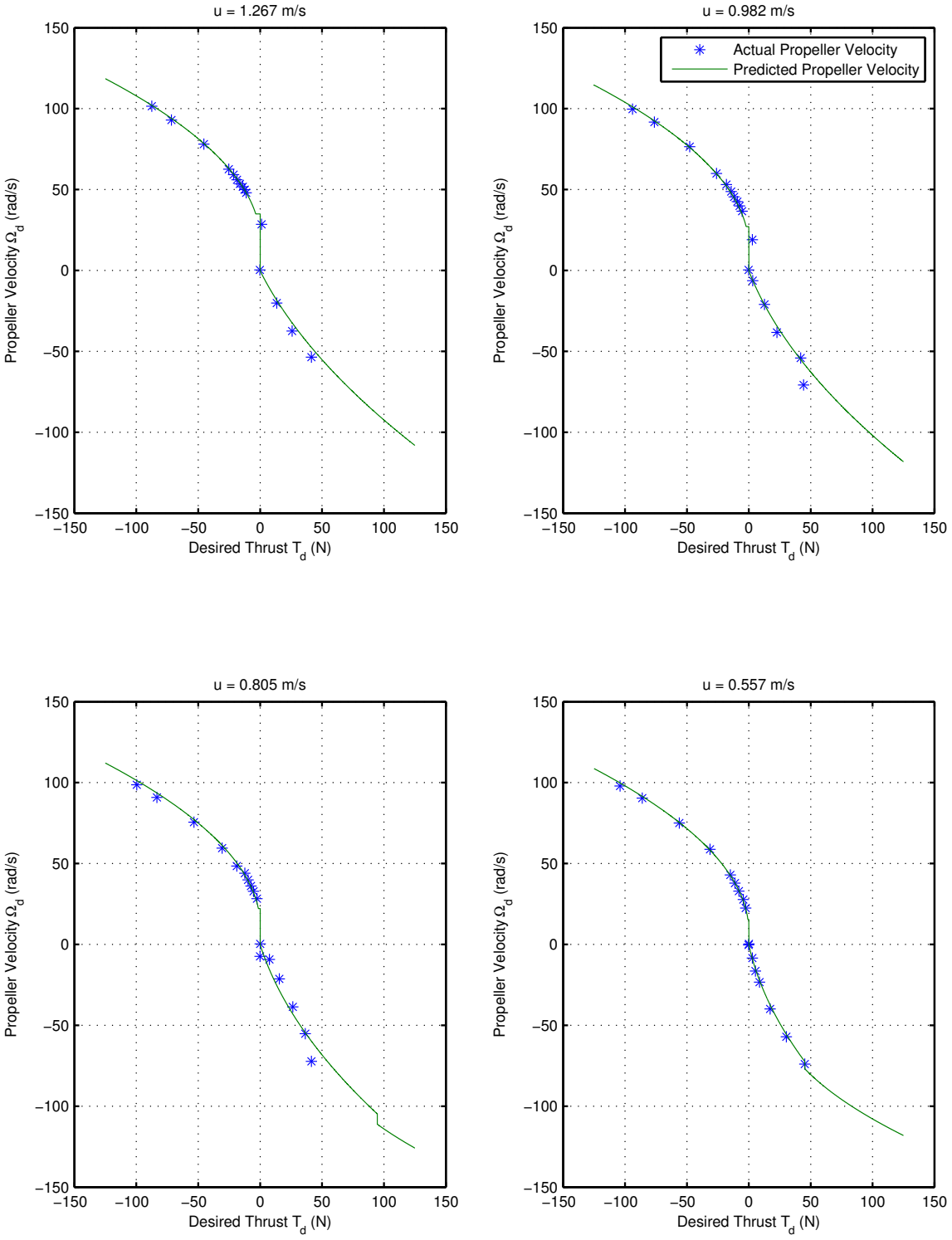

Figure 5.2 – Predicted Ω_d compared with the experimental data, for specified ambient water velocities.

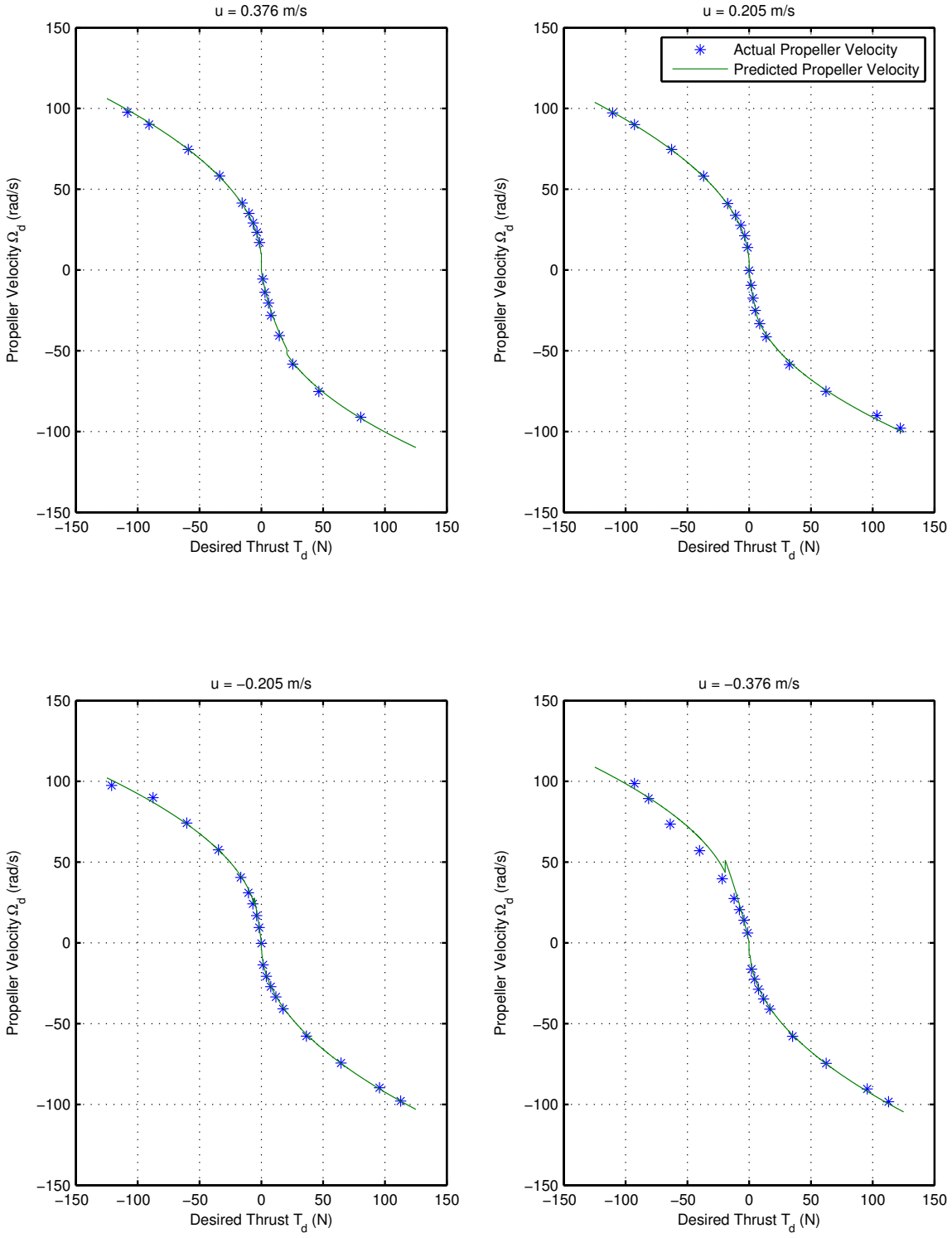

Figure 5.3 – Predicted Ω_d compared with the experimental data, for specified ambient water velocities.

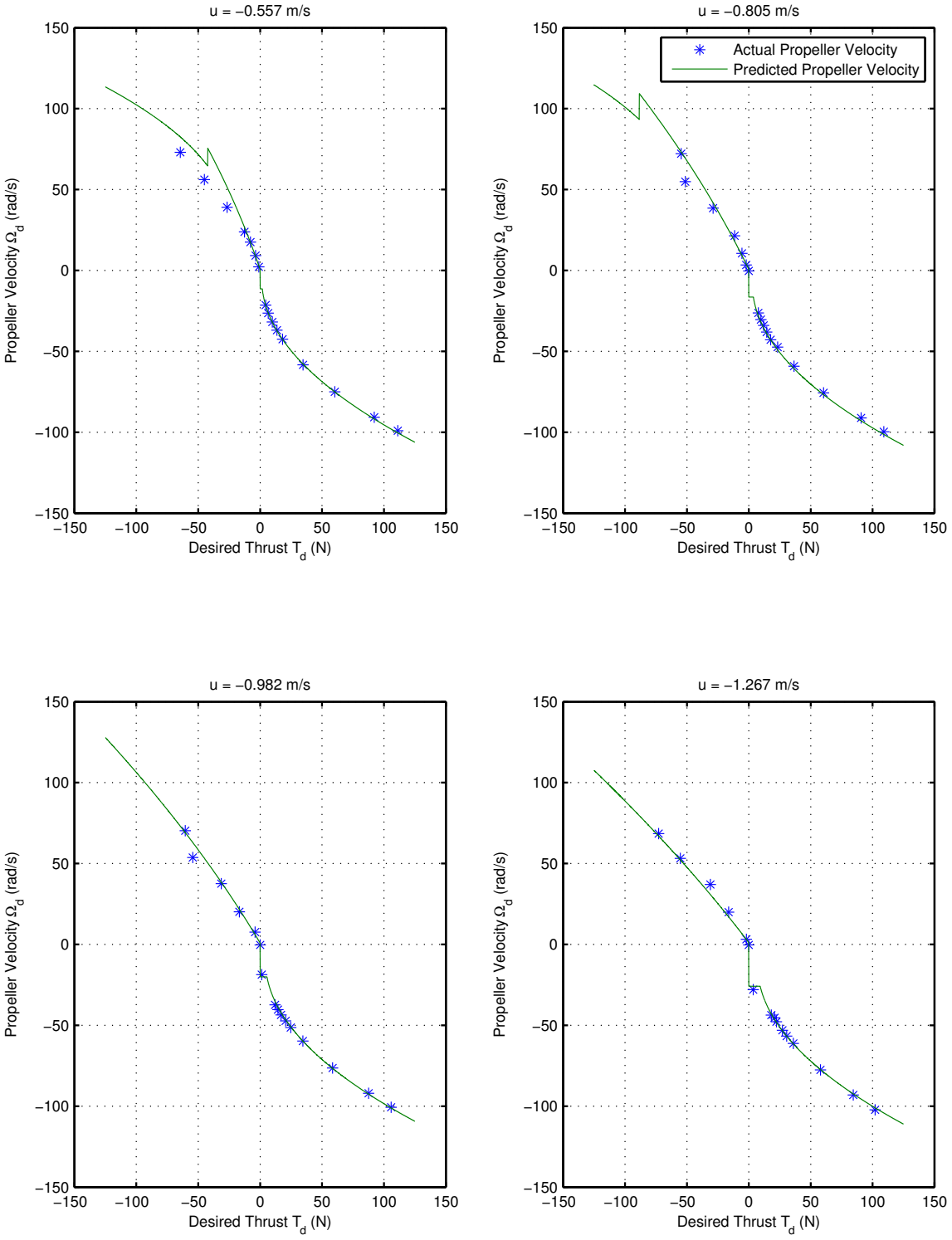

Figure 5.4 – Predicted Ω_d compared with the experimental data, for specified ambient water velocities.

5.2 Determining the Thruster Input Command

Once the desired propeller angular velocity was determined, the thruster input command was calculated. As stated in Section 4.2, the propeller angular velocity is dependent on both the thruster input command and the ambient water velocity in a way that is difficult to model. To circumvent this problem, a PID control loop is used. The drawback of using a PID control loop is that the output of the PID control loop is linear with respect to the desired propeller angular velocity, but the actual relationship between the thruster control input and the propeller angular velocity is not linear. This makes accurate control across the entire range of the thruster difficult. This section describes some methods of compensating for this problem.

One of the thruster properties discovered during testing is the difference in the thruster mechanics between situations of increasing and decreasing propeller angular velocity. Imagine the case where ζ_0 is the initial positive thruster input command driving the thruster, and another positive value, ζ_1, is then commanded. If $\zeta_1 > \zeta_0$, the thruster internal controller drives the motor up to the final propeller angular velocity. If $\zeta_1 < \zeta_0$, the propeller angular velocity is decreased. In this case, the thruster internal controller allows the propeller to slow down via hydrodynamic drag rather than applying a torque to drive it down. The difference between these two cases is shown in Figure 5.5. The same effects are seen for negative values of ζ_1 and ζ_2 as well.

The change with respect to propeller angular velocity for a percent point of input diminishes as the thruster input command reaches $\pm 100\%$. In addition there is a deadband that the PID controller is oblivious to. The deadband is asymmetric in the forward and reverse directions and changes for static and dynamic propeller conditions, as discussed in Section 4.3. To compensate for these two conditions an iso-parametric mapping between the linear PID output, η, and the non-linear thruster input command, ζ, is created. This mapping sits between the output of the PID controller and the input to the thruster.

An iso-parametric mapping takes an input, locks down some control points, and uses a

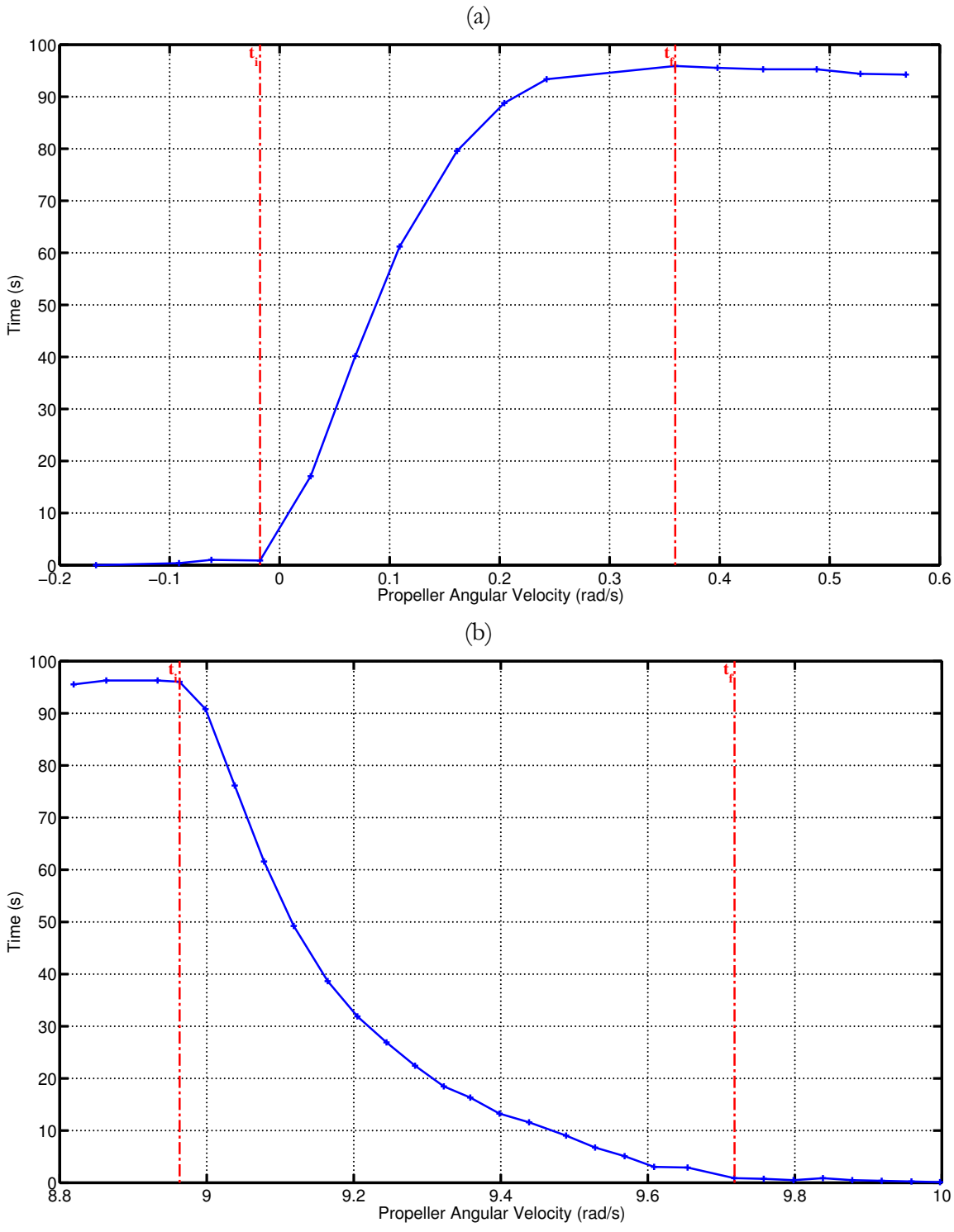

Figure 5.5 – The difference in rise and fall times for the propeller angular velocity. (a) The propeller angular velocity is driven up to the final speed, with a rise time of 0.377s. (b) The propeller angular velocity is allowed to slow without being driven, with a fall time of 0.755s. The fall time is over twice the rise time.

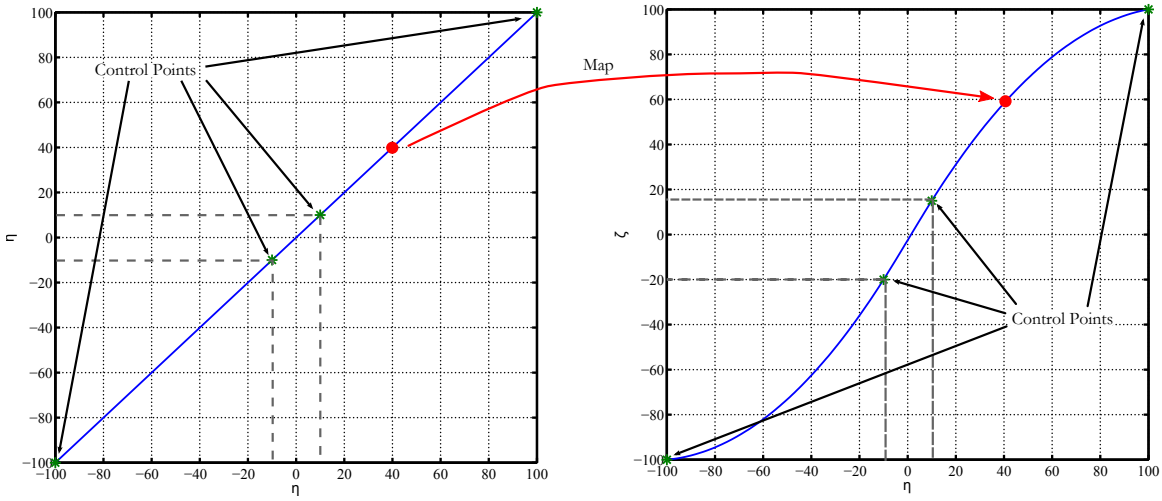

Figure 5.6 – Example of an iso-parametric mapping. The linear output on the left is mapped to a cubic polynomial on the right using the control points shown. The value 40 from the input on the left is mapped to the value 60 in the output on the right.

function to map between those control points to create an output. As an example, a mapping might take a 1-d line as an input and map it to a curve in 2-d space, as shown in Figure 5.6. The goal is to create a mapping that takes the linear PID output, η, and map it to the non-linear thruster input, ζ, to compensate for the deadband and the diminishing thrust at higher thruster input commands.

One benefit of the mapping is to limit the time spent in the deadband. The deadband changes based on whether the propeller is moving, thus it is undesirable to entirely skip this area. Instead, a sharp slope is created between the control points. Another benefit of using the iso-parametric mapping is to make the deadband look symmetric to the PID control loop. The deadband on the thruster is asymmetrical, but through the mapping this space may be made symmetrical for the PID output. This allows the deadband zone to be reachable, but small changes to η have a larger impact to ζ in this space.

The mapping also stretches the effective input zones for the thruster. On the thruster, a change from 30% to 40% thruster input command has a much larger impact on the propeller angular velocity than the change from 80% to 90%. A mapping can smooth these differences

such that a 10% of full scale change in the PID output results in a similar change in the propeller angular velocity throughout the range of thruster input commands. This difference can be in the results shown in Figure 4.3. While such a perfect 1:1 mapping cannot be achieved, some of the non-linearity in the ζ to Ω relationship could be removed.

The thruster input command is bounded by $\pm 100\%$. For convenience, these values are chosen for two of the control points for η. The thruster input command is limited to integer values within this range. This is handled by rounding the output from the iso-parametric map to the nearest integer within this range. However, the PID output is not rounded before being fed back into the PID input. This prevents stabilization problems with the PID controller.

Four iso-parametric mappings were created for testing. Each has a different construction, with potentially different effects on the conversion from η to ζ. The equations for these mappings are all based on third-order polynomials. Third order polynomials were chosen because they are the lowest order that provides a good match for the translation of ζ to Ω, shown in Figure 4.3. All four mappings have control points at ± 100 to match the top and bottom range of the thruster input. These four mappings are:

- 3rd-order iso-parametric map with two locked control points for the deadband.
- 3rd-order iso-parametric map with two shifting control points based on whether the propeller is moving or stationary.
- Three 3rd-order B-Spline sections, connected at C^0 continuous points at two locked control points for the deadband.
- Three 3rd-order B-Spline sections, connected at C^2 continuous points at two locked control points for the deadband.

The third-order iso-parametric map was constructed using the deadband points as control points for ζ, which were shown in Figure 4.5 from Section 4.3. These points were mapped to the two points $[-10, 10]$ in the PID controller output η, as shown in Table 5.1. With respect to the PID controller output η, these two control points create a symmetric deadband about

$\eta = 0$. In addition, the effective deadband is made smaller with respect to the PID output η. The other two control points are $[-100, 100]$ for each of ζ and η. These are the limits of the thruster input, and it makes it easy to conceptualize the translation from η to ζ. The control points are summarized in Table 5.1. Four cubic polynomials, $N_i(\eta)$, are created for each of these four control points, constructed to fit the equation (5.7).

$$\zeta(\eta) = [N_1(\eta)\ N_2(\eta)\ N_3(\eta)\ N_4(\eta)] \begin{bmatrix} -100 \\ -20 \\ 15 \\ 100 \end{bmatrix} \tag{5.7}$$

The control points are used to create constraints for this equation at each of the set points, and are solved to determine the coefficients. The functions built in this way are shown in equations (5.8a) through (5.8d).

$$N_1(\eta) = -\frac{1}{1936000}\eta^3 + \frac{49}{968000}\eta^2 + \frac{1}{6050}\eta - \frac{3}{484} \tag{5.8a}$$

$$N_2(\eta) = \frac{1}{216832}\eta^3 - \frac{5}{108416}\eta^2 - \frac{625}{13552}\eta + \frac{3125}{6776} \tag{5.8b}$$

$$N_3(\eta) = -\frac{1}{217800}\eta^3 - \frac{1}{18150}\eta^2 + \frac{50}{1089}\eta + \frac{200}{363} \tag{5.8c}$$

$$N_4(\eta) = \frac{1}{2016000}\eta^3 + \frac{17}{336000}\eta^2 + \frac{1}{25200}\eta - \frac{1}{168} \tag{5.8d}$$

To use these functions, the value from the PID controller, η, is evaluated for each of these four functions to create four values. These are then multiplied by the four ζ_i control points, as shown in equation (5.7). This gives a single value for ζ, which is used as the thruster input command. This process maps the linear PID output to a non-linear thruster input command. The curve created by this mapping is shown in Figure 5.7.

The second mapping under test is constructed using a similar process as the first. The

Table 5.1 – Mapping control points used for the third-order iso-parametric map. These control points are used to solve for the four third-order polynomials in equation (5.7).

i	η	ζ
1	-100	-100
2	-10	-20
3	10	15
4	100	100

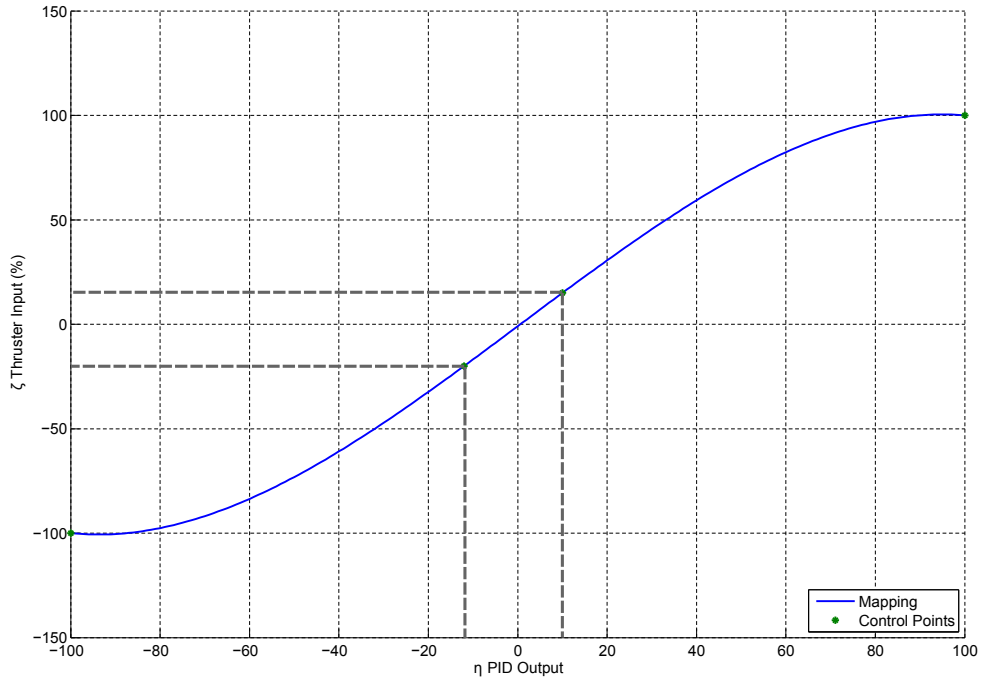

Figure 5.7 – Iso-parametric map, with control points from Table 5.1.

Table 5.2 – Mapping control points used for the third-order iso-parametric map with shifting control points. The same third-order polynomials given in equation (5.7) can be used because the PID output η control points have not changed, only the ζ control points.

Range	i	η	ζ
Static Deadband	1	-100	-100
	2	-10	-20
	3	10	15
	4	100	100
Dynamic Deadband	1	-100	-100
	2	-10	-20
	3	10	15
	4	100	100

process of generating and using the four $N_i(\eta)$ functions is the same as previous. Different control points in η_i would necessitate re-calculating the values for the functions. However, changing the set of control points for ζ_i has no effect on the set of $N_i(\eta)$ functions. Given a static set of η_i control points, they may be mapped to any arbitrary set of ζ_i control points.

To make the shifting map, the four ζ_i control points are changed based on the condition of the propeller angular velocity. One set is used when the propeller velocity $|\Omega_d| < 0.5$rad/s; the other set is used when $|\Omega_d| \geq 0.5$rad/s. This shifting map uses the same control points for η_i as the previous case. These control points are summarized in Table 5.2. Since both mappings use the same control points, they can both use the same set of $N_i(\eta)$ equations given in equations (5.8a) through (5.8d). Because the deadband changes based on whether the propeller is in motion, this mapping shifts based on the propeller angular velocity to map the deadband into a larger or smaller space. This effectively creates two mappings, shown in Figure 5.8.

The next mapping is a C^0 continuous B-spline. In this mapping three 3rd order polynomials are created such that they connect at the control points but no other restrictions are placed. The result is a system of three third order equations with 12 unknowns. With each equation, a reasonable fit is made empirically based on the output data. Unlike in the previous two map-

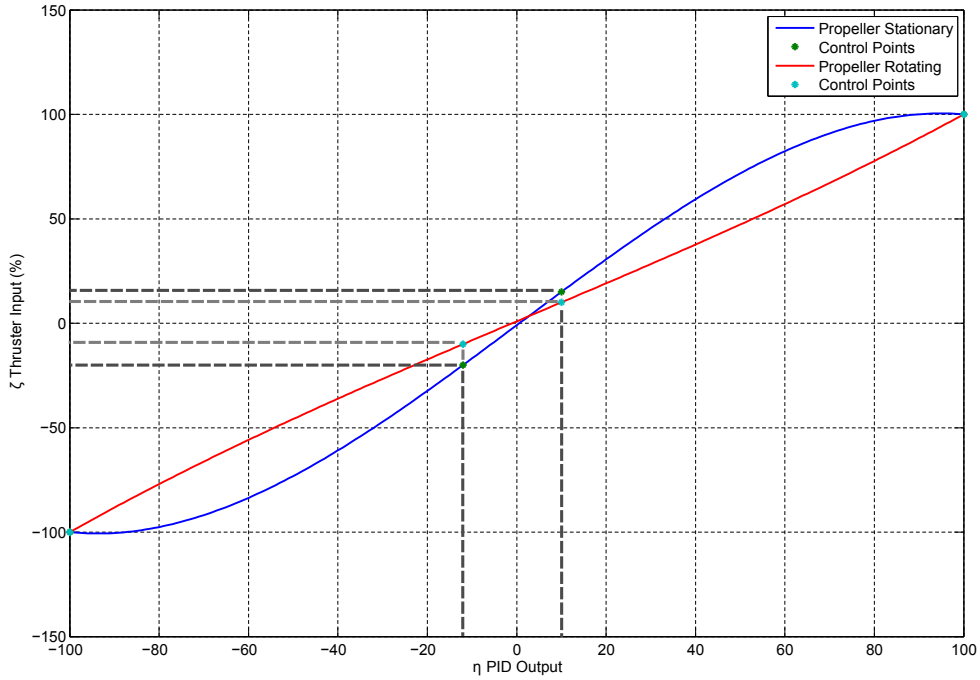

Figure 5.8 – Iso-parametric map with two sets of control points from Table 5.2.

pings, the three B-splines are independent outside of the control points used to clamp down the edges. This creates a single C^0 continuous line over the entirety of the control space, as shown in Figure 5.9.

The C^2 mapping is built in a similar manner as the C^0 mapping: with three 3rd order B-Splines that are connected at the control points. However, this mapping adds the constraints that each of the B-Spline curves are C^2 continuous, which makes for a much smoother curve over the entire mapping. In the C^0 mapping above, there are discontinuities in the output as the PID output η crosses the control points. The purpose of the C^2 mapping is to smooth the transitions when crossing control points. The mapping is shown in Figure 5.10. In both of the B-Spline mappings the control points used are the same as in the iso-parametric mapping, shown in Table 5.1.

Each of these four cases are constructed using the results from Chapter 4. They were tested

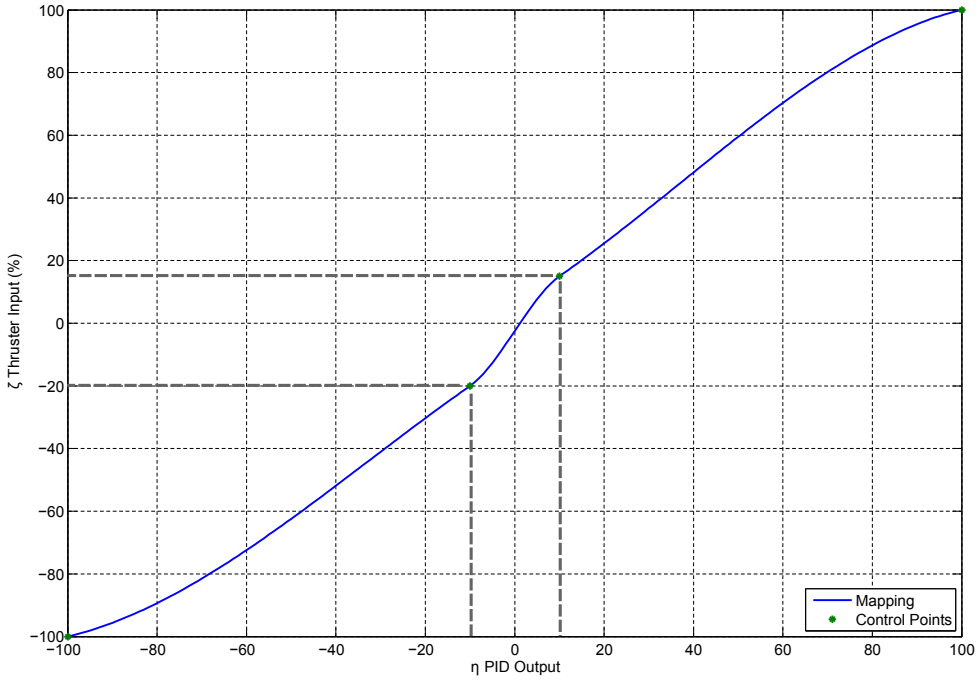

Figure 5.9 – Iso-parametric mapping, C^0 continuous B-Spline.

in the free-standing tank, described in Section 3.2.6, and compared against the PID controller without a mapping. The results of these tests are shown in Chapter 6.

5.3 Integrating the Controller

Once the controller was designed as described in this chapter, it was implemented on the thruster and tested. The inputs to the controller are the ambient water velocity and the desired thrust force. The full control diagram is shown in Figure 5.11.

When the low-level thruster controller described here is used as part of a navigation system aboard a ROV, some method of measuring the ambient water velocity in the vicinity of the thruster must be found. One method is to determine the ROV's velocity, then transform that into a velocity in the reference frame of each thruster. This method is outlined by Proctor [15]. The thruster controller here may be used in conjunction with the guidance, navigation and

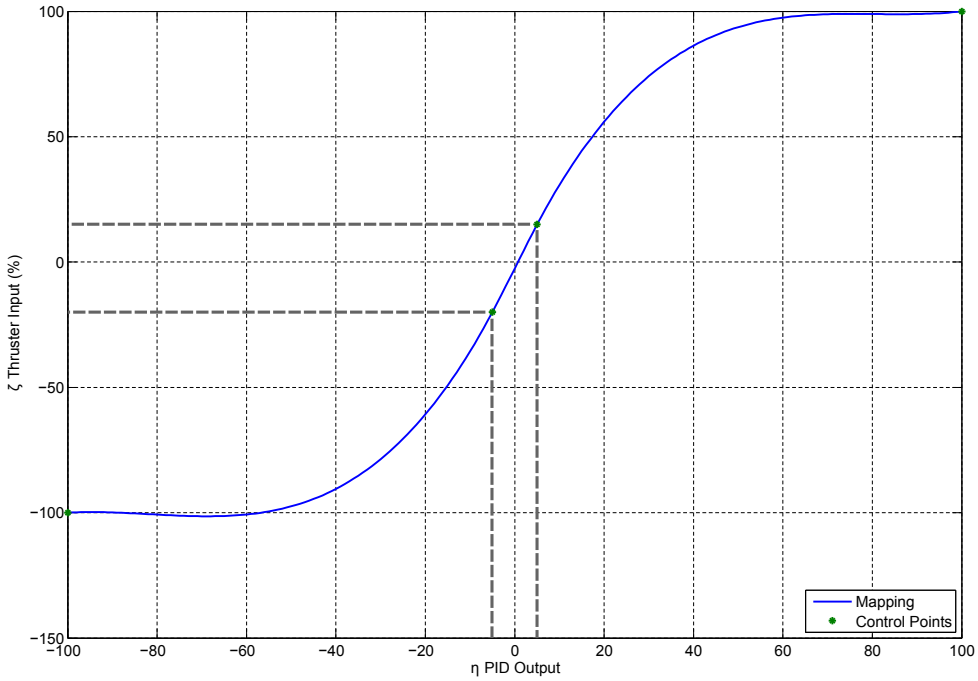

Figure 5.10 – Iso-parametric mapping, C^2 continuous B-Spline.

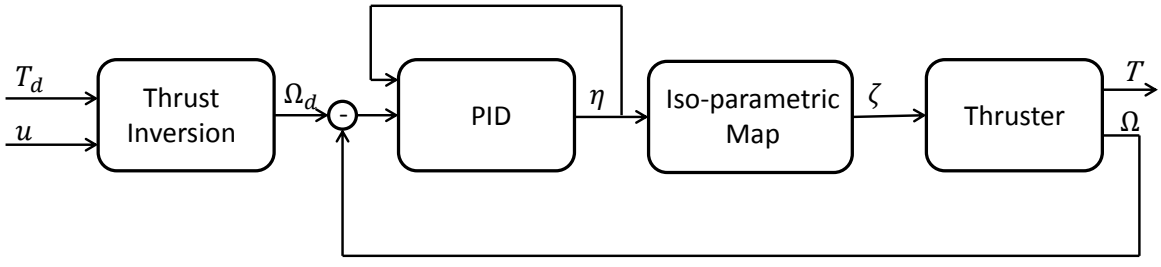

Figure 5.11 – Diagram of the thruster controller.

control systems as outlined in Chapter 1. The results of testing the controller outlined here are given in Chapter 6.

Chapter 6

Results of Thruster Controller Testing

This chapter summarizes the results gathered during trials of the thruster controller described in Chapter 5. Five controller configurations were tested: the PID controller without a mapping, and the PID controller with each of the four mappings. The tests performed were step-response tests and six target propeller angular velocities that covered a sample of the entire range of the thruster. The test results for the no-mapping case are described in Section 6.1, and the results of tests that include the various mappings are given in Section 6.2.

Robustness tests of the controller's ability to provide a constant thrust force are described in Section 6.3. These tests involved returning the thruster to the flume tank and testing the controller's ability to regulate propeller angular velocity subject to a range of ambient water velocities. The results of the robustness testing are described in Section 6.3.

6.1 PID Controller Response Test Results

Before testing the PID controller developed in Section 5.2, the PID control gains tuned. The tuning process involved measuring the response of the propeller angular velocity to a step thruster input command. Three measurements were minimized through this process:

Rise time: This is the time it takes the propeller velocity to reach 95% of the desired output

from the step input.

Overshoot percentage: This is the amount by which the output overshoots the desired propeller angular velocity. As discussed in Section 5.2 on page 86, propeller deceleration takes much longer than acceleration. For propeller deceleration, large overshoots result in disproportionately longer settling times.

Settling time: This is the length of time it takes the propeller angular velocity to stabilize within an envelope of 5% of the desired propeller angular velocity. If the output oscillates around the set point but never remains within the 5% bound, this value is not defined.

Of these three metrics, minimizing the overshoot percentage is expected to have the single greatest positive impact on the efficiency of the controller. The Ziegler-Nichols method was used as a starting point for the gain tuning. From there the PID gains were heuristically adjusted based on observation of the system outputs.

A basic PID equation is shown in equation (6.1).

$$\eta_{i+1}=K_P\mathcal{E}_i + K_D\frac{\mathcal{E}_i - \mathcal{E}_{i-1}}{\Delta t_i} + K_I\left(\sum_i \mathcal{E}_i\Delta t_i\right) \tag{6.1}$$

$$\mathcal{E}_i=\Omega_d - \Omega_i \tag{6.2}$$

$$\Delta t_i=t_i - t_{i-1} \tag{6.3}$$

When using the PID equation without a mapping, the PID output $\eta = \zeta$. In equation (6.1), the value of i is the most recent output from the PID controller, and $i + 1$ is the next output. The error between the desired propeller velocity, Ω_d , and the measured propeller velocity, Ω_i, is given by $\mathcal{E}_i$. The required gains are the proportional gain, K_P, the derivative gain, K_D, and the integral gain, K_I.

The PID equation in equation (6.1) had to be modified to account for the noise in the propeller angular velocity output. Specifically, the derivative term was smoothed. Instead of

using a single sample of the changing error, the derivative term was calculated using the mean of several successive samples, as shown in equation (6.4).

$$\eta_{i+1}=K_P\mathcal{E}_i + \frac{K_D}{n}\left(\sum_{j=0}^{n}\frac{\mathcal{E}_{i-j}-\mathcal{E}_{i-j-1}}{\Delta t_{i-j}}\right) + K_I\left(\sum_i \mathcal{E}_i\Delta t_i\right),\ n=5 \qquad (6.4)$$

The tuning was completed in the free-standing tank described in Section 3.2.6. The resulting gains from the tuning are shown in Table 6.1. Six target propeller angular velocities were chosen as set points for the controller gain tuning. These target propeller angular velocities were 52rad/s, 26rad/s, 9rad/s, −9rad/s, −26rad/s and −52rad/s.

While tuning the gains for the PID controller for the three positive set points, the following observations were made:

- Equation (6.4) improved the accuracy of the derivative term as the output settled, but it caused inaccuracies while the output was rising. These inaccuracies could increase the rise time. In testing, adding this smoothing prevented some of the unwanted oscillation around the set point and improved settling time.
- When the PID controller was tuned for the mid range desired propeller angular velocity, 26rad/s, the step response would overshoot at higher desired angular velocities.
- Tuning the PID controller for the higher desired propeller angular velocity, 52rad/s, caused the lower desired propeller angular velocity to be over-damped. n
- In the case of low desired propeller angular velocity, 9rad/s, the thruster is near its deadband. The PID controller raised the thruster input command until the propeller angular velocity rose out of the static deadband, at which point the propeller angular velocity would significantly overshoot the desired value.

A balanced set of values was used to minimize the effects of the mid- and high-range desired propeller angular velocities while mitigating the effects near the deadband. Once the gains were tuned for the positive propeller angular velocity set points, the negative set points were tested.

Table 6.1 – The resulting gains from tuning the PID controller with no mappings.

Gain	Units	Value
K_P	$\%/s$	0.05
K_D	$\%/s^2$	0.00
K_I	$\%$	1.80

Table 6.2 – Step response data for the PID controller without a mapping across the set of test desired propeller angular velocities.

Set Point (rad/s)	95% Rise Time (s)	Overshoot (%)	Settling Time (s)
52	0.851	3.270	1.025
26	0.962	5.184	1.397
9	1.515	16.295	9.093
− 9	1.544	22.940	8.797
−26	0.983	5.184	1.576
−52	0.773	4.745	1.030

The asymmetrical nature of the thruster in positive and negative directions added another level of difficulty in finding the PID gains. The thruster input commands map differently in the negative direction from the positive, so the gains don't respond the same in each direction. This can be seen in the difference in the hysteresis response from Section 4.3, shown in Figure 4.5. Only one set of gains were used for the entire range of the thruster, so a balance had to be struck between the positive and negative set points. The set of gains that were found to give reasonable results across all the test set points are shown in Table 6.1. The result of the step responses are given in Table 6.2, and shown in Figures 6.1 and 6.2.

The difficulties in using the PID controller outlined above can be mitigated by several methods. One method is to use different gains for the positive and negative propeller angular velocities. Another is to use a function for the parameters rather than constant gains. One method is to map the output of the PID controller through iso-parametric maps, which is the method described in Section 5.2 and used in this work. The results of the map testing are compared to the PID results in the following section.

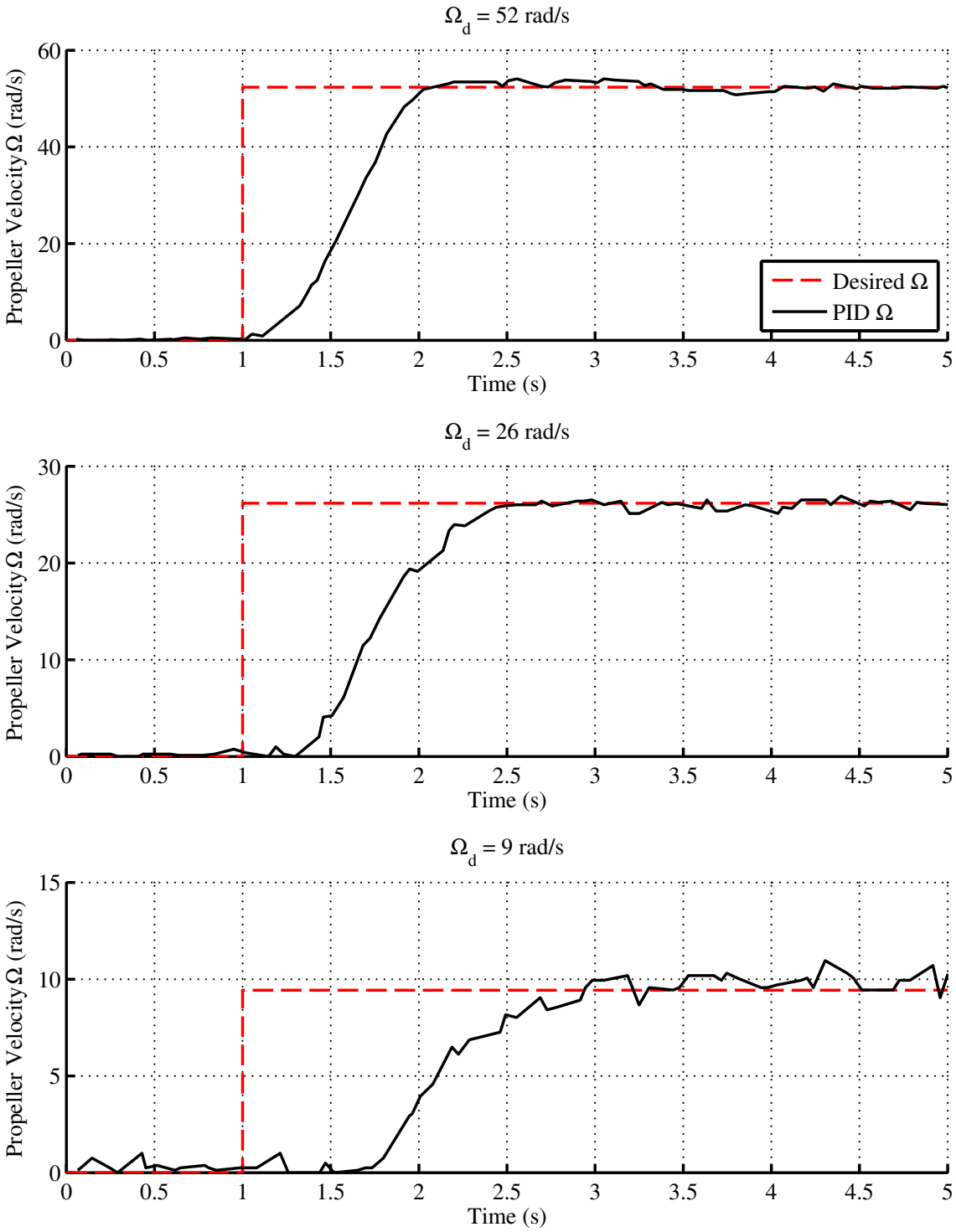

Figure 6.1 – Results of the tuned PID controller for positive propeller angular velocities.

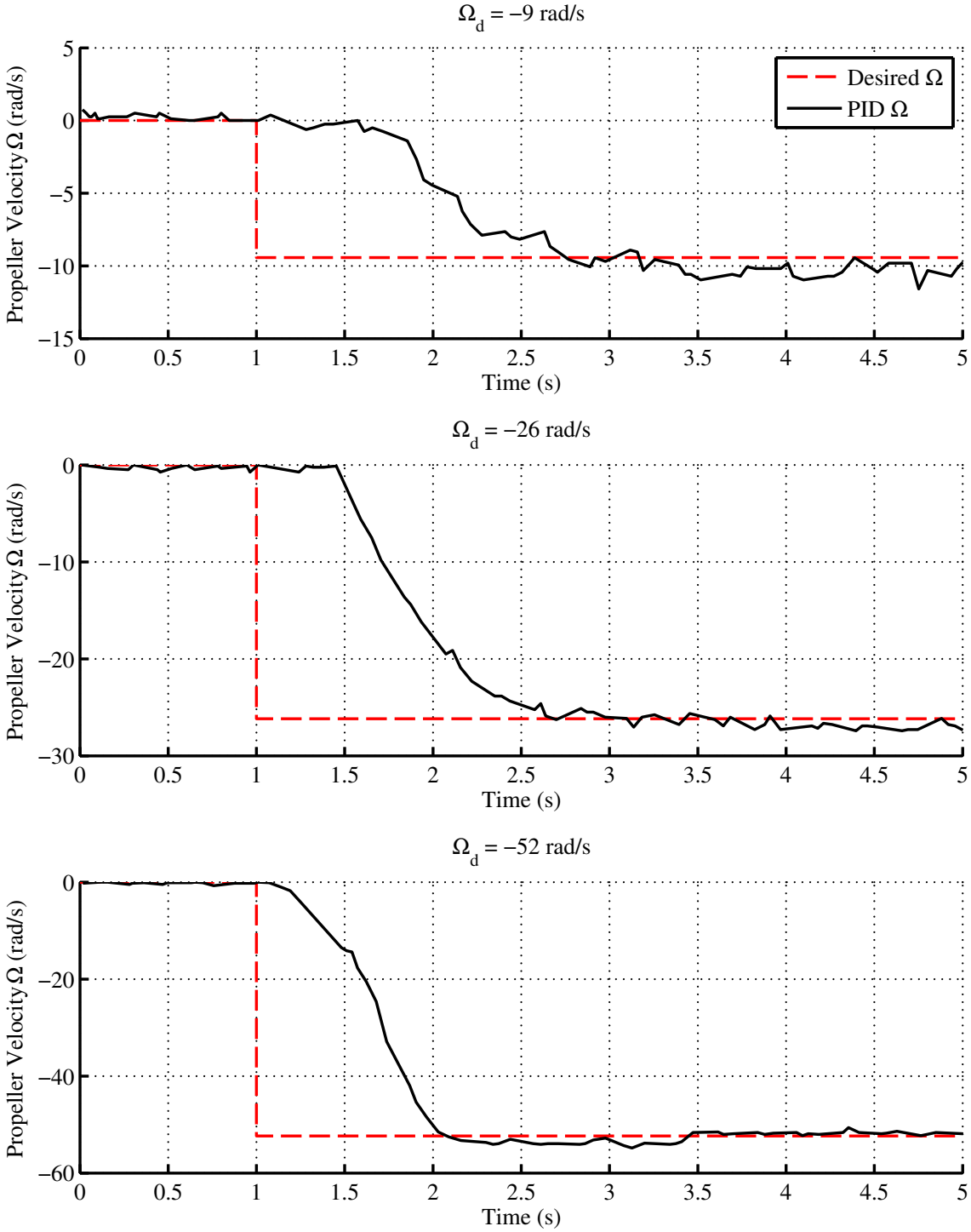

Figure 6.2 – Results of the tuned PID controller for negative propeller angular velocities.

6.2 Iso-Parametric Mapping Results

The previous section outlined problems in the PID controller surrounding the difference in forward and reverse propeller angular velocity directions, especially around the asymmetric deadband. To mitigate these problems, the control input mappings were added to the system. The four control input mappings were:

- 3rd-order iso-parametric map (Figure 5.7).
- 3rd-order iso-parametric map with two shifting control points (Figure 5.8).
- Third order B-Splines with C^0 connections (Figure 5.9).
- Third order B-Splines with C^2 connections (Figure 5.10).

For each mapping, the PID gains had to be tuned in the manner as described in the previous section. The gains from the PID controller without a mapping were used as a starting place for each of the four mappings. The final PID gain results are shown in Table 6.3. A comparison of the rise time, overshoot percentage and settling time achieved with each of the four mapping strategies is given in Table 6.4 and shown in Figures 6.3, 6.4 and 6.5, below.

The first mapping considered was the third-order iso-parametric map, shown originally in Figure 5.7. This mapping showed depressed rise times and settling times over the larger propeller angular velocity set points without much change in the response to the lower set points. The overshoot percentage showed a small amount of improvement with lower angular velocity set points, but not at the higher points. This mapping showed no improvement over the straight PID controller, instead this mapping retarded the response of the controller.

The third-order iso-parametric map with shifting control points, shown originally in Figure 5.8, provided similar results as the third-order iso-parametric map. It improved the overshoot for the higher thruster velocities, but not by much. It tended toward longer settling times and showed no improvement in any of the three performance metrics for the smaller set points. One of the consequences of the shifting control points were instabilities around very

Table 6.3 – The resulting gains from tuning the PID controller with each of the four mappings. The gains from the PID controller without mappings has been reproduced for reference.

Mapping	Gain	Units	Value
PID, No Mapping	K_P	$\%/s$	0.05
	K_D	$\%/s^2$	0.00
	K_I	$\%$	1.80
Third-Order Iso-Parametric	K_P	$\%/s$	0.20
	K_D	$\%/s^2$	0.00
	K_I	$\%$	1.80
Shifting Iso-Parametric	K_P	$\%/s$	0.10
	K_D	$\%/s^2$	0.00
	K_I	$\%$	1.30
Third-Order C^0 B-Splines	K_P	$\%/s$	0.20
	K_D	$\%/s^2$	0.01
	K_I	$\%$	2.00
Third-Order C^2 B-Splines	K_P	$\%/s$	0.20
	K_D	$\%/s^2$	0.01
	K_I	$\%$	1.00

low propeller angular velocities, less than the lowest test point of $\pm$9rad/s. This is caused by the thruster having an inconsistent speed around the threshold velocity of 0.5 rad/s. There may be a way to fix this by changing the threshold, or shifting the threshold based on whether the propeller is speeding up or slowing down. However, this was not pursued as other control input mappings provided better results.

The third-order C^0 B-Spline mapping, shown originally in Figure 5.9, improved on the response time from the PID controller, and the settling time in some cases. The response improvement is prominent at the lower thruster velocities; at 9rad/s it responded much faster than the PID controller. However, at $-$9rad/s it was very unstable. The controller was unable to find an input that kept the propeller out of the deadband at this set point. This may be a consequence of the discontinuity at the control point, which caused a sudden change in the output of the controller. It performed better than the PID controller in terms of the rise time,

but otherwise the results were similar to the PID controller with the exception of the stability problem at lower propeller angular velocities.

The third-order C^2 B-Spline mapping, shown originally in Figure 5.10, showed a small improvement and the greatest consistency across all six target velocities. Like the C^0 mapping, the C^2 mapping improved the thruster response to the step input, but the settling times were on par with the PID controller. At higher target velocities, the output was over-damped compared to the PID controller without a mapping. This reduced overshooting the target, but slowed the settling time. This mapping gave a consistent response across the range of desired propeller angular velocities.

Figures 6.3 ($\pm$9rad/s), 6.4 ($\pm$26rad/s) and 6.5 ($\pm$52rad/s), below, show a comparison of all five of the mappings together. The combination of PID controller and the C^2 B-Spline mapping gives the best results, as shown. For the purposes of the low-level thruster controller used in the rest of this chapter, the third-order C^2 B-Splines mapping was used due to the high consistency and thruster response times. Iso-parametric mapping showed some promise in taking the PID output to the thruster input and could be further explored. In specific, the creation and tuning of the mappings could be revisited to better match the thruster input command curve.

6.3 Steady-State Thrust Results

Steady-state thrust force testing was performed in the flume tank to determine the efficacy of the low-level thruster controller in practise. Given the ambient water velocity and a desired thrust force, the open-loop low-level thruster controller commanded the thruster to accurately provide the desired thrust force. The thrust was measured independently using the experimental setup described in Chapter 3. Ambient water velocities used for testing were chosen as a mix of previously used values (0m/s, $\pm$0.205m/s, $\pm$0.376m/s, $\pm$0.557m/s and $\pm$0.805m/s) and new values ($\pm$0.285m/s, $\pm$0.669m/s). This was done to compare the new results to ex-

Table 6.4 – Step response data for the PID controller and for each of the four mappings across the set of test desired propeller angular velocities.

Mapping	Set Point (rad/s)	95% Rise Time (s)	Overshoot (%)	Settling Time (s)
PID, No Mapping	52	0.851	3.270	1.025
	26	0.962	5.184	1.397
	9	1.515	16.295	9.093
	−9	1.544	22.940	8.797
	−26	0.983	5.184	1.576
	−52	0.773	4.745	1.030
Third-Order Iso-Parametric	52	0.839	1.316	1.098
	26	0.836	4.227	7.377
	9	0.913	24.491	11.594
	52	0.839	1.316	1.098
	−26	0.738	6.140	1.492
	−52	0.516	2.791	0.603
Shifting Iso-Parametric	52	0.782	3.509	1.005
	26	0.934	4.227	1.155
	9	0.671	21.611	16.950
	−9	0.395	28.478	13.828
	−26	0.788	6.619	1.858
	−52	0.851	1.316	1.111
Third-Order C^0 B-Splines	52	0.431	1.794	0.642
	26	0.938	4.705	1.209
	9	1.358	21.611	8.634
	−9	0.364	66.357	—
	−26	0.495	8.134	1.601
	−52	0.739	2.034	0.884
Third-Order C^2 B-Splines	52	0.991	1.794	1.092
	26	1.496	4.705	1.680
	9	0.561	16.295	10.801
	−9	4.529	16.295	10.849
	−26	0.660	4.227	0.762
	−52	1.099	1.794	1.183

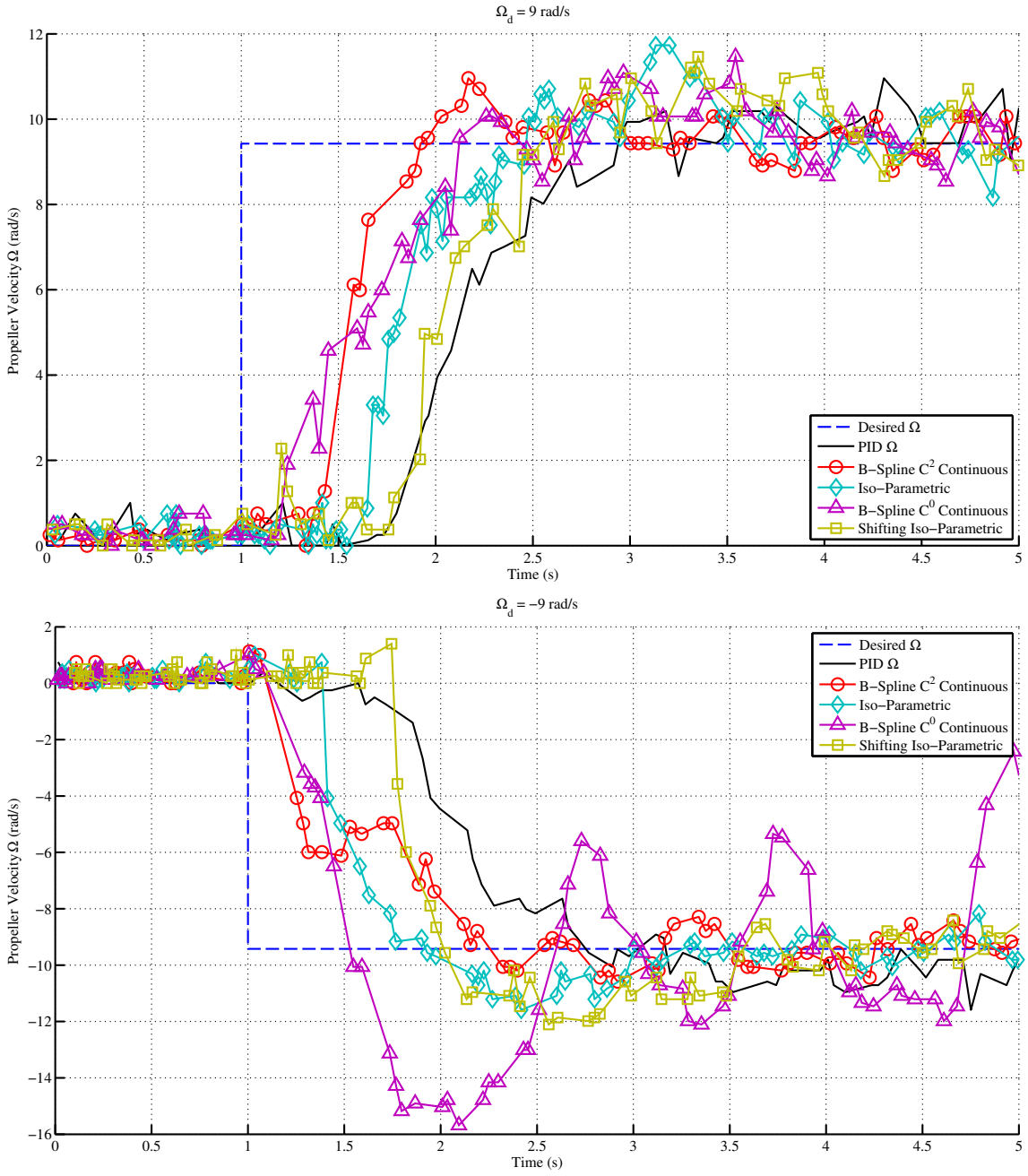

Figure 6.3 – Summary of thruster controller step responses, ± 9 rad/s.

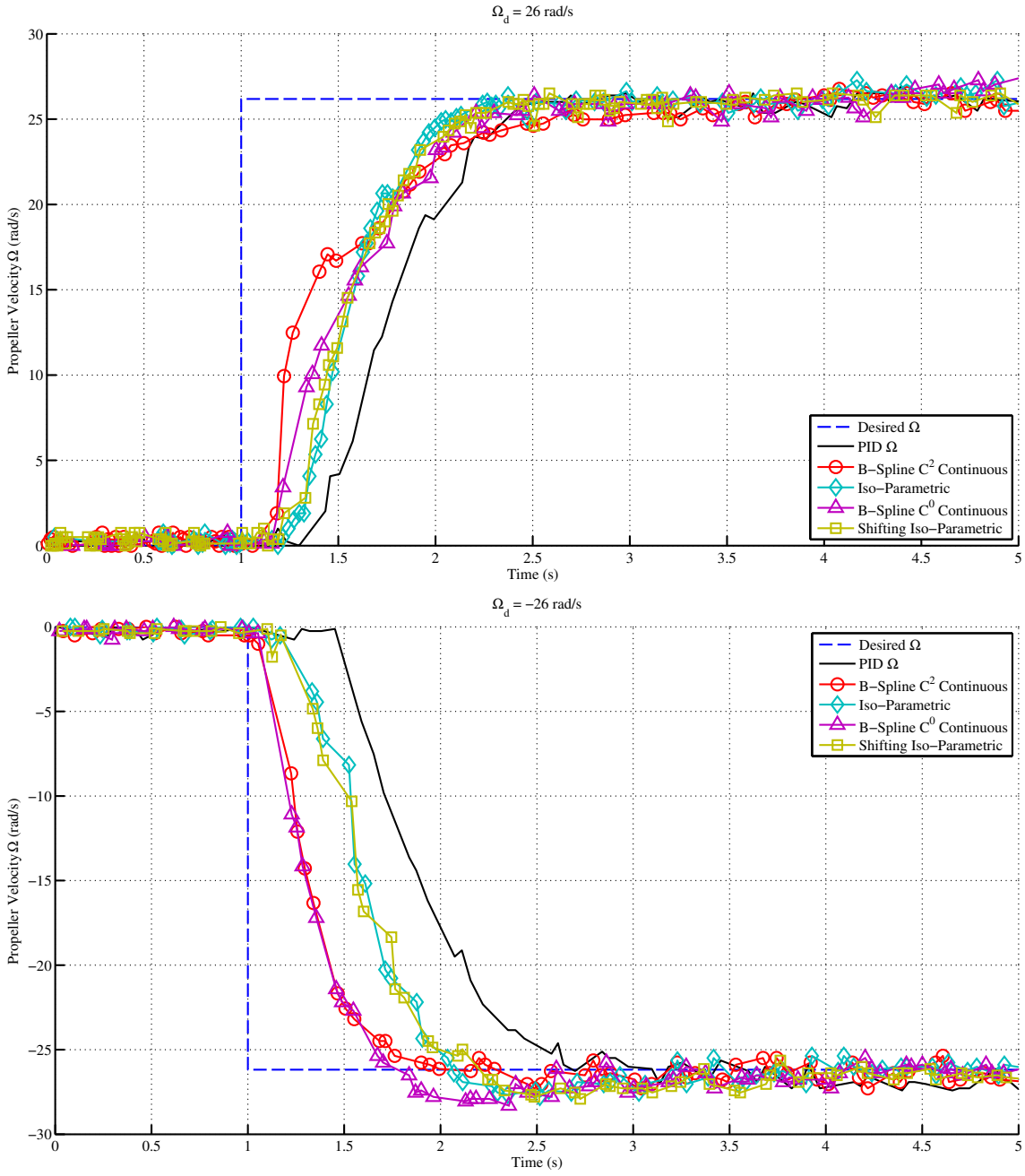

Figure 6.4 – Summary of thruster controller step responses, ± 26 rad/s.

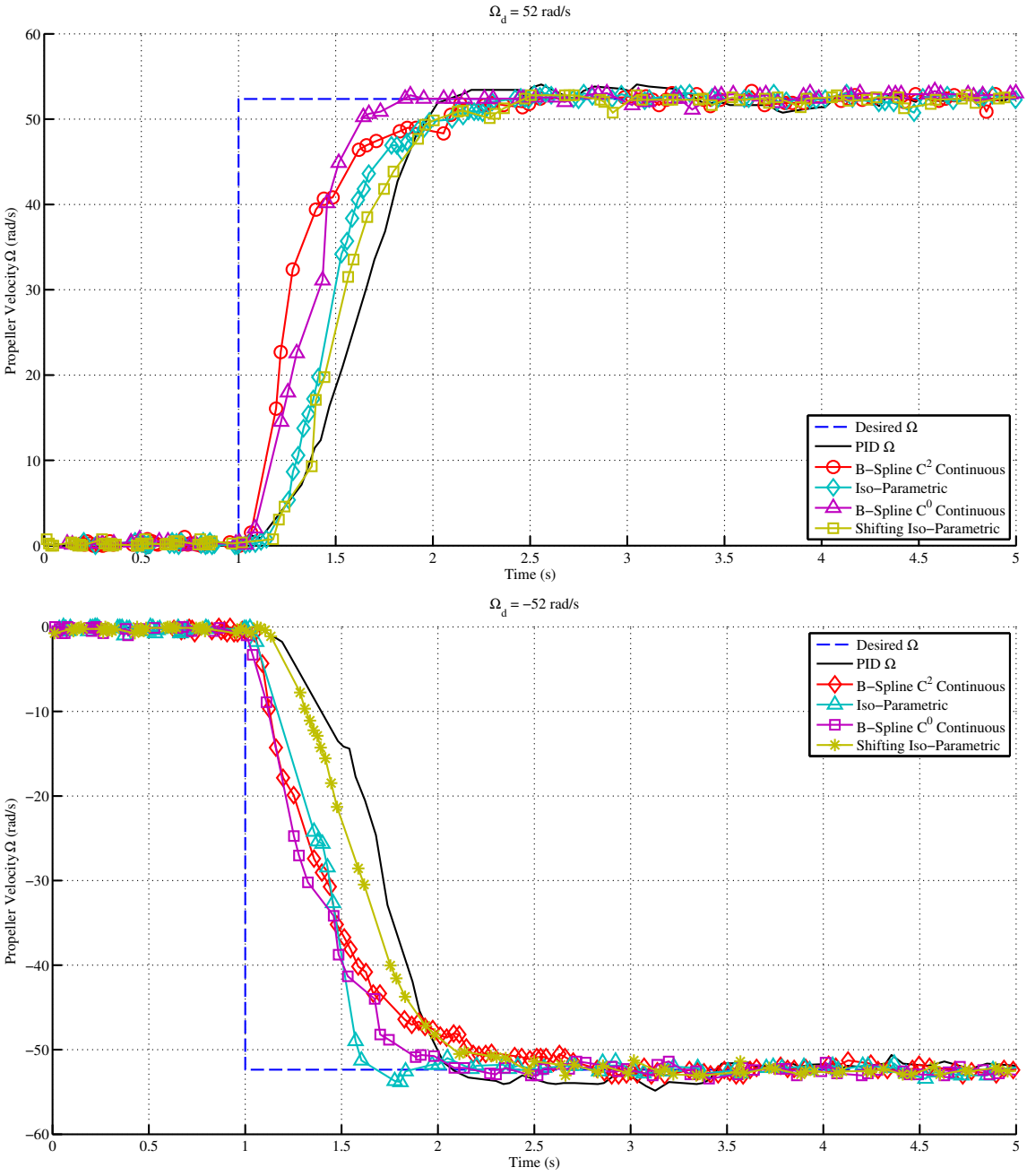

Figure 6.5 – Summary of thruster controller step responses, ± 52 rad/s.

isting data, and show how well the results could be generalized. In addition, lower speeds were chosen to reduce the near-surface effects that occurred at high propeller angular velocities, as described in Chapter 4. The controller successfully generated the desired thrust based on the model over a range of speeds. There was some asymmetry in the results, shown in Table 6.5. Steady-state results were found for each of the target ambient water velocities, which are shown in the graphs on the following pages.

The results are shown in two ways. The first is to show the measured thrust force against the desired thrust force. The desired thrust force is given by the dashed line on the graph. Another graph shows the absolute error of each measured thrust force, in Newtons. This gives a comparison of how the thruster performs across the range of desired thrust values. In most cases, larger desired thrust force corresponded with larger error, though this wasn't universal as described below.

With no ambient water velocity the desired thrust and actual thrust showed very low error, as seen in Figure 6.6. This is the simplest form of the model, and an indicator that the controller works at a basic level. The maximum error without ambient water velocity is 4.04 N. As the ambient water velocity changed, the results became less clear. At higher ambient water velocities, the error increases. But, the negative thrust force direction is consistently less error-prone than the positive thrust force direction, regardless of the ambient water velocity. This is shown in the case of $\Omega_d > 0$ from Table 6.5. There was a larger error in the thruster for this case for every test ambient water velocity. This error did not consistently correspond to the equi-directional, anti-directional or vague-directional states, but on the direction of force. In the thruster orientation as it is used on the vehicle, this corresponds to reverse thrust. This may be attributed in part to a bias of the thruster itself; it may be designed for more efficiency or power in one direction over the other. More investigation is required to determine the root cause of this difference.

The results here show that the low-level thruster controller is valid, but requires further

Table 6.5 – Average steady-state thrust force error for different water speeds, in Newtons. Note that the case where $J_0 > 0$ is the equi-directional state and $J_0 < 0$ covers both anti-directional and vague-directional states. Larger errors are consistently seen in the case where $\Omega_d > 0$, regardless of the ambient water velocity.

u (m/s)	$J_0 > 0$ (N)	$J_0 < 0$ (N)	$\Omega_d > 0$ (N)	$\Omega_d < 0$ (N)	$\Omega_d = 0$ (N)
−0.805	4.445	5.880	5.899	5.880	0.084
−0.669	8.369	8.908	12.535	8.908	0.038
−0.557	7.718	4.811	10.289	4.811	0.006
−0.376	4.362	2.061	5.817	2.061	0.000
−0.285	4.211	0.630	5.614	0.630	0.000
−0.205	5.035	2.097	6.713	2.097	0.000
0.000	1.316	0.000	1.554	0.984	0.024
0.205	1.093	14.983	14.983	1.276	0.001
0.285	1.095	6.841	6.841	1.277	0.000
0.376	0.318	6.522	6.522	0.473	0.009
0.557	0.800	5.792	5.792	0.933	0.002
0.669	0.677	9.756	9.756	0.789	0.000
0.805	0.707	8.420	8.420	0.824	0.005
Mean	3.088	5.900	7.749	2.380	0.013

investigation to improve accuracy. However, the benefit of this controller is that it can be built with minimal information about the thruster and a minimal amount of characterization time. It provides open-loop control that can be used as part of a higher-level guidance controller on the vehicle.

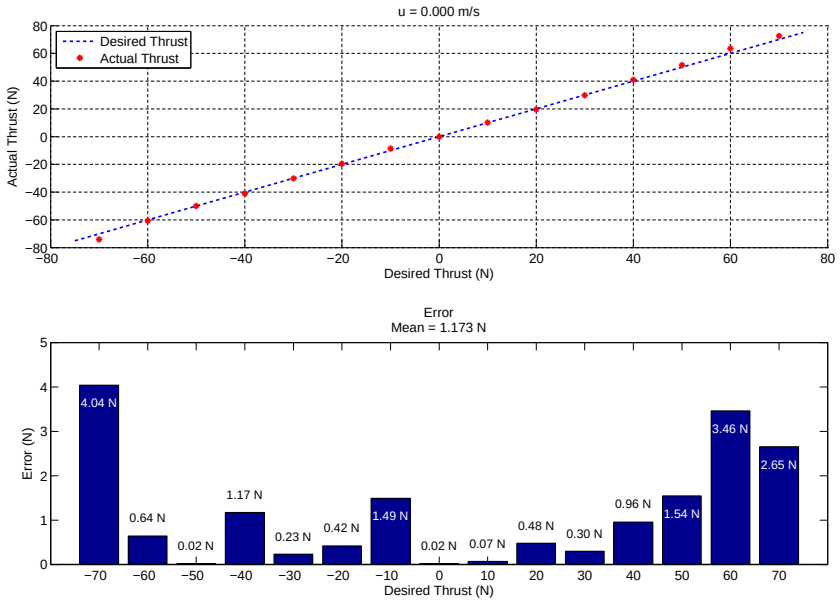

Figure 6.6 – Steady-state thrust and error for ambient water speed $u = 0.000$ m/s.

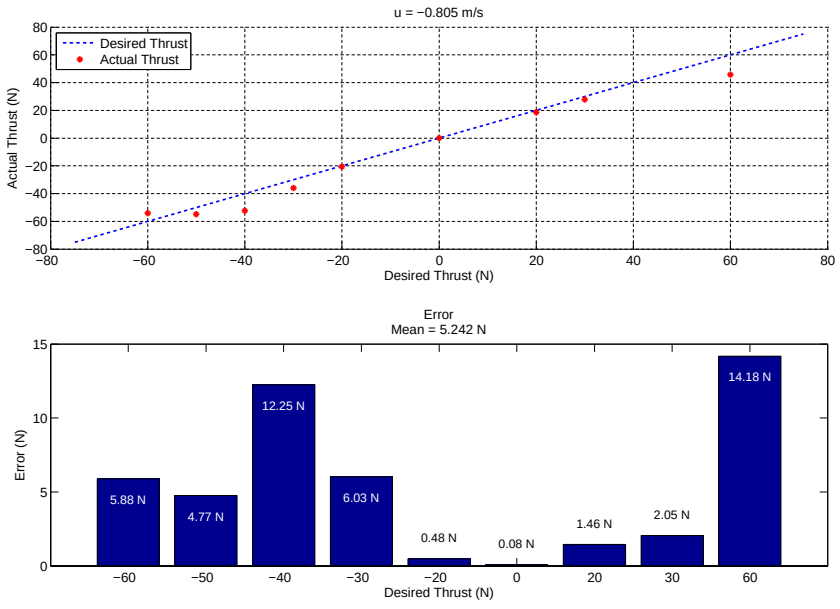

Figure 6.7 – Steady-state thrust and error for ambient water speed $u = -0.805$ m/s.

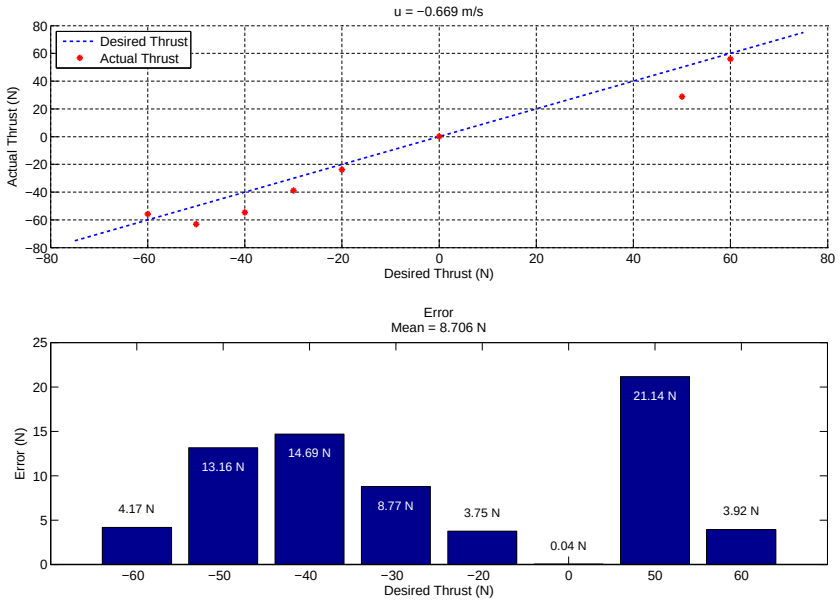

Figure 6.8 – Steady-state thrust and error for ambient water speed $u = -0.669$ m/s.

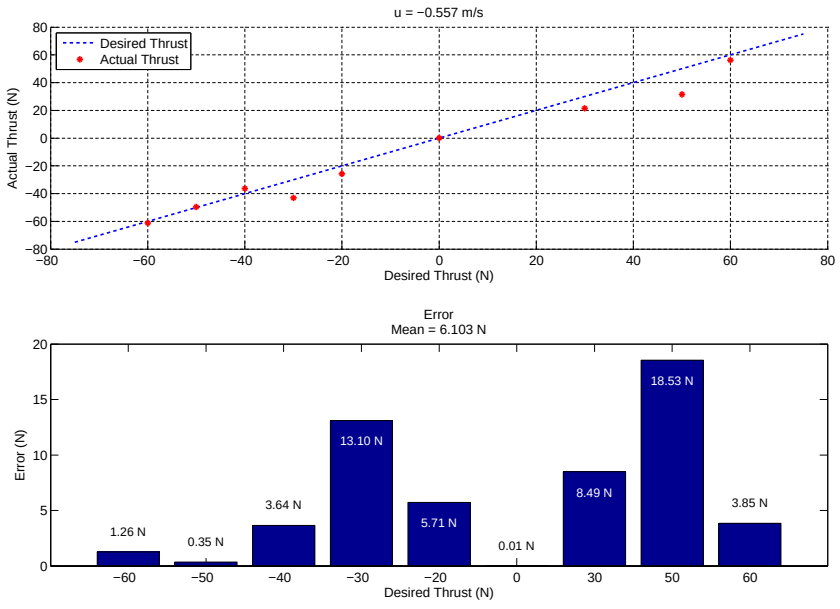

Figure 6.9 – Steady-state thrust and error for ambient water speed $u = -0.557$ m/s.

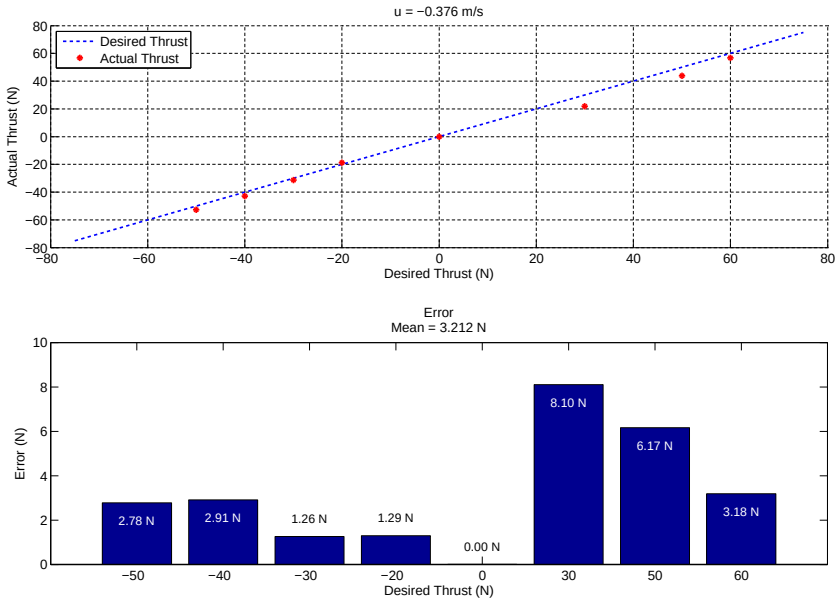

Figure 6.10 – Steady-state thrust and error for ambient water speed $u = -0.376$ m/s.

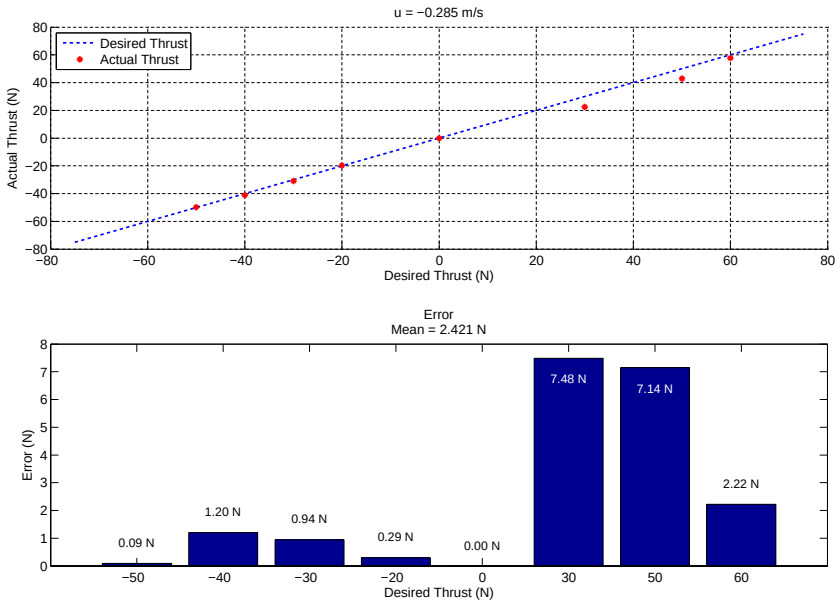

Figure 6.11 – Steady-state thrust and error for ambient water speed $u = -0.285$ m/s.

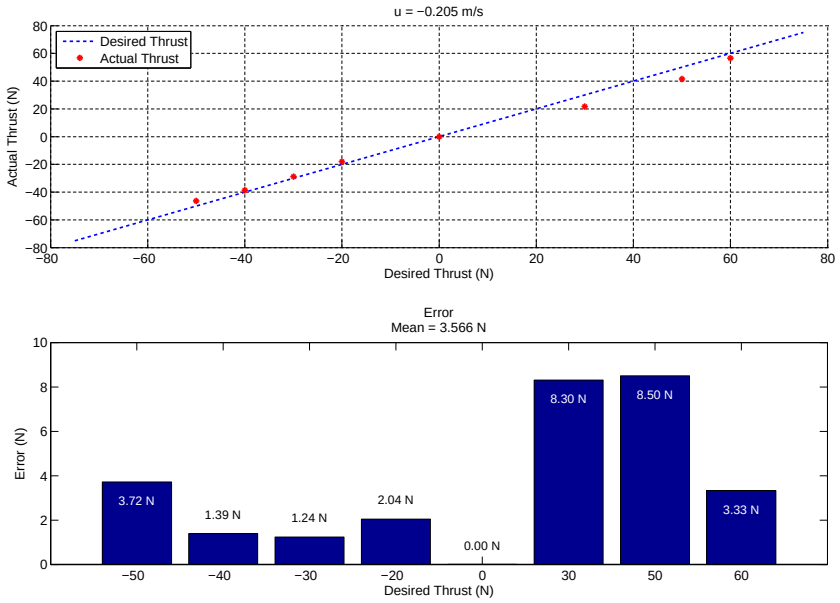

Figure 6.12 – Steady-state thrust and error for ambient water speed $u = -0.205$ m/s.

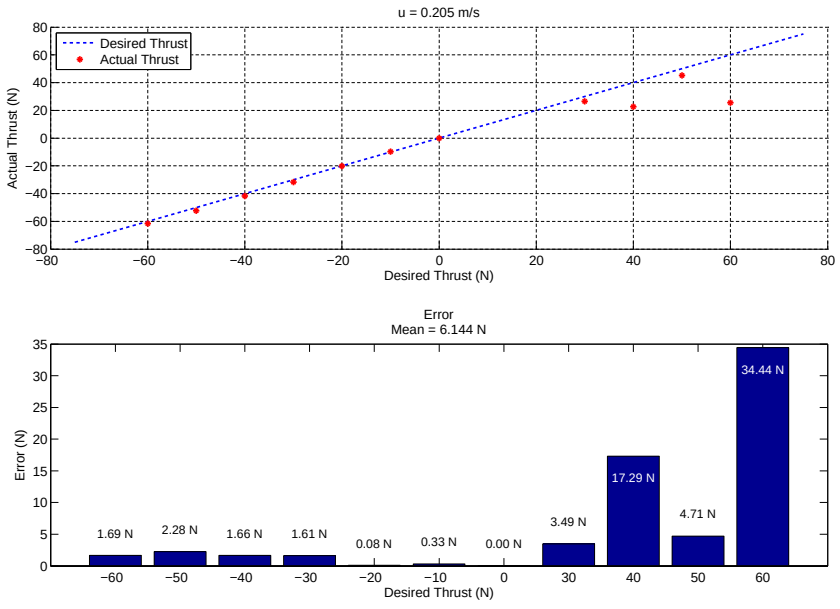

Figure 6.13 – Steady-state thrust and error for ambient water speed $u = 0.205$ m/s.

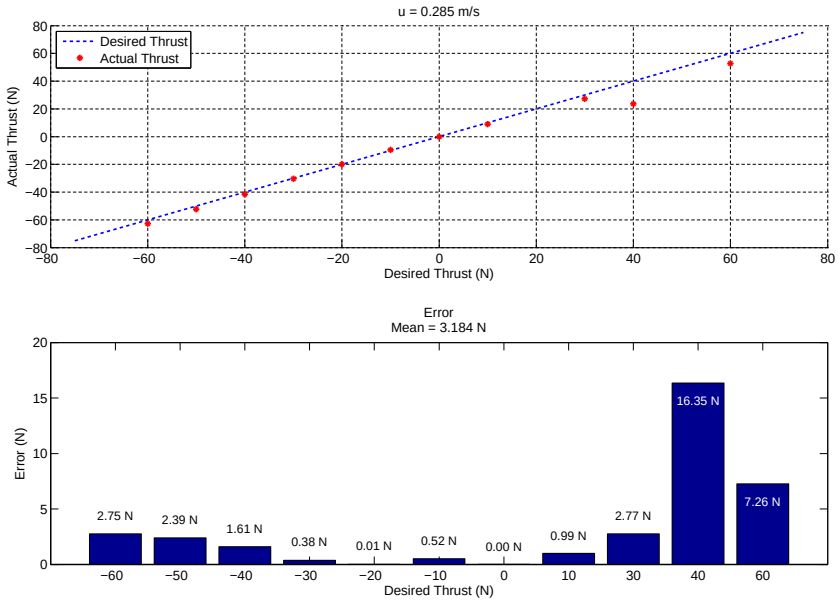

Figure 6.14 – Steady-state thrust and error for ambient water speed $u = 0.285$ m/s.

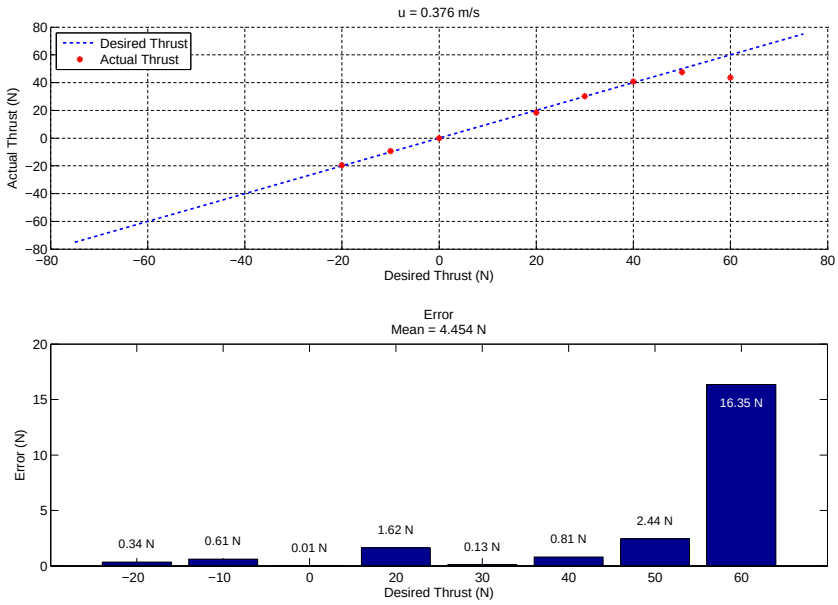

Figure 6.15 – Steady-state thrust and error for ambient water speed $u = 0.376$ m/s.

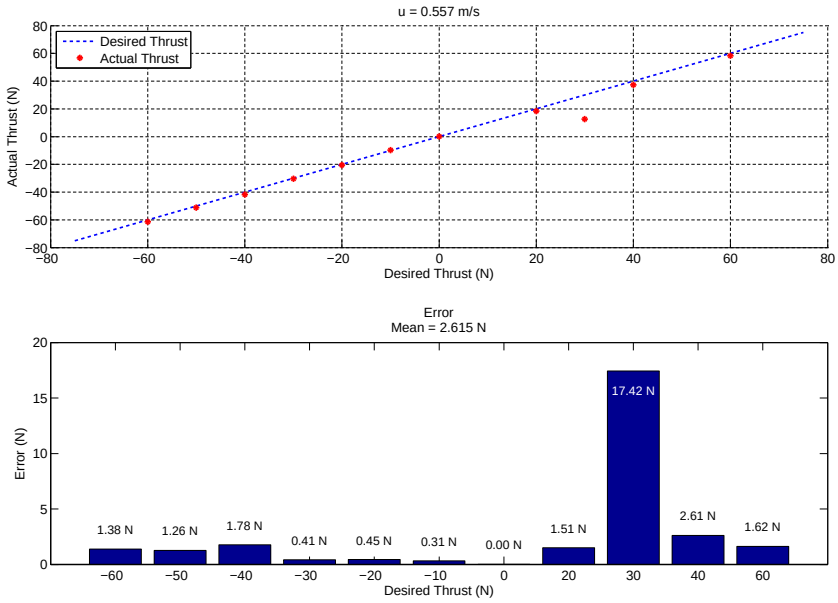

Figure 6.16 – Steady-state thrust and error for ambient water speed $u = 0.557$ m/s.

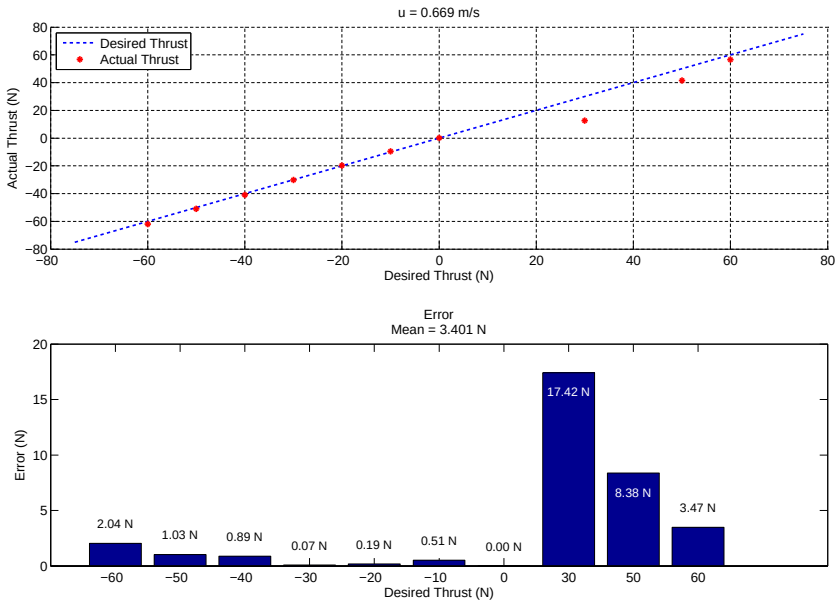

Figure 6.17 – Steady-state thrust and error for ambient water speed $u = 0.669$ m/s.

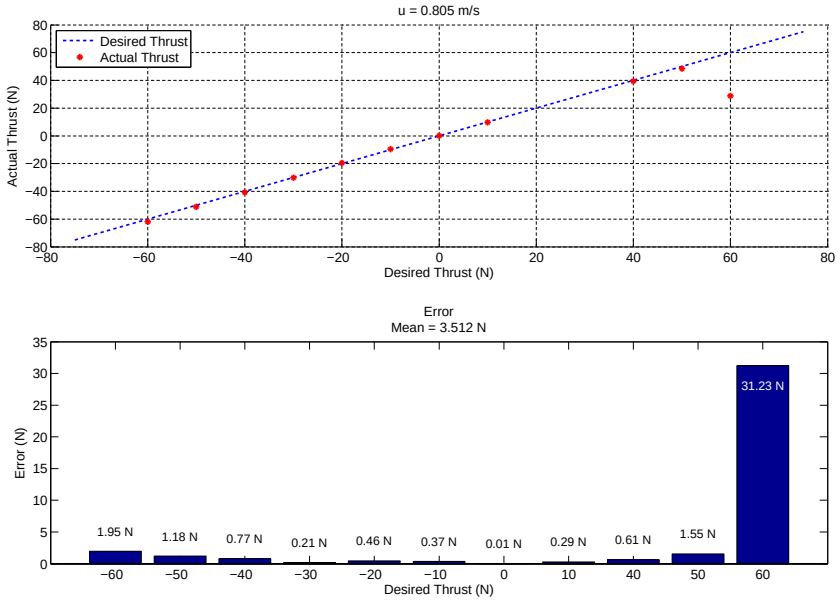

Figure 6.18 – Steady-state thrust and error for ambient water speed $u = 0.805$ m/s.

Chapter 7

Conclusions

The purpose of this work is to use the knowledge from existing literature and apply it to a practical thruster controller for the Saab Seaeye Falcon. Section 7.1 summarizes the findings and its applicability. Section 7.2 describes ways this work may be extended in future projects.

7.1 Summary of Findings

A thruster dynamics model was described by the thrust coefficient and the advance ratio. The thrust force is based on the axial water velocity. This axial water velocity is not known and is difficult to measure; however, it is related to a value that is more readily measured: the ambient water velocity. The thruster dynamics model is built around this relationship and based on models for similar thrusters on other underwater vehicles. This model incorporates three states: the equi-directional, anti-directional and vague-directional states. These states correspond to the direction of axial water velocity with respect to the direction of propeller angular velocity. Each of these three states is described by a 2nd order polynomial.

A sensor frame was built and tests were designed to determine the model coefficients. The model predicted the output thrust force for a given propeller angular velocity and ambient water speed. The experiment also showed the propeller angular velocity is not directly related

to the thruster input command and there is an unknown dependence on the ambient water velocity.

Once the model was fully created and verified, it was inverted for use as a low-level thruster controller. The inverted model provided the desired propeller angular velocity based on the ambient water speed and desired thrust force. A PID controller was used to translate the desired propeller angular velocity into the thruster input command. The PID alone did not scale consistently through the entire range of the thruster input command. The output of the PID is linear and the relationship between the thruster input command and the propeller angular velocity is not. To compensate for the difference, an iso-parametric mapping was created to sit between the PID controller and the thruster input. This mapping took the linear PID output and mapped it to a non-linear curve. Testing of this aspect of the controller followed two steps:

- The first verified the efficacy of the PID controller alone, and showed the need for the iso-parametric mappings.
- The second tested each of four mappings, and compared them to the base PID controller.

The use of an iso-parametric mapping was justified by using these tests. Not all of them improved on the PID results, but one of them did give a more consistent response over the PID controller. The one that gave the most accurate results in the span tested was used in the final controller.

The ability of the controller to generate the desired thrust force in the presence of a known ambient water velocity was also tested. This test considered the thrust force at several ambient water velocities representing the range of the Saab Seaeye Falcon's expected velocity range. The error in the steady-state thrust was documented for each ambient water velocity. The results showed an asymmetry in the model's prediction based on the direction of the propeller angular velocity: the mean error increased for the positive propeller angular velocity. However, the open-loop controller was able to provide near the desired thrust across the range of ambient

water velocities.

The low-level thruster controller described in this work can be implemented on an existing vehicle without costly retrofits for feedback control. This controller provides an accurate thrust force on demand while under a range of ambient water velocities, which is invaluable to an accurate high-level vehicle controller.

7.2 Future Work

The result of this work is a low-level thruster controller for the Saab Seaeye Falcon ROV. There are some limitations in this low-level thruster controller that can be explored in future research. One is in knowing the ambient water velocity. In the test environment this is easy to measure. When the thruster is attached to the ROV, this value must be estimated based on the vehicle velocity and orientation of the thruster. This doesn't take into account unknowns, such as the water speed due to fluid being expelled from other nearby thrusters. There is asymmetry in the controller thrust force that may be further analyzed to determine the root cause and whether it may be reduced. The efficiency of the low-level thruster controller calculations were not considered as part of this research, which is important if it is to be integrated on a ROV. The following other other avenues of research can also be explored.

Ambient Water Velocity Observer One of the limitations of the controller is its reliance on an external measurement of the ambient water velocity. The propeller velocity and thruster current responses will change for a given thruster input command based on the ambient water velocity. It may be possible to use this interdependence to build an observer for the ambient water velocity.

Thrust Force Response The thrust force response to a thruster input command was not considered as part of this work; only the steady state thrust force. The response time may be considered and improved as part of an extension to this controller. This may allow

the thruster to respond faster to environment changes. This change would necessitate including the acceleration term of equation (2.1).

PID Controller Improvements The thruster speed output is very noisy, as shown in Chapter 4. Some compensation was added to the PID controller in equation (6.4) by smoothing the derivative term. This solution helps smooth the response, but as noted there is a cost in the rise time from this smoothing. An alternative method of smoothing out this input or otherwise compensating for the noise in the thruster reported speed measurements may be found.

Controller Options There are other options for controllers. The PID controller was chosen in part for the ease of implementation and testing for the efficacy of creating this model. Other controllers may provide different benefits with different associated costs, and these may be explored in future work.

Thruster Controller Response Time Chapter 4.9 showed a clear difference in response time between retrieving the electrical current and the thruster reported speed; the current request had a consistently lower response time, on the order of 1-1.5 ms. This has an impact on the speed of a low-level thruster controller. If the current can be used in lieu of the thruster reported speed, the low-level thruster controller response time may be improved. However, this may result in reduced accuracy in determining the desired thruster input command. This is something that may be explored as part of integrating the low-level thruster controller on a vehicle.

Vehicle Integration The timing of the controller was not considered as part of this work. The five thrusters on the vehicle are controlled in serial, and the timing is such that there is a limited window in which a controller may operate. The controller must be able to fit within this window. It may also be integrated into a vehicle automated guidance and navigation system, such as the one proposed by Proctor [15].

There may be other avenues to explore in thruster characterization with the sensor frame and the water tanks used as part of this work.

Appendix A

Variables

Variable	Units	Description
Ω	rad/s	Actual propeller angular velocity
Ω_d	rad/s	Desired propeller angular velocity
Ω_{rd}	rad/s	Independent propeller angular velocity
α_i	–	Constants in linear thrust model
γ	–	Thruster speed scaling factor
ζ	%	Thruster input command
η	–	PID Controller Output
θ	rad	Propeller angle
ρ	kg/m^3	Water density
ω	–	Thruster reported speed
A_p	m^2	Area of propeller
D	m	Diameter of propeller
I	A	Thruster current draw

Variable	Units	Description
J_0	–	Advance ratio
J_0^*	–	Critical Advance Ratio (CAR)
K_T	–	Thrust coefficient
T	N	Total measured thrust
T_d	N	Desired thrust set-point
V_i	V	Voltage input to thruster
k_i, k_i'	–	Constants in quadratic thrust model
m_f	kg	Mass of fluid
u	m/s	Ambient water velocity
u_p	m/s	Axial water velocity
v	m/s	Vehicle velocity

Bibliography

[1] R. Bachmayer, L.L. Whitcomb, and M.A..+ + Grosenbaugh. "An accurate four-quadrant nonlinear dynamical model for marine thrusters: theory and experimental validation". In: *Oceanic Engineering, IEEE Journal of* 25.1 (2000), pp. 146–159. ISSN: 0364-9059. DOI: 10.1109/48.820747.

[2] M. Caccia and G. Veruggio. "Guidance and control of a reconfigurable unmanned underwater vehicle". In: *Control Engineering Practice* 8.1 (2000), pp. 21–37. ISSN: 0967-0661. DOI: http://dx.doi.org/10.1016/S0967-0661(99)00125-2. URL: http://www.sciencedirect.com/science/article/pii/S0967066199001252.

[3] Steven E. Cody. "An Experimental Study of the Response of Small Tunnel Thrusters to Triangular and Square Wave Inputs". English. MA thesis. Montery, CA: Naval Postgraduate School, Dec. 1992.

[4] John Gordon Cooke. "Incorporating Thruster Dynamics in the Control of an Underwater Vehicle". MA thesis. Massachusetts Institute of Technology/Woods Hole Oceanographic Institution, 1989.

[5] Ola-Erik Fjellstad, Thor I. Fossen, and Olav Egeland. "Adaptive control of ROVs with actuator dynamics and saturation". In: *Modeling, Identification and Control* 13.3 (1992), pp. 175–188.

[6] T. I. Fossen. *Guidance and Control of Ocean Vehicles*. John Wiley & Sons, Ltd., 1994.

[7] T.I. Fossen and M. Blanke. "Nonlinear output feedback control of underwater vehicle propellers using feedback form estimated axial flow velocity". In: *Oceanic Engineering, IEEE Journal of* 25.2 (2000), pp. 241–255. ISSN: 0364-9059. DOI: 10.1109/48.838987.

[8] A.J. Healey et al. "Toward an improved understanding of thruster dynamics for underwater vehicles". In: *Oceanic Engineering, IEEE Journal of* 20.4 (1995), pp. 354–361. ISSN: 0364-9059. DOI: 10.1109/48.468252.

[9] Jinhyun Kim and Wan Kyun Chung. "Accurate and practical thruster modeling for underwater vehicles". In: *Ocean Engineering* 33.5–6 (2006), pp. 566–586. ISSN: 0029-8018. DOI: http://dx.doi.org/10.1016/j.oceaneng.2005.07.008. URL: http://www.sciencedirect.com/science/article/pii/S0029801805001836.

[10] A. Leonessa and R. Poirrier. "Adaptive control of marine thrusters". In: *OCEANS, 2001. MTS/IEEE Conference and Exhibition*. Vol. 1. 2001, 474–481 vol.1. DOI: 10.1109/OCEANS.2001.968770.

[11] Saab Seaeye Ltd. *Seaeye Falcon Technical Manual*.

[12] Michael B. McLean. "Dynamic Performance of Small Diameter Tunnel Thrusters". MA thesis. Montery, CA: Naval Postgraduate School, Mar. 1991.

[13] K. Muljowidodo et al. "Design and testing of underwater thruster for SHRIMP ROV-ITB". In: *Indian Journal of Marine Sciences* 38.3 (Sept. 2009). doi: pmid: pp. 338–345.

[14] Edin Omerdic and Geoff Roberts. "Thruster fault diagnosis and accommodation for open-frame underwater vehicles". In: *Control Engineering Practice* 12.12 (2004). Guidance and control of underwater vehicles, pp. 1575–1598. ISSN: 0967-0661. DOI: http://dx.doi.org/10.1016/j.conengprac.2003.12.014. URL: http://www.sciencedirect.com/science/article/pii/S0967066103002892.

[15] Alison Proctor. "Semi-Autonomous Guidance and Control of a Saab SeaEye Falcon ROV". PhD thesis. University of Victoria, 2014.

[16] D. Shea et al. "Design and testing of the Marport SQX-500 Twin-Pod AUV". In: *Autonomous Underwater Vehicles (AUV), 2010 IEEE/OES*. 2010, pp. 1–9. DOI: 10.1109/AUV.2010.5779657.

[17] Forum Energy Technologies. *XLX 150 HP*. Forum Energy Technologies. Nov. 2014. URL: http://www.f-e-t.com/our_products_technologies/subsea-solutions/rovs-work-class/triton-xlx-heavy-duty-work-class-rov.

[18] L.L. Whitcomb and D.R. Yoerger. "Preliminary experiments in model-based thruster control for underwater vehicle positioning". In: *Oceanic Engineering, IEEE Journal of* 24.4 (1999), pp. 495–506. ISSN: 0364-9059. DOI: 10.1109/48.809273.

[19] D.R. Yoerger, J.G. Cooke, and J.-J.E. Slotine. "The influence of thruster dynamics on underwater vehicle behavior and their incorporation into control system design". In: *Oceanic Engineering, IEEE Journal of* 15.3 (1990), pp. 167–178. ISSN: 0364-9059. DOI: 10.1109/48.107145.