An Investigation into Advanced Predictive Control Methodologies for the REMUS AUV

by

Brad Chevarie

Bachelor of Science in Mechanical Engineering, Mechatronics Option, UNB, 2017

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master's of Science in Engineering

In the Graduate Academic Unit of Mechanical Engineering

Supervisor(s):	Rickey Dubay, Ph.D., Mechanical Engineering
Examining Board:	Andy Simoneau, Ph.D., Mechanical Engineering (Chair)
	Tiger Jeans, Ph.D., Mechanical Engineering
	Clodualdo Aranas, Ph.D., Mechanical Engineering
	Ian Church, Ph.D., Geodesy and Geomatics Engineering

This thesis is accepted by the
Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

June, 2022

© Brad Chevarie, 2022

Abstract

The health of the ocean ecosystem is vitally important to humans and the planet. In order to maintain the health of these environments, and reduce the human impact upon them, efforts must be made to monitor them. A direct method for monitoring ocean environments is the autonomous underwater vehicle (AUV). A well verified AUV model (the REMUS AUV) is chosen as the platform for this thesis. This vehicle model represents a highly nonlinear, underactuated, highly coupled system in three spatial dimensions. The three dimensional path-following of this system is addressed in this work.

This thesis examines the effectiveness of various advanced predictive control algorithms in controlling simplified systems exhibiting similar properties to those listed above, to be extended to the AUV. The full nonlinear model of the AUV is run through various simulations and tuning is discussed. A simplified Line of Sight (LOS) waypoint-based guidance system is presented, along with an advanced virtual vehicle guidance system, controlled in two stages. The error model for the virtual vehicle is shown, along with the derivation of the first stage control method (constrained model-based predictive control). The second stage control is then developed using the simplified nonlinear model-based predictive controller. A unique hybrid guidance system is introduced, combining these two guidance methods. Simulations on the hybrid guidance system demonstrate the good path-following capability of both control systems and the guidance systems introduced.

Table of Contents

Abstract **ii**

Table of Contents **iii**

List of Figures **vi**

1 Background **1**

1.1 Introduction 1

1.2 Literature Review 4

1.2.1 Linear and Nonlinear Dynamics 4

1.3 Control Methodologies 5

1.3.1 Classical Control - PID 6

1.3.2 Predictive Control 7

1.3.3 Adaptive Control 8

1.4 Autonomous Underwater Vehicles 10

2 AUV System Modeling **12**

2.1 Mathematical Modeling of an AUV 12

2.1.1 Environmental Assumptions 13

2.1.2 Vehicle Assumptions 13

2.1.3 Reference Frames 14

2.1.4 Dynamics 16

3 Controller Theory **21**

3.1 Dynamic Matrix Control 21

3.2 Nonlinear Model Predictive Control 23

3.2.1 Simplified Nonlinear MPC 24

3.3 Complex Profile Tracking 27

4 Preliminary Simulations **30**

4.1 Single Inverted Pendulum 31

4.2 Complex Profile Tracking 35

4.3 AUV Simulation 39

5 AUV Path Following **48**

5.1 Path Following Strategies 48

5.1.1 Line of Sight 48

5.1.2 Virtual Target Vehicle 50

5.1.3 Hybrid Guidance 52

5.2 AUV Control 53

5.2.1 Virtual Vehicle Control 54

5.2.2 LOS Control 60

6 Simulations **62**

6.1 LOS Guidance Simulations 62

6.2 Virtual Vehicle Guidance Simulation 66

6.3 Hybrid Guidance Simulation 68

7 Conclusions **71**

7.1 Summary 71

7.2 Contributions 72

7.3 Future Work 73

7.3.1 LOS Guidance Improvement 73
7.3.2 Controller Robustness and Model Flexibility 73
7.3.3 Experimental Validation 74

Bibliography **80**

A Additional AUV Plots **81**

A.1 LOS Guidance (Single Waypoint) 81
A.2 LOS Guidance (Waypoint Planar Curve) 83
A.3 LOS Guidance (Waypoint Spiral) 84
A.4 Virtual Vehicle Guidance 86
A.5 Hybrid Guidance 87

Curriculum Vitae

List of Figures

1.1 PID Control Loop 6

1.2 Traditional Controller Design [19] 9

1.3 Adaptive Controller Design [19] 9

2.1 REMUS 100 AUV 13

2.2 Earth-Fixed and Body-Fixed Coordinate Frames 15

3.1 nMPC Block Diagram [15] 25

4.1 Single Inverted Pendulum Diagram on a Cart [35] 31

4.2 LQR Control of SIP (25° Initial Angle) 33

4.3 SnMPC Control of SIP (25° Initial Angle) 34

4.4 Sinusoidal Input Comparison 35

4.5 Sinusoidal Input Comparison with η-correction 36

4.6 Step Input Comparison 37

4.7 System Simulation for Varied N_u 38

4.8 AUV Trajectory (Maximum Rudder Fin Angle) 40

4.9 AUV Linear Displacements and Velocities (Maximum Rudder Fin Angle) 40

4.10 AUV Angular Displacements and Velocities (Maximum Rudder Fin Angle) 41

4.11 AUV Forces and Moments (Maximum Rudder Fin Angle) 41

4.12 AUV Trajectory (Oscillating Input) 42

4.13 AUV Linear Displacements and Velocities (Oscillating Input) 43

4.14 AUV Angular Displacements and Velocities (Oscillating Input) 43
4.15 AUV Forces and Moments (Oscillating Input) 44
4.16 AUV Trajectory (Pitch Down) 45
4.17 AUV Linear Displacements and Velocities (Pitch Down) 45
4.18 AUV Angular Displacements and Velocities (Pitch Down) 46
4.19 AUV Forces and Moments (Pitch Down) 46

6.1 AUV Trajectory (Single Waypoint) 63
6.2 AUV Control Actions (Single Waypoint) 63
6.3 AUV Trajectory (Waypoint Planar Curve) 64
6.4 AUV Control Actions (Waypoint Planar Curve) 64
6.5 AUV Trajectory (Waypoint 3D Curve) 65
6.6 AUV Control Actions (Waypoint 3D Curve) 66
6.7 AUV Trajectory (Virtual Vehicle) 67
6.8 AUV Control Actions (Virtual Vehicle) 67
6.9 AUV Trajectory (Hybrid Guidance) 69
6.10 AUV Control Actions (Hybrid Guidance) 69

A.1 AUV Linear Displacements and Velocities (Single Waypoint) 81
A.2 AUV Angular Displacements and Velocities (Single Waypoint) 82
A.3 AUV Forces and Moments (Single Waypoint) 82
A.4 AUV Linear Displacements and Velocities (Waypoint Planar Curve) . 83
A.5 AUV Angular Displacements and Velocities (Waypoint Planar Curve) 83
A.6 AUV Forces and Moments (Waypoint Planar Curve) 84
A.7 AUV Linear Displacements and Velocities (Waypoint 3D Curve) . . . 84
A.8 AUV Angular Displacements and Velocities (Waypoint 3D Curve) . . 85
A.9 AUV Forces and Moments (Waypoint 3D Curve) 85
A.10 AUV Linear Displacements and Velocities (Virtual Vehicle) 86

A.11 AUV Angular Displacements and Velocities (Virtual Vehicle) 86
A.12 AUV Forces and Moments (Virtual Vehicle) 87
A.13 AUV Linear Displacements and Velocities (Hybrid Guidance) 87
A.14 AUV Angular Displacements and Velocities (Hybrid Guidance) 88
A.15 AUV Forces and Moments (Hybrid Guidance) 88

Chapter 1

Background

1.1 Introduction

The health of ocean ecosystems is of vital importance to the overall health of the planet. Important factors such as algae growth, oil spills, marine wildlife populations, temperatures, and many others are critical to the future of these ecosystems [1]. One measure for the state of global oceans gives an index score (this index is comprised of ten public goals ranging from food provision, to biodiversity, to carbon storage and clean water) of the ocean as an exclusive economic zone of 60 out of 100. These index scores varied greatly by country, with some as low as 36 out 100, while highly developed countries scored as high as 86 out of 100 [1].

With approximately 60% of the global population living in coastal areas, and about two billion people relying on seafood as a source of nutrition, it is also clear that the health of ocean ecosystems is directly related to human health and development [2]. In recent years there has been an increased focus placed on monitoring these ecosystems, for both sustainability and for commercial development. The data collected for analyzing ocean health often excludes the deep water oceans, as these environments are very difficult to investigate through conventional means. It was famously stated

by Dr. Paul Snelgrove that "We know more about the surface of the Moon and about Mars than we do about the deep sea floor, despite the fact that we have yet to extract a gram of food, a breath of oxygen or a drop of water from those bodies" [3]. This perfectly illustrates the need to better monitor oceans and other aquatic ecosystems. Of the different techniques to accomplish this, satellites [4], acoustic monitoring [5], and seemingly increasingly autonomous underwater vehicles (AUVs) [6].

The case for the use of autonomous underwater vehicles in various applications has grown much stronger in the last several decades with the explosion of cost-efficient computing. With the ability to carry sensor payloads on-board, these vehicles are able to better maneuver than traditional towed sensor arrays, and with the added advantage of the avoidance of large signal noise from the surface [7]. These systems tend to operate within a horizontal plane, utilizing angled cameras for visual data, along with more specialized sensors for other applications. This suggests that the capability, application, and functionality of these vehicles has not yet reached its full potential [7].

The research sector has seen large scale funding of autonomous underwater vehicles of different varieties and for various applications. In 2007, NASA and the NSF funded an arctic expedition to investigate and explore the previously inaccessible Gakkel Ridge, in an attempt to study an isolated arctic environment. The success of this mission was attributed to AUV technology [8] and several new microbial species were discovered along with volcanic anomalies. Another interesting research development project funded by NASA and executed by Stone Aerospace is the DEPTHX AUV. This AUV was built with the intention of creating a robot to explore the underwater environment of Jupiter's moon Europa. This robot was successful in various autonomous missions including the use of a robotic manipulator to collect microbiological samples in 2007 [9].

Autonomous underwater vehicles have also seen adoption by the military. Aside from

the early motivation to utilize these platforms for oceanographic mapping [7], the military has found use for AUVs in mine-countermeasure operations. Citing their particular usefulness in shallow water operations, the military employed the REMUS 100 AUV for mine clearance operations in Umm Qasr Harbor, calling the AUV system "invaluable" [10]. The U.S. Navy has gone so far as to lay out a fifty year plan for the future development of unmanned underwater vehicles [7].

The ocean environment is inherently very complex, due to uncertainty of models, highly nonlinear dynamics, and disturbances which are very difficult to estimate [11]. This demands a control scheme that is very robust and highly stable to be able to handle intense hydrodynamic applications [12]. Traditional control schemes typically struggle with these challenges, and thus new methods and techniques must be applied. Control systems have increased in complexity with the increasing availability of computational power. This has been fueled by an increase in the demand for such systems, as automation is becoming more and more prevalent in the world. These factors explain the growing demand for control schemes which can be applied to highly nonlinear and under-actuated systems, as well as be adaptive to changing dynamics. This makes the control of an AUV an excellent candidate for the application of advanced methods, given that the vast majority of AUVs are under-actuated robotic-type systems, and that their operating environment is dynamically very complex.

This research is performed with funding from the NSERC Interdisciplinary Marine Engineering Research and Industrial Training (iMERIT) program based at Dalhousie University in Halifax, Nova Scotia and in conjunction with researchers and students at several Atlantic Canadian universities. A focus of this program is on various aspects of AUV design and ocean monitoring systems, including improvements to control systems and control theory of AUV systems.

1.2 Literature Review

1.2.1 Linear and Nonlinear Dynamics

When discussing dynamical systems in relation to control engineering of robotics, it is usually most common to find a continuous time representation in the form of differential equations of motion. In the case of AUVs, as the primary concern is temporal behavior of the system, these equations of motion will take the form of ordinary differential equations. These can be either linear or nonlinear, and the distinction is an extremely important one.

Traditional control methods assume linear system models. In the case of real systems, which are all nonlinear to some degree, this assumption can be satisfied by linearizing about a point, or assuming linearity in a specified operating range. Linear equations can often be solved analytically and are thus considered much simpler than their nonlinear counterparts, an example being a traditional damped oscillator:

$$m\frac{d^2x}{dt^2} + b\frac{dx}{dt} + kx = u(t) \tag{1.1}$$

This allows the equations of motions to be packaged neatly as a linear combination, in the following form:

$$\dot{x} = \mathbf{A}x + \mathbf{B}u \tag{1.2}$$

Where x and $\dot{x}$ are system states and derivatives thereof, and $\mathbf{A}$ and $\mathbf{B}$ is the system matrix. The terms in $\mathbf{A}$ are all first order and involve no products, powers, or functions of states.

By contrast, a nonlinear system will by nature involve nonlinear terms (x_1x_2, x^2, $\sin x$). What this causes is a system where the change in output will not be proportional to a change in input. A test for this is whether the system satisfies the

superposition principle, where the system exhibits the properties of homogeneity and additivity. An example of this is an unforced swinging pendulum (without friction), whose nonlinearity can be observed in the $\sin x$ term:

$$\ddot{x} + \frac{g}{L} \sin x = 0 \tag{1.3}$$

Nonlinear systems will generally not display simple behavior, and analytical solutions are usually out of reach. Some examples are pendulums, robotic manipulators, or marine robots. When improperly controlled these systems can become unstable, they can be dangerous to both operators and the systems themselves. This highlights the need for effective control schemes.

1.3 Control Methodologies

Control methods for robotic systems necessitate both high precision and guaranteed stability, as failure of either can be disastrous. High complexity systems, such as an AUV, may also require a fast computational speed to remain adaptive in the actual physical environment. Another important factor to consider is level of actuation of the system.

Put simply, a system is underactuated if for every degree of freedom of the system, there does not exist an actuator to produce a desired response [13]. Defined mathematically using the general form for a second-order control dynamical system:

$$\ddot{\mathbf{q}} = \mathbf{f}_1(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{f}_2(\mathbf{q}, \dot{\mathbf{q}}, t)u \tag{1.4}$$

where $\mathbf{q}$ is a vector of positions, $\dot{\mathbf{q}}$ is a vector of corresponding velocities, and $\mathbf{u}$ is a vector of inputs. A control system is underactuated if:

$$\text{rank}[\mathbf{f}_2(\mathbf{q}, \dot{\mathbf{q}}, t)] < \dim[\mathbf{q}] \tag{1.5}$$

This displays mathematically the above statement that a system is underactuated if the number of control inputs (rank) of the system is less than the degrees of freedom ($\dim[\mathbf{q}]$).

This section will discuss very briefly several control methodologies from traditional to advanced.

1.3.1 Classical Control - PID

Usually one of the first control schemes learned, and popular on a vast majority of systems is the classical Proportional-Integral-Derivative (PID) controller, at once simple to design and apply. PID control architecture is very intuitive in the how it seeks to reduce the error signal in a closed loop system.

This scheme uses a proportional gain (K_p), an integral gain (K_i), and a derivative gain (K_d) to reduce the error in the output, error being the difference between the output and the set point. This is seen below in Laplace form in Figure 1.1

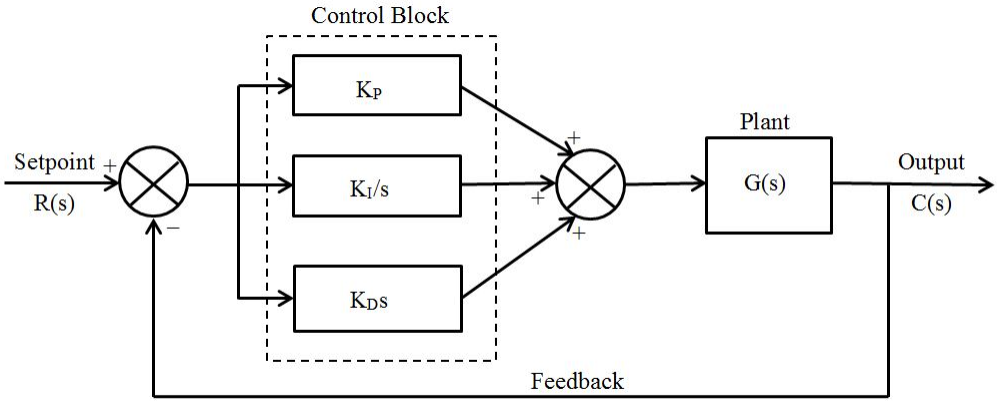

Figure 1.1: PID Control Loop

This diagram results in the following equation describing the control action to the system:

$$u(t) = K_p e(t) + K_i \int_{t_0}^{t} e(t)dt + K_d \frac{de(t)}{dt} \tag{1.6}$$

Where the first term is the proportional term, proportional to the error and thus scaling with it. The second term represents an integrator, which sums the error terms from $t = t_0$ up to the current time t, resulting in a consistent control action that will reduce steady-state error. The third term is the derivative term, essentially a measure of the slope of the error curve, helps with the transient error.

While this is a basic control method, it can be quite powerful. This control method is widely used in industrial applications such as pulp and paper, chemicals, and mining, accounting for as much as 97% of the controllers used in these industries as of 2000 [14]. While this number has likely dropped some, there is no denying it is an incredibly useful and simple control scheme.

For more advanced and nonlinear applications, simple PID control on its own will not suffice, however there are many examples of PID controllers being integrated into more advanced controllers. Wilson et al. shows a PID Nonlinear Adaptive Model Predictive Controller (PIDnMPC) for the control of a two link robot [15], Khodayari and Balochian implemented a self adaptive fuzzy PID controller on an AUV [16], and Zhao et al. used an optimized fuzzy PID for trajectory tracking of a flexible joint robot arm [17].

1.3.2 Predictive Control

Predictive control algorithms, mainly Model-based Predictive Control (MPC) are a classification of control algorithms that attempt to reduce error in the output by leveraging a mathematical model to make predictions about the system behavior. The first form of this controller was conceived in the late 1970s, and is more specifically called Dynamic Matrix Control (DMC). This utilizes a discrete model of the system

to function as follows:

- A prediction horizon N is used to create a dynamic matrix $\mathbf{A}$ which captures the dynamics of the system.
- A vector of predicted outputs $\hat{y}$ is built using the dynamic matrix and previous change in control move Δu.
- The minimization of an objective function J is used to optimize Δu to reduce the future error, defined as the difference between future set point and predicted plant output.
- Δu is added to control action u and sent to the plant.
- The measured plant output y_m is sent back to the controller to use as a model correction factor ϕ between the predicted and measured outputs.

This will be explored more deeply in Chapter 3, however the basis of this method rests on the minimization of J in order to determine control moves u to minimize future errors over a horizon. This is in contrast to a more traditional control scheme where the controller is attempting to minimize the error of previous and current time steps.

These controllers are seeing a large adoption in both industrial applications as well as in academia, with applications in robotic manipulators, industrial control processes, unmanned aerial vehicles, and many others [18].

1.3.3 Adaptive Control

Adaptive control is a fast growing area of control theory that relies on the ability of the controller to adjust itself in real time based on the system conditions. This typically requires high computational effort, and as such the widespread implementation of such

control methods is relatively new. The ability to adapt to changing system conditions is clearly very desirable among many applications such as aviation, robotics, AUVs, and industrial processes among others. Most commonly, adaptive control theory is used to augment other control strategies.

Adaptive controllers are used when a system has unknown parameters or parameters that change in time. For example, if a linearization of a system about a certain operating point is considered, a change in operation state could lead to a different linearized model; or the effects of velocity on the dynamics of an AUV. These situations show a system that cannot be tuned precisely for a single operation state, as tuning requirements will change with the operation state, hence the necessity of adaptive control [19]. The diagrams below, Figure 1.2 and Figure 1.3, show the differences in the design/functionality between a more traditional control system and an adaptive one.

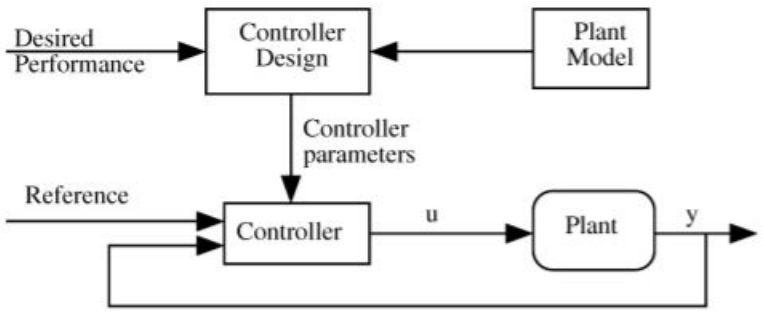

Figure 1.2: Traditional Controller Design [19]

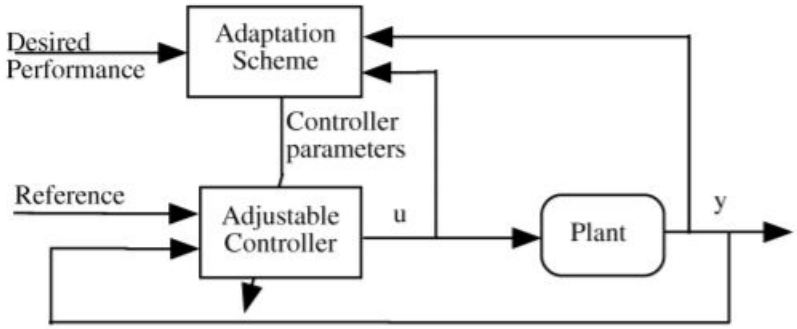

Figure 1.3: Adaptive Controller Design [19]

Figures 1.2 and 1.3 show the difference in operating methodology. The adaptive control design utilizes, on-line, the system output and its own output to adjust its control parameters throughout operation. This is frequently applied to complex robotics and nonlinear systems [19].

1.4 Autonomous Underwater Vehicles

In the area of marine engineering, manned underwater vehicles have been in use in one form or another for hundreds of years, with the first submersible vehicle being developed in 1775 [20]. The advent of modern electronics has brought with it the ability to automate these vehicles, allowing for Unmanned Underwater Vehicles (UUVs). This eliminates the need for a human to be on board the vehicle, and also reduces operational costs, allowing for use at greater depths and in harsher conditions [21]. While their exists a classification of UUVs known as Remotely Operated underwater Vehicles (ROVs), the focus of this research is on the aforementioned robotic systems, AUVs.

The AUV robotic system presents many unique control challenges. Chief among them is motion control for trajectory tracking. While trajectory tracking is difficult for a variety of reasons, it is especially so for AUVs, due to the following factors:

- AUVs generally operate freely in a three dimensional environment, giving them six degrees of freedom (surge, sway, heave, roll, pitch, and yaw).
- AUVs exhibit inherently nonlinear behaviour.
- The underwater environment, often a lake or ocean, is by nature highly unpredictable due to waves and currents, among other things [22].
- The overwhelming majority of AUVs are underactuated systems, without actuation in the heave and sway axis.

While this list is by no means exhaustive, it serves to illustrate that the motion control problem of underactuated AUVs is one of the most challenging in robotics [22].

There is a growing body of research into different control methods for AUVs, mostly focused on trajectory tracking. Until the mid 2000s much of this was performed using planar paths, which greatly simplified the problem by constraining the motion to two-dimensions (2-D). Many control techniques are also employed on systems with a greater number of actuators than the single thruster underactuated AUV design [12]. These control problems have seen some success using Lyapunov-based model predictive control as well as heuristic fuzzy control [12][22]. These are still underactuated, however not to the degree of many commonly available systems, where there is a single propeller in the surge axis, along with moment actuators providing independent control of pitch and yaw [23]. For this configuration, it appears that an output feedback controller based on backstepping technique for velocity control has been successful in up to this point [23].

Chapter 2 will now describe in depth the dynamic model that will be used as a simulation platform for implementing control methodologies.

Chapter 2

AUV System Modeling

2.1 Mathematical Modeling of an AUV

This section will cover the mathematical modeling of an autonomous underwater vehicle. As there are many variations of AUVs available for study ranging from robotic fish to fully actuated [21], it is important to note that the type of AUV being considered is an underactuated system, with a single propeller providing thrust, and sets of pitch and yaw angle fins to orient the AUV. This is a very common classification of AUV, as discussed in Chapter 1. It is also important to note that the focus of this research is neither to derive nor improve new or existing models for AUVs. Instead, the goal is to use a verified mathematical model of an underactuated AUV as a platform for the investigation of control methodologies, with a vision to better enable the monitoring of marine environments for health and discovery and to utilize an AUV as a platform for the testing of control methodologies.

The model chosen for this research is of the REMUS 100 AUV, a small, low-cost, modular underwater vehicle developed at the Woods Hole Oceanographic Institution, seen in Figure 2.1. The full derivation of this mathematical model, along with all vehicle dynamics examined and vehicle coefficients can be seen in [24]. The following

is a description of the equations used in the simulation of the AUV.

Figure 2.1: REMUS 100 AUV

It is important to note of the model in question that several assumptions must be made regarding the derivation and simulation of the model. These are directly from [24], divided into environmental and vehicle assumptions, and are as follows:

2.1.1 Environmental Assumptions

- The vehicle is deeply submerged in a homogeneous, unbounded fluid.
- The vehicle does not experience memory effects.
- The vehicle does not experience underwater currents.

2.1.2 Vehicle Assumptions

- The vehicle is a rigid body of constant mass.
- Control fins do not stall regardless of angle
- Instantaneous fin response.

- Simplified propulsion model, where the propeller is a source of constant thrust and torque.

With these items in mind, the mathematical model for an underactuated AUV can be developed.

2.1.3 Reference Frames

To begin, it is important to understand the relationship between several reference frames that will be used for modeling. The first is the earth-fixed (or inertial) reference frame and the second is the body-fixed (or AUV) reference frame. The following are used to describe the position:

$$\begin{gathered}\mathbf{p} = [x, y, z]^T \\ \Theta = [\phi, \theta, \psi]^T \\ \eta = [\mathbf{p}, \Theta]^T\end{gathered}$$

where $\mathbf{p}$ is the position of the AUV in the earth-fixed frame, Θ represents the orientation of the AUV relative to the earth-fixed frame in which ϕ, θ, and ψ are the roll, pitch, and yaw Euler angles, respectively. Combining these yields η, the complete position of the AUV relative to the earth-fixed frame. The velocities of the AUV are described by:

$$\begin{gathered}\mathbf{v} = [u, v, w]^T \\ \omega = [p, q, r]^T \\ \nu = [\mathbf{v}, \omega]^T\end{gathered}$$

where $\mathbf{v}$ is the body-fixed translational velocity vector, comprised of surge u, sway v, and heave w, along the body-fixed x, y, and z axes. The vector ω contains the body-fixed angular velocities p, q, and r, rotating about the surge, sway, and heave axes respectively. These vectors are combined into ν to represent the complete velocities in the body-fixed frame. This demonstrates the six degrees of freedom of the AUV.

A diagram of these coordinate frames is depicted in Figure 2.2 (This figure is edited, the original can be seen in [25]).

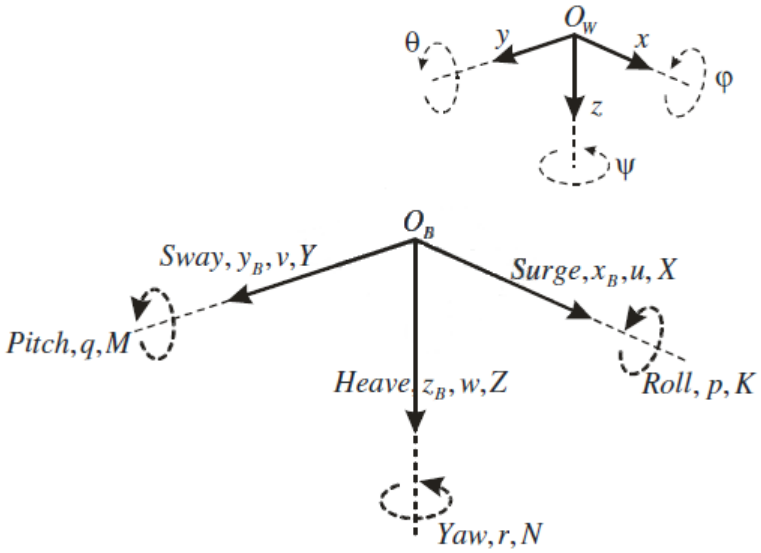

Figure 2.2: Earth-Fixed and Body-Fixed Coordinate Frames

It is important to note that in this figure, the ϕ, θ, and ψ angles depicted on the O_W reference frame are only accurate in isolation. As the orientation of the AUV is described using an Euler angle representation, following any rotations made this diagram is no longer accurate.

Since the translation and angular velocities are expressed with respect to the body-fixed reference frame, they must be transformed to the earth-fixed reference frame to be used. The following equation is used to perform this transformation:

$$\dot{\eta} = J(\eta)\nu \tag{2.1}$$

where $\dot{\eta}$ contains the translation and angular velocities of the AUV in the earth-fixed reference frame and $J(\eta)$ is a diagonal matrix of transformation matrices as follows:

$$J(\eta) = \begin{bmatrix} J_1(\eta) & 0_{33} \\ 0_{33} & J_2(\eta) \end{bmatrix} \tag{2.2}$$

the diagonal components of which are represented by the following equations:

$$J_1(\eta) = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \tag{2.3}$$

$$J_2(\eta) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \tag{2.4}$$

Euler angle representation of the orientation of a rigid body with respect to an earth-fixed reference frame using the xyz convention through the Euler angles ϕ, θ, and ψ (the rotation sequence being ψ, θ, ϕ) is shown by the succession of rotations $J_1(\eta)$. These Euler angles also yield $J_2(\eta)$, used to transform the angular velocities from the body-fixed to earth-fixed reference frame. It should be noted that s, c, and t represent the sin, cos, and tan of the denoted angle, respectively, and that 0_{33} is a zero matrix of size 3x3. The calculation of these is shown in [25].

This parameterisation is part of a unified framework for the nonlinear modeling of marine vehicles in six degrees of freedom, presented by Fossen and Fjellstad, 1994 [26]. With this kinematic representation of the AUV, the dynamic equations can now be considered.

2.1.4 Dynamics

The dynamics of a six degree of freedom body in fluid have been studied extensively, and formulations specific to this type of AUV have been presented many times [24][26][25][27]. The rigid-body dynamics, the inertial forces, are developed first in [24]. To begin, the vehicle centers of buoyancy and gravity with respect to the body-fixed reference frame are denoted as:

$$\mathbf{r}_B = [x_b, y_b, z_b]^T; \quad \mathbf{r}_G = [x_g, y_g, z_g]^T$$

respectively. The formulation of the equations of motions for this vehicle assume the body-fixed reference frame to be coincident with the center of buoyancy $\mathbf{r}_B$ of the vehicle. It is assumed that under typical ballasting conditions for this specific AUV simulation that y_g will be negligible, in order to simplify the rigid-body dynamic equations of the vehicle.

In order to further simplify the vehicle equations of motion with negligible effects in simulation accuracy, it is assumed that the vehicle has two planes of symmetry. This assumes that the products of inertia are zero, leaving the inertia tensor at the vehicle origin:

$$\mathbf{I}_o = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{2.5}$$

The values for these can be found in [24]. The following are the rigid-body dynamic equations after the simplifications described:

$$\begin{aligned} m[\dot{u} - vr + wq - x_g(q^2 + r^2) + z_g(pr + \dot{q})] &= \sum X_{ext} \\ m[\dot{v} - wp + ur + z_g(qr - \dot{p}) + x_g(qp + \dot{r})] &= \sum Y_{ext} \\ m[\dot{w} - uq + vp - z_g(p^2 + q^2) + x_g(rp - \dot{q})] &= \sum Z_{ext} \\ I_{xx}\dot{p} + (I_{zz} - I_{yy})qr + m[-z_g(\dot{v} - wp + ur)] &= \sum K_{ext} \\ I_{yy}\dot{q} + (I_{xx} - I_{zz})rp + m[z_g(\dot{u} - vr + wq) - x_g(\dot{w} - uq + vp)] &= \sum M_{ext} \\ I_{zz}\dot{r} + (I_{yy} - I_{xx})pq + m[x_g(\dot{v} - wp + ur)] &= \sum N_{ext} \end{aligned} \tag{2.6}$$

The vehicle mechanics represent the external forces and moments on the vehicle causing motion, shown in the following equation:

$$\sum F_{ext} = F_{hydrostatic} + F_{hydrodynamic} + F_{control} \tag{2.7}$$

where $\sum F_{ext}$ is the sum of external forces shown in equation 2.6, while the other terms are the hydrostatic, hydrodynamic due to motion of the AUV (including lift, drag, and added mass), and control forces, respectively.

Hydrostatic forces and moments are the result of the weight and buoyancy of the vehicle as the orientation changes, and are summarised as follows:

$$\begin{aligned}
X_{HS} &= -(W - B)\sin\theta \\
Y_{HS} &= (W - B)\cos\theta\sin\phi \\
Z_{HS} &= (W - B)\cos\theta\cos\phi \\
K_{HS} &= (y_g W - y_b B)\cos\theta\cos\phi - (z_g W - z_b B)\cos\theta\sin\phi \\
M_{HS} &= -(z_g W - z_b B)\sin\theta - (x_g W - x_b B)\cos\theta\cos\phi \\
N_{HS} &= (x_g W - x_b B)\cos\theta\sin\phi + (y_g W - y_b B)\sin\theta
\end{aligned} \tag{2.8}$$

Where W is weight of the AUV in newtons, and B is the force of buoyancy in newtons. The AUV is typically ballasted to achieve positive buoyancy, as is the case in this simulation.

The derivation for the other external forces acting on the AUV can be seen in [24], a brief description of each will be given here. The body lift force is the result of the AUV angle of attack while moving through water. This is modeled as a force applied at the center of pressure of the AUV, which is not coincident with the center of buoyancy, resulting in an additional moment on the vehicle. The drag force represents all combined sources of drag acting on the body of the vehicle. These forces are nonlinear and coupled, creating complexity. There are several assumptions made to simplify the modeling of the drag that are used in this formulation that are not shown here. The added mass forces are the forces that arise due to the added mass phenomenon observed when moving a rigid body in a viscous fluid. When a rigid body

is accelerated in fluid, the surrounding fluid will also be accelerated. This increase in inertia can be seen as an apparent increase in mass, thus the term "added mass". Finally, the control forces $F_{control}$ are the forces on the AUV due to the control fins (rudder and pitch fins) and propeller, creating both an axial thrust and a rolling torque.

These represent the external forces and moments on the AUV that will be present in the combined equations of motion, to be used in the simulation of the vehicle. The final form of the combined nonlinear external forces is presented as follows:

$$\begin{aligned}
\sum X_{ext} = \;& X_{HS} + X_{u|u|}u|u| + X_{\dot{u}}\dot{u} + X_{wq}wq + X_{qq}qq + X_{vr}vr + X_{rr}rr \\
& + X_{prop} \\
\sum Y_{ext} = \;& Y_{HS} + Y_{v|v|}v|v| + Y_{r|r|}r|r| + Y_{\dot{v}}\dot{v} + Y_{\dot{r}}\dot{r} + Y_{ur}ur + Y_{wp}wp \\
& + Y_{pq}pq + Y_{uv}uv + Y_{uu\delta_r}uu\delta_r \\
\sum Z_{ext} = \;& Z_{HS} + Z_{w|w|}w|w| + Z_{q|q|}q|q| + Z_{\dot{w}}\dot{w} + Z_{\dot{q}}\dot{q} + Z_{uq}uq + Z_{vp}vp \\
& + Z_{rp}rp + Z_{uw}uw + Z_{uu\delta_s}uu\delta_s \qquad (2.9) \\
\sum K_{ext} = \;& K_{HS} + K_{p|p|}p|p| + K_{\dot{p}}\dot{p} + K_{prop} \\
\sum M_{ext} = \;& M_{HS} + M_{w|w|}w|w| + M_{q|q|}q|q| + M_{\dot{w}}\dot{w} + M_{\dot{q}}\dot{q} + M_{uq}uq \\
& + M_{vp}vp + M_{rp}rp + M_{uw}uw + M_{uu\delta_s}uu\delta_s \\
\sum N_{ext} = \;& N_{HS} + N_{v|v|}v|v| + N_{r|r|}r|r| + N_{\dot{v}}\dot{v} + N_{\dot{r}}\dot{r} + N_{ur}ur + N_{wp}wp \\
& + N_{pq}pq + N_{uv}uv + N_{uu\delta_r}uu\delta_r
\end{aligned}$$

where the control inputs are δ_r and δ_s, which are the angles in radians for the rudder and pitch fins, respectively. This model was designed with the assumption that the AUV propeller is held to a constant 1500 rpm, thus X_{prop} is a constant and not a control input. The descriptions, derivations, and values of all coefficients are found in [24]. In order to simulate the motion of the AUV, the accelerations in the body-fixed frame must be solvable. With both sides of the dynamic model developed in equation

2.6 and equation 2.9, the accelerations are isolated and the resulting equations are collected into a matrix representation:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 & mz_g & -my_g \\ 0 & m - Y_{\dot{v}} & 0 & -mz_g & 0 & mx_g - Y_{\dot{r}} \\ 0 & 0 & m - Z_{\dot{w}} & my_g & -mx_g - Z_{\dot{q}} & 0 \\ 0 & -mz_g & my_g & I_{xx} - K_{\dot{p}} & 0 & 0 \\ mz_g & 0 & -mx_g - M_{\dot{w}} & 0 & I_{yy} - M_{\dot{q}} & 0 \\ -my_g & mx_g - N_{\dot{v}} & 0 & 0 & 0 & I_{zz} - N_{\dot{r}} \end{bmatrix}^{-1} \begin{bmatrix} \sum X \\ \sum Y \\ \sum Z \\ \sum K \\ \sum M \\ \sum N \end{bmatrix} \tag{2.10}$$

With this final more compact form, along with the kinematic transformations described in equation 2.1, the full model of the AUV is represented. This allows for the calculation of the body-fixed accelerations in all six degrees of freedom along with the translational and rotational velocities relative to the earth-fixed reference frame, given the body-fixed translational and rotational velocities and the AUV orientation. With the calculated accelerations and velocities, a numerical integration method can be used (in this case a fourth-order Runge-Kutta approximation) to simulate AUV position, orientation, and velocities through time. This model was compared with experimental data for the REMUS 100 AUV in [24], and the open-loop tests are reattempted and shown in Chapter 4 to verify the capabilities of the model and to attempt to determine appropriate parameters and initial conditions for subsequent simulations.

The following section describes the control schemes to be used.

Chapter 3

Controller Theory

This chapter provides an in-depth explanation of the relevant control theory introduced in the first chapter. The focus of this chapter will be on model-based predictive control, in its basic form, as well as augmentations made in order to handle the various demands of more complex systems. The AUV system has a number of complexities (underactuation and nonlinearity chief among them), meaning enhanced control tools are required for achieving accurate control.

3.1 Dynamic Matrix Control

As previously discussed, MPC strategies are a classification of advanced control algorithm which use a predicted future error as a means to achieve the desired system output, as opposed to traditional feedback control which uses past and current signal error. Some of the advantages of predictive control methods are that they are easily extensible from SISO to MIMO systems (dynamic matrices simply grow in size), constraints on operation are conceptually simple to handle during controller design, there is intrinsic compensation for plant dead time, simple handing of non-minimum phase systems, and the methodology itself is based on basic principles that allow for extension [18].

To formalize the explanation given in Chapter 1, the formulation of the calculation of the control moves will be given for a dynamic matrix controller, developed during in the 1970s at Shell Oil [18], which serves as an excellent theoretical base upon which more advanced algorithms are built.

The DMC algorithm uses the minimization of an objective function to optimize control moves. This is performed using a vector of predicted outputs over a prediction horizon, built from a matrix that captures the system dynamics. The error signal in this case is the difference between the predicted plant outputs and the reference trajectory over the prediction horizon. It is important to note that this formulation uses a discrete-time system model, and is based on a linear system. The dynamic matrix $\mathbf{A}$ is constructed from the normalized response to an open-loop step input or impulse to the system or its model:

$$\mathbf{A} = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ a_2 & a_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_N & a_{N-1} & \cdots & a_{N-n_u} \end{bmatrix} \tag{3.1}$$

The response vector A is stored in n_u (the control horizon) down-shifted columns of N (the prediction horizon) length. This results in a matrix of size N by n_u. For a linear plant this matrix is constant and is calculated outside of the control loop, for nonlinear systems this will not be the case. From this a vector of predicted plant outputs of length N for time t is constructed:

$$\hat{\mathbf{y}}|_t = \hat{\mathbf{y}}|_{t-1} + \mathbf{A}\Delta\mathbf{u} + \phi|_t \tag{3.2}$$

where $\hat{\mathbf{y}}|_{t-1}$ is the predicted output vector from the previous time step $(t-1)$, $\Delta\mathbf{u}$ is a vector of changes in control moves, of length n_u, and $\phi|_t$ is a model mismatch correction value, calculated through the equation:

$$\phi|_t = y_m - \hat{\mathbf{y}}_1|_{t-1} \tag{3.3}$$

in which y_m is the measured output from the plant from the previous time step. The next step is to calculate a new $\Delta\mathbf{u}$. This is accomplished using the following objective function, which is a simplification of a more general objective function [18]:

$$J = \sum_{j=1}^{N} [\hat{\mathbf{y}}(t+j \mid t) - \hat{\mathbf{r}}(t+j)]^2 + \sum_{j=1}^{n_u} \lambda(j)[\Delta\mathbf{u}(t+j-1)]^2 \tag{3.4}$$

where $\hat{\mathbf{r}}$ is the reference trajectory, and $\lambda(j)$ is often a constant and represents the move suppression factor, chosen to weight the future behavior of the control moves. In order to solve for $\Delta\mathbf{u}$, for most linear systems, the solution can be formed analytically through the minimization of J, by setting its derivative to zero and solving for $\Delta\mathbf{u}$, yielding the following:

$$\Delta\mathbf{u} = (\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{A}^T\mathbf{E} \tag{3.5}$$

in which $\mathbf{I}$ is an identity matrix of size n_u by n_u, and $\mathbf{E}$ represents the future error $\hat{\mathbf{y}} - \hat{\mathbf{r}}$. From this vector, only the first value is actually sent to the plant (subsequent control values are used to form the prediction vector).

3.2 Nonlinear Model Predictive Control

The early model-based predictive control strategy, dynamic matrix control, and its later evolution the generalized predictive control strategy, utilize linear models for their formulations. Real systems are nonlinear, and although many can be approximated with linear models or operated in an approximately linear range, for certain systems (such as the autonomous underwater vehicle system that is being explored in this research), such methods will not suffice. This necessitates augmentations of

the control strategies in order to handle, by various means, the nonlinearities and complicated dynamics of these systems.

There are countless ways to approach this, relying on varying mathematical and computational complexities. Since this research will focus on studying and seeking a model-predictive approach to the control of an AUV system, in depth formulations for several of these methods will be shown here.

In Chapter 2 the mathematical model for a general underactuated AUV system is given. This model has six degrees of freedom and displays nonlinear dynamics. In order to control this using a predictive approach, the prediction process must be able to handle nonlinearity. This can be handled in several ways; the first is to linearize about various operating points throughout the operation of the system, and to generate the necessary dynamic matrices inside of the control loop (or if operating conditions are known, multiple times prior to entering the control loop). The number of points to linearize about will increase with higher nonlinearity in the system. The second method involves computing a nonlinear model of the plant, by making assumptions about the dynamics [28]. The more accurate the nonlinear model, the better the predictions to minimize future error, and thus improved performance of the controller.

3.2.1 Simplified Nonlinear MPC

Nonlinear Adaptive Model Predictive Control (nMPC) represents a basic approach to making adaptive a canonical DMC controller. The formulation presented by Wilson et al. [15] alters the formulation of the dynamic matrix **A** from being a constant to being calculated at each time step. A block diagram is shown in Figure 3.1.

A step response of the simulated system is used at each interval, making this a very computationally intensive algorithm. The resulting optimal control move is:

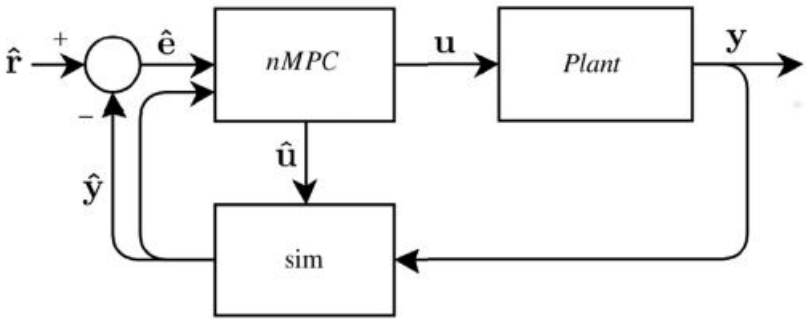

Figure 3.1: nMPC Block Diagram [15]

$$\Delta \mathbf{u} = (\mathbf{A}|_t^T \mathbf{A}|_t - \lambda \mathbf{I})^{-1} \mathbf{A}|_t^T \mathbf{E} \tag{3.6}$$

The least-squares optimization of the cost function J (which remains the same) still holds, since the adaptation is a linearization at each time step.

This was extended in the same paper to a Simplified nonlinear Model Predictive Control (SnMPC). The goal being to decrease computational effort by avoiding the use of a dynamic matrix to allow for a similar level of control of more complex systems. This approach uses a different cost function:

$$J = \sum_{i=1}^{N_u} (\hat{\mathbf{w}} - \hat{\mathbf{y}})^2 \alpha_i \tag{3.7}$$

where N_u is the number of control steps into the future that will be used considered in error optimization, $\hat{\mathbf{w}}$ is the predicted setpoint, and α_i is a vector of weights to adjust the aggressiveness of the controller.

The prediction vector is calculated in a different way as well, being split into two parts. First the full prediction vector:

$$\hat{\mathbf{y}}_i = \hat{\mathbf{y}}_i' + \frac{\Delta u|_t}{\delta} \hat{\mathbf{y}}_i'' \tag{3.8}$$

where $\hat{\mathbf{y}}_i'$ is the prediction vector assuming the control action remains constant and $\hat{\mathbf{y}}_i''$ is the change in the prediction vector due to a small change in control action. These

parts of the prediction vector are calculated using the following equations:

$$\hat{\mathbf{y}}'_i = f(u|_{t-1})_i \tag{3.9}$$

$$\hat{\mathbf{y}}''_i = f(u|_{t-1} + \delta)_i - f(u|_{t-1})_i \tag{3.10}$$

These predictions rely on a simulation of the controlled system represented by $f(u)$, as shown in Figure 3.1. What this method for prediction achieves is essentially a sort of linearization based on a small change in control action, represented by δ, which must be kept small enough for this linearization to be accurate [15].

Substituting the full prediction vector shown in equation 3.8 into the cost function shown in equation 3.7 yields the full objective cost function to be used:

$$J = \sum_{i=1}^{N_u} (\hat{\mathbf{w}} - \hat{\mathbf{y}}'_i - \frac{\Delta u|_t}{\delta} \hat{\mathbf{y}}''_i)^2 \alpha_i \tag{3.11}$$

This equation is expanded and then minimized in the same way as the standard DMC formulation, by setting the derivative to zero and solving for optimal changes in control moves. The following expression is the final control law for the SnMPC:

$$\Delta u|_t = \frac{\delta \sum_{i=1}^{N_u} \alpha_i (\hat{\mathbf{w}}_i - \hat{\mathbf{y}}'_i) \hat{\mathbf{y}}''_i}{\sum_{i=1}^{N_u} \alpha_i \hat{\mathbf{y}}''_i} \tag{3.12}$$

It is important to note that this method does not rely on matrix multiplication or utilize any matrix inversion, which are extremely computationally costly. While there is more computational effort required in building the prediction vector, where a simulation of the nonlinear system is preferred, this is made up for by avoiding costly matrix operations, while achieving high performance [15] in controlling nonlinear systems.

3.3 Complex Profile Tracking

Depending on the system and application, it is often necessary to track complex setpoint profiles, which is the case for path following for AUV systems. Historically, MPC algorithms were used in applications where simple reference trajectories were acceptable [18][29][30]. With the more recent application of MPC algorithms to fields such as aerospace, marine systems, and robotics, the necessity of complex setpoint tracking was made more apparent.

A typical MPC or even nMPC algorithm as discussed above often struggles with steady-state error, or tracking offset when attempting to adhere to complex setpoints. It is useful to address this problem in the control of an AUV, as it is often desired to track complex profiles. A simple algorithm is proposed as a strategy to address this tracking offset in [30] and is detailed here. For the full derivation and a more detailed explanation of the development process of this algorithm, see [30].

The method used to compensate for tracking offset using an MPC algorithm is denoted in [30] as η-MPC. This method proposes a correction to the prediction vector $\hat{\mathbf{y}}$ before it is used to generate optimal changes in control action. The mechanism to calculate the adjustment to the prediction vector is similar in spirit to the integrator component of a PID controller, where steady-state error is eliminated by summing past errors over time to generate an additional gain to overcome the offset.

First, the slope of the reference trajectory is calculated for the current time step, assuming that the slope remains constant from time $(t-1)$ to time $(t+1)$:

$$\zeta_t = r|_t - r|_{t-1} \tag{3.13}$$

under that assumption that:

$$r|_t - r|_{t-1} = r|_{t+1} - r|_t \tag{3.14}$$

where ζ is the slope of the setpoint vector at time t. The correction value η to be applied to the prediction is then obtained through the following:

$$\eta_t = \eta_{t-1} + \sigma e_{t-1} \tag{3.15}$$

where η_{t-1} is the correction value from the previous time instant (initialized at zero in the formulation given in [30]), σ is a weighting factor chosen to determine the aggressiveness of the controller (more on the selection of this value can be found in the reference given), and e_{t-1} is the error value from the previous time instant (the difference between the setpoint and system output at that time instant). The correction factor is then applied to the prediction as a product with the setpoint slope, which yields:

$$\hat{\mathbf{y}}_\eta|_t = \hat{\mathbf{y}}|_t + |\eta_t|\zeta_t \tag{3.16}$$

This is applied to the prediction vector at the current time t, following which the rest of the control loop is executed normally.

In the course of building simulations to investigate this approach for use on an AUV system (this preliminary work can be seen Chapter 4), it was determined that this method could exhibit adverse effects under certain operating conditions. The observation was made that depending upon the setpoint and the prediction at the current time t in a simulation, that the application of the correction factor could cause a sign change in some of the values of the prediction vector. This caused undesired behaviour in the system output. To account for this, a modification is made to the algorithm where instead of adjusting the prediction vector, the correction value is instead applied to the setpoint at the current time instant. This has the effect of using an artificial reference in the calculation of the optimal changes in control action, resulting in the desired behavior from the controller. The following is the modified

form of equation 3.16:

$$r_\eta|_t = r|_t + |\eta_t|\zeta_t \tag{3.17}$$

This is the form used in all applications in this research.

The following chapter presents the preliminary simulations of the AUV system presented in Chapter 2 and simulations of the control algorithms discussed in this chapter.

Chapter 4

Preliminary Simulations

The purpose of this chapter is to present and discuss the preliminary simulations performed as part of this research. These include an example of the improvement gained over a classical control strategy by implementing a predictive control strategy, various simulations demonstrating the effectiveness of a nonlinear predictive control strategy used in conjunction with modifications discussed in chapter 3, and various open-loop simulations of the REMUS 100 AUV, in order to examine the motion of the AUV under different conditions.

The control simulations in this chapter are performed with the intention of examining and displaying the capabilities of the chosen control schemes in overcoming the challenges (nonlinearity, underactuation, etc.) that are present in the full AUV model, as previously discussed. The systems discussed in this chapter are simple when compared to the full AUV model, allowing the development and testing of control strategies with fewer obstacles. These algorithms are then extended to the AUV in the following chapters.

4.1 Single Inverted Pendulum

This section demonstrates the effectiveness of two control algorithms in controlling a Single Inverted Pendulum (SIP) on a cart, and serves to test the capability of the SnMPC to control a challenging system. The SIP system is an excellent example of a system that is nonlinear, unstable, exhibits non-minimum phase behaviour, and is underactuated, while being constrained enough in its motion that less complex controllers are capable of achieving control. The SIP system is a well known and widely used control problem. A diagram of this system is shown below in Figure 4.1. The pendulum is free to swing, while the cart is constrained to one line of motion, and the only actuation available is a force u applied to the cart, making the system underactuated. The objective of this system is typically to recover either initial angles or disturbances to the pendulum back to the unstable equilibrium (pendulum up). Various constraints can be imposed to model real systems such as cart position limits or control force available to the cart.

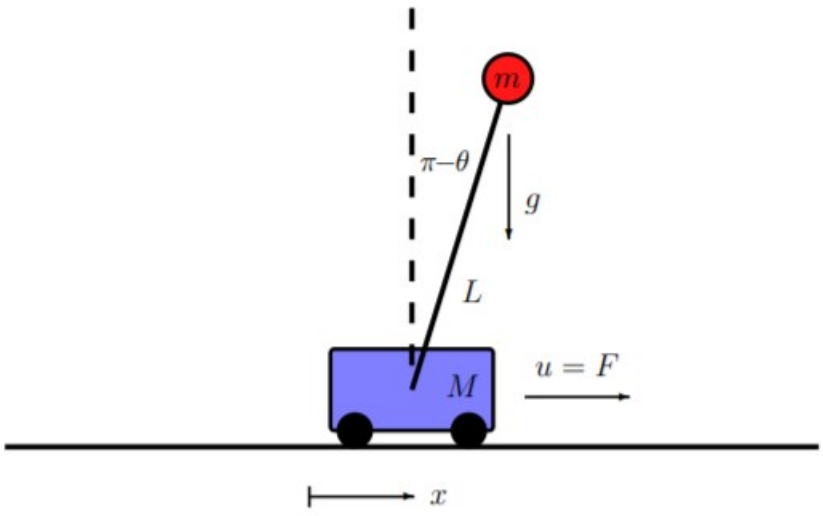

Figure 4.1: Single Inverted Pendulum Diagram on a Cart [35]

The following are the complete nonlinear state equations, from [31]:

$$
\begin{aligned}
\dot{x} &= v \\
\ddot{x} &= \frac{-m^2L^2g\cos\theta\sin\theta + mL^2(mL\dot{\theta}^2\sin\theta - d\dot{x}) + mL^2u}{mL^2(M + m(1-\cos^2\theta))} \\
\dot{\theta} &= \omega \\
\ddot{\theta} &= \frac{(m+M)mgL\sin\theta - mL\cos\theta(mL\dot{\theta}^2\sin\theta - d\dot{x}) - mL\cos\theta u}{mL^2(M + m(1-\cos^2\theta))}
\end{aligned}
\tag{4.1}
$$

where x and θ are the cart position and pendulum angle, M is the cart mass, m is the pendulum mass (modeled as a point mass at the end of the pendulum), L is the pendulum length, g is the acceleration due to gravity, and d represents a damping coefficient.

To get an idea of the nature of the dynamics of this system, open-loop testing was performed on the simulated system. The motion exhibited by the system is fairly complex, showing free oscillation of the pendulum, and the cart being moved back and forth as a result.

The Linear Quadratic Regulator (LQR) is a well-documented controller using optimal control theory to minimize a quadratic cost function, allowing the tuning of $\mathbf{Q}$ and $\mathbf{R}$ matrices to penalize deviations from the setpoints of the states and the control actions respectively. In order to calculate the LQR gain for the system, a linear model is required. From [31], the inverted pendulum is linearized about its upright equilibrium point. The control gain is calculated using this linearized model and then applied to the full nonlinear system.

Formulated from the parameters in [31], simulations were built to show the LQR controller operating on a SIP system. For brevity, only one is shown here in Figure 4.2 as it is consistent with other simulations.

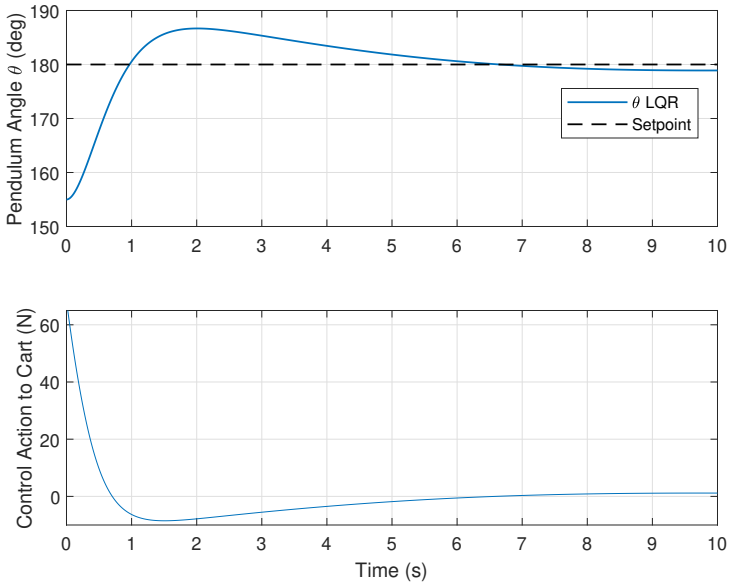

Figure 4.2: LQR Control of SIP (25° Initial Angle)

These simulations display the ability of the LQR to recover to the initial position of the cart and the pendulum to its equilibrium position from an initial angle away from the equilibrium. The LQR performs fairly well, recovering and maintaining the equilibrium point within approximately 7 seconds.

Following this, the nonlinear predictive control strategy discussed in chapter 3 (Sn-MPC) is applied to the same SIP system with the same initial angle. A prediction horizon $N_u = 20$ is chosen, along with a tuning value of $\delta = 0.1$. The following Figure 4.3 presents the results.

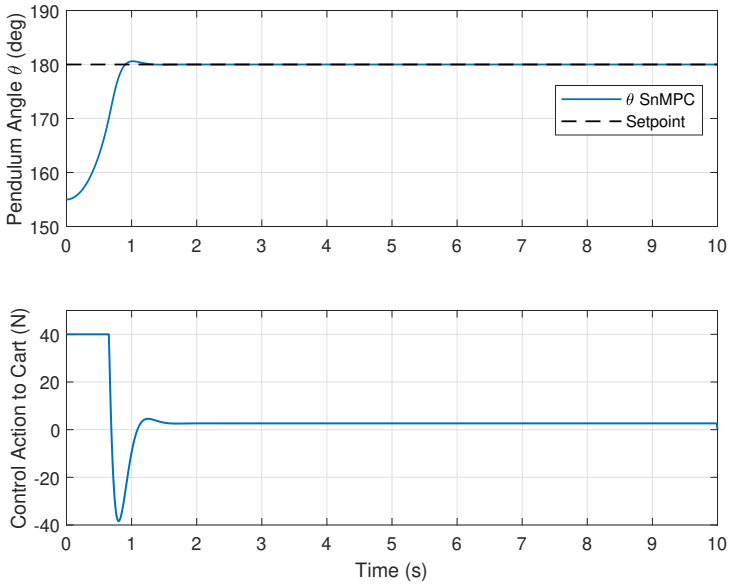

Figure 4.3: SnMPC Control of SIP (25° Initial Angle)

This simulation demonstrates the effectiveness of moving to a more advanced control strategy in order to better accommodate more complex systems. The SnMPC was able to recover the equilibrium position within just over 1 second, showing a massive improvement over the LQR control. Further, it was able to accomplish this with a hard constraint on the available control action, resulting in a spike of a lower magnitude. This improvement can be attributed to the fact that the LQR controller relies on a linearization about the equilibrium point, where the SnMPC utilizes a simulation of the nonlinear dynamics, allowing not only a greater operating range, but improved performance over that range. The ability of the SnMPC to predict future behaviour of the system and react accordingly also contributes to the improved control performance. This simulation works to demonstrate the effectiveness of the SnMPC strategy in the presence of nonlinearities, under-actuation, and open-loop unstable conditions.

4.2 Complex Profile Tracking

The purpose of this section is to highlight the advantages of utilizing the complex profile tracking technique discussed in chapter 3 when controlling a nonlinear system, while also demonstrating the advantages gained in utilizing a nonlinear predictive strategy (SnMPC) over a traditional predictive strategy (DMC).

The system used for testing in this section is from [32] and is described using a discrete-time form as follows:

$$y_t = \frac{y_{t-1}y_{t-2}(y_{t-1}+2.5)}{1+y_{t-1}^2+y_{t-2}^2} + u_t \tag{4.2}$$

This system was chosen because it exhibits a fairly high degree of nonlinearity as well as having dead-time-like behaviour. Several simulations were done using both DMC and SnMPC strategies. The pertinent examples that show the advantage of a nonlinear formulation over a linearized control model, as well as the effectiveness of the complex profile tracking technique are displayed here.

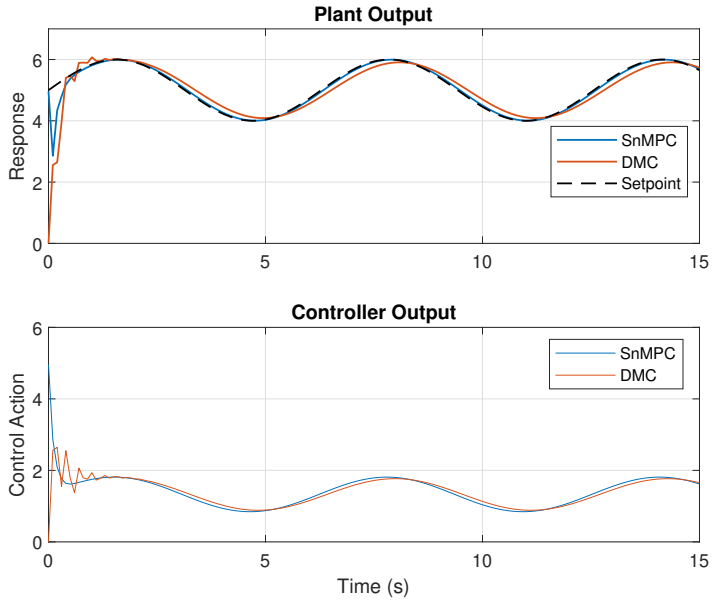

Figure 4.4: Sinusoidal Input Comparison

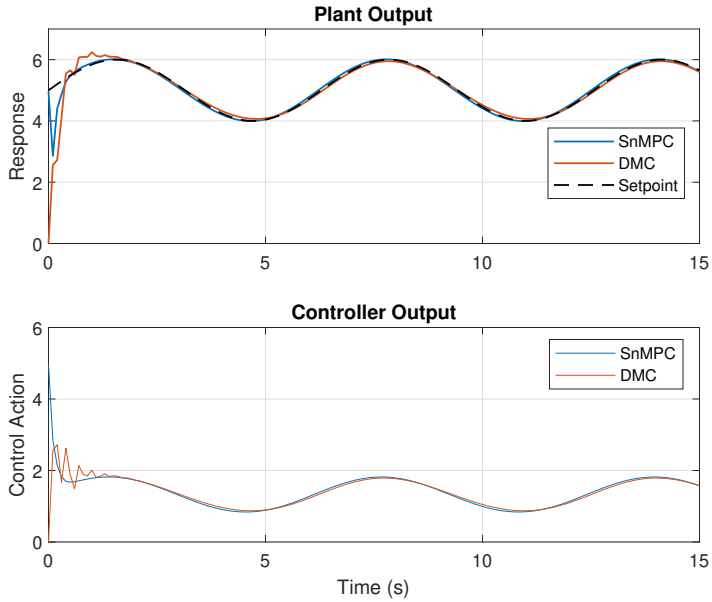

Figure 4.5: Sinusoidal Input Comparison with η-correction

Figures 4.4 and 4.5 show the tracking of a sinusoidal profile by both controllers. Figure 4.4 shows the simulation of the system before applying the η-correction factor to artificially alter the setpoint. While the SnMPC performs quite well even before the modification, the DMC clearly expresses some difficulty in maintaining the setpoint without some constant offset. Figure 4.5 shows the tracking produced after the application of the η-correction. While the while the DMC still shows some initial error, the constant offset has been effectively eliminated. The mean-squared error values (collected in Table 4.2) show that the SnMPC performance is also improved by the setpoint adjustment.

To better highlight the utility of a controller that simulates the full nonlinear model as opposed to capturing the plant dynamics outside of the control loop, Figure 4.6 displays the performance of the SnMPC over the DMC in various step inputs to the system.

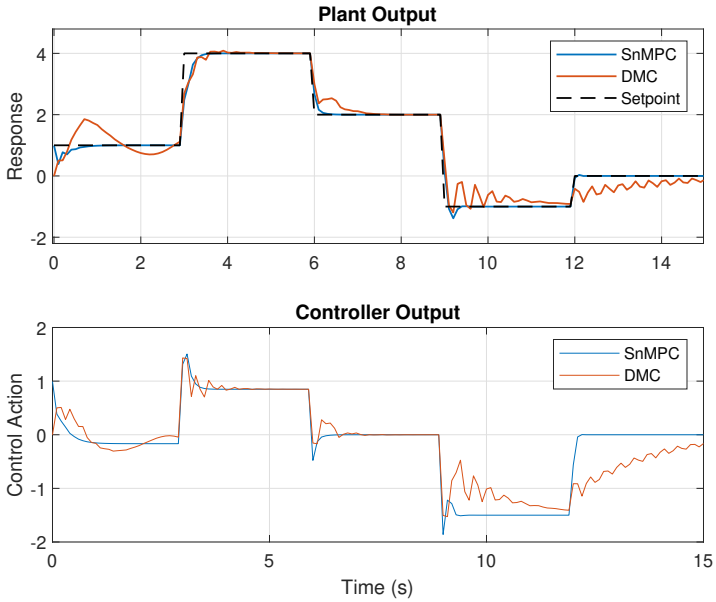

Figure 4.6: Step Input Comparison

In this simulation, the DMC shows the ability to track the setpoint fairly well when it is further away from a zero value, however a breakdown in control can be observed due to the inability to accurately capture the system dynamics at a near zero setpoint value. The SnMPC shows no such difficulty, instead utilizing the full simulation of the system over the prediction horizon to generate control actions. Tables 4.1 and 4.2 collect parameters for the simulations and controllers used and the mean-squared error values for various tests performed (including those displayed in Appendix A) respectively. In addition, in all simulations the tuning parameter for the η-correction, σ, is set to 0.3.

Table 4.2 shows an improvement in mean-squared error in almost all cases when utilizing η-correction. The exception to this being in the varying steps setpoint controlled by the DMC strategy. This could be attributed to tuning of the controller, or to the functionality of the η-correction, as it can be subject to an integral windup effect sometimes observed in control schemes utilizing an integral function.

The final simulation in this section serves to illustrate the importance of properly

Table 4.1: Controller Parameters

Controller	Parameter	Value
DMC	N	20
	n_u	3
	λ	1.2
SnMPC	N_u	4
	δ	0.01
	α	1

Table 4.2: Mean-Squared Errors for Various Simulations

Setpoint Profile	**Controller**			
	DMC	**η-DMC**	**SnMPC**	**η-SnMPC**
Steps	0.1348	0.1610	0.0394	0.0359
Sine	0.2901	0.2657	0.0414	0.0406
Ramps	0.0456	0.0344	0.0062	0.0041

choosing tuning parameters by varying the N_u prediction horizon for the SnMPC strategy:

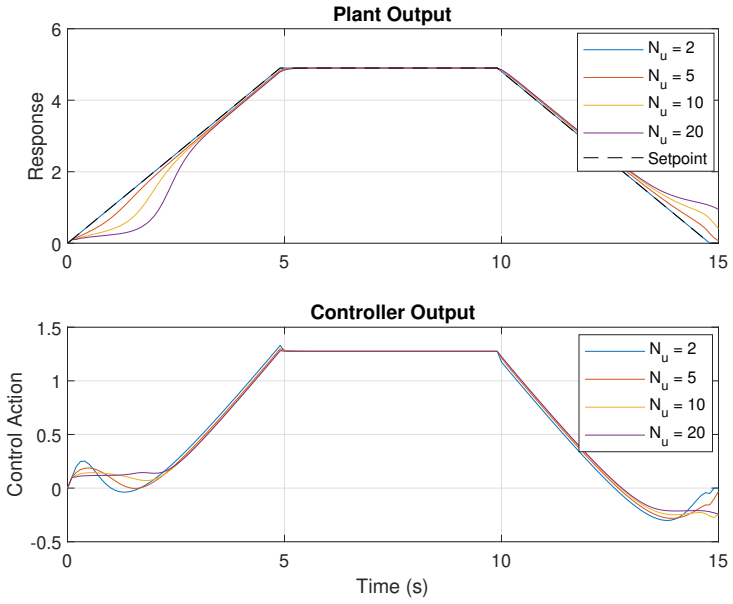

Figure 4.7: System Simulation for Varied N_u

The mean-squared errors for each value of N_u are collected in the Table 4.3.

Table 4.3: Mean-Squared Errors for Varying N_u

$\mathbf{N_u}$	**Mean-Squared Error**
2	7.068e-31
5	0.0135
10	0.0746
20	0.1918

From Figure 4.7 and the corresponding data in Table 4.3, the effectiveness of choosing a low value of N_u are apparent. This will not be the case for all systems, however, and must be investigated based on the system dynamics. For example, a system exhibiting non-minimum phase behaviour would likely be difficult or impossible to control with such a short prediction horizon, as there would not be enough data to accurately predict the behaviour of the system.

4.3 AUV Simulation

Following the simulations that display the advantages of the control strategies selected for this research, this section covers the simulation of the AUV kinematic and dynamic equations presented in Chapter 2. The objective here is to observe how the AUV system reacts to various input conditions without control, in order to better understand the system and the limitations it may present. These simulations utilize a fourth-order Runge-Kutta approximation to evaluate the system differential equations, with a time-step Δt of 0.1 seconds.

The first simulation shown is an open-loop simulation of the AUV system with a nominal maximum rudder fin angle of 15 degrees for 28 seconds to determine a minimum turning radius for the AUV when control is applied. The coefficients and physical parameters are as close as possible to those mentioned in [24]. The AUV is given a

constant thrust, with an initial surge velocity u of 1.54 m/s, and an initial roll angle ϕ of -5 degrees. The initial roll angle is an initial condition given in the reference [24]. Figures 4.8 - 4.11 depict this maneuver.

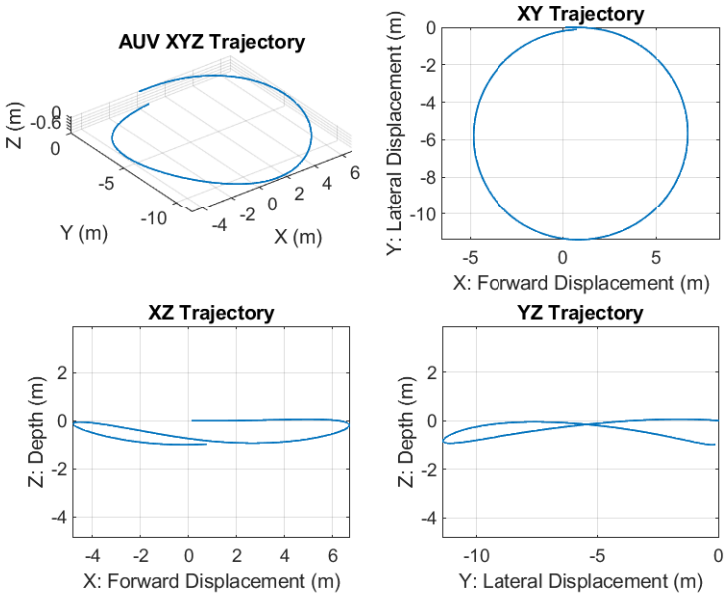

Figure 4.8: AUV Trajectory (Maximum Rudder Fin Angle)

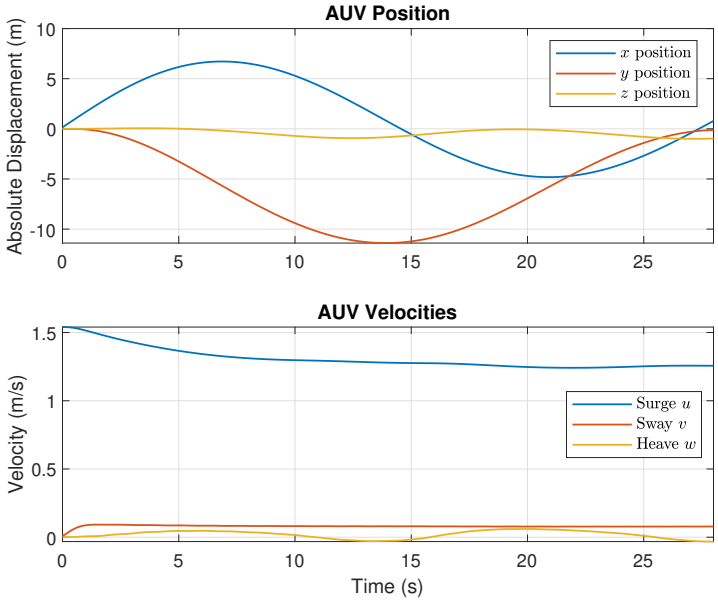

Figure 4.9: AUV Linear Displacements and Velocities (Maximum Rudder Fin Angle)

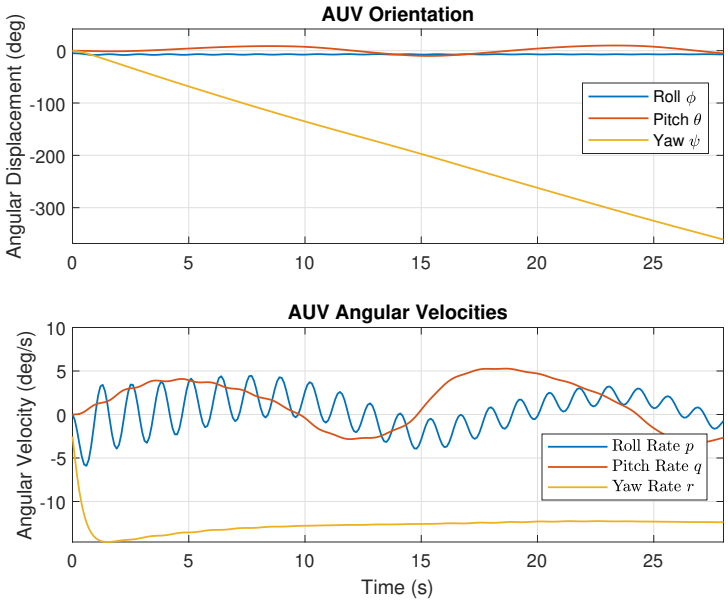

Figure 4.10: AUV Angular Displacements and Velocities (Maximum Rudder Fin Angle)

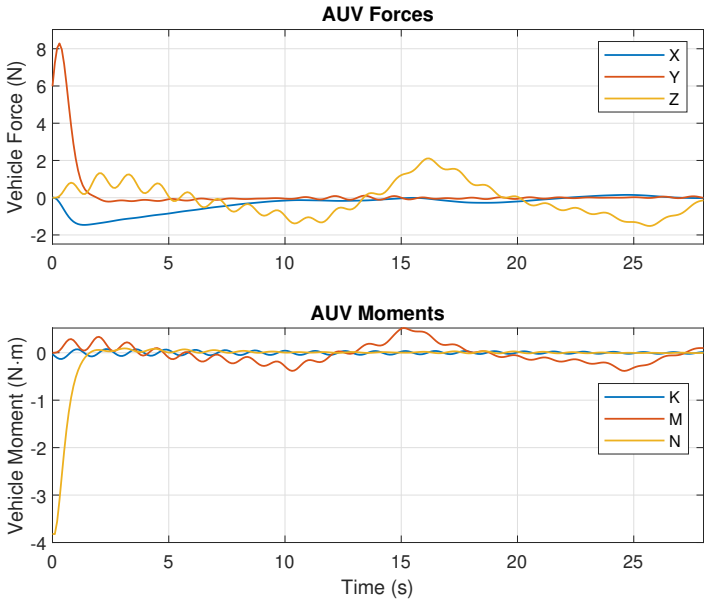

Figure 4.11: AUV Forces and Moments (Maximum Rudder Fin Angle)

From this, it is determined that the minimum turning radius for the vehicle is approximately 6.5 meters at a maximum rudder fin angle of 15 degrees. This is important as it will determine which target points are unattainable under typical operating con-

ditions, without modification to the path being followed. In order to avoid actuator saturation, the minimum turning radius for the vehicle should be assumed to be larger than what is shown here.

The next simulation shows the system reacting to an oscillating rudder fin input within ±5 degrees over 35 seconds to display the kinematic and dynamic reactions of changing fin angles. Note that it is assumed that fin angles change instantaneously (noted also in Chapter 2). This maneuver is seen in Figures 4.12 - 4.15.

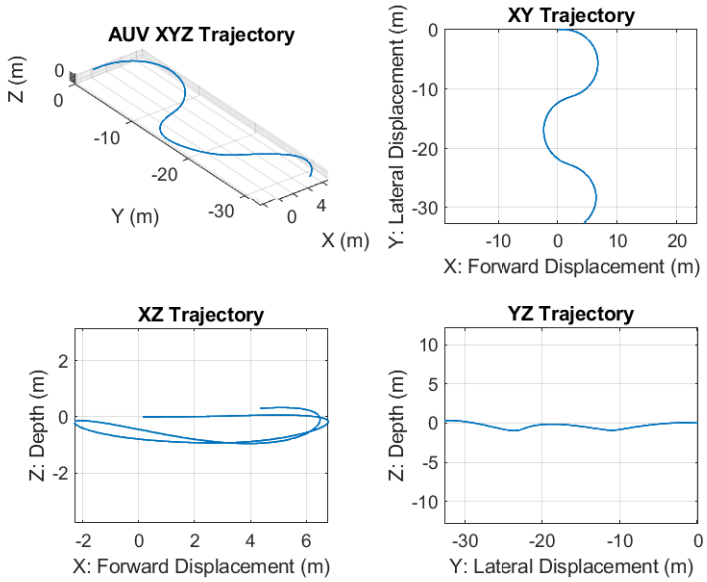

Figure 4.12: AUV Trajectory (Oscillating Input)

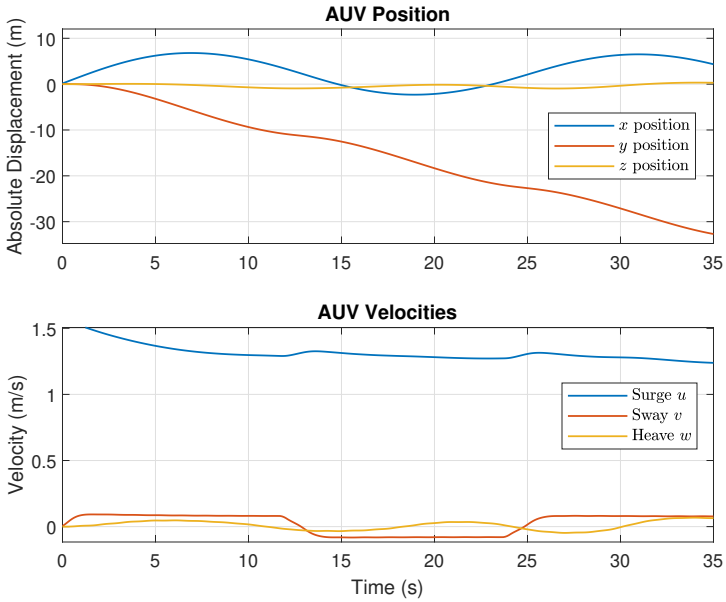

Figure 4.13: AUV Linear Displacements and Velocities (Oscillating Input)

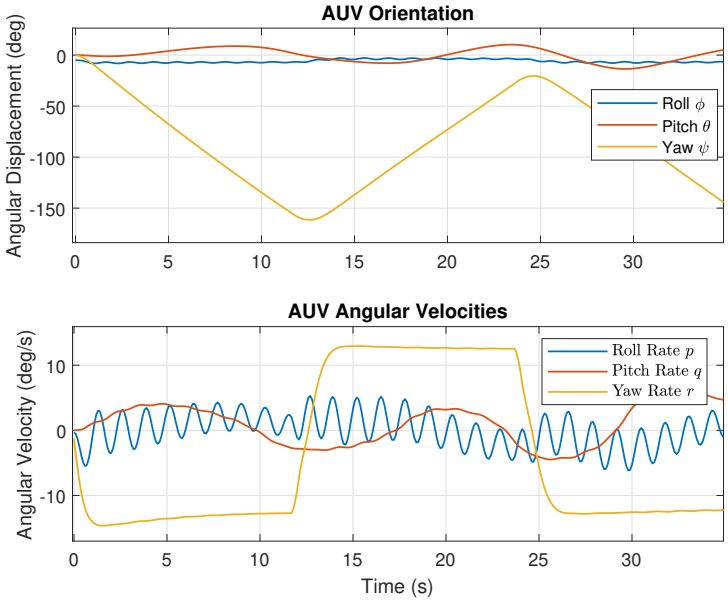

Figure 4.14: AUV Angular Displacements and Velocities (Oscillating Input)

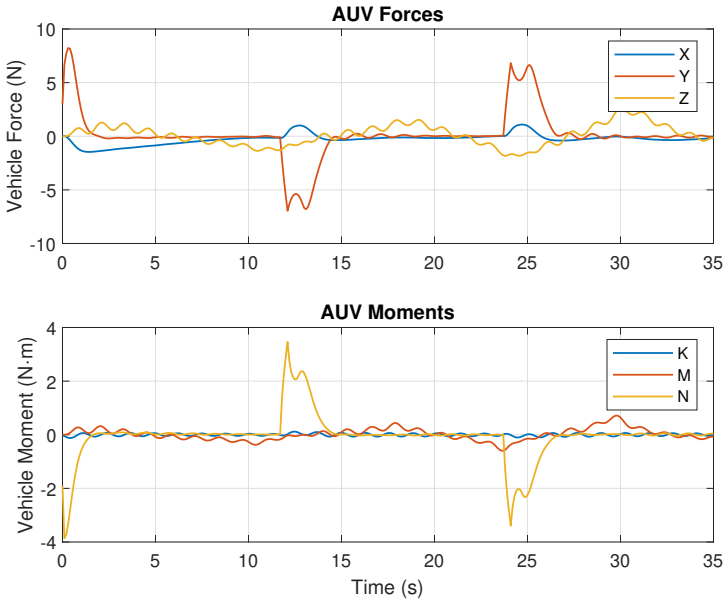

Figure 4.15: AUV Forces and Moments (Oscillating Input)

At the time instants when fin angle changes are occurring, spikes in the force along the Y direction of the AUV and the moment about the Z direction (M) are seen in Figure 4.15. These are directly resultant from the fin input, and lead to the sharp changes in yaw rate r, leading to angular velocities oscillating between approximately ± 13 degrees/second. The results from this simulation are consistent with a similar simulation performed in [24] which in turn was consistent with experimental AUV tests.

The final simulation shown in this section (Figures 4.16 - 4.19) shows a depth change resulting from a pitch-down maneuver over 5 seconds, with a pitch fin angle of 5 degrees over 2 seconds, similar to that depicted in [24].

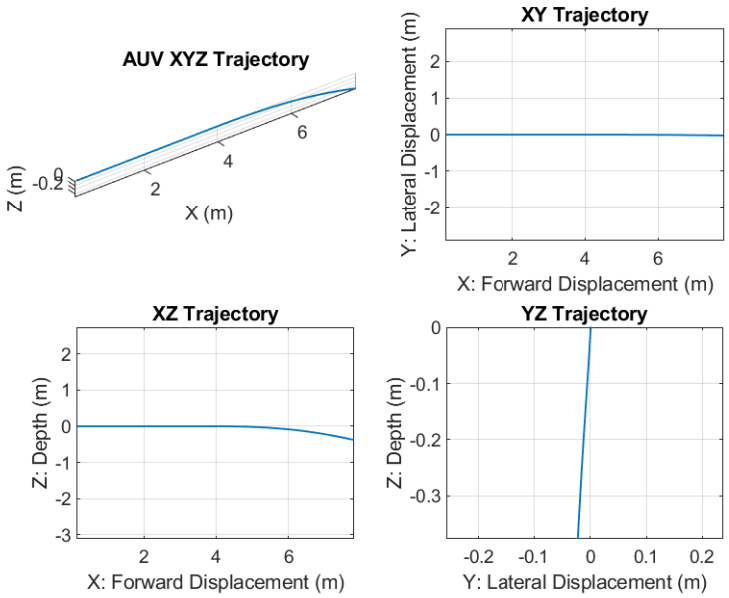

Figure 4.16: AUV Trajectory (Pitch Down)

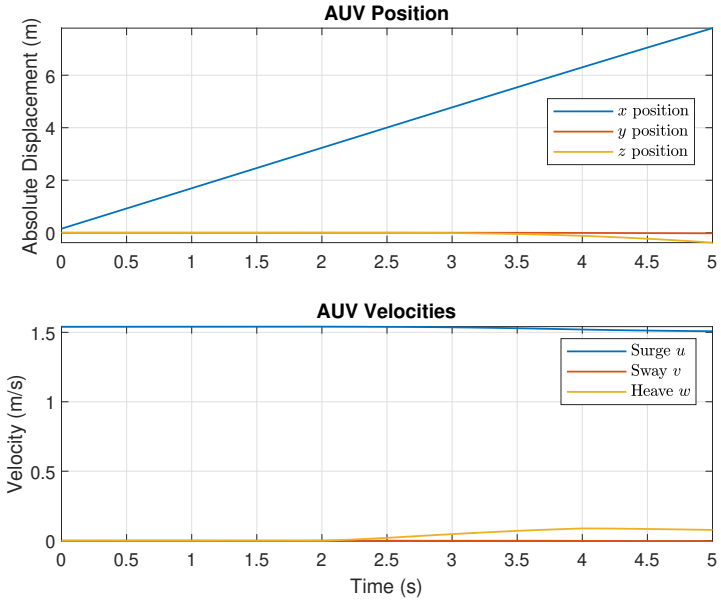

Figure 4.17: AUV Linear Displacements and Velocities (Pitch Down)

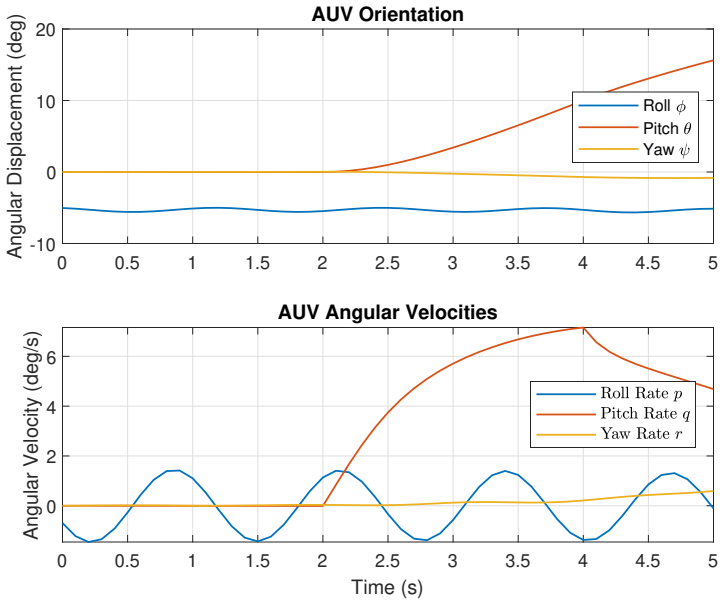

Figure 4.18: AUV Angular Displacements and Velocities (Pitch Down)

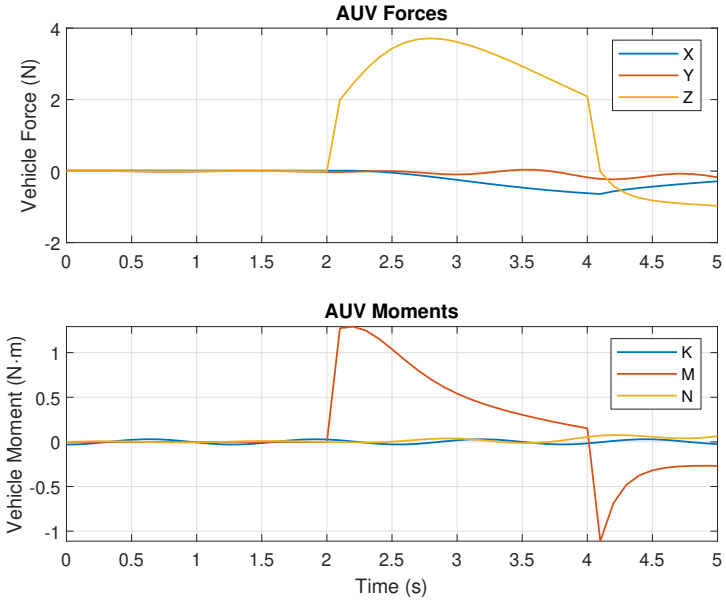

Figure 4.19: AUV Forces and Moments (Pitch Down)

This action results in a depth change of approximately 0.3 meters. The results from this simulation are consistent with [24], which includes experimental validation. This would be drastically influenced by the ballast of the AUV, which should be taken to

be a design variable when running simulations or designing control strategies for this vehicle.

This chapter covered the simulations that outline the basis for the path following control that is described and demonstrated in the following chapter. Different control methods in the traditional LQR and the more advanced predictive strategies DMC and SnMPC are displayed to highlight the advantages of predictive control. The effectiveness of utilizing an explicitly nonlinear strategy and generating predictions from a nonlinear model of a system is demonstrated, as well as the improvements in control that can obtained by using the η-correction technique to influence the setpoint the controller sees. Finally, simulations were performed without control on the REMUS 100 AUV model in order to glean useful information about the behaviour of the system prior to moving on to a path following controller implementation.

Chapter 5

AUV Path Following

This chapter covers the control strategy used for the path following control of the REMUS 100 AUV presented in this research. Several path following strategies and control strategies are discussed, and the development of the chosen strategy is shown.

5.1 Path Following Strategies

5.1.1 Line of Sight

Line of sight (LOS) guidance systems are commonly used in marine surface vessel as well as underwater vessel path following control systems. These systems are conceptually simple, acting in the same way a helmsman might, by utilizing a waypoint (on its own or as part of a path) to derive angles of yaw and pitch (for 3D applications) to drive the vehicle towards the waypoint. These angles can then be used to determine error signals for a controller to convert into fin angles and thrust (in the case of an underactuated AUV). This method in its most basic form has the advantages of being simple to implement, computationally lightweight, flexible in the type of controller that can be used, and versatile in how it is able to work with single waypoints, predefined paths, or path generating algorithms.

Positional error is calculated in the earth-fixed reference frame as:

$$\mathbf{P}_e = [x_e \quad y_e \quad z_e]^T = \mathbf{P}_{SF} - \mathbf{P}_B \tag{5.1}$$

where $\mathbf{P}_{SF} = [x_F \quad y_F \quad z_F]^T$ is the position of the current target point on the desired path and $\mathbf{P}_B = [x \quad y \quad z]^T$ is the position of the AUV. Using these errors, the target pitch and yaw angles denoted by θ_t and ψ_t are determined using the following equations:

$$\theta_t = -\frac{\pi}{2} + \arccos\left(\frac{z_e}{||\mathbf{P}_e||}\right) \tag{5.2}$$

$$\psi_t = \arctan\left(\frac{y_e}{x_e}\right) \tag{5.3}$$

The chosen controller then works to drive the error between the target pitch and yaw angles and the AUV pitch and yaw angles to zero. To advance the waypoint the AUV must come within a zone around the current target waypoint, in this case a predefined three dimensional Euclidean distance. This distance can be treated as a design variable and is denoted γ, to be selected based on the accuracy required, the AUV capability, the vehicle velocity, or other factors. In order for this algorithm to maintain its function, waypoints must be achievable; that is to say it is within the line of sight of the vehicle, line of sight being subject to the vehicle constraints (velocity, turning radius).

In this form, the LOS guidance method has the drawback of not being able to reliably converge to a path when not started on or near the target waypoint with an appropriate orientation. There have been attempts made to solve this by modifying the algorithm by choosing the target point as the closest point on the path, however this too can be unreliable when there is not a single closest point. Other methods for convergence include using control-Lyapunov functions (CLF) to control the transient

behaviour of the vehicle, or generating a spline of points that can be attained on the way to a new waypoint on the path. Some of these methods can be seen in [33] [34]. In [35], the various potential pitfalls of these methods are discussed in more detail.

5.1.2 Virtual Target Vehicle

The virtual target vehicle guidance system is commonly used as a workaround for the issues an LOS system presents. This method sees a virtual target vehicle that moves along the desired path. This allows the control of the AUV to be performed in two phases. The first is to determine the three dimensional positional error between the AUV and the virtual vehicle, and reduce it to zero, while also aligning the AUV with the orientation of the virtual target vehicle. This is accomplished by controlling virtual control signals for the desired surge, yaw rate, and pitch rate u_d, r_d, and q_d respectively. These are then used as inputs into the second phase of the controller to adjust thrust and fin angles to reach these desired values.

The virtual target vehicle is represented as a moving reference frame constrained in its motion to the desired path. Also called the Frenet-Serret frame, the unit vectors that define this frame at each point on the path are the unit vector tangent to the path, the unit normal vector (parameterized by arc length s), and the unit binormal vector. In keeping with common notation, these are denoted as x_{SF}, y_{SF}, and z_{SF} respectively. This will be used to calculate the error model of the AUV as seen in [36].

Calculating the position and orientation error between the virtual vehicle and the AUV is done as follows:

$$\mathbf{P}_e = [x_e \quad y_e \quad z_e \quad \theta_e \quad \psi_e]^T = \mathbf{R}_I^F(\mathbf{P}_B - \mathbf{P}_{SF}) \tag{5.4}$$

where $\mathbf{P}_B = [x \quad y \quad z \quad \theta \quad \psi]^T$ is the position and orientation of the AUV in the

earth-fixed reference frame, and $\mathbf{P}_{SF} = [x_F \quad y_F \quad z_F \quad \theta_F \quad \psi_F]^T$ is the position and orientation of the virtual vehicle in the earth-fixed reference frame. To express the position error in the SF frame, the following rotation matrix $\mathbf{R}_I^F$ from the earth-fixed to virtual vehicle frame is used:

$$\mathbf{R}_I^F = \begin{bmatrix} \cos\psi_F \cos\theta_F & \sin\psi_F \cos\theta_F & -\sin\theta_F & 0 & 0 \\ -\sin\psi_F & \cos\psi_F & 0 & 0 & 0 \\ \cos\psi_F \sin\theta_F & \sin\psi_F \sin\theta_F & \cos\theta_F & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cos\theta_F \end{bmatrix} \tag{5.5}$$

The pitch θ_F and yaw ψ_F of the virtual vehicle are calculated for each time step using the following equations:

$$\theta_F = -\arctan\left(\frac{\dot{z_F}}{\sqrt{\dot{x}_F^2 + \dot{y}_F^2}}\right) \tag{5.6}$$

$$\psi_F = \arctan\left(\frac{\dot{y}_F}{\dot{x}_F}\right) \tag{5.7}$$

Using these error values, the kinematic error model expressed as a system of first-order differential equations can be derived. The full derivation is relatively concise and can be seen in [35]. A derivation can alternatively be seen in [37] in the AUV body-fixed reference frame, or in [36] from the virtual vehicle reference frame, as it is presented here. The following equations show the final form of the three dimensional error model, ignoring heave and sway, as well as roll, as these are small enough to be considered insignificant for the first phase of control.

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{z}_e \\ \dot{\theta}_e \\ \dot{\psi}_e \end{bmatrix} = \begin{bmatrix} u\cos\psi_e\cos\theta_e - u_F - q_F z_e + r_F y_e \\ u\sin\psi_e\cos\theta_e - r_F x_e \\ -u\sin\theta_e + q_F x_e \\ q - q_F \\ -q_F\psi_e\tan\theta_F + r\cos\theta_F/\cos\theta - r_F \end{bmatrix} \tag{5.8}$$

Where q_F and r_F are the pitch rate and yaw rate of the virtual vehicle frame. In both [35] and [36], these are calculated using the desired path curvature and torsion, respectively. This works well when the path is known in advance, and when the curvature and torsion are known at each point on the path, however in practical application this may not be possible. As such, this implementation uses a numerical approach to calculating these values, in order to maintain flexibility in path choice. Due to the control architecture associated with this path following method, there is typically a much higher computation time when compared with a LOS system (depending on the algorithm used), and in part due to the formulation of the error model associated with the virtual target vehicle.

5.1.3 Hybrid Guidance

The virtual vehicle guidance system has been shown to be highly effective for AUV path following, especially when there is a high initial position and orientation error between the AUV and the desired path. This method can fall short, however, when used in an automated setting, where the path to be followed is not necessarily known in advance. Path planning algorithms in three dimensions can be very computationally expensive, and can be designed to generate waypoints that are achievable by the AUV within its operational constraints. This method lends itself well to a LOS guidance system, where the control of the AUV is computationally lighter weight, to offset the need for the path planning algorithm.

The ability to both converge to a path and be able to adapt to having a single target waypoint at a time when a desired position and orientation is achieved is the inspiration for the hybrid guidance system shown in this research. This guidance method affords the flexibility to work with both predefined paths as well as target waypoints generated in real time.

Given the LOS guidance system and the virtual target vehicle guidance system discussed previously, the hybrid guidance system proposed here simply allows the switching of one control system to another based on some condition being met. The control system begins by utilizing the multistage control based on a virtual vehicle to ensure convergence to the path. In the case of this research, the condition being used to switch between the two path following strategies is simply a suitably small value of cross-track error (error in the yz-plane of the body-fixed frame), which can be considered a design variable. When this condition has been met, the LOS guidance system is activated and will continue tracking the path or use well generated waypoints from a path generation algorithm. If error were to again grow outside of the allowable range, the path following strategy would swap back to the virtual vehicle guidance system.

5.2 AUV Control

This section details how the previously discussed control systems are implemented for the AUV. The development is divided into sections for each of the path following strategies separately, and then a short discussion on the hybrid guidance system is presented. The virtual vehicle control system is shown first due to certain details of the LOS control system being developed in that section.

5.2.1 Virtual Vehicle Control

The virtual vehicle control strategy is a two stage predictive control strategy, consisting of an MPC which gives the optimal optimal pitch rate q_d and yaw rate r_d to reduce positional and orientation error in the AUV to zero, and then the modified SnMPC to control the fin angles to achieve the desired pitch and yaw rates. The kinematic control (first stage) is based on [36], with several important differences. The most important difference is that in this research AUV propeller thrust is fixed. This reduces the number of controllable inputs, causing the AUV to be further underactuated in this case. Thrust is fixed to ensure model simulation accuracy, as the mathematical model shown in Chapter 2 is predicated upon a fixed thrust. Constraining thrust in this way results in a much more accurate simulation. Another key difference is absence of delay in the second stage controller, which slightly changes the system equations as will be seen.

To calculate q_d and r_d, the 3D error model equation 5.8 must be modified to include control inputs to influence the desired outputs. This takes the following form:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \tag{5.9}$$

The equations are based on those in [36], modified to exclude surge control, and arranged to be factored into a state-space form are as follows:

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{z}_e \\ \dot{\theta}_e \\ \dot{\psi}_e \\ \dot{q}_e \\ \dot{r}_e \end{bmatrix} = \begin{bmatrix} u\cos\theta_e + u(\cos\theta_e(\cos\psi_e - 1)/\psi_e)\psi_e - u_F - q_F z_e + r_F y_e \\ u(\cos\theta_e \sin\psi_e/\psi_e)\psi_e - r_F x_e \\ -u(\sin\theta_e/\theta_e)\theta_e + q_F x_e \\ q_e \\ -q_F\psi_e \tan\theta_F + (r_e + r_F)\cos\theta_F/\cos\theta - r_F \\ (q_{ed} - q_e)/T - \dot{q}_F \\ (r_{ed} - r_e)/T - \dot{r}_F \end{bmatrix} \tag{5.10}$$

and the control inputs are defined as:

$$\begin{aligned} q_{ed} &= q_d - q_F \\ r_{ed} &= r_d - r_F \\ \mathbf{u} &= [q_{ed} \quad r_{ed}]^T \end{aligned} \tag{5.11}$$

where q_{ed} and r_{ed} are the difference variables between the desired pitch and yaw rates and those of the virtual vehicle, used to calculate the desired variables for the second stage controller. The MPC input vector $\mathbf{u}$ is comprised of these values. The addition of the states $\dot{q}_e$ and $\dot{r}_e$ to the original error model is required to include control inputs. These are added by the author of [36] in order to account for delay in the dynamic controller used in that research. As the controller implemented here exhibits no delay, these equations are presented in a modified form where the approximations for the acceleration terms performed using sampling time T will leave only the control inputs q_{ed} and r_{ed}, seen below when the model is discretized.

The AUV has physical constraints as discussed in previous chapter. To avoid saturation of the control signals (fin angles) in the second phase of control, it is advantageous to apply constraints to the control actions in the MPC phase. As described in [18], the formulation of an MPC in the presence of constraints requires the use of a different objective function, which is developed here. The controlled error model can be closely approximated in a linearized state-space form about the zero-error equilibrium point as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{5.12}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & r_F & -q_F & 0 & uk_x & 0 & 0 \\ -r_F & 0 & 0 & 0 & uk_y & 0 & 0 \\ q_F & 0 & 0 & -u\sin\theta_e/\theta_e & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -q_F\tan\theta_F & 0 & \cos\theta_F/\cos\theta \\ 0 & 0 & 0 & 0 & 0 & -1/T & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1/T \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1/T & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/T \end{bmatrix}^T$$

where $k_x = \cos\theta_e(\cos\psi_e - 1)/\psi_e$ and $k_y = \cos\theta_e \sin\psi_e/\psi_e$. In the presence of constraints, the optimal control moves must be obtained using a quadratic cost function solved using quadratic programming as per [18]. This necessitates the use of an incremental state-space MPC controller. The major difference between this form and the basic DMC form shown in chapter 3 is how the the system is modeled to obtain predictions. In function there is essentially no difference, where before the dynamic matrix $\mathbf{A}$ (not to be confused with system matrix $\mathbf{A}$ used in this chapter) was used to form predictions, here the matrices that construct the state-space model are discretized and reorganized to form an incremental prediction, serving the exact same function as before by then calculating future errors, to optimize control moves.

The system in equation 5.12 is discretized using a zero-order hold approximation with sampling time T. This is calculated using $\mathbf{A}_d = \mathbf{I} + \mathbf{A}T$ and $\mathbf{B}_d = \mathbf{B}T$, where I is the identity matrix with the same dimensions as $\mathbf{A}$. These are then used to form an incremental controlled error model:

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}_d\mathbf{x}(t) + \mathbf{B}_d\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) \end{aligned} \tag{5.13}$$

where

$$\mathbf{A}_d = \begin{bmatrix} 1 & Tr_F & -Tq_F & 0 & Tuk_x & 0 & 0 \\ -Tr_F & 1 & 0 & 0 & Tuk_y & 0 & 0 \\ Tq_F & 0 & 1 & -Tu\sin\theta_e/\theta_e & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & 1 - Tq_F\tan\theta_F & 0 & T\cos\theta_F/\cos\theta \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{B}_d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T$$

and $\mathbf{C} = \mathbf{I}_{n\times n}$, with n being number of system outputs and m being system inputs by convention. As previously discussed, the approximation of the acceleration terms using T in the full error model allows the control signals to pass through the equations unaffected by any delays, in contrast with [36]. The general form of the state-space model is obtained by defining a state vector $\bar{\mathbf{x}}(t) = [\mathbf{x}(t) \quad \mathbf{u}(t-1)]^T$ and the change in control action $\Delta\mathbf{u}(t) = \mathbf{u}(t) - \mathbf{u}(t-1)$ to construct:

$$\begin{aligned} \bar{\mathbf{x}}(t+1) &= \bar{\mathbf{A}}\bar{\mathbf{x}}(t) + \bar{\mathbf{B}}\Delta\mathbf{u}(t) \\ \mathbf{y}(t) &= \bar{\mathbf{C}}\bar{\mathbf{x}}(t) \end{aligned} \tag{5.14}$$

where

$$\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_d & \mathbf{B}_d \\ 0 & \mathbf{I}_{m\times m} \end{bmatrix}, \quad \bar{\mathbf{B}} = \begin{bmatrix} \mathbf{B}_d \\ \mathbf{I}_{m\times m} \end{bmatrix}, \quad \bar{\mathbf{C}} = \begin{bmatrix} \mathbf{C} & 0 \end{bmatrix}$$

Using this general state-space form, the predictions $\mathbf{Y}$ can now be formed. When using an incremental model this is performed by recursive substitution as shown in

[18] where the predictions take the following form (using notation from [36]):

$$\mathbf{Y} = \mathbf{\Psi}\mathbf{x}(t) + \mathbf{\Theta}\Delta\mathbf{U} \tag{5.15}$$

where

$$\mathbf{Y} = \begin{bmatrix} y(t+1|t) \\ y(t+2|t) \\ y(t+3|t) \\ \vdots \\ y(t+N|t) \end{bmatrix}, \quad \mathbf{\Psi} = \begin{bmatrix} \bar{\mathbf{C}}\bar{\mathbf{A}} \\ \bar{\mathbf{C}}\bar{\mathbf{A}}^2 \\ \bar{\mathbf{C}}\bar{\mathbf{A}}^3 \\ \vdots \\ \bar{\mathbf{C}}\bar{\mathbf{A}}^N \end{bmatrix}, \quad \Delta\mathbf{U} = \begin{bmatrix} \Delta\mathbf{u}(t) \\ \Delta\mathbf{u}(t+1|t) \\ \Delta\mathbf{u}(t+2|t) \\ \vdots \\ \Delta\mathbf{u}(t+N_u-1|t) \end{bmatrix},$$

$$\mathbf{\Theta} = \begin{bmatrix} \bar{\mathbf{C}}\bar{\mathbf{B}} & 0 & 0 & \cdots & 0 \\ \bar{\mathbf{C}}\bar{\mathbf{A}}\bar{\mathbf{B}} & \bar{\mathbf{C}}\bar{\mathbf{B}} & 0 & \cdots & 0 \\ \bar{\mathbf{C}}\bar{\mathbf{A}}^2\bar{\mathbf{B}} & \bar{\mathbf{C}}\bar{\mathbf{A}}\bar{\mathbf{B}} & \bar{\mathbf{C}}\bar{\mathbf{B}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{C}}\bar{\mathbf{A}}^{N-1}\bar{\mathbf{B}} & \bar{\mathbf{C}}\bar{\mathbf{A}}^{N-2}\bar{\mathbf{B}} & \bar{\mathbf{C}}\bar{\mathbf{A}}^{N-3}\bar{\mathbf{B}} & \cdots & \bar{\mathbf{C}}\bar{\mathbf{A}}^{N-N_u}\bar{\mathbf{B}} \end{bmatrix}$$

Note that the predictions depend on the current states $\mathbf{x}(t)$ and future control actions $\Delta\mathbf{U}$. This is similar to the earlier DMC formulation, and is where the optimization occurs. As this is a model of the 3D errors, the goal of the optimization is to select $\Delta\mathbf{u}$ that minimizes predicted output, since the "setpoint" in this case is always zero. This can be considered a stabilization problem, allowing the use of a quadratic cost function with with Q and R weighting matrices (much like the LQR formulation):

$$J = \mathbf{Y}^T\mathbf{Q}\mathbf{Y} + \Delta\mathbf{U}^T\mathbf{R}\Delta\mathbf{U} \tag{5.16}$$

Expanding this by substituting equation 5.15 gives the following:

$$J = \mathbf{x}(t)^T\mathbf{\Psi}^T\mathbf{Q}\mathbf{\Psi}\mathbf{x}(t) + 2\mathbf{x}(t)^T\mathbf{\Psi}^T\mathbf{Q}\mathbf{\Theta}\Delta\mathbf{U} + \Delta\mathbf{U}^T(\mathbf{\Theta}^T\mathbf{Q}\mathbf{\Theta} + \mathbf{R})\Delta\mathbf{U} \tag{5.17}$$

Then defining $\mathbf{E} = \mathbf{\Theta}^T\mathbf{Q}\mathbf{\Theta} + \mathbf{R}$ and $\mathbf{F} = \mathbf{\Theta}^T\mathbf{Q}\mathbf{\Psi}\mathbf{x}(t)$ and dropping the term that is unaffected by control action, as it does not affect the minimization, the general form of the objective function is as follows:

$$\begin{aligned} &\min J = \tfrac{1}{2}\Delta\mathbf{U}^T\mathbf{E}\Delta\mathbf{U} + \mathbf{F}^T\Delta\mathbf{U} \\ &\text{s.t.} \mathbf{\Gamma}\Delta\mathbf{U} \leq \mathbf{G} \end{aligned} \tag{5.18}$$

The constraints on the control action $\mathbf{u}$ and control increment $\Delta\mathbf{u}$ are:

$$\begin{aligned} &\mathbf{u}_{\min} \leq \mathbf{u}(t+k) \leq \mathbf{u}_{\max}, \quad k = 0, 1, \cdots, N_u - 1 \\ &\Delta\mathbf{u}_{\min} \leq \Delta\mathbf{u}(t+k) \leq \Delta\mathbf{u}_{\max}, \quad k = 0, 1, \cdots, N_u - 1 \end{aligned} \tag{5.19}$$

and are represented in matrix form as the inequality constraint in equation 5.18. Details on how to arrange the matrix $\mathbf{\Gamma}$ and vector $\mathbf{G}$ can be found seen most clearly in [18] and more succinctly in [36]. The calculation for the desired pitch and yaw rates q_d and r_d to be sent to the second stage controller done by rearranging equations 5.11 resulting in the following:

$$\begin{bmatrix} q_d \\ r_d \end{bmatrix}_t = \begin{bmatrix} q_F \\ r_F \end{bmatrix}_t + \Delta\mathbf{u}(t) \tag{5.20}$$

where only the first control moves from $\Delta\mathbf{u}(t)$ are used. As this implementation is further restricted in that there is no speed control, an alternative method for evolving the speed of the virtual vehicle is necessary. This is similar in spirit to a method used in [37], and uses the following equation:

$$u_r = 1 + \tanh\left(\frac{x_e}{\sigma}\right) \tag{5.21}$$

where σ is a design variable to scale the magnitude of the virtual vehicle speed. This function ensures that the virtual vehicle speeds up when the error x_e is small, and slows down when the error grows. Using this speed, the path parameter increment

is calculated as $ds = u_r \cdot dt$ and the path parameter is updated using $s_t = s_{t-1} + ds$. Note that the path parameter s is independent of time, and evolves as a consequence of the AUV's proximity to the path. This is a primary feature of path following (time independent) algorithms over trajectory following (time dependent) algorithms.

In the second stage controller, using the desired variables calculated in the first stage of control, the SnMPC as formulated in chapter 3 is used. The full nonlinear model of the AUV is simulated over the control horizon chosen for the SnMPC. This has the advantage of having the most accurate predictions possible from which optimal control moves can be determined. From this simulation, the predicted pitch and yaw rates are compared with the desired pitch and yaw rates from the first stage of the controller to determine future errors. These future errors are used in separate pitch and yaw control law equations. These are then used to update the control action that is sent to the AUV system. In contrast with the first stage of control, the SnMPC uses unconstrained optimization, instead relying on the limits placed on control action sent to the AUV to avoid actuator saturation.

5.2.2 LOS Control

The hybrid guidance system will switch the active control system from the virtual vehicle method to the LOS method upon reaching a desired position and orientation error, determined as part of design. This stage will vastly reduce the computation required to control the AUV and allow for flexibility of continued control, by accepting either a continuation of the predefined path (as shown parameterized by s in the previous section) or accepting discrete waypoints. The details of both are shown here.

In the case of following a predefined path, a "waypoint" must be chosen at each time instant to calculate target pitch and yaw. This is generated in a similar way as the virtual vehicle control method, where a target on the path is chosen based on the

error x_e as it is calculated in the LOS section (in the earth-fixed frame) to evolve the virtual vehicle on the path. The equation governing this is the same as equation 5.21, calculated from a different reference; thus:

$$u_r = 1 - \tanh\left(\frac{x_e}{\sigma}\right) \tag{5.22}$$

this equation can be scaled if necessary (for different AUV speeds). This virtual speed u_r is used to calculate the path parameter increment ds, as before. In order to advance the waypoint on the path, the positional error vector $\mathbf{P}_e$ is calculated and using this the total distance from the AUV to the waypoint is determined. If this is outside of the design radius γ, this point on the path becomes the waypoint. If the error distance is within the design radius, the path parameter s is successively updated using ds until an acceptable waypoint is found.

The use of this control method using a set of waypoints is performed in the same way. Given an ordered set of waypoints, check positional error between the current waypoint and the AUV using $\mathbf{P}_e$, if this distance is outside of design radius γ, use this as the target waypoint, otherwise check the next waypoint.

Whichever method is employed, the target pitch and yaw angles θ_t and ψ_t are used to determine the control action using the SnMPC formulation. The AUV mathematical model is simulated over the control horizon and in slight contrast with the virtual vehicle method, the predicted pitch and yaw angles are sent to the pitch and yaw controllers along with the target values. Using this predicted error, the optimal control moves are calculated, used to update control actions, and sent to the AUV system as has been seen several times.

Chapter 6

Simulations

This chapter shows simulations building to the hybrid guidance method described in chapter 5. This includes control of the AUV in straight line, planar, and three dimensional complex paths.

As a note, all plots in this section are of the 3D path or waypoints followed by the AUV. This is displayed in a four-panel configuration as seen in chapter 4, where the waypoint or path are displayed in orange, while the path of the AUV is displayed in blue. Also included are AUV control input plots. All other plots including linear and angular displacements and velocities as well as AUV forces and moments for each simulation are omitted from the text for brevity. These plots are included in the Appendix A.

6.1 LOS Guidance Simulations

The first simulation shown displays the capability of the simplified waypoint guidance system controlled by the SnMPC to reach a single waypoint in the line of sight of the AUV:

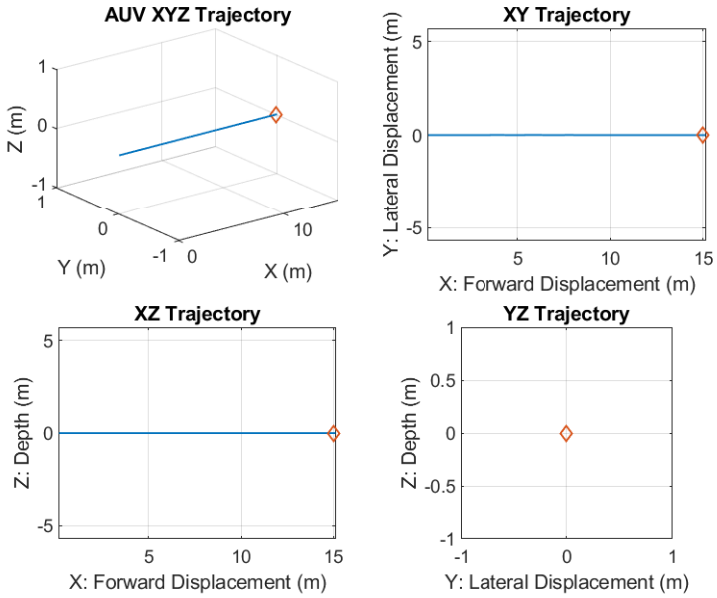

Figure 6.1: AUV Trajectory (Single Waypoint)

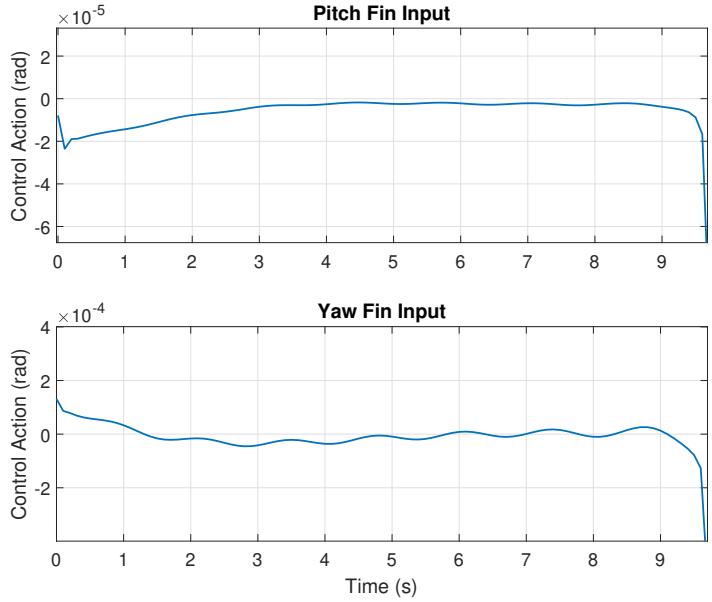

Figure 6.2: AUV Control Actions (Single Waypoint)

The 3D plot shown in figure 6.1 shows that when oriented towards a waypoint, the guidance and control systems can drive the AUV to the waypoint. The control inputs in figure 6.2 show some control chattering. In cases where the AUV is well oriented,

this is of very low amplitude, and thus not an issue. As will be seen when using the virtual vehicle guidance method, this can be well mitigated.

The next test case is that of a planar curve. This is performed using a series of waypoints with the simplified waypoint guidance system:

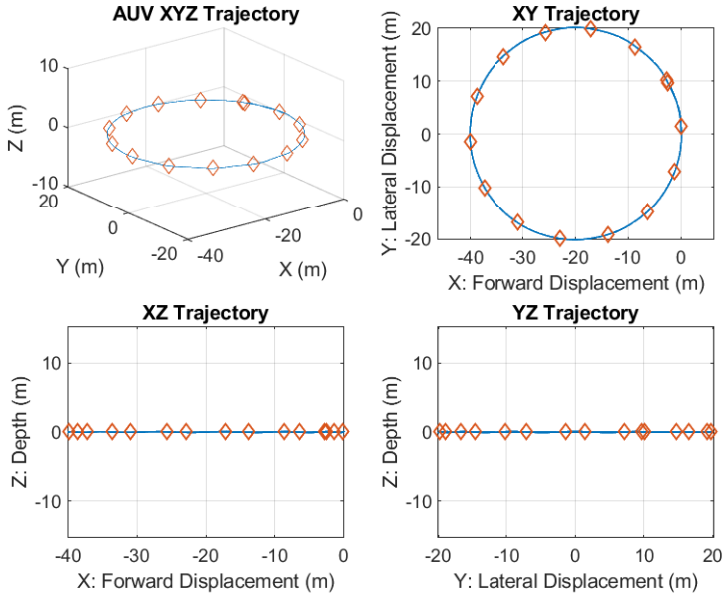

Figure 6.3: AUV Trajectory (Waypoint Planar Curve)

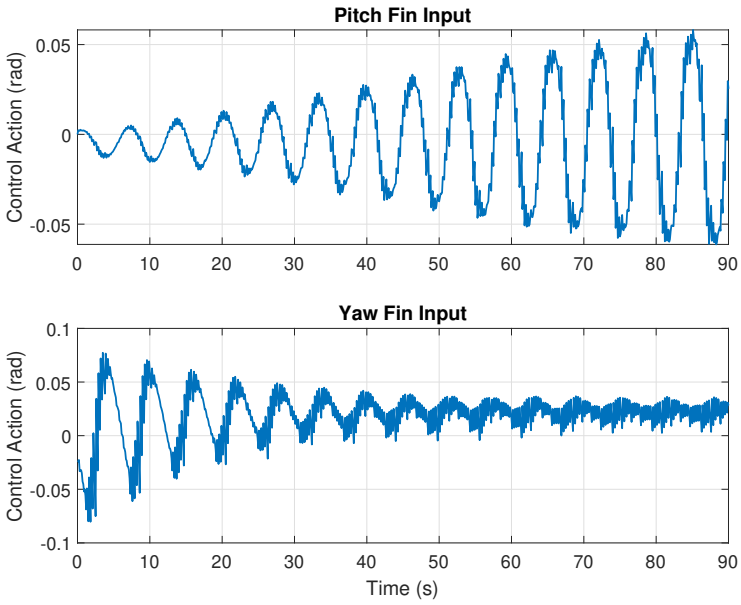

Figure 6.4: AUV Control Actions (Waypoint Planar Curve)

Figures 6.3 and 6.4 show a series of waypoints arranged in a circle on the XY-plane, starting at the origin, along with the plots of the control actions. The control system is able to track this series of waypoints well, maintaining a constant depth while adjusting the yaw fins to follow the waypoints. Note that only a subset of waypoints are shown in the plot, to reduce visual clutter.

The final simulation utilizing only the LOS guidance system is a descending spiral, an example of a complex 3D path:

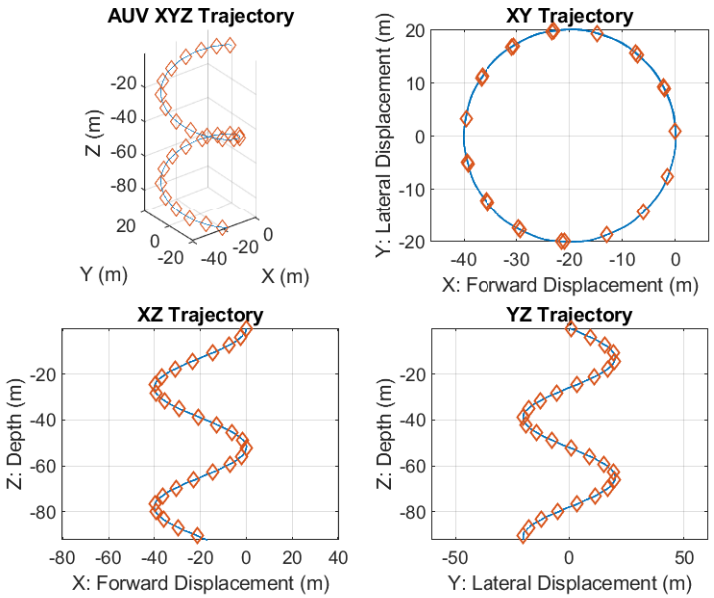

Figure 6.5: AUV Trajectory (Waypoint 3D Curve)

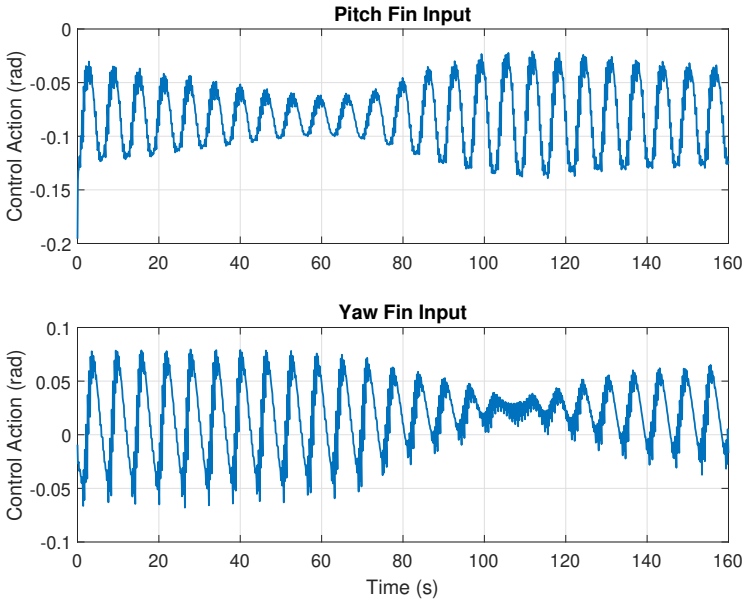

Figure 6.6: AUV Control Actions (Waypoint 3D Curve)

Figures 6.5 and 6.6 demonstrate the ability of the LOS guidance system using the SnMPC to achieve each successive waypoint as part of a complex path. It is important to emphasize that with this control system no knowledge of future or prior waypoints is required. This is a key differentiating factor between this system and more complex variants of LOS guidance or virtual vehicle methods. This is predicated upon the AUV being well-oriented in order to attain each waypoint. All three of these simulations have shown the AUV beginning at time $t = 0$ already oriented approximately in line with the first waypoint. In practice this is a severe constraint on usability.

6.2 Virtual Vehicle Guidance Simulation

The next simulation demonstrates the virtual vehicle guidance system converging to a straight line path in 3D space:

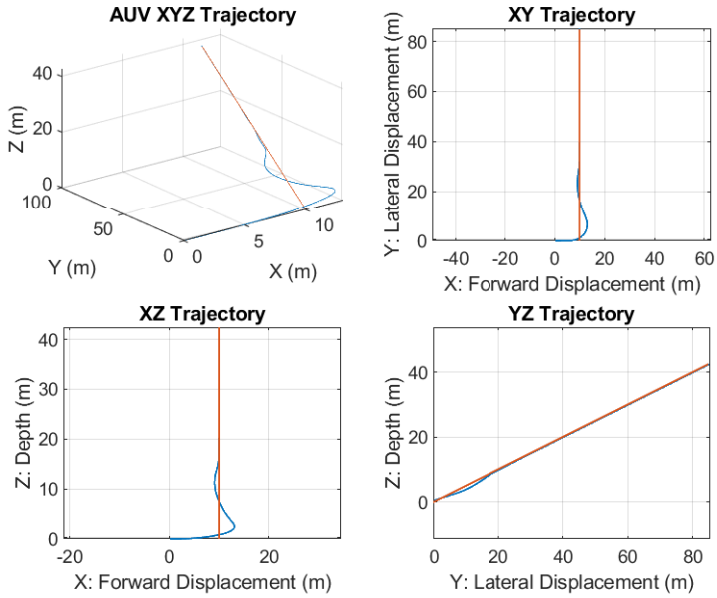

Figure 6.7: AUV Trajectory (Virtual Vehicle)

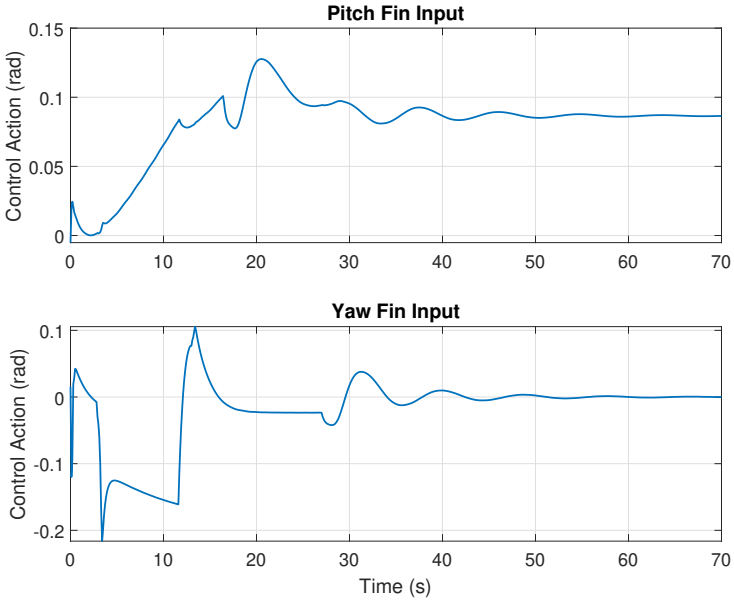

Figure 6.8: AUV Control Actions (Virtual Vehicle)

This simulation (figures 6.7 and 6.8) display the power of the virtual vehicle guidance system in converging to and following a path in 3D space. As thrust is constant in this AUV model, there is some overshoot present, however the controller is able to

overcome this effectively. While there is no control along several dimensions of the AUV, because of the prediction taking place in the second stage (SnMPC) of control using the full nonlinear model, the predictions used in the pitch and yaw control action calculations are highly accurate. This helps to account for the simplified AUV model used in formulating the error model of the path following problem in the first stage of control. Note that control actions for this simulation are much smoother than when using the LOS guidance system. As previously discussed, there are compromises to this guidance and control system; foremost among these being that some information of the path must be known, and a parameterized path is necessary, as well as far higher computation time as compared with the simplified LOS system presented here.

6.3 Hybrid Guidance Simulation

The hybrid guidance system discussed in chapter 5 is introduced in the final simulation below. The LOS guidance system has demonstrated the ability to follow a series of discrete waypoints (or waypoints selected from a parameterized path) in complex 3D configurations provided the AUV is well-oriented to the next waypoint. The virtual vehicle guidance system has shown the ability to converge to a path in 3D space. Given these, the following simulation demonstrates the effectiveness and flexibility of the hybrid guidance system:

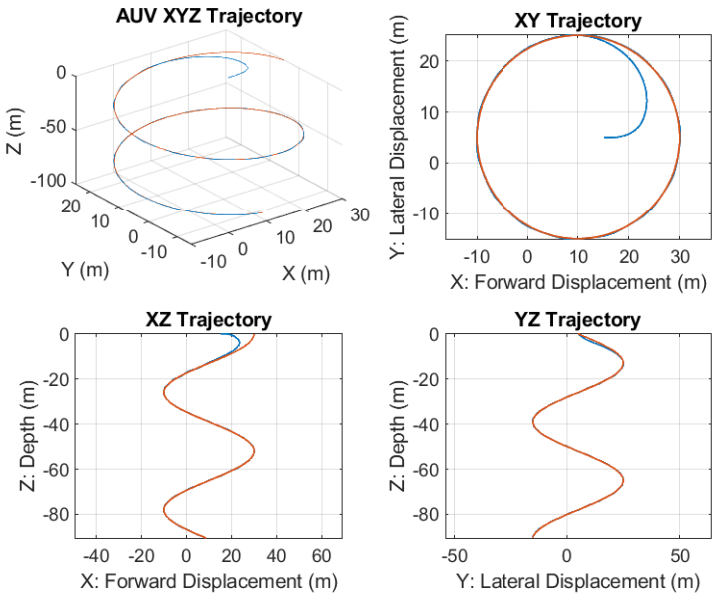

Figure 6.9: AUV Trajectory (Hybrid Guidance)

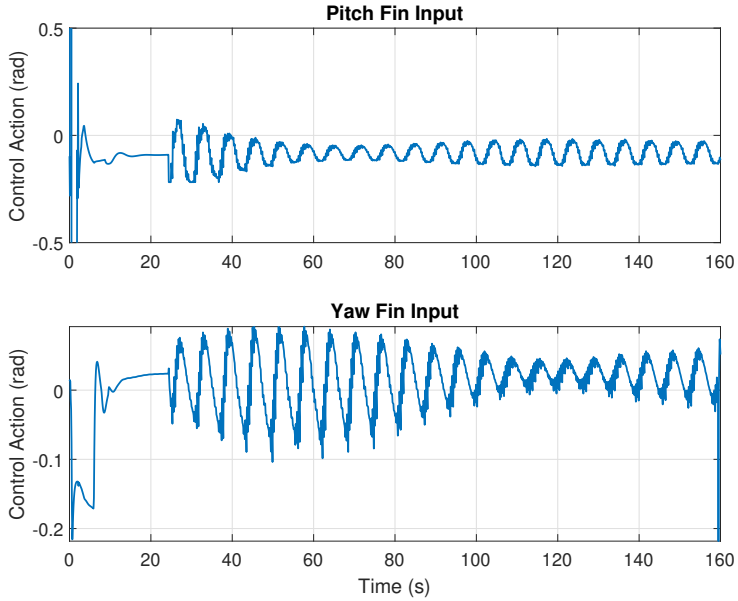

Figure 6.10: AUV Control Actions (Hybrid Guidance)

Figures 6.9 and 6.10 show the AUV starting off of a descending spiral path, converging to the path, and then maintaining the path over the allotted time. Figure 6.10 shows the fin inputs to the AUV, and makes clear where the guidance switching

takes place, after which the inputs begin to stutter. As previously mentioned, this stuttering is within the design constraints of the AUV model, and the amplitudes are relatively small. With this switch comes more efficient computation and the ability to accommodate different path inputs, a feature that would pair well with some form of path planning algorithm.

Chapter 7

Conclusions

This chapter concludes the research laid out in this thesis by: giving a brief summary of the work in this document, discussing the original contributions of this work, and discussing some ideas for future work based on this research.

7.1 Summary

The preceding work in this document has:

- Given a brief introduction to mathematical modeling of an Autonomous Underwater Vehicle (AUV) (specifically the REMUS 100).
- Described the mathematical basis for several predictive control algorithms and augmentations that can be made.
- Demonstrated simulations of the open-loop AUV as well as preliminary control algorithm simulations.
- Outlined several AUV guidance methods and the control variables and algorithms that accompany them.

- Shown simulation results of various guidance and control strategies on the full nonlinear AUV system.

This section serves to highlight the conclusions from this work, original contributions, and recommendations for expansions to this work.

7.2 Contributions

The primary goal of this research is to investigate the capability of a predictive control approach in controlling a highly nonlinear, highly coupled, non-minimum phase, underactuated system (the AUV). To this end, preliminary simulations are used to illustrate some of these system characteristics and the effectiveness of various predictive control strategies and modifications to improve performance.

The simulations in this chapter illustrate the effectiveness of the LOS, virtual vehicle, and hybrid guidance systems coupled with their respective error models and the SnMPC to follow waypoints and paths to varying degrees. The capabilities and limitations of these systems are discussed and detailed. The simulations in the LOS guidance section make clear that the AUV can follow a series of waypoints in 3D configuration or a parameterized path. The virtual vehicle guidance section displays the ability of the AUV to converge to and follow a parameterized path. The hybrid guidance system shows the two guidance systems working in concert to first converge to the path using a virtual vehicle model, and then hand off control to the LOS system, offering greater flexibility in future control.

The following are the contributions of this work, itemized:

- Demonstrate the effectiveness of predictive control strategies and modifications and extend these to a highly complex system.

- Investigate the viability of predictive control methods as applied to an autonomous underwater vehicle.

- Introduce a unique hybrid guidance system to leverage the best features of existing AUV guidance systems.

7.3 Future Work

This section outlines several recommendations for future work using this research as a basis.

7.3.1 LOS Guidance Improvement

As discussed in chapter 5, the LOS guidance system utilized here is simplified, and will fail at a sufficiently high error. This can be mitigated in the case of a parameterized path as an input with the hybrid switching system, however may cause issue given a single waypoint (or waypoints in series) as an input. There have been attempts at more advanced waypoint guidance systems [33] seeking to overcome some of these challenges, which could be explored as part of the LOS guidance system in the hybrid approach.

7.3.2 Controller Robustness and Model Flexibility

The AUV mathematical model used in this research has been experimentally verified [24] under certain operating conditions. As has been discussed, in order to keep the model as accurate as possible, a constant thrust is used in all simulations in this work. The robustness of the controller to handle changing thrust, or more extreme conditions could be investigated in future research. This could also include the presence of underwater currents or other disturbances which would be present in most applications.

The investigation of the robustness of the guidance and control systems and possible enhancements may also allow for the simplification of the model used for predictions

in the controller, leading to more efficient computation. This area of research could also allow for the possibility of some flexibility in the model used, from other similar AUV models to a data based model.

7.3.3 Experimental Validation

Experimental validation of the guidance and control systems described here is crucial to the application of these methods to real-world AUV systems. As all conclusions here are in relation to simulated systems, such research would illuminate possible sources of error, and avenues for improvement that may otherwise be difficult to predict.

References

[1] B. S. Halpern, C. Longo, D. Hardy, K. L. McLeod, J. F. Samhouri, S. K. Katona, K. Kleisner, S. E. Lester, J. O'Leary, M. Ranelletti, A. A. Rosenberg, C. Scarborough, E. R. Selig, B. D. Best, D. R. Brumbaugh, F. S. Chapin, L. B. Crowder, K. L. Daly, S. C. Doney, C. Elfes, M. J. Fogarty, S. D. Gaines, K. I. Jacobsen, L. B. Karrer, H. M. Leslie, E. Neeley, D. Pauly, S. Polasky, B. Ris, K. St Martin, G. S. Stone, U. R. Sumaila, and D. Zeller, "An index to assess the health and benefits of the global ocean," *Nature*, vol. 488, pp. 615–620, Aug. 2012. Number: 7413 Publisher: Nature Publishing Group.

[2] A. Knap, E. Dewailly, C. Furgal, J. Galvin, D. Baden, R. E. Bowen, M. Depledge, L. Duguay, L. E. Fleming, T. Ford, F. Moser, R. Owen, W. A. Suk, and U. Unluata, "Indicators of ocean health and human health: developing a research and monitoring framework.," *Environmental Health Perspectives*, vol. 110, pp. 839–845, Sept. 2002.

[3] J. Carstensen, "Need for monitoring and maintaining sustainable marine ecosystem services," *Frontiers in Marine Science*, vol. 1, 2014. Publisher: Frontiers.

[4] M. Kubat, R. C. Holte, and S. Matwin, "Machine Learning for the Detection of Oil Spills in Satellite Radar Images," *Machine Learning*, vol. 30, pp. 195–215, Feb. 1998.

[5] N. O. Handegard, L. d. Buisson, P. Brehmer, S. J. Chalmers, A. D. Robertis, G. Huse, R. Kloser, G. Macaulay, O. Maury, P. H. Ressler, N. C. Stenseth, and O. R. Godø, "Towards an acoustic-based coupled observation and modelling system for monitoring and predicting ecosystem dynamics of the open ocean," *Fish and Fisheries*, vol. 14, no. 4, pp. 605–615, 2013.

[6] R. B. Wynn, V. A. I. Huvenne, T. P. Le Bas, B. J. Murton, D. P. Connelly, B. J. Bett, H. A. Ruhl, K. J. Morris, J. Peakall, D. R. Parsons, E. J. Sumner, S. E. Darby, R. M. Dorrell, and J. E. Hunt, "Autonomous Underwater Vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience," *Marine Geology*, vol. 352, pp. 451–468, June 2014.

[7] J. W. Nicholson and A. J. Healey, "The Present State of Autonomous Underwater Vehicle (AUV) Applications and Technologies," *Marine Technology Society Journal*, vol. 42, no. 1, pp. 44–51, 2008.

[8] "Explorers to use new robotic vehicles to hunt for life and hydrothermal vents on Arctic seafloor : 2007 Arctic Seafloor Expedition : Woods Hole Oceanographic Institution (WHOI)."

[9] R. Tuttle, "Visionary Counting on Unmanned Vehicles," p. 5, 2007.

[10] C. v. Alt, "REMUS 100 transportable mine countermeasure package," in *Oceans 2003. Celebrating the Past ... Teaming Toward the Future (IEEE Cat. No.03CH37492)*, vol. 4, pp. 1925–1930 Vol.4, Sept. 2003.

[11] D. Yoerger and J. Slotine, "Robust trajectory control of underwater vehicles," *IEEE Journal of Oceanic Engineering*, vol. 10, pp. 462–470, Oct. 1985.

[12] C. Shen, Y. Shi, and B. Buckham, "Trajectory Tracking Control of an Autonomous Underwater Vehicle Using Lyapunov-Based Model Predictive Con-

trol," *IEEE Transactions on Industrial Electronics*, vol. 65, pp. 5796–5805, July 2018. Conference Name: IEEE Transactions on Industrial Electronics.

[13] R. Tedrake, *Underactuated Robotics: Learning, Planning, and Control for Efficient and Agile Machines*. 2009.

[14] L. Desborough and R. Miller, "Increasing Customer Value of Industrial Control Performance Monitoring—Honeywell's Experience," p. 22, 2002.

[15] J. Wilson, M. Charest, and R. Dubay, "Non-linear model predictive control schemes with application on a 2 link vertical robot manipulator," *Robotics and Computer-Integrated Manufacturing*, vol. 41, pp. 23–30, Oct. 2016.

[16] M. H. Khodayari and S. Balochian, "Modeling and control of autonomous underwater vehicle (AUV) in heading and depth attitude via self-adaptive fuzzy PID controller," *Journal of Marine Science and Technology*, vol. 20, pp. 559–578, Sept. 2015.

[17] J. Zhao, L. Han, L. Wang, and Z. Yu, "The fuzzy PID control optimized by genetic algorithm for trajectory tracking of robot arm," in *2016 12th World Congress on Intelligent Control and Automation (WCICA)*, pp. 556–559, June 2016.

[18] E. Camacho and C. Bordons, *Model Predictive Control*. Springer, 2nd ed., 2007.

[19] I. D. Landau, R. Lozano, and M. M'Saad, "Introduction to Adaptive Control," in *Adaptive Control* (I. D. Landau, R. Lozano, and M. M'Saad, eds.), Communications and Control Engineering, pp. 1–30, London: Springer, 1998.

[20] G. P. Harnwell, "Submarine Physics," *American Journal of Physics*, vol. 16, pp. 127–139, Mar. 1948.

[21] A. Sahoo, S. K. Dwivedy, and P. S. Robi, "Advancements in the field of autonomous underwater vehicle," *Ocean Engineering*, vol. 181, pp. 145–160, June 2019.

[22] X. Xiang, C. Yu, and Q. Zhang, "Robust fuzzy 3D path following for autonomous underwater vehicle subject to uncertainties," *Computers & Operations Research*, vol. 84, pp. 165–177, Aug. 2017.

[23] Y. Li, C. Wei, Q. Wu, P. Chen, Y. Jiang, and Y. Li, "Study of 3 dimension trajectory tracking of underactuated autonomous underwater vehicle," *Ocean Engineering*, vol. 105, pp. 270–274, Sept. 2015.

[24] T. Prestero, *Verification of a six-degree of freedom simulation model for the REMUS autonomous underwater vehicle*. Woods Hole, MA: Massachusetts Institute of Technology and Woods Hole Oceanographic Institution, 2001.

[25] J. H. A. M. Vervoort, "Modeling and Control of an Unmanned Underwater Vehicle," 2008.

[26] T. I. Fossen and O.-E. Fjellstad, "Nonlinear modelling of marine vehicles in 6 degrees of freedom," *Mathematical Modelling of Systems*, vol. 1, pp. 17–27, Jan. 1995.

[27] C. Wang, F. Zhang, and D. Schaefer, "Dynamic modeling of an autonomous underwater vehicle," *Journal of Marine Science and Technology*, vol. 20, pp. 199–212, June 2015.

[28] D. Soloway and P. Haley, "Neural generalized predictive control," in *Proceedings of the 1996 IEEE International Symposium on Intelligent Control*, pp. 277–282, Sept. 1996. ISSN: 2158-9860.

[29] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control—Part I. The basic algorithm," *Automatica*, vol. 23, pp. 137–148, Mar. 1987.

[30] M. Charest and R. Dubay, "MPC enhancement for tracking of complex profiles — The basic technique," *Control Engineering Practice*, vol. 33, pp. 136–147, Dec. 2014.

[31] S. Brunton and J. N. Kutz, *Data Driven Science and Engineering*. 2017.

[32] M. Abu-Ayyad and R. Dubay, "Improving the Performance of Generalized Predictive Control for Nonlinear Processes," *Industrial & Engineering Chemistry Research*, vol. 49, pp. 4809–4816, May 2010. Publisher: American Chemical Society.

[33] M. Breivik and T. Fossen, "Guidance-based path following for autonomous underwater vehicles," in *Proceedings of OCEANS 2005 MTS/IEEE*, pp. 2807–2814 Vol. 3, Sept. 2005. ISSN: 0197-7385.

[34] S.-R. Oh and J. Sun, "Path following of underactuated marine surface vessels using line-of-sight based model predictive control," *Ocean Engineering*, vol. 37, pp. 289–295, Feb. 2010.

[35] L. Lapierre and D. Soetanto, "Nonlinear path-following control of an AUV," *Ocean Engineering*, vol. 34, pp. 1734–1744, Aug. 2007.

[36] X. Yao, X. Wang, L. Zhang, and X. Jiang, "Model predictive and adaptive neural sliding mode control for three-dimensional path following of autonomous underwater vehicle with input saturation," *Neural Computing and Applications*, vol. 32, Nov. 2020.

[37] X. Liang, X. Qu, L. Wan, and Q. Ma, "Three-Dimensional Path Following of an Underactuated AUV Based on Fuzzy Backstepping Sliding Mode Control," *International Journal of Fuzzy Systems*, vol. 20, pp. 640–649, Feb. 2018.

Appendix A

Additional AUV Plots

A.1 LOS Guidance (Single Waypoint)

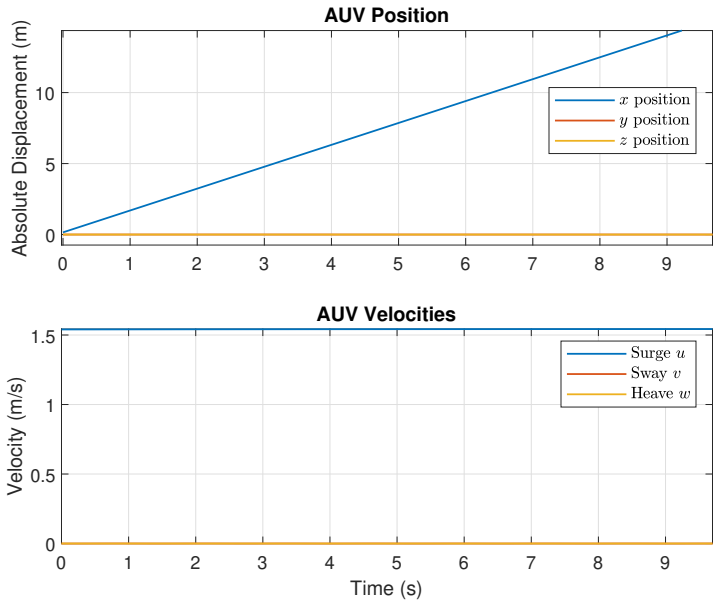

Figure A.1: AUV Linear Displacements and Velocities (Single Waypoint)

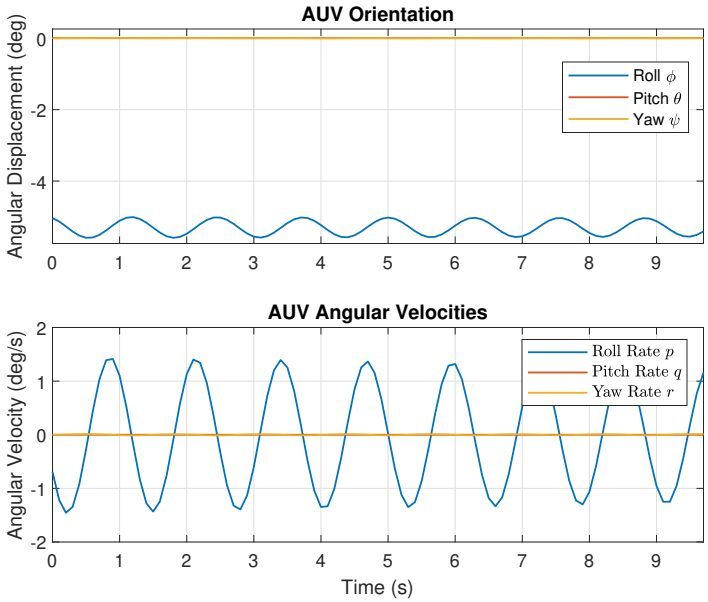

Figure A.2: AUV Angular Displacements and Velocities (Single Waypoint)

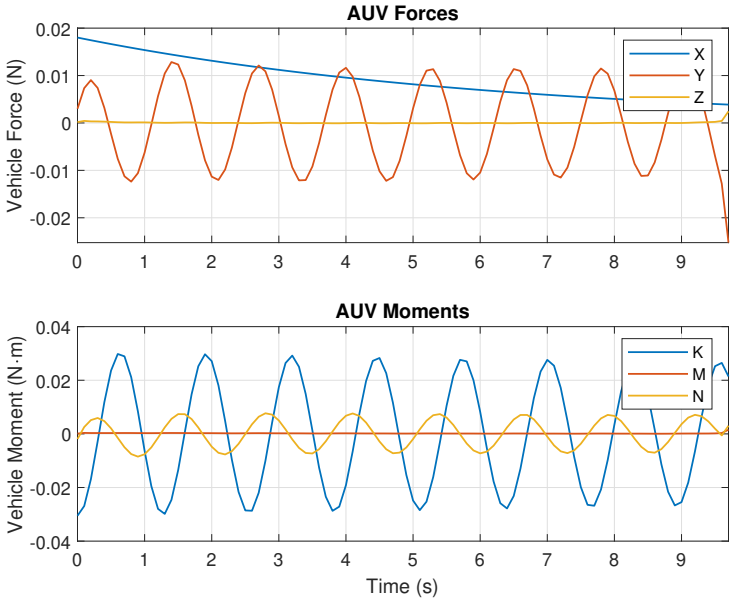

Figure A.3: AUV Forces and Moments (Single Waypoint)

A.2 LOS Guidance (Waypoint Planar Curve)

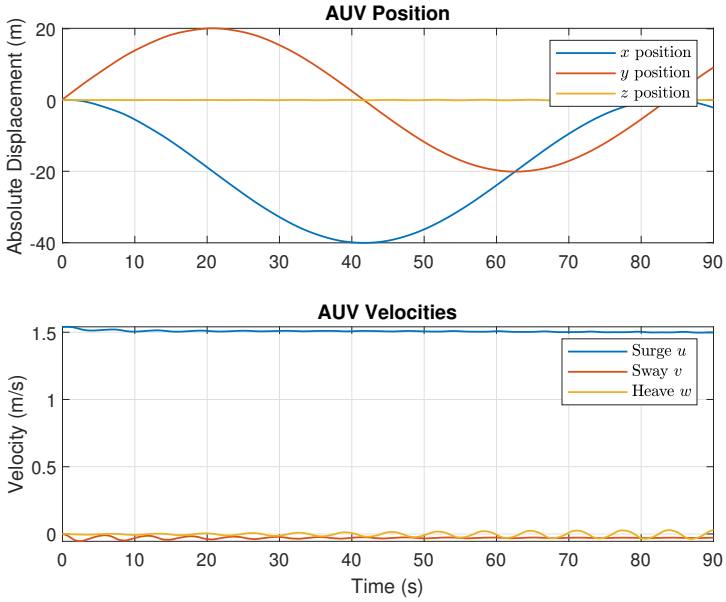

Figure A.4: AUV Linear Displacements and Velocities (Waypoint Planar Curve)

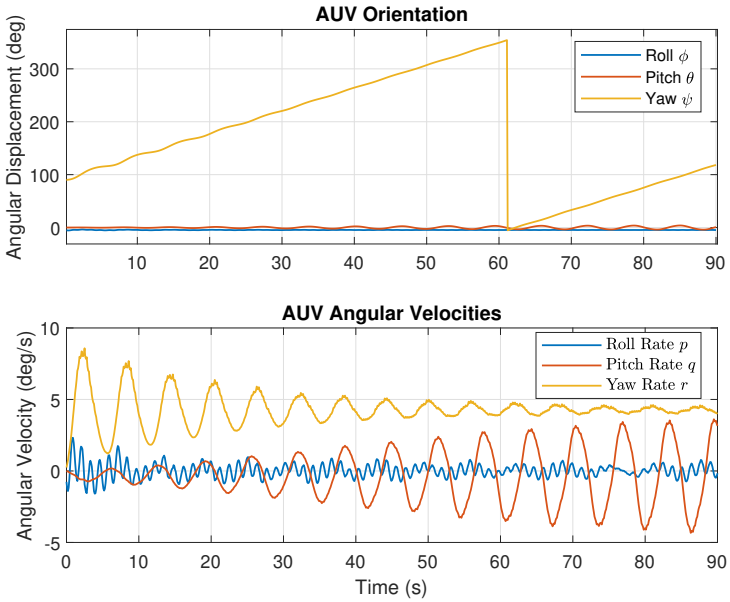

Figure A.5: AUV Angular Displacements and Velocities (Waypoint Planar Curve)

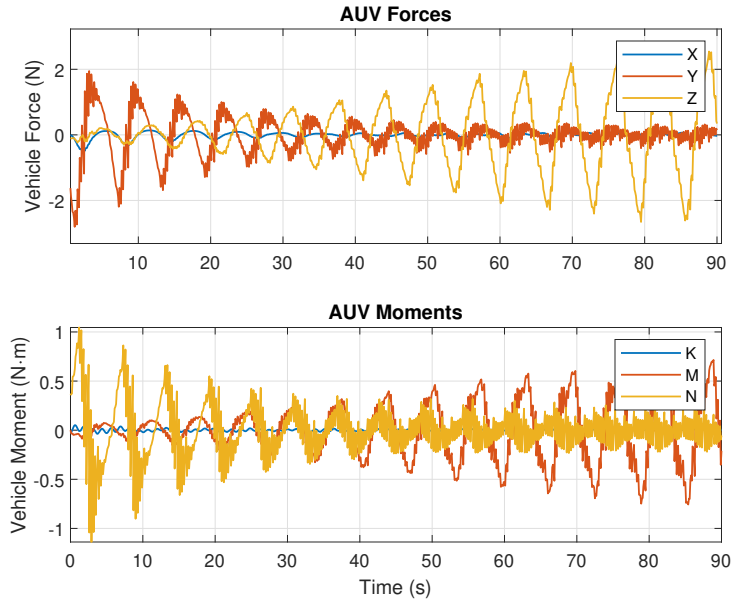

Figure A.6: AUV Forces and Moments (Waypoint Planar Curve)

A.3 LOS Guidance (Waypoint Spiral)

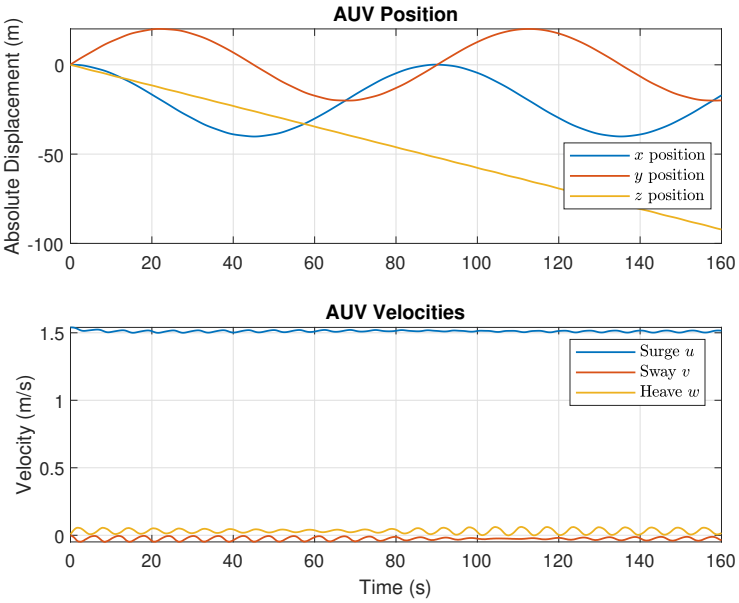

Figure A.7: AUV Linear Displacements and Velocities (Waypoint 3D Curve)

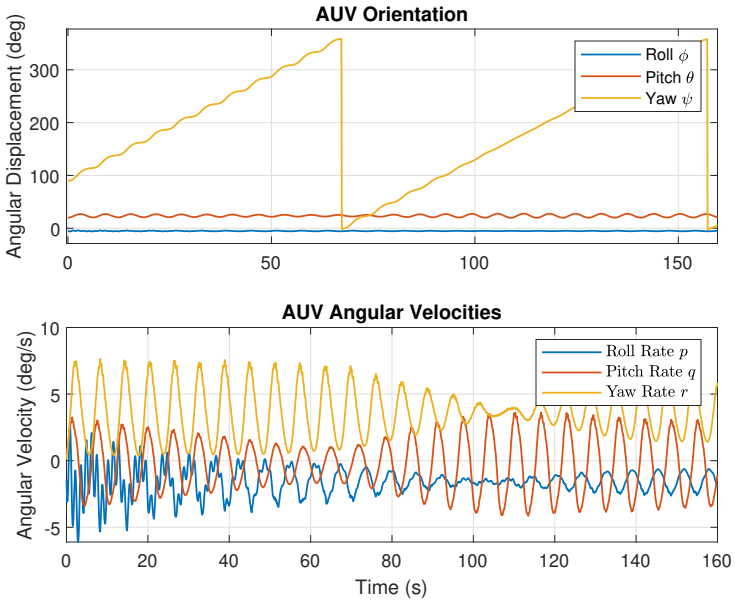

Figure A.8: AUV Angular Displacements and Velocities (Waypoint 3D Curve)

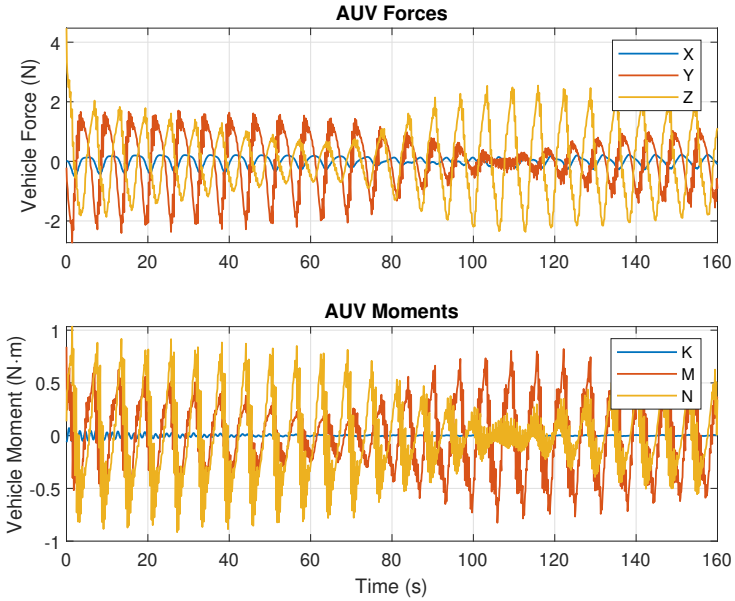

Figure A.9: AUV Forces and Moments (Waypoint 3D Curve)

A.4 Virtual Vehicle Guidance

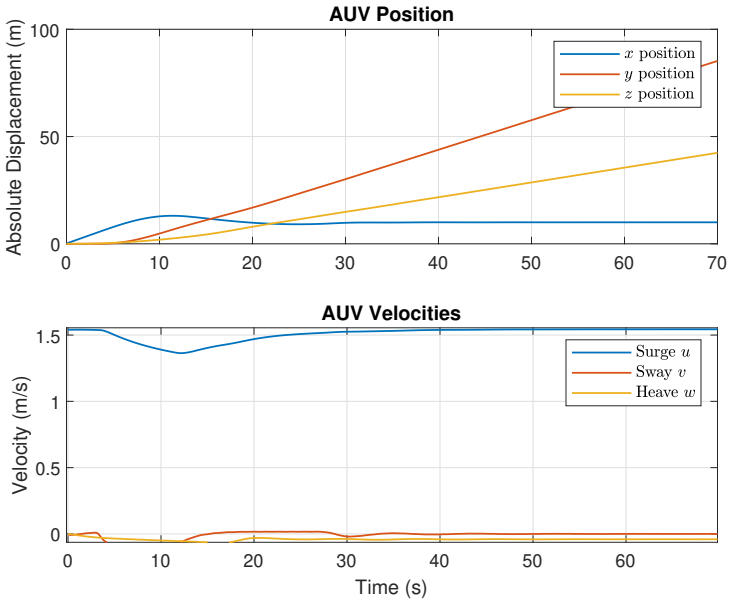

Figure A.10: AUV Linear Displacements and Velocities (Virtual Vehicle)

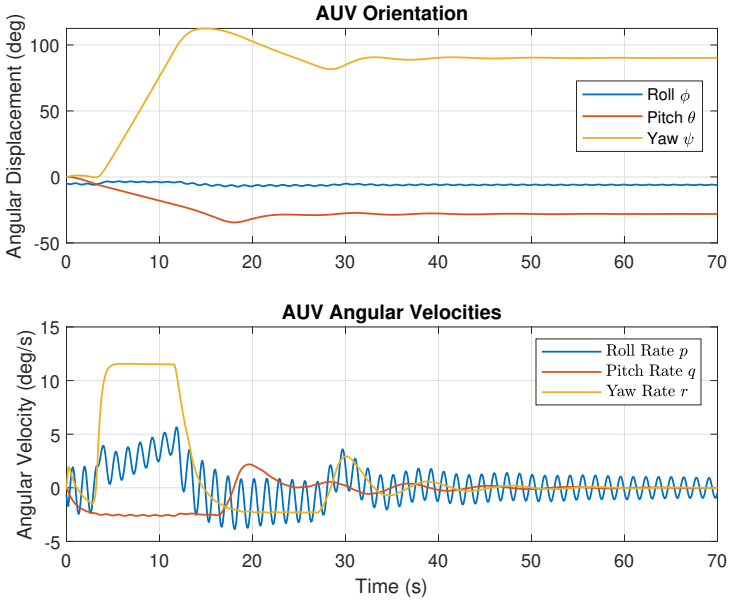

Figure A.11: AUV Angular Displacements and Velocities (Virtual Vehicle)

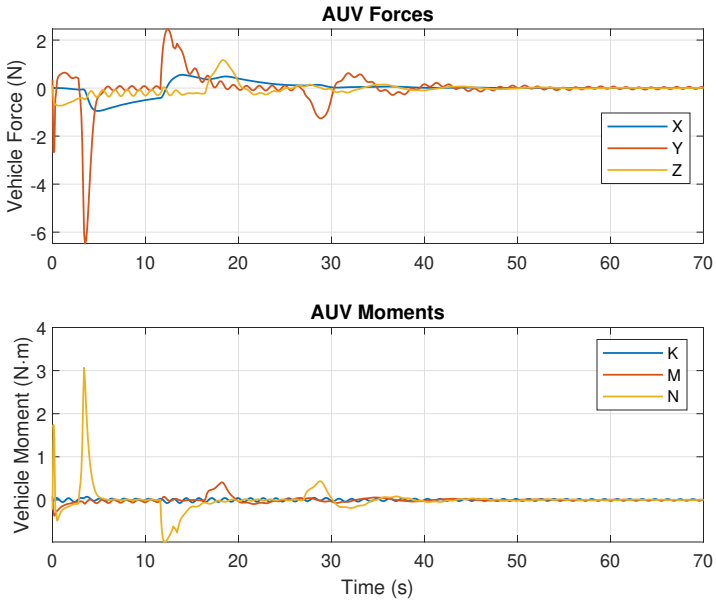

Figure A.12: AUV Forces and Moments (Virtual Vehicle)

A.5 Hybrid Guidance

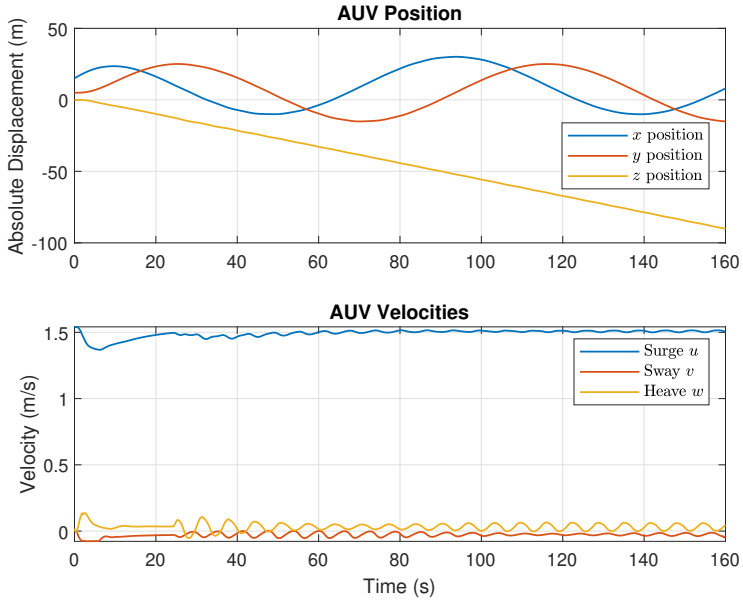

Figure A.13: AUV Linear Displacements and Velocities (Hybrid Guidance)

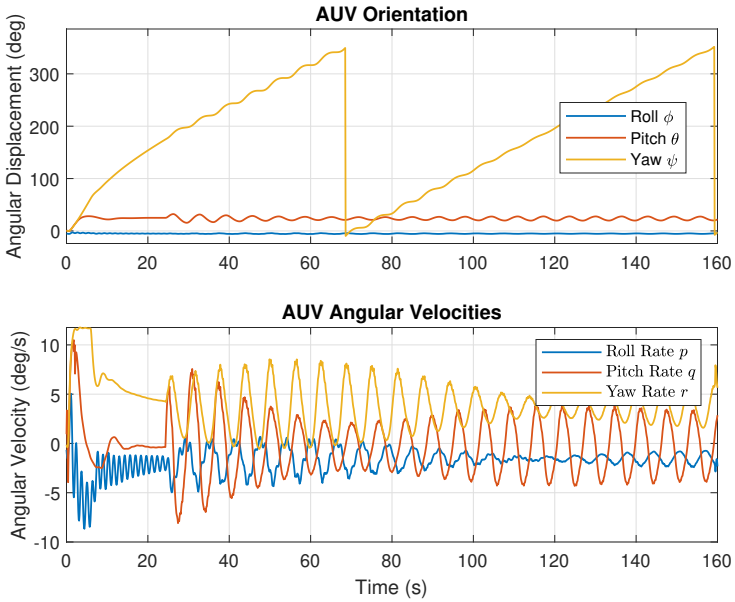

Figure A.14: AUV Angular Displacements and Velocities (Hybrid Guidance)

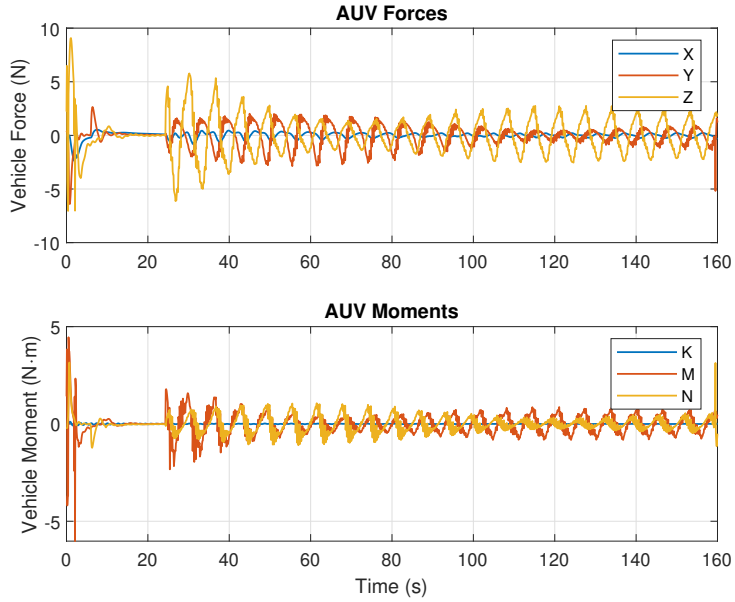

Figure A.15: AUV Forces and Moments (Hybrid Guidance)

CURRICULUM VITAE

Candidate's full name: Brad Chevarie

Universities attended:

BScE. Mechanical Engineering, September 2012 – May 2017
University of New Brunswick
First Division, Dean's List
Diploma in Technology, Management, and Entrepreneurship (TME)

MScE. Mechanical Engineering, September 2019 – June 2022
University of New Brunswick
Thesis: An Investigation into Advanced Predictive Control Methodologies for the REMUS AUV

Publications: None

Conference Papers:

A Survey of Advanced Dynamic Matrix Control Algorithms for Improved Performance, CSME 2020