GUIDANCE AND CONTROL OF MARINE CRAFT: AN ADAPTIVE DYNAMIC PROGRAMMING APPROACH

By

PATRICK S. WALTERS

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2015

To my parents Carl and Susan Walters, and my sisters Heather and Lindsay for their unwavering support and encouragement

ACKNOWLEDGMENTS

I would like to express sincere gratitude to my advisor, Dr. Warren E. Dixon, whose support and encouragement have been vital to my academic success. As an advisor, he has provided me with guidance in my research and support in developing my ideas. As a mentor, he has helped me to develop professionally, teaching me lessons that will be invaluable in future pursuits. I would also like to extend my gratitude to my committee members Dr. Carl Crane, Dr. Prabir Barooah, Dr. William Hager, and Dr. Eric M. Schwartz from whom I have drawn considerable knowledge and inspiration. I would also like to thank my colleagues at the University of Florida for their encouragement and criticism, which helped shape the ideas of this dissertation. I acknowledge that this dissertation would not have been possible without the support and encouragement provided by my family and friends.

TABLE OF CONTENTS

page

CHAPTER

APPENDIX

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADP	adaptive dynamic programming
AUV	autonomous underwater vehicle
CL	concurrent learning
DOF	degree-of-freedom
DVL	Doppler velocity log
HJB	Hamiliton-Jacobi-Bellman
LP	linear-in-the-parameters
MPC	model predictive control
NN	neural network
RL	reinforcement learning
StaF	state following

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

GUIDANCE AND CONTROL OF MARINE CRAFT: AN ADAPTIVE DYNAMIC PROGRAMMING APPROACH

By

Patrick S. Walters

May 2015

Chair: Warren E. Dixon
Major: Mechanical Engineering

Advances in sensing and computational capabilities have enabled autonomous vehicles to become vital assets across multiple disciplines. These improved capabilities have led to increased interest in autonomous marine craft. As the technologies of these vehicles mature, there is a desire to improve the performance of their motion control systems so that these vehicles can better achieve their mission objectives. Path planning, station keeping, and path following are critical control objectives of marine craft that enable autonomous docking, surveying, etc. Improving the performance of these control objectives for marine craft directly correspond to improved range, endurance, accuracy, and robustness in different environmental conditions.

Model-based adaptive dynamic programming has also seen considerable attention in the last decade, as a method of generating approximate optimal policies for classes of general uncertain nonlinear systems. Recent advances have made model-based adaptive dynamic programming a viable option for the control of uncertain complex systems such as marine craft. These recent advances motivate the exploration of model-based adaptive dynamic programming, as a means of improvement for motion control systems of autonomous marine craft.

This dissertation focuses on the application of adaptive dynamic programming to the motion control of marine craft. Specifically, Chapter 1 provides motivation for the application of model-based adaptive dynamic programming to control objectives

commonly faced by marine craft. Chapter 2 introduces the state of the art in model-based adaptive dynamic programming, which enables the results throughout this body of work. Chapter 3 presents the development of approximate optimal station keeping for a marine craft in the presents of a time-varying irrotational current, where the hydrodynamic drift dynamics are assumed to be unknown. The developed strategy is validated by experiments on an autonomous underwater vehicle. Chapter 4 presents approximate optimal path following of an arbitrarily parametrized two-dimensional path achieved by optimally tracking a virtual target placed on the desired path. As a kinematic analog to a marine craft, the developed strategy is validated by experiments on a wheeled mobile robot. Chapter 5 details an approximate optimal path planner that respects input (e.g., actuator saturation) and state (e.g., obstacles) constraints. The developed planner tackles the challenges associated with avoiding obstacles not known a priori and optimally re-planning in real-time to avoid collisions.

CHAPTER 1
INTRODUCTION

1.1 Motivation

Marine craft, which include ships, floating platforms, autonomous underwater vehicles (AUVs), autonomous surface vehicle, etc, play an vital role in commercial, military, and recreational activities. As the technologies of marine craft mature, there is a desire to improve the performance of marine craft (e.g., range, endurance, operation in a wider range of operating conditions), improving their ability to achieve mission objectives [1]. This increased interest has drawn considerable attention to motion control systems of marine craft over the last few decades. Motion control systems of marine craft are typically represented as three interconnected blocks denoted as guidance, navigation, and control [2]. The roles of the subsystems are included in the following, where the guidance and control subsystems are the focus of the work in this dissertation.

- Navigation is the process of determining a vehicle's position and orientation, and if necessary velocity and acceleration. The navigation subsystems utilize on-board sensors, such as a global navigation satellite system and inertial motion sensors, to determine vehicle state information.

- Guidance is the process of determining a reference position, velocity, and acceleration for the vehicle. An advanced guidance subsystem can compute an optimal trajectory or path while satisfying secondary objectives such as minimizing fuel consumption, minimizing transit time, and avoiding collisions. Advanced guidance subsystems may use navigation and weather data to compute these reference signals, while basic open-loop guidance systems may only use known vehicle model information.

- Control is the process of determining the required force and moment input to enable the vehicle to follow the reference trajectory. The control algorithm is constructed of feedback laws using navigation data and can exploit information from the guidance and navigation subsystems and other sensors (e.g. wind and current sensors) to incorporate feedforward laws.

1.2 Literature Review

Marine craft are often required to remain on a station for an extended period of time, e.g., floating oil platforms, support vessels, and AUVs acting as a communication link for multiple vehicles or persistent environmental monitors. The success of the vehicle often relies on the vehicle's ability to hold a precise station (e.g., station keeping near structures or underwater features). The cost of holding that station is correlated to the energy expended for propulsion through consumption of fuel and wear on mechanical systems, especially when station keeping in environments with a persistent current. Therefore, by reducing the energy expended for station keeping objectives, the cost of holding a station can be reduced.

Precise station keeping of a marine craft is challenging because of nonlinearities in the dynamics of the vehicle. A survey of station keeping for surface vessels can be found in [3]. Common approaches employed to control a marine craft include robust and adaptive control methods [4–7]. These methods provide robustness to disturbances and/or model uncertainty; however, they do not explicitly account for the cost of the control effort. Motivated by the desire to balance energy expenditure and the accuracy of the vehicle's station, approximate optimal control methods are examined to minimize a user defined cost function of the control effort (energy expended) and state error (station accuracy). Because of the difficulties associated with finding closed-form analytical solutions to optimal control problems for marine craft, efforts such as [8] numerically approximate the solution to the Hamilton-Jacobi-Bellman (HJB) equation using an iterative application of Galerkin's method, and efforts in [9] implement a model predictive control (MPC) policy.

Mission objectives of a marine craft may also require the vehicle to navigate a cluttered environment or accurately execute a search pattern. Similar to station keeping, vehicle endurance could determine the difference between a success and failure. Path following control is ideal for applications intolerant of spatial error. Path following heuristically yields smoother convergence to a desired path and reduces the risk of control saturation (cf. [10–12]). A path following control structure can also alleviate difficulties in the control of nonholonomic vehicles (cf. [11] and [13]).

Optimal control techniques have been applied to path following to improve path following performance. Nonlinear MPC is used in [14] to develop an optimal path following controller over a finite time horizon. Dynamic programming was applied to the path following problem in [15] where an approximation of the value function is computed offline to implement an approximate optimal feedback path following control law. The survey in [16] cites additional examples of MPC, linear quadratic regulation, and dynamic programming controllers applied to the path following problem.

In this dissertation, we pose station keeping and path following as optimal control problems. Solving optimal control problems is difficult, especially when the system dynamics are complex and may be uncertain as is the case with many marine craft. Adaptive dynamic programming (ADP) is one method that can be used to generate an approximate optimal solution to problems with uncertain nonlinear dynamics. An optimal control problem may be characterized by the HJB equation. ADP approximates the solution to the HJB equation using parametric function approximation techniques. ADP-based techniques have been used to approximate optimal control policies for point regulation (e.g., [17–21]) and trajectory tracking (e.g., [22–25]) of general nonlinear systems. An introduction to model-based ADP is presented in Chapter 2.

In addition to station keeping and path following, the route that a marine craft selects effects certain performance criteria, e.g., reducing the energy required to travel to an objective has a direct improvement on the marine craft's overall range and endurance. This motivates the investigation of optimal paths for marine craft.

Path planning approaches can be divided into two types, pregenerative and reactive [26]. Pregenerative methods compute a path before a mission begins (c.f [26–29]), while reactive methods determine a path as the marine craft progresses through its environment. From an optimality perspective, a reactive method (feedback motion planner) that is optimal has the advantage of generating a policy that provides optimal feedback even if the vehicle is forced off its original path, where a pregenerative method would require the craft to take a non-optimal trajectory to return to the original optimal path.

In developing an optimal path for marine craft, it is often necessary to consider the vehicle's dynamics. In general, it is difficult to develop optimal path planing strategies for nonlinear dynamics. One method of dealing with the challenges of path planning under differential constraints is to pose the problem as an optimal control problem. The corresponding HJB equation can be numerically approximated to produce feedback policies. Dynamic programming has been used as a feedback motion planner to compute an approximate optimal path through value iteration in results such as [30]. However, similar to pregenerative graph search methods (e.g., A*, Dijkstra), difficulties arise related to the state discretization as the order of the dynamics increase [31]. In results such as [32, 33], feedback-based path planning is generated offline by solving the HJB equation numerically. In the event of a change in the environment, such results would be required to recalculate a new approximate optimal plan offline.

Further complicating the task of optimal path planning, state constraints (e.g., obstacles) are often present en route to an objective. Marine craft are not able to sense all obstacles a priori, e.g., obstacles may remain undiscovered until they fall within a given sensing range. A recent advance in ADP bases the parametric approximation of the solution to the HJB equation on state following (StaF) kernels [34]. These StaF kernels yield a local approximation of the HJB equation around the current state. By only utilizing information near the current state to approximate the solution, StaF does not require knowledge of obstacles outside an approximation window.

In addition to state constraints, input constraints inherent to the marine craft (e.g., maximum speed) are also important to consider. Results such as [20, 21, 35] have considered input constraints within the ADP framework. Utilizing a generalized non-quadratic local cost [36], the results in [20, 21, 35] yield a bounded approximate optimal controller. As with path following and station keeping, we are motivated to utilize ADP because of the method's ability to approximate the solution of the HJB equation of general nonlinear systems online with parametric function approximation techniques. We are further motivated to leverage the concepts in [20, 34, 35] to help address challenges introduced by the unknown obstacles and input constraints.

1.3 Contributions

The contributions of Chapters 3-5 are indicated in the following.

1.3.1 Station Keeping with in the Presence of a Current

The contribution in Chapter 3 is an approximate optimal station keeping policy that captures the desire to balance the need to accurately hold a station and the cost of holding that station through a quadratic performance criterion. The developed controller differs from results such as [19, 37] in that it tackles the challenges associated with the introduction of a time-varying irrotational current. Since the hydrodynamic parameters of a marine craft are often difficult to determine, a concurrent learning (CL) system identifier is developed. As outlined in [38, 39], CL uses additional information from recorded data to remove the persistence of excitation requirement associated with traditional system identifiers. The developed model-based ADP method simultaneously learns and implements an approximate optimal station keeping policy using a combination of on-policy and off-policy data, eliminating the need for physical exploration of the state space. A Lyapunov-based stability analysis is presented which guarantees ultimately bounded convergence of the marine craft to its station and of the approximated policy to the optimal policy. The developed strategy is validated for planar motion of an autonomous underwater vehicle, where experiments are conducted in a second-magnitude spring located in central Florida.

1.3.2 Planar Path Following Guidance Law

The contribution in Chapter 4 is a guidance law that provides approximate optimal path following of an arbitrarily parametrized two-dimensional path. Path following is achieved by tracking a virtual target placed on the desired path. The motion of the virtual target is described by a predefined state-dependent ordinary differential equation (cf. [12, 40, 41]). The state associated with the virtual target's location along the path is unbounded due to the infinite time horizon of the guidance law, which presents several technical challenges. The motion of the virtual target is redefined to facilitate the use of a parametric approximation of the optimal policy. The cost function is formulated in terms of the redefined virtual target motion, a unique challenge that is not addressed in previous ADP literature. A Lyapunov-based stability analysis is presented to establish ultimately bounded convergence of the approximate policy to the optimal policy and the vehicle state to the path while maintaining a desired speed profile. Simulation results compare the policy obtained using the developed technique to an offline numerical optimal solution. The proposed method is also experimentally validated on a differential mobile robot, which is used as a kinematic analog to a marine craft neglecting side slip.

1.3.3 Path Planning with Static Obstacles

Inspired by the advances in [20, 21, 34, 35], the contribution in Chapter 5 is the development of an approximate optimal feedback-based motion planner that respects input and state constraints. The developed planner differs from previous ADP literature in that it tackles the challenges specific to obstacle avoidance. The local approximation in [34] enables the handling of obstacles not known a priori. The result in [34] also introduces time-varying parameters in the parametric representation of the solution to the HJB equation. The time-varying parameters cause the estimation of the parameters to be in a near constant transient state making it difficult to prove that the generated feedback plan avoids the obstacles. This technical challenge motivates the introduction of an auxiliary feedback term to assist in navigating a marine craft around obstacles, and a scheduling function to switch between the approximate optimal feedback plan and the auxiliary feedback plan. Switching to the auxiliary feedback plan when the craft risks hitting an obstacle ensures obstacle avoidance. The proposed model-based ADP method approximates optimal paths using a combination of on-policy and off-policy data, eliminating the need for physical exploration of the state space. A Lyapunov-based stability analysis is presented which guarantees ultimately bounded convergence of the approximate path to the optimal path. Simulation results compare the path generated using the developed technique to a numerical pregenerative planner.

CHAPTER 2
PRELIMINARIES

2.1 Notation

Unless otherwise specified, the domain of all functions is $[0,\infty)$. Functions with the domain $[0,\infty)$ are specified only by their image, e.g., the function $h:[0,\infty)\to\mathbb{R}^n$ is denoted by $h\in\mathbb{R}^n$. By abuse of notation, state variables are also used to denote state trajectories, e.g., the state variable x in the equation $\dot{x}=f(x)+u$ is also used to denote the state trajectory $x(t)$. Unless otherwise specified, all mathematical quantities are assumed to be time-varying. The partial derivative with respect to the first argument $\partial f(x,y)/\partial x$ is denoted as $\nabla f(x,y)$. An $n\times m$ matrix of zeros is denoted by $0_{n\times m}$, and a $n\times n$ identity matrix be denoted by $I_{n\times n}$.

2.2 Problem Formulation

The focus of this dissertation is to develop an online approximate solution to the infinite-horizon total-cost optimal control problem for marine craft. To facilitate the formulation of the optimal control problem, consider a control-affine nonlinear system given by

$$\dot{\zeta}=f(\zeta)+g(\zeta)u, \tag{2–1}$$

where $\zeta\in\mathbb{R}^n$ denotes the system state, $f:\mathbb{R}^n\to\mathbb{R}^n$ denotes the drift dynamics, $g:\mathbb{R}^n\to\mathbb{R}^{n\times m}$ denotes the control effectiveness, and $u\in\mathbb{R}^m$ denotes the control input. The functions f and g must be locally Lipschitz continuous functions such that $f(0_{n\times 1})=0_{n\times 1}$ and the partial derivative $\nabla f(\zeta)$ is continuous.

The control objective is to find the solution to the infinite-horizon problem online, i.e., to simultaneously learn and utilize a control signal u online to minimize the cost functional

$$J(\zeta,u)\triangleq\int\limits_{t_o}^{\infty}r(\zeta(\tau),u(\tau))\,d\tau,$$

subject to the dynamic constraints in (2–1) where t_0 denotes the initial time and $r : \mathbb{R}^n \times \mathbb{R}^m \to [0, \infty)$ is the local cost given as

$$r(\zeta, u) \triangleq Q(\zeta) + u^T R u,$$

where $Q : \mathbb{R}^n \to \mathbb{R}$ is a positive definite function , and $R \in \mathbb{R}^{m\times m}$ is constant symmetric positive definite matrix.

2.3 Exact Solution

It is well known that if the functions f, g, and Q are stationary and the time-horizon is infinite, then the optimal control input $u = u^*(\zeta)$ is a stationary state-feedback policy, where $u^* : \mathbb{R}^n \to \mathbb{R}^m$. Furthermore, the value function, which maps each state to the total accumulated cost associated with following the stationary state-feedback policy from the given state, is also a stationary function. Assuming an optimal controller exists, the value function $V : \mathbb{R}^n \to [0, \infty)$ is written as

$$V(\zeta) = \min_{u\in\mathcal{U}} \int\limits_{t_0}^{\infty} r(\zeta(\tau), u(\tau))\, d\tau,$$

where $\mathcal{U}$ is the set of admissible control policies. The optimal value function is characterized by the HJB equation, which is given as

$$\nabla V(\zeta)(f(\zeta) + g(\zeta)u^*(\zeta)) + r(\zeta, u^*(\zeta)) = 0 \tag{2–2}$$

with the boundary condition $V(0) = 0$. Provided the HJB equation admits a continuously differentiable solution, the HJB equation constitutes a necessary and sufficient condition for optimality. The optimal control policy can be determined from (2–2) as

$$u^*(\zeta) = -\frac{1}{2}R^{-1}g^T(\nabla V(\zeta))^T. \tag{2–3}$$

2.4 Approximate Solution

The analytical expression for the optimal controller in (2–3) requires knowledge of the value function which is the solution to the HJB equation in (2–2). The HJB equation is a partial differential equation which is generally infeasible to solve; hence, an approximate solution is sought. In an approximate actor-critic-based solution, the value function V is replaced by a parametric approximation $\hat{V}\left(\zeta,\hat{W}_c\right)$ and the optimal policy u^* is replaced by a parametric approximation $\hat{u}\left(\zeta,\hat{W}_a\right)$. The objective of the critic is to learn the parameters $\hat{W}_c\in\mathbb{R}^l$, while the objective of the actor is to learn the parameters $\hat{W}_a\in\mathbb{R}^l$. Substituting the approximations $\hat{V}$ and $\hat{u}$ for V and u^* in (2–2), respectively, results in a residual error $\delta:\mathbb{R}^n\times\mathbb{R}^l\times\mathbb{R}^l\to\mathbb{R}$ called the Bellman error given as

$$\delta\left(\zeta,\hat{W}_c,\hat{W}_a\right)=\nabla\hat{V}\left(\zeta,\hat{W}_c\right)\left(f\left(\zeta\right)+g\left(\zeta\right)\hat{u}\left(\zeta,\hat{W}_a\right)\right)+r\left(\zeta,\hat{u}\left(\zeta,\hat{W}_a\right)\right). \tag{2–4}$$

To solve the optimal control problem, the critic and actor aim to find the set of parameters $\hat{W}_c$ and $\hat{W}_a$, respectively, that eliminate the Bellman error; hence, $\delta\left(\zeta,\hat{W}_c,\hat{W}_a\right)=0$. Since the parametric approximation does not exactly represent the value function, the set of parameters that minimize the Bellman error is sought. In particular, it is desirable to find a set of parameters that minimize the integral error $E_s:\mathbb{R}^l\times\mathbb{R}^l\to\mathbb{R}$ defined as

$$E_s\left(\hat{W}_c,\hat{W}_a\right)\triangleq\int\limits_{\zeta\in\mathcal{D}}\delta^2\left(\zeta,\hat{W}_c,\hat{W}_a\right)d\zeta,$$

where the domain $\mathcal{D}\subset\mathbb{R}^n$ is the domain of operation. In an online implementation of the actor-critic method, it is desirable to minimize the cumulative instantaneous error defined as

$$E\left(t\right)\triangleq\int_{t_0}^{t}E_s\left(\hat{W}_c\left(\tau\right),\hat{W}_a\left(\tau\right)\right)d\tau. \tag{2–5}$$

The Bellman error in (2–4) requires model knowledge to compute. The requirement of exact model knowledge is often a restrictive requirement. Three approaches are used to free the control design of model certainty in the system drift dynamics: integral reinforcement learning (RL) (c.f. [21] and [42]), state derivative estimation (c.f. [18]

and [24]), and CL-based model identification (c.f. [19] and [43]). Integral RL and state derivative methods can only compute the Bellman error along the state trajectory, while the Bellman error can be evaluated over the entire operating domain $\mathcal{D}$ using an identified or known model. Bellman errors computed off the state trajectory have been shown to improve controller performance as well as remove exploration conditions inherent in controllers designed with integral RL and state derivative estimation. This motivates the use of an identified system model for the body of this work when a known model is not available.

Using a identified model, the approximated Bellman error is given as

$$\hat{\delta}\left(\zeta,\hat{\theta},\hat{W}_c,\hat{W}_a\right)=\nabla\hat{V}\left(\zeta,\hat{W}_c\right)\left(\hat{f}\left(\zeta,\hat{\theta}\right)+g\left(\zeta\right)\hat{u}\left(\zeta,\hat{W}_a\right)\right)+r\left(\zeta,\hat{u}\left(\zeta,\hat{W}_a\right)\right)$$

where $\hat{f}\ :\ \mathbb{R}^n\times\mathbb{R}^p\ \rightarrow\ \mathbb{R}^n$ is a uniform approximation of the drift dynamics f and $\hat{\theta}\in\mathbb{R}^p$ denotes the vector of model parameter estimates. Given the parameters $\hat{\theta}$, $\hat{W}_c$, and $\hat{W}_a$, the Bellman error can be evaluated at any point $\zeta_k\in\mathcal{D}$[1] . The Bellman error serves as an indirect measure of how close the parameter $\hat{W}_c$ is to its ideal value. The critic performs updates to the parameter based on an approximation of the cumulative instantaneous error in (2–5) given as

$$\hat{E}\left(t\right)=\int_{t_0}^{t}\left(\hat{\delta}\left(\zeta\left(\tau\right),\hat{\theta}\left(\tau\right),\hat{W}_c\left(\tau\right),\hat{W}_a\left(\tau\right)\right)^2+\sum_{k=1}^{N}\hat{\delta}\left(\zeta_k\left(\tau\right),\hat{\theta}\left(\tau\right),\hat{W}_c\left(\tau\right),\hat{W}_a\left(\tau\right)\right)^2\right)d\tau, \tag{2–6}$$

using a steepest descent update law. Note that for exact model knowledge, the approximated Bellman error in (2–6) is replaced by the true Bellman error in (2–4).

[1] Note the sampled state ζ_k can be either stationary or evolve continuously in the state space that is $\dot{\zeta}_k\ =\ h\left(\zeta\right)$ were $h\ :\ \mathbb{R}^n\ \rightarrow\ \mathbb{R}^n$ is a bounded function of the current state.

2.5 Online Implementation

For feasibility of implementation, the value function parametric approximation is a linear-in-the-parameters (LP) approximation given as

$$\hat{V}\left(\zeta,\hat{W}_c\right)=\hat{W}_c^T\sigma\left(\zeta\right),$$

where $\sigma:\mathbb{R}^n\to\mathbb{R}^l$ is a bounded, continuously differentiable activation function. The activation function satisfies the properties $\sigma\left(0\right)=0$ and $\nabla\sigma\left(0\right)=0$. From (2–3), the LP approximation of the optimal control policy is given as

$$\hat{u}\left(\zeta,\hat{W}_a\right)=-\frac{1}{2}R^{-1}g\left(\zeta\right)^T\nabla\sigma\left(\zeta\right)^T\hat{W}_a.$$

For a LP approximation, the regressor vector $\omega:\mathbb{R}^n\times\mathbb{R}^p\times\mathbb{R}^l\to\mathbb{R}^l$ is given as

$$\omega\left(\zeta,\hat{\theta},\hat{W}_a\right)=\nabla\sigma\left(\zeta\right)\left(\hat{f}\left(\zeta,\hat{\theta}\right)+g\left(\zeta\right)\hat{u}\left(\zeta,\hat{W}_a\right)\right).$$

The critic update law is given as

$$\dot{\hat{W}}_c=-\Gamma\left(k_{c1}\frac{\omega_t}{\rho_t}\hat{\delta}_t+\frac{k_{c2}}{N}\sum_{k=1}^{N}\frac{\omega_k}{\rho_k}\hat{\delta}_k\right),\tag{2–7}$$

$$\dot{\Gamma}=\begin{cases}\beta\Gamma-k_{c1}\Gamma\frac{\omega_t{\omega_t}^T}{\rho_t^2}\Gamma, & \left\Vert\Gamma\right\Vert\leq\overline{\Gamma}\\ 0 & \text{otherwise}\end{cases},\tag{2–8}$$

where $k_{c1},k_{c2}\in\mathbb{R}$ are a positive constant adaptation gains, $\hat{\delta}_t\triangleq\hat{\delta}\left(\zeta,\hat{\theta},\hat{W}_c,\hat{W}_a\right)$ and $\hat{\delta}_k\triangleq\hat{\delta}\left(\zeta_k,\hat{\theta},\hat{W}_c,\hat{W}_a\right)$ are the Bellman errors, $\omega_t=\omega\left(\zeta,\hat{\theta},\hat{W}_a\right)$ and $\omega_k=\omega\left(\zeta_k,\hat{\theta},\hat{W}_a\right)$ denote the regressor vectors, $\left\Vert\Gamma\left(t_0\right)\right\Vert=\left\Vert\Gamma_0\right\Vert\leq\bar{\Gamma}$ is the initial adaptation gain in (2–8), $\bar{\Gamma}\in\mathbb{R}$ is a positive saturation constant, $\beta\in\mathbb{R}$ is a positive constant forgetting factor, and

$$\rho_t=\sqrt{1+k_\rho{\omega_t}^T\omega_t},$$

$$\rho_k=\sqrt{1+k_\rho{\omega_k}^T\omega_k},$$

are normalization terms with $k_\rho \in \mathbb{R}$ as a positive constant gain. The actor update law is given as

$$\dot{\hat{W}}_a = \text{proj}\left\{-k_a\left(\hat{W}_a - \hat{W}_c\right)\right\}, \tag{2–9}$$

where $k_a \in \mathbb{R}$ is a positive constant gain, and proj $\{\cdot\}$ is a smooth projection operator[2] .

Assumption 2.1. [46] There exists a strictly positive constant $\underline{c}$ such that

$$\underline{c} = \inf_{t\in[t_0,\infty)}\left[\lambda_{min}\left(\sum_{k=1}^{N}\frac{\omega_k{\omega_k}^T}{{\rho_k}^2}\right)\right],$$

where the operator $\lambda_{\min}$ denotes the minimum eigenvalue of a matrix.

In general, the Assumption in (2.1) cannot be guaranteed to hold a priori; however, heuristically, the condition can be met by sampling redundant data, i.e., $N \gg l$.

Let $\tilde{W}_c \triangleq W - \hat{W}_c$ and $\tilde{W}_a \triangleq W - \hat{W}_a$ denote the parameter estimation error, $W \in \mathbb{R}^l$ denotes the ideal parametric weight for $\hat{W}_c$ and $\hat{W}_a$. Provided Assumption (2.1) and sufficient learning gain conditions are satisfied, the candidate Lyapunov function

$$V_L\left(\zeta, \hat{W}_c, \hat{W}_a\right) = V\left(\zeta\right) + \frac{1}{2}\tilde{W}_c^T\Gamma^{-1}\tilde{W}_c + \frac{1}{2}\tilde{W}_a^T\tilde{W}_a$$

can be used to establish convergence of ζ, $\hat{W}_c$, and $\hat{W}_a$ to a neighborhood of zero as $t \to \infty$ when the system in (2–1) is controlled by the control law $u = \hat{u}\left(\zeta\right)$ and the parameters $\hat{W}_c$ and $\hat{W}_a$ are updated by (2–7) and (2–9), respectively.

[2] See Section 4.4 in [44] or Remark 3.6 in [45] for details of the projection operator.

CHAPTER 3
STATION KEEPING IN THE PRESENCE OF A CURRENT

The focus of this chapter is to develop an online approximation of the optimal station keeping strategy for a fully actuated marine craft subject to a time-varying irrotational ocean current. The hydrodynamic drift dynamics of the dynamic model are assumed to be unknown; therefore, a CL system identifier is developed to identify the unknown model parameters. Using the identified model, an adaptive update law is used to estimate the unknown value function and generate the optimal policy.

3.1 Vehicle Model

Consider the nonlinear equations of motion for a marine craft including the effects of irrotational ocean current given in Section 7.5 of [2] as

$$\dot{\eta} = J_E\left(\eta\right)\nu, \tag{3–1}$$

$$M_{RB}\dot{\nu} + C_{RB}\left(\nu\right)\nu + M_A\dot{\nu}_r + C_A\left(\nu_r\right)\nu_r + D_A\left(\nu_r\right)\nu_r + G\left(\eta\right) = \tau_b, \tag{3–2}$$

where $\nu \in \mathbb{R}^n$ is the body-fixed translational and angular velocity vector, $\nu_c \in \mathbb{R}^n$ is the body-fixed irrotational current velocity vector, $\nu_r = \nu - \nu_c$ is the relative body-fixed translational and angular fluid velocity vector, $\eta \in \mathbb{R}^n$ is the earth-fixed position and orientation vector, $J_E : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the coordinate transformation between the body-fixed and earth-fixed coordinates[1] , $M_{RB} \in \mathbb{R}^{n\times n}$ is the constant rigid body inertia matrix, $C_{RB} : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the rigid body centripetal and Coriolis matrix, $M_A \in \mathbb{R}^{n\times n}$ is the constant hydrodynamic added mass matrix, $C_A : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the unknown hydrodynamic centripetal and Coriolis matrix, $D_A : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the unknown hydrodynamic damping and friction matrix, $G : \mathbb{R}^n \to \mathbb{R}^n$ is the gravitational

[1] The orientation of the vehicle may be represented as Euler angles, quaternions, or angular rates. In this development, the use of Euler angles is assumed, see Section 7.5 in [2] for details regarding other representations.

and buoyancy force and moment vector, and $\tau_b \in \mathbb{R}^n$ is the body-fixed force and moment control input.

In the case of a three degree-of-freedom (DOF) planar model with orientation represented as Euler angles, the state vectors in (3–1) and (3–2) are further defined as

$$\eta \triangleq \begin{bmatrix} x & y & \psi \end{bmatrix}^T,$$

$$\nu \triangleq \begin{bmatrix} u & v & r \end{bmatrix}^T,$$

where x, $y \in \mathbb{R}$, are the earth-fixed position vector components of the center of mass, $\psi \in \mathbb{R}$ represents the yaw angle, u, $v \in \mathbb{R}$ are the body-fixed translational velocities, and $r \in \mathbb{R}$ is the body-fixed angular velocity. The irrotational current vector is defined as

$$\nu_c \triangleq \begin{bmatrix} u_c & v_c & 0 \end{bmatrix}^T,$$

where u_c, $v_c \in \mathbb{R}$ are the body-fixed current translational velocities. The coordinate transformation $J_E(\eta)$ is given as

$$J_E(\eta) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Assumption 3.1. The marine craft is neutrally buoyant if submerged and the center of gravity is located vertically below the center of buoyancy on the vertical axis if the vehicle model includes roll and pitch[2] .

3.2 System Identifier

Since the hydrodynamic effects pertaining to a specific marine craft may be unknown, an online system identifier is developed for the vehicle drift dynamics. Consider

[2] This assumption simplifies the subsequent analysis and can often be met by trimming the vehicle.

the control affine form of the vehicle model,

$$\dot{\zeta} = Y\left(\zeta, \nu_c\right)\theta + f_0\left(\zeta, \dot{\nu}_c\right) + g\tau_b, \tag{3–3}$$

where $\zeta \triangleq \begin{bmatrix} \eta & \nu \end{bmatrix}^T \in \mathbb{R}^{2n}$ is the state vector. The unknown hydrodynamics are LP with p unknown parameters where $Y : \mathbb{R}^{2n} \times \mathbb{R}^n \to \mathbb{R}^{2n\times p}$ is the regression matrix and $\theta \in \mathbb{R}^p$ is the vector of unknown parameters. The unknown hydrodynamic effects are modeled as

$$Y\left(\zeta, \nu_c\right)\theta = \begin{bmatrix} 0 \\ -M^{-1}C_A\left(\nu_r\right)\nu_r - M^{-1}D_A\left(\nu_r\right)\nu_r \end{bmatrix},$$

and known rigid body drift dynamics $f_0 : \mathbb{R}^{2n} \times \mathbb{R}^n \to \mathbb{R}^{2n}$ are modeled as

$$f_0\left(\zeta, \dot{\nu}_c\right) = \begin{bmatrix} J_E\left(\eta\right)\nu \\ M^{-1}M_A\dot{\nu}_c - M^{-1}C_{RB}\left(\nu\right)\nu - M^{-1}G\left(\eta\right) \end{bmatrix},$$

where $M \triangleq M_{RB} + M_A$, and the body-fixed current velocity ν_c, and acceleration $\dot{\nu}_c$ are assumed to be measurable[3] . The known constant control effectiveness matrix $g \in \mathbb{R}^{2n\times n}$ is defined as

$$g \triangleq \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix}.$$

An identifier is designed as

$$\dot{\hat{\zeta}} = Y\left(\zeta, \nu_c\right)\hat{\theta} + f_0\left(\zeta, \dot{\nu}_c\right) + g\tau_b + k_\zeta\tilde{\zeta}, \tag{3–4}$$

[3] The body-fixed current velocity ν_c may be trivially measured using sensors commonly found on marine craft, such as a Doppler velocity log, while the current acceleration $\dot{\nu}_c$ may be determined using numerical differentiation and smoothing.

where $\tilde{\zeta} \triangleq \zeta - \hat{\zeta}$ is the measurable state estimation error, and $k_\zeta \in \mathbb{R}^{2n \times 2n}$ is a constant positive definite, diagonal gain matrix. Subtracting (3–4) from (3–3), yields

$$\dot{\tilde{\zeta}} = Y\left(\zeta, \nu_c\right)\tilde{\theta} - k_\zeta \tilde{\zeta},$$

where $\tilde{\theta} \triangleq \theta - \hat{\theta}$ is the parameter identification error.

3.2.1 Parameter Update

Traditional adaptive control techniques require persistence of excitation to ensure the parameter estimates $\hat{\theta}$ converge to their true values θ (cf. [44] and [47]). Persistence of excitation often requires an excitation signal to be applied to the vehicle's input resulting in unwanted deviations in the vehicle state. These deviations are often in opposition to the vehicle's control objectives. Alternatively, a CL-based system identifier can be developed (cf. [38] and [39]). The CL-based system identifier relaxes the persistence of excitation requirement through the use of a prerecorded history stack of state-action pairs[4] .

Assumption 3.2. There exists a prerecorded data set of sampled data points $\{\zeta_j, \nu_{cj}, \dot{\nu}_{cj}, \tau_{bj} \in \chi | j = 1, 2, \ldots, M\}$ with a numerically calculated state derivatives $\dot{\bar{\zeta}}_j$ at each recorded state-action pair such that $\forall t \in [0, \infty)$,

$$\operatorname{rank}\left(\sum_{j=1}^{M} Y_j^T Y_j\right) = p,$$

[4] In this development, it is assumed that a data set of state-action pairs is available a priori. Experiments to collect state-action pairs do not necessarily need to be conducted in the presence of a current (e.g., the data may be collected in a pool). Since the current affects the dynamics only through the ν_r terms, data that is sufficiently rich and satisfies Assumption 3.2 may be collected by merely exploring the ζ state space. Note, this is the reason the body-fixed current ν_c and acceleration $\dot{\nu}_c$ are not considered a part of the state. If state-action data is not available for the given system then it is possible to build the history stack in real-time and the details of that development can be found in Appendix A of [43].

$$\left\|\dot{\bar{\zeta}}_j - \dot{\zeta}_j\right\| < \bar{d}, \forall j,$$

where $Y_j \triangleq Y\left(\zeta_j, \nu_{cj}\right)$, $f_{0j} \triangleq f_0\left(\zeta_j\right)$, $\dot{\zeta}_j = Y_j\theta + f_{0j} + g\tau_{bj}$, and $\bar{d} \in [0, \infty)$ is a constant.

The parameter estimate update law is given as

$$\dot{\hat{\theta}} = \Gamma_\theta Y\left(\zeta, \nu_c\right)^T \tilde{\zeta} + \Gamma_\theta k_\theta \sum_{j=1}^{M} Y_j^T \left(\dot{\bar{\zeta}}_j - f_{0j} - g\tau_{bj} - Y_j\hat{\theta}\right), \tag{3–5}$$

where Γ_θ is a positive definite, diagonal gain matrix, and k_θ is a positive, scalar gain matrix. To facilitate the stability analysis, the parameter estimate update law is expressed in the advantageous form

$$\dot{\hat{\theta}} = \Gamma_\theta Y\left(\zeta, \nu_c\right)^T \tilde{\zeta} + \Gamma_\theta k_\theta \sum_{j=1}^{M} Y_j^T \left(Y_j\tilde{\theta} + d_j\right),$$

where $d_j = \dot{\bar{\zeta}}_j - \dot{\zeta}_j$.

3.2.2 Convergence Analysis

Consider the candidate Lyapunov function $V_P : \mathbb{R}^{2n+p} \times [0, \infty)$ given as

$$V_P\left(Z_P\right) = \frac{1}{2}\tilde{\zeta}^T\tilde{\zeta} + \frac{1}{2}\tilde{\theta}^T\Gamma_\theta^{-1}\tilde{\theta}, \tag{3–6}$$

where $Z_P \triangleq \left[\begin{array}{cc} \tilde{\zeta}^T & \tilde{\theta}^T \end{array}\right]$. The candidate Lyapunov function can be bounded as

$$\frac{1}{2}\min\left\{1, \underline{\gamma_\theta}\right\}\left\|Z_P\right\|^2 \leq V_P\left(Z_P\right) \leq \frac{1}{2}\max\left\{1, \overline{\gamma_\theta}\right\}\left\|Z_P\right\|^2 \tag{3–7}$$

where $\underline{\gamma_\theta}, \overline{\gamma_\theta}$ are the minimum and maximum eigenvalues of Γ_θ, respectively.

The time derivative of the candidate Lyapunov function in (3–6) is

$$\dot{V}_P = -\tilde{\zeta}^T k_\zeta \tilde{\zeta} - k_\theta\tilde{\theta}^T \sum_{j=1}^{M} Y_j^T Y_j \tilde{\theta} - k_\theta\tilde{\theta}^T \sum_{j=1}^{M} Y_j^T d_j.$$

The time derivative may be upper bounded by

$$\dot{V}_P \leq -\underline{k_\zeta}\left\|\tilde{\zeta}\right\|^2 - k_\theta\underline{y}\left\|\tilde{\theta}\right\|^2 + k_\theta d_\theta\left\|\tilde{\theta}\right\|, \tag{3–8}$$

where $\underline{k_\zeta}, \underline{y}$ are the minimum eigenvalues of k_ζ and $\sum_{j=1}^{M} Y_j^T Y_j$, respectively, and $d_\theta = \bar{d}\sum_{j=1}^{M}\|Y_j\|$. Completing the squares, (3–8) may be upper bounded by

$$\dot{V}_P \leq -\underline{k_\zeta}\left\|\tilde{\zeta}\right\|^2 - \frac{k_\theta \underline{y}}{2}\left\|\tilde{\theta}\right\|^2 + \frac{k_\theta d_\theta^2}{2\underline{y}},$$

which may be further upper bounded by

$$\dot{V}_P \leq -\alpha_P \left\|Z_P\right\|^2, \forall \left\|Z_P\right\| \geq K_P > 0, \tag{3–9}$$

where $\alpha_P \triangleq \frac{1}{2}\min\left\{2\underline{k_\zeta}, k_\theta \underline{y}\right\}$ and $K_P \triangleq \sqrt{\frac{k_\theta d_\theta^2}{2\alpha_P \underline{y}}}$. Using (3–7) and (3–9), $\tilde{\zeta}$ and $\tilde{\theta}$ can be shown to exponentially decay to an ultimate bound as $t \to \infty$. The ultimate bound may be made arbitrarily small depending on the selection of the gains k_ζ and k_θ.

3.3 Problem Formulation

3.3.1 Residual Model

The presence of a time-varying irrotational current yields unique challenges in the formulation of the optimal regulation problem. Since the current renders the system non-autonomous, a residual model that does not include the effects of the irrotational current is introduced. The residual model is used in the development of the optimal control problem in place of the original model. A disadvantage of this approach is that the optimal policy is developed for the current-free model[5] . In the case where the earth-fixed current is constant, the effects of the current may be included in the development of the optimal control problem as detailed in Appendix A.

The residual model can be written in a control affine form as

$$\dot{\zeta} = Y_{res}\left(\zeta\right)\theta + f_{0_{res}}\left(\zeta\right) + gu, \tag{3–10}$$

[5] To the author's knowledge, there is no method to generate a policy with time-varying inputs (e.g., time-varying irrotational current) that guarantees optimally and stability.

where the unknown hydrodynamics are linear-in-the-parameters with p unknown parameters where $Y_{res}:\mathbb{R}^{2n}\to\mathbb{R}^{2n\times p}$ is a regression matrix, the function $f_{0_{res}}:\mathbb{R}^{2n}\to\mathbb{R}^{2n}$ is the known portion of the dynamics, and $u\in\mathbb{R}^{n}$ is the control vector. The drift dynamics, defined as $f_{res}\left(\zeta\right)=Y_{res}\left(\zeta\right)\theta+f_{0_{res}}\left(\zeta\right)$, can be shown to satisfy $f_{res}\left(0\right)=0$ when Assumption 3.1 is satisfied. The drift dynamics in (3–10) are modeled as

$$Y_{res}\left(\zeta\right)\theta=\begin{bmatrix}0\\-M^{-1}C_{A}\left(\nu\right)\nu-M^{-1}D\left(\nu\right)\nu\end{bmatrix},$$

$$f_{0_{res}}\left(\zeta\right)=\begin{bmatrix}J_{E}\left(\eta\right)\nu\\-M^{-1}C_{RB}\left(\nu\right)\nu-M^{-1}G\left(\eta\right)\end{bmatrix},\tag{3–11}$$

and the virtual control vector u is defined as

$$u=\tau_{b}-\tau_{c}\left(\zeta,\nu_{c},\dot{\nu}_{c}\right),\tag{3–12}$$

where $\tau_{c}:\mathbb{R}^{2n}\times\mathbb{R}^{n}\times\mathbb{R}^{n}\to\mathbb{R}^{n}$ is a feedforward term to compensate for the effect of the variable current, which includes cross-terms generated by the introduction of the residual model and is given as

$$\tau_{c}\left(\zeta,\nu_{c},\dot{\nu}_{c}\right)=C_{A}\left(\nu_{r}\right)\nu_{r}+D\left(\nu_{r}\right)\nu_{r}-M_{A}\dot{\nu}_{c}-C_{A}\left(\nu\right)\nu-D\left(\nu\right)\nu.$$

The current feedforward term is represented in the advantageous form

$$\tau_{c}\left(\zeta,\nu_{c},\dot{\nu}_{c}\right)=-M_{A}\dot{\nu}_{c}+Y_{c}\left(\zeta,\nu_{c}\right)\theta,$$

where $Y_{c}:\mathbb{R}^{2n}\times\mathbb{R}^{n}\to\mathbb{R}^{2n\times p}$ is the regression matrix and

$$Y_{c}\left(\zeta,\nu_{c}\right)\theta=C_{A}\left(\nu_{r}\right)\nu_{r}+D\left(\nu_{r}\right)\nu_{r}-C_{A}\left(\nu\right)\nu-D\left(\nu\right)\nu.$$

Since the parameters are unknown, an approximation of the compensation term τ_{c} given by

$$\hat{\tau}_{c}\left(\zeta,\nu_{c},\dot{\nu}_{c},\hat{\theta}\right)=-M_{A}\dot{\nu}_{c}+Y_{c}\hat{\theta}\tag{3–13}$$

is implemented, and the approximation error is defined by

$$\tilde{\tau}_c \triangleq \tau_c - \hat{\tau}_c.$$

3.3.2 Nonlinear Optimal Regulation Problem

The performance index for the optimal regulation problem is defined as

$$J\left(\zeta, u\right) \triangleq \int_{t_0}^{\infty} r\left(\zeta\left(\tau\right), u\left(\tau\right)\right) d\tau, \tag{3–14}$$

where t_0 denotes the initial time, and $r : \mathbb{R}^{2n} \to [0, \infty)$ is the local cost defined as

$$r\left(\zeta, u\right) \triangleq \zeta^T Q \zeta + u^T R u. \tag{3–15}$$

In (3–15), $Q \in \mathbb{R}^{2n \times 2n}$, $R \in \mathbb{R}^{n \times n}$ are symmetric positive definite weighting matrices, and u is the virtual control vector. The matrix Q has the property $\underline{q} \left\|\xi_q\right\|^2 \leq \xi_q^T Q \xi_q \leq \overline{q} \left\|\xi_q\right\|^2$, $\forall \xi_q \in \mathbb{R}^{2n}$ where $\underline{q}$ and $\overline{q}$ are positive constants. The infinite-time scalar value function $V : \mathbb{R}^{2n} \to [0, \infty)$ for the optimal solution is written as

$$V\left(\zeta\right) = \min_{u \in \mathcal{U}} \int_{t_0}^{\infty} r\left(\zeta\left(\tau\right), u\left(\tau\right)\right) d\tau. \tag{3–16}$$

where $\mathcal{U}$ is the set of admissible control policies. The objective of the optimal control problem is to find the optimal policy $u^* : \mathbb{R}^{2n} \to \mathbb{R}^n$ that minimizes the performance index (3–14) subject to the dynamic constraints in (3–10). The optimal value function is characterized by the HJB equation, which is given as

$$\nabla V\left(\zeta\right)\left(Y_{res}\left(\zeta\right)\theta + f_{0_{res}}\left(\zeta\right) + g u^*\left(\zeta\right)\right) + r\left(\zeta, u^*\left(\zeta\right)\right) = 0 \tag{3–17}$$

with the boundary condition $V\left(0\right) = 0$. The optimal policy can be determined from (3–17) as

$$u^*\left(\zeta\right) = -\frac{1}{2} R^{-1} g^T \nabla V\left(\zeta\right)^T. \tag{3–18}$$

The analytical expression for the optimal controller in (3–18) requires knowledge of the value function which is the solution to the HJB equation in (3–17). The HJB equation is a partial differential equation which is generally infeasible to solve; hence, an approximate solution is sought.

3.4 Approximate Solution

The subsequent development is based on a neural network (NN) approximation of the value function and optimal policy. Differing from previous ADP literature with model uncertainty (e.g., [20, 21, 37]) that seeks a NN approximation using the integral form of the HJB, the following development seeks a NN approximation using the differential form. The differential form of the HJB coupled with the identified model allows off-policy learning, which relaxes the persistence of excitation condition previously required.

Over any compact domain $\chi \subset \mathbb{R}^{2n}$, the value function $V : \mathbb{R}^{2n} \to [0, \infty)$ can be represented by a single-layer NN with l neurons as

$$V\left(\zeta\right) = W^{T}\sigma\left(\zeta\right) + \epsilon\left(\zeta\right), \tag{3–19}$$

where $W \in \mathbb{R}^{l}$ is the constant ideal weight vector bounded above by a known positive constant, $\sigma : \mathbb{R}^{2n} \to \mathbb{R}^{l}$ is a bounded, continuously differentiable activation function, and $\epsilon : \mathbb{R}^{2n} \to \mathbb{R}$ is the bounded, continuously differential function reconstruction error. Using (3–18) and (3–19), the optimal policy can be represented by

$$u^{*}\left(\zeta\right) = -\frac{1}{2}R^{-1}g^{T}\left(\nabla\sigma\left(\zeta\right)^{T}W + \epsilon'\left(\zeta\right)^{T}\right). \tag{3–20}$$

Based on (3–19) and (3–20), NN approximations of the value function and the optimal policy are defined as

$$\hat{V}\left(\zeta, \hat{W}_{c}\right) = \hat{W}_{c}^{T}\sigma\left(\zeta\right), \tag{3–21}$$

$$\hat{u}\left(\zeta, \hat{W}_{a}\right) = -\frac{1}{2}R^{-1}g^{T}\nabla\sigma\left(\zeta\right)^{T}\hat{W}_{a}, \tag{3–22}$$

where $\hat{W}_{c},\ \hat{W}_{a} \in \mathbb{R}^{l}$ are estimates of the constant ideal weight vector W. The weight estimation errors are defined as $\tilde{W}_{c} \triangleq W - \hat{W}_{c}$ and $\tilde{W}_{a} \triangleq W - \hat{W}_{a}$. Substituting

(3–21), (3–22), and the approximation of (3–10) into (3–17), results in a residual error $\delta : \mathbb{R}^{2n} \times \mathbb{R}^{p} \times \mathbb{R}^{l} \times \mathbb{R}^{l} \to \mathbb{R}$ called the Bellman error given as

$$\delta\left(\zeta, \hat{\theta}, \hat{W}_c, \hat{W}_a\right) = r\left(\zeta, \hat{u}\left(\zeta, \hat{W}_a\right)\right) + \hat{W}_c^T \omega\left(\zeta, \hat{\theta}, \hat{W}_a\right), \tag{3–23}$$

where $\omega : \mathbb{R}^{2n} \to \mathbb{R}^{l}$ is given by

$$\omega\left(\zeta, \hat{\theta}, \hat{W}_a\right) = \nabla\sigma\left(\zeta\right)\left[Y_{res}\left(\zeta\right)\hat{\theta} + f_{0_{res}}\left(\zeta\right) + g\hat{u}\left(\zeta, \hat{W}_a\right)\right].$$

The online implementation of the approximation is presented in Section 2.5, where the parameters $\hat{W}_c$ and $\hat{W}_a$ are updated by (2–7) and (2–9), respectively. Using the definition in (3–12), the force and moment applied to the vehicle, described in (3–3), is given in terms of the approximated optimal virtual control (3–22) and the compensation term approximation in (3–13) as

$$\hat{\tau}_b = \hat{u}\left(\zeta, \hat{W}_a\right) + \hat{\tau}_c\left(\zeta, \hat{\theta}, \nu_c, \dot{\nu}_c\right). \tag{3–24}$$

3.5 Stability Analysis

For notational brevity, all function dependencies from previous sections will be henceforth suppressed. An unmeasurable form of the Bellman error can be written as

$$\delta = -\tilde{W}_c^T \omega - W^T \nabla\sigma Y_{res}\tilde{\theta} - \nabla\epsilon\left(Y_{res}\theta + f_{0_{res}}\right) + \frac{1}{4}\tilde{W}_a^T G_\sigma \tilde{W}_a + \frac{1}{2}\nabla\epsilon G \nabla\sigma^T W + \frac{1}{4}\nabla\epsilon G \nabla\epsilon^T, \tag{3–25}$$

where $G \triangleq gR^{-1}g^T \in \mathbb{R}^{2n\times 2n}$ and $G_\sigma \triangleq \nabla\sigma G \nabla\sigma^T \in \mathbb{R}^{l\times l}$ are symmetric, positive semi-definite matrices. Similarly, the Bellman error at the sampled data points can be written as

$$\delta_k = -\tilde{W}_c^T \omega_k - W^T \nabla\sigma_k Y_{res_k}\tilde{\theta} + \frac{1}{4}\tilde{W}_a^T G_{\sigma k} \tilde{W}_a + E_k, \tag{3–26}$$

where

$$E_k \triangleq \frac{1}{2}\nabla\epsilon_k G \nabla{\sigma_k}^T W + \frac{1}{4}\nabla\epsilon_k G \nabla\epsilon_k^T - \nabla\epsilon_k\left(Y_{res_k}\theta + f_{0_{res_k}}\right) \in \mathbb{R}$$

is a constant at each data point, and the notation F_k denotes the function $F(\zeta, \cdot)$ evaluated at a sampled state, i.e., $F_k(\cdot) = F(\zeta_k, \cdot)$. The functions Y_{res} and $f_{0_{res}}$ on the compact set χ are Lipschitz continuous and can be bounded by

$$\|Y_{res}\| \leq L_{Y_{res}} \|\zeta\|, \; \forall \zeta \in \chi,$$

$$\|f_{0_{res}}\| \leq L_{f0_{res}} \|\zeta\|, \; \forall \zeta \in \chi,$$

respectively, where $L_{Y_{res}}$ and $L_{f0_{res}}$ are positive constants.

To facilitate the subsequent stability analysis, consider the candidate Lyapunov function $V_L : \mathbb{R}^{2n} \times \mathbb{R}^l \times \mathbb{R}^l \times \mathbb{R}^p \to [0, \infty)$ given as

$$V_L(Z) = V(\zeta) + \frac{1}{2}\tilde{W}_c^{\,T}\Gamma^{-1}\tilde{W}_c + \frac{1}{2}\tilde{W}_a^T\tilde{W}_a + V_P(Z_P),$$

where $Z \triangleq \begin{bmatrix} \zeta^T & \tilde{W}_c^T & \tilde{W}_a^T & Z_P^T \end{bmatrix}^T \in \chi \times \mathbb{R}^l \times \mathbb{R}^l \times \mathbb{R}^p$. Since the value function V in (3–16) is positive definite, V_L can be bounded by

$$\underline{v_L}(\|Z\|) \leq V_L(Z) \leq \overline{v_L}(\|Z\|) \tag{3–27}$$

using Lemma 4.3 of [48] and (3–7), where $\underline{v_L}$, $\overline{v_L} : [0, \infty) \to [0, \infty)$ are class $\mathcal{K}$ functions. Let $\beta_L \subset \chi \times \mathbb{R}^l \times \mathbb{R}^l \times \mathbb{R}^p$ be a compact set, where the notation $\overline{\|(\cdot)\|}$ is defined as $\overline{\|(\cdot)\|} = \sup_{Z \in \beta_L} \|(\cdot)\|$, then

$$\varphi_\zeta = \underline{q} - \frac{k_{c1}\overline{\|\nabla\epsilon\|}\left(L_{Y_{res}}\|\theta\| + L_{f_{0_{res}}}\right)}{2} - \frac{L_{Y_c}\|g\|\left(\|W\|\,\overline{\|\nabla\sigma\|} + \overline{\|\nabla\epsilon\|}\right)}{2},$$

$$\varphi_c = \frac{k_{c2}}{N}\underline{c} - \frac{k_a}{2} - \frac{k_{c1}\overline{\|\nabla\epsilon\|}\left(L_{Y_{res}}\|\theta\| + L_{f_{0_{res}}}\right)}{2} - \frac{k_{c1}L_Y\overline{\|\zeta\|\|\nabla\sigma\|}\,\|W\|}{2} \\ - \frac{\frac{k_{c2}}{N}\sum_{j=1}^{n}\left(\left\|Y_{res_j}\nabla\sigma_j\right\|\right)\|W\|}{2},$$

$$\varphi_a = \frac{k_a}{2},$$

$$\varphi_\theta = k_\theta \underline{y} - \frac{\frac{k_{c2}}{N}\sum_{k=1}^{N}\left(\left\|Y_{res_k}\nabla\sigma_k\right\|\right)\left\|W\right\|}{2} - \frac{L_{Y_c}\left\|g\right\|\left(\left\|W\right\|\overline{\left\|\nabla\sigma\right\|}+\overline{\left\|\nabla\epsilon\right\|}\right)}{2} - \frac{k_{c1}L_{Y_{res}}\left\|W\right\|\overline{\left\|\zeta\right\|\left\|\nabla\sigma\right\|}}{2},$$

$$\iota_c = \overline{\left\|\frac{k_{c2}}{4N}\sum_{j=1}^{N}\tilde{W}_a^T G_{\sigma_j}\tilde{W}_a + \frac{k_{c1}}{4}\tilde{W}_a^T G_\sigma \tilde{W}_a + k_{c1}\nabla\epsilon G\nabla\sigma^T W + \frac{k_{c1}}{4}\nabla\epsilon G\nabla\epsilon^T + \frac{k_{c2}}{N}\sum_{k=1}^{N}E_k\right\|},$$

$$\iota_a = \overline{\left\|\frac{1}{2}W^T G_\sigma + \frac{1}{2}\nabla\epsilon G\nabla\sigma^T\right\|},$$

$$\iota_\theta = k_\theta d_\theta,$$

$$\iota = \overline{\left\|\frac{1}{4}\nabla\epsilon G\nabla\epsilon^T\right\|}.$$

When Assumption 2.1 and 3.2, and the sufficient gain conditions

$$\underline{q} > \frac{k_{c1}\overline{\left\|\nabla\epsilon\right\|}\left(L_{Y_{res}}\left\|\theta\right\| + L_{f_{0_{res}}}\right)}{2}, + \frac{L_{Y_c}\left\|g\right\|\left(\left\|W\right\|\overline{\left\|\nabla\sigma\right\|}+\overline{\left\|\nabla\epsilon\right\|}\right)}{2}, \tag{3–28}$$

$$\underline{c} > \frac{N}{k_{c2}}\left(\frac{k_{c1}\overline{\left\|\nabla\epsilon\right\|}\left(L_{Y_{res}}\left\|\theta\right\| + L_{f_{0_{res}}}\right)}{2} + \frac{k_a}{2} + \frac{k_{c1}L_Y\overline{\left\|\zeta\right\|\left\|\nabla\sigma\right\|}\left\|W\right\|}{2} + \frac{\frac{k_{c2}}{N}\sum_{k=1}^{N}\left(\left\|Y_{res_k}\nabla\sigma_k\right\|\right)\left\|W\right\|}{2}\right),$$

$$\underline{y} > \frac{1}{k_\theta}\left(\frac{\frac{k_{c2}}{N}\sum_{k=1}^{N}\left(\left\|Y_{res_k}\nabla\sigma_k\right\|\right)\left\|W\right\|}{2} + \frac{L_{Y_c}\left\|g\right\|\left(\left\|W\right\|\overline{\left\|\nabla\sigma\right\|}+\overline{\left\|\nabla\epsilon\right\|}\right)}{2} + \frac{k_{c1}L_{Y_{res}}\left\|W\right\|\overline{\left\|\zeta\right\|\left\|\nabla\sigma\right\|}}{2}\right), \tag{3–29}$$

are satisfied, the constant $K \in \mathbb{R}$ defined as

$$K \triangleq \sqrt{\frac{\iota_c^2}{2\alpha\varphi_c} + \frac{\iota_a^2}{2\alpha\varphi_a} + \frac{\iota_\theta^2}{2\alpha\varphi_\theta} + \frac{\iota}{\alpha}}$$

is positive, where $\alpha \triangleq \frac{1}{2}\min\left\{\varphi_\zeta, \varphi_c, \varphi_a, \varphi_\theta, 2\underline{k_\zeta}\right\}$.

Theorem 3.1. *Provided Assumptions 2.1, 3.1, and 3.2 are satisfied along with the sufficient conditions in* (3–28), (3–29), *and*

$$K < \underline{\upsilon_L}^{-1}\left(\overline{\upsilon_L}\left(r_L\right)\right), \tag{3–30}$$

where $r_L \in \mathbb{R}$ is the radius of the compact set β_L, then the policy in (3–22) *with the update laws in* (2–7)-(2–9) *guarantee uniformly bounded regulation of the state ζ and uniformly bounded convergence of the approximated policies $\hat{u}$ to the optimal policy u^*.*

Proof. The time derivative of the candidate Lyapunov function is

$$\dot{V}_L = \frac{\partial V}{\partial \zeta}\left(Y\theta + f_0\right) + \frac{\partial V}{\partial \zeta} g\left(\hat{u} + \hat{\tau}_c\right) - \tilde{W}_c^T \Gamma^{-1} \dot{\hat{W}}_c - \frac{1}{2}\tilde{W}_c^T \Gamma^{-1}\dot{\Gamma}\Gamma^{-1}\tilde{W}_c - \tilde{W}_a^T \dot{\hat{W}}_a + \dot{V}_P.$$

Substituting (2–7)-(2–9) and (3–17), yields

$$\begin{aligned}\dot{V}_L = \frac{\partial V}{\partial \zeta}\left(Y\theta + f_0\right) + \frac{\partial V}{\partial \zeta} g\left(\hat{u} + \hat{\tau}_c\right) + \tilde{W}_c^T\left[k_{c1}\frac{\omega_t}{\rho_t}\delta_t + \frac{k_{c2}}{N}\sum_{j=1}^{N}\frac{\omega_k}{\rho_k}\delta_k\right] + \tilde{W}_a^T k_a\left(\hat{W}_a - \hat{W}_c\right)\\ - \frac{1}{2}\tilde{W}_c^T\Gamma^{-1}\left[\left(\beta\Gamma - k_{c1}\Gamma\frac{\omega_t {\omega_t}^T}{\rho_t}\Gamma\right)\mathbf{1}_{\|\Gamma\|\leq\overline{\Gamma}}\right]\Gamma^{-1}\tilde{W}_c + \dot{V}_P.\end{aligned}$$

Using Young's inequality, (3–19), (3–20), (3–22), (3–25), and (3–26) the Lyapunov derivative can be upper bounded as

$$\dot{V}_L \leq -\varphi_\zeta \left\|\zeta\right\|^2 - \varphi_c\left\|\tilde{W}_c\right\|^2 - \varphi_a\left\|\tilde{W}_a\right\|^2 - \varphi_\theta\left\|\tilde{\theta}\right\|^2 - \underline{k_\zeta}\left\|\tilde{\zeta}\right\|^2 + \iota_a\left\|\tilde{W}_a\right\| + \iota_c\left\|\tilde{W}_c\right\| + \iota_\theta\left\|\tilde{\theta}\right\| + \iota.$$

Completing the squares, the upper bound on the Lyapunov derivative may be written as

$$\dot{V}_L \leq -\frac{\varphi_\zeta}{2}\left\|\zeta\right\|^2 - \frac{\varphi_c}{2}\left\|\tilde{W}_c\right\|^2 - \frac{\varphi_a}{2}\left\|\tilde{W}_a\right\|^2 - \frac{\varphi_\theta}{2}\left\|\tilde{\theta}\right\|^2 - \underline{k_\zeta}\left\|\tilde{\zeta}\right\|^2 + \frac{\iota_c^2}{2\varphi_c} + \frac{\iota_a^2}{2\varphi_a} + \frac{\iota_\theta^2}{2\varphi_\theta} + \iota,$$

which can be further upper bounded as

$$\dot{V}_L \leq -\alpha\left\|Z\right\|, \; \forall \left\|Z\right\| \geq K > 0. \tag{3–31}$$

Using (3–27), (3–30), and (3–31), Theorem 4.18 in [48] is invoked to conclude that Z is ultimately bounded, in the sense that $\limsup_{t\to\infty}\left\|Z\left(t\right)\right\| \leq \underline{\upsilon_L}^{-1}\left(\overline{\upsilon_L}\left(K\right)\right)$.

Based on the definition of Z and the inequalities in (3–27) and (3–31), $\zeta, \tilde{W}_c, \tilde{W}_a \in \mathcal{L}_\infty$. Using the fact that W is upper bounded by a bounded constant and the definition of the NN weight estimation errors, $\hat{W}_c, \hat{W}_a \in \mathcal{L}_\infty$. Using the policy update laws in (2–9), $\dot{\hat{W}}_a \in \mathcal{L}_\infty$. Since $\hat{W}_c, \hat{W}_a, \zeta \in \mathcal{L}_\infty$ and $\sigma, \nabla\sigma$ are continuous functions of ζ, it follows that $\hat{V}, \hat{u} \in \mathcal{L}_\infty$. From the dynamics in (3–11), $\dot{\zeta} \in \mathcal{L}_\infty$. By the definition in (3–23), $\delta \in \mathcal{L}_\infty$. By the definition of the normalized value function update law in (2–7), $\dot{\hat{W}}_c \in \mathcal{L}_\infty$. □

3.6 Experimental Results

Validation of the proposed controller is demonstrated with experiments conducted at Ginnie Springs in High Springs, FL. Ginnie Springs is a second-magnitude spring discharging 142 million liters of freshwater daily with a spring pool measuring 27.4 m in diameter and 3.7 m deep [49]. Ginnie Springs was selected to validate the proposed controller because of its relatively high flow rate and clear waters for vehicle observation. For clarity of exposition[6] and to remain within the vehicle's depth limitations[7] , the developed method is implemented on an AUV, where the surge, sway, and yaw are controlled by the algorithm developed in (3–24).

3.6.1 Experimental Platform

Experiments were conducted on an AUV, SubjuGator 7, developed at the University of Florida. The AUV, shown in Figure 3-1, is a small two man portable AUV with a mass of 40.8 kg. The vehicle is over-actuated with eight bidirectional thrusters.

[6] The number of basis functions and weights required to support a six DOF model greatly increases from the set required for the three DOF model. The increased number of parameters and complexity reduces the clarity of this proof of principal experiment.

[7] The vehicle's Doppler velocity log has a minimum height over bottom of approximately 3 m that is required to measure water velocity. A minimum depth of approximately 0.5 m is required to remove the vehicle from surface effects. With the depth of the spring nominally 3.7 m, a narrow window of about 20 cm is left operate the vehicle in heave.

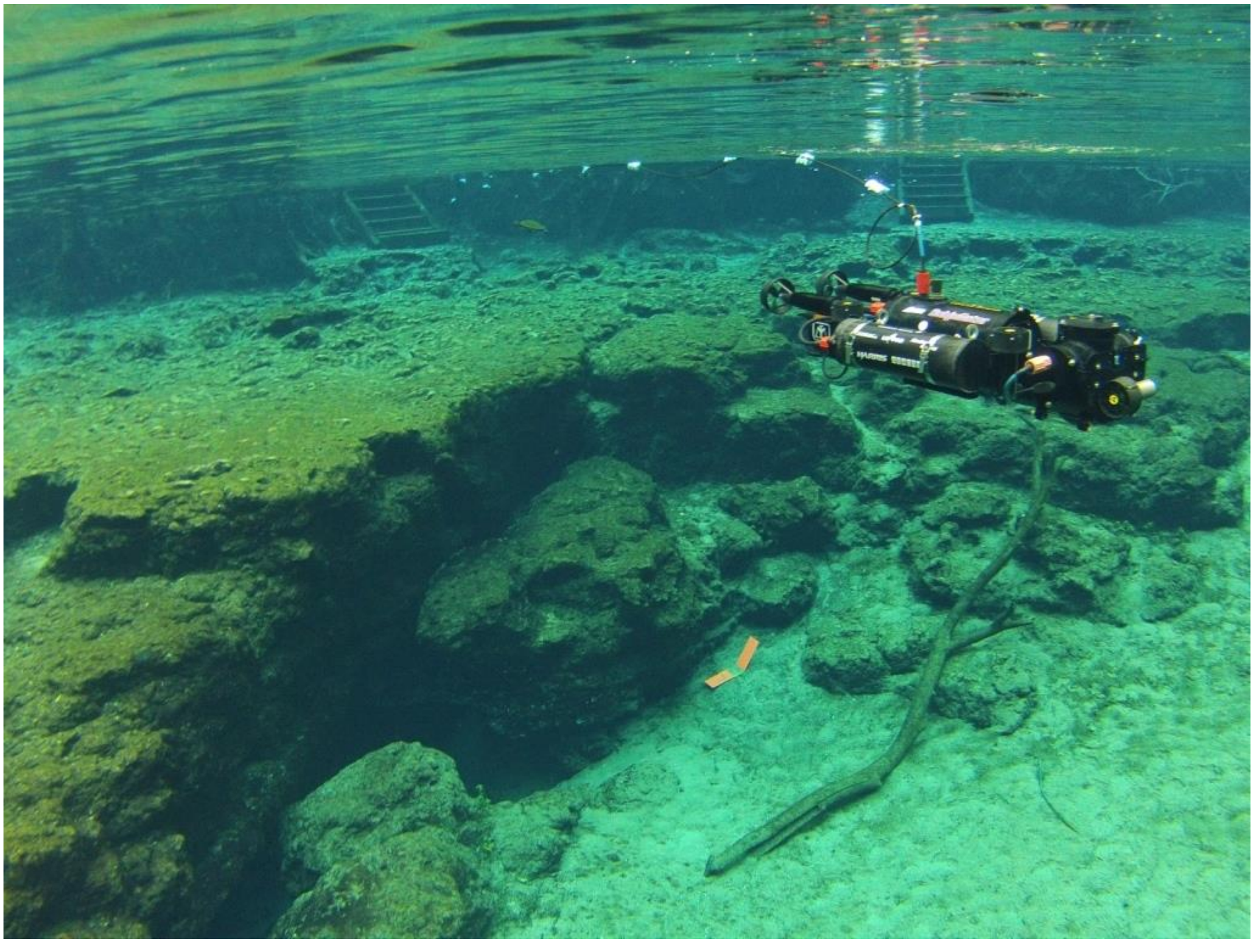

Figure 3-1. SubjuGator 7 AUV operating at Ginnie Springs, FL. Photo courtesy of author.

Designed to be modular, the vehicle has multiple specialized pressure vessels that house computational capabilities, sensors, batteries, and mission specific payloads. The central pressure vessel houses the vehicle's motor controllers, network infrastructure, and core computing capability. The core computing capability services the vehicles environmental sensors (e.g. visible light cameras, scanning sonar, etc.), the vehicles high-level mission planning, and low-level command and control software. A standard small form factor computer makes up the computing capability and utilizes a 2.13 GHz server grade quad-core processor. Located near the front of the vehicle, the navigation vessel houses the vehicle's basic navigation sensors. The suite of navigation sensors include an inertial measurement unit, a Doppler velocity log (DVL), a depth sensor, and a digital compass. The navigation vessel also includes an embedded 720 MHz processor for preprocessing and packaging navigation data. Along the sides of the central pressure vessel, two vessels house 44 Ah of batteries used for propulsion and electronics.

The vehicle's software runs within the Robot Operating System framework in the central pressure vessel. For the experiment, three main software nodes were used: navigation, control, and thruster mapping nodes. The navigation node receives packaged navigation data from the navigation pressure vessel where an extended Kalman filter estimates the vehicle's full state at 50Hz. The controller node contains the proposed controller and system identifier. The desired force and moment produced by the controller are mapped to the eight thrusters using a least squares minimization algorithm in the thruster mapping node.

3.6.2 Controller Implementation

The implementation of the developed method involves: system identification, value function iteration, and control iteration. Implementing the system identifier requires (3–4), (3–5), and the data set described in Assumption 3.2. The data set in Assumption 3.2 was collected in a swimming pool. The vehicle was commanded to track an exciting trajectory with a robust integral of the sign of the error controller [7] while the state-action pairs were recorded. The recorded data was trimmed to a subset of 40 sampled points that were selected to maximize the minimum singular value of $\begin{bmatrix} Y_1 & Y_2 & \ldots & Y_j \end{bmatrix}$ as in Algorithm 1 of [39].

Evaluating the extrapolated Bellman error in (3–23) with each control iteration is computational expensive. Due to the limited computational resources available on-board the AUV, the value function weights were updated at a slower rate (i.e., 5Hz) than the main control loop (implemented at 50 Hz). The developed controller was used to control the surge, sway, and yaw states of the AUV, and a nominal controller was used to regulate the remaining states.

The vehicle uses water profiling data from the DVL to measure the relative water velocity near the vehicle in addition to bottom tracking data for the state estimator. By using the state estimator, water profiling data, and recorded data, the equations used to implement the proposed controller, i.e., (2–7)-(2–9), (3–4), (3–5), (3–22), (3–23), and (3–24), only contain known or measurable quantities.

3.6.3 Results

The vehicle was commanded to hold a station near the vent of Ginnie Spring. An initial condition of

$$\zeta\left(t_0\right) = \begin{bmatrix} 4\,\mathrm{m} & 4\,\mathrm{m} & \frac{\pi}{4}\,\mathrm{rad} & 0\,\mathrm{m/s} & 0\,\mathrm{m/s} & 0\,\mathrm{rad/s} \end{bmatrix}^T$$

was given to demonstrate the method's ability to regulate the state. The optimal control weighting matrices were selected to be $Q = \mathrm{diag}\left([20, 50, 20, 10, 10, 10]\right)$ and $R = I_{3\times3}$. The system identifier adaptation gains were selected to be $k_\zeta = 25 \times I_{6\times6}$, $k_\theta = 12.5$, and $\Gamma_\theta =$ diag $\left([187.5, 937.5, 37.5, 37.5, 37.5, 37.5, 37.5, 37.5]\right)$. The parameter estimate was initialized with $\hat{\theta}\left(t_0\right) = 0_{8\times1}$. The NN weights were initialized to match the ideal values for the linearized optimal control problem, which is obtained by solving the algebraic Riccati equation with the dynamics linearized about the station. The policy adaptation gains were selected to be $k_{c1} = 0.25$, $k_{c2} = 0.5$, $k_a = 1$, $k_\rho = 0.25$, and $\beta = 0.025$. The adaptation matrix was initialized to $\Gamma_0 = 400 \times I_{21\times 21}$. The Bellman error was extrapolated to 2025 sampled states. The sampled states where uniformly selected throughout the state space in the vehicle's operating domain.

Figures 3-2 and 3-3 illustrate the ability of the generated policy to regulate the state in the presence of the spring's current. Figure 3-6 illustrates the total control effort applied to the body of the vehicle, which includes the estimate of the current compensation term and approximate optimal control. Figure 3-8 illustrates the output of the approximate optimal policy for the residual system. Figure 3-7 illustrates the convergence of the parameters of the system identifier, and Figure 3-4 and 3-5 illustrate convergence of the NN weights representing the value function.

The anomaly seen at ~70 seconds in the total control effort, Figure 3-6, is attributed to a series of incorrect current velocity measurements. The corruption of the current velocity measurements is possibly due in part to the extremely low turbidity in the spring and/or relatively shallow operating depth. Despite the presence of unreliable current velocity measurements the vehicle was able to regulate the vehicle to its station. The results demonstrate the developed method's ability to concurrently identify the unknown hydrodynamic parameters and generate an approximate optimal policy using the identified model. The vehicle follows the generated policy to achieve its station keeping objective using industry standard navigation and environmental sensors (i.e., inertial measurement unit, DVL).

3.7 Summary

The online approximation of an optimal control strategy is developed to enable station keeping by a marine craft. The solution to the HJB equation is approximated using ADP. The hydrodynamic effects are identified online with a CL-based system identifier. Leveraging the identified model, the developed strategy simulates exploration of the state space to learn the optimal policy without the need of a persistently exciting trajectory. A Lyapunov-based stability analysis concludes uniformly bounded convergence of the states and of the approximated policies to the optimal polices. Experiments in a central Florida second-magnitude spring demonstrate the ability of the controller to generate and execute an approximate optimal policy in the presence of a time-varying irrotational current.

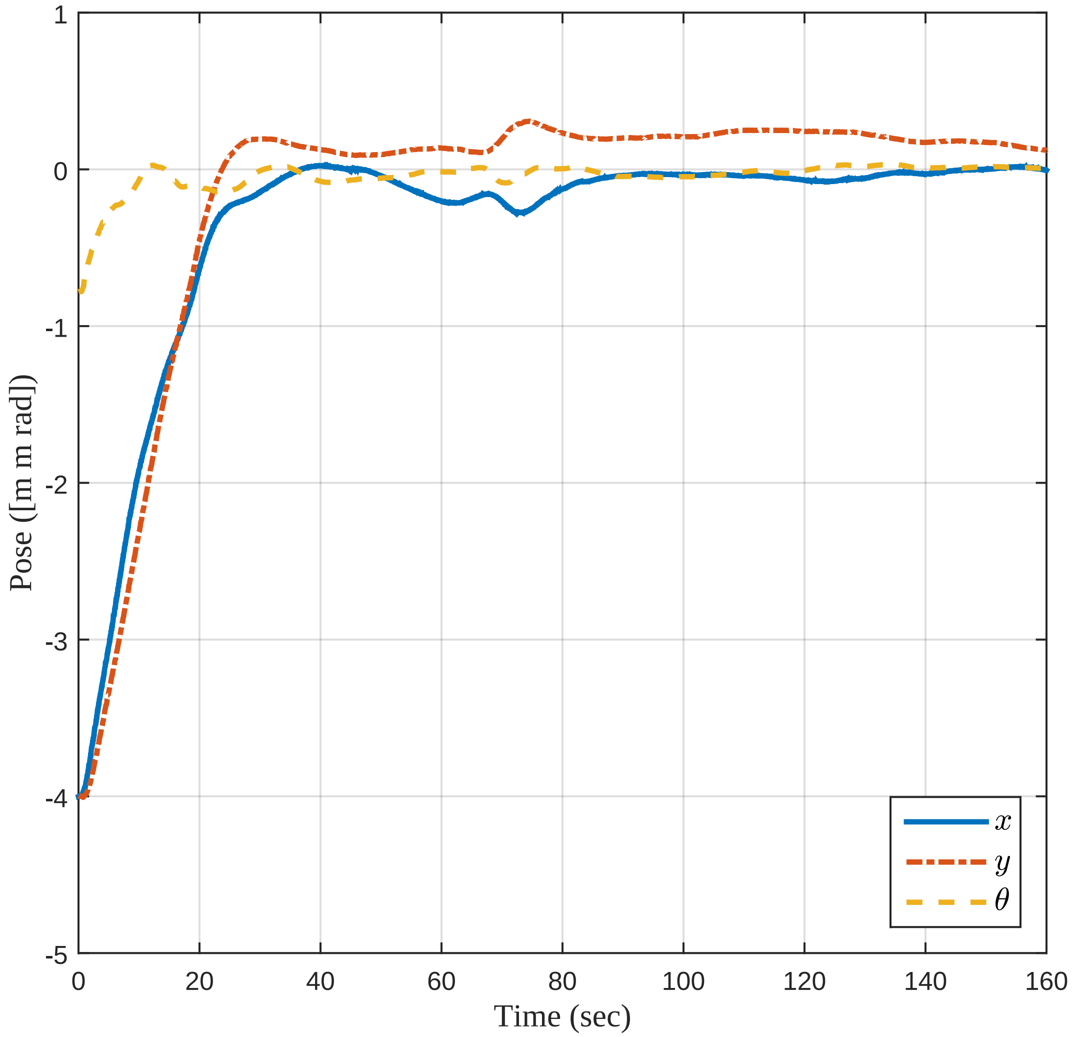

Figure 3-2. Positional error state trajectory generated by developed station keeping method implemented on SubjuGator.

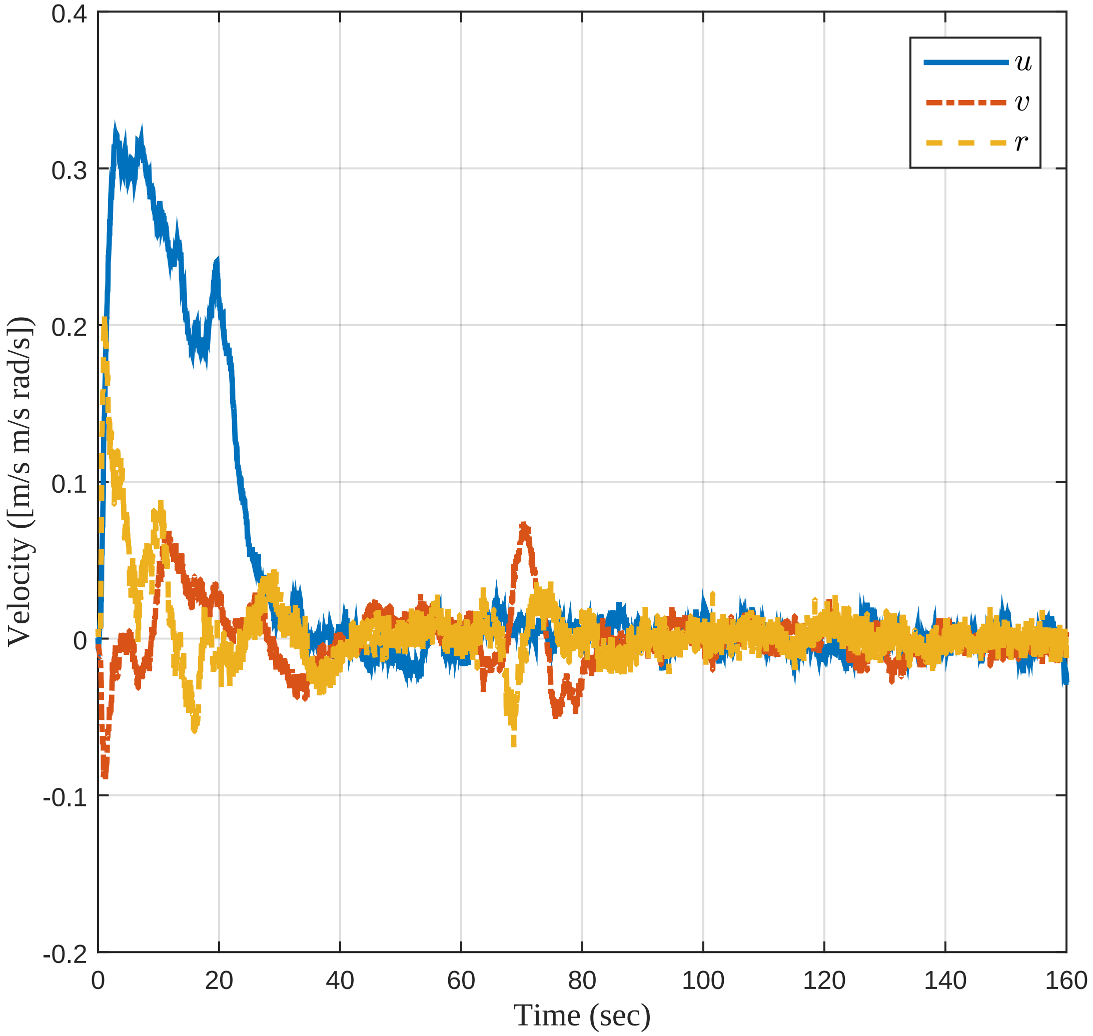

Figure 3-3. Velocity error state trajectory generated by developed station keeping method implemented on SubjuGator.

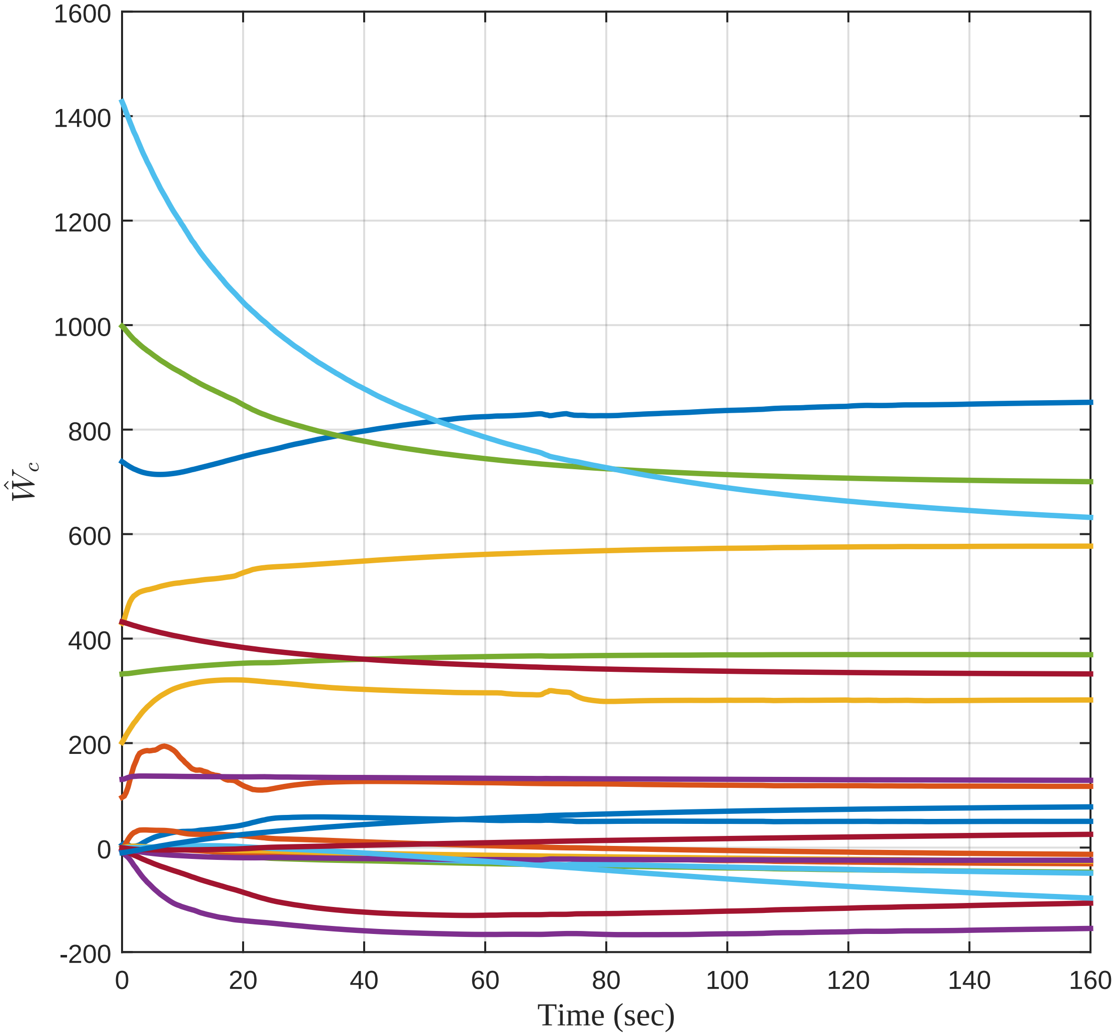

Figure 3-4. Estimated critic weight trajectories generated by developed station keeping method implemented on SubjuGator.

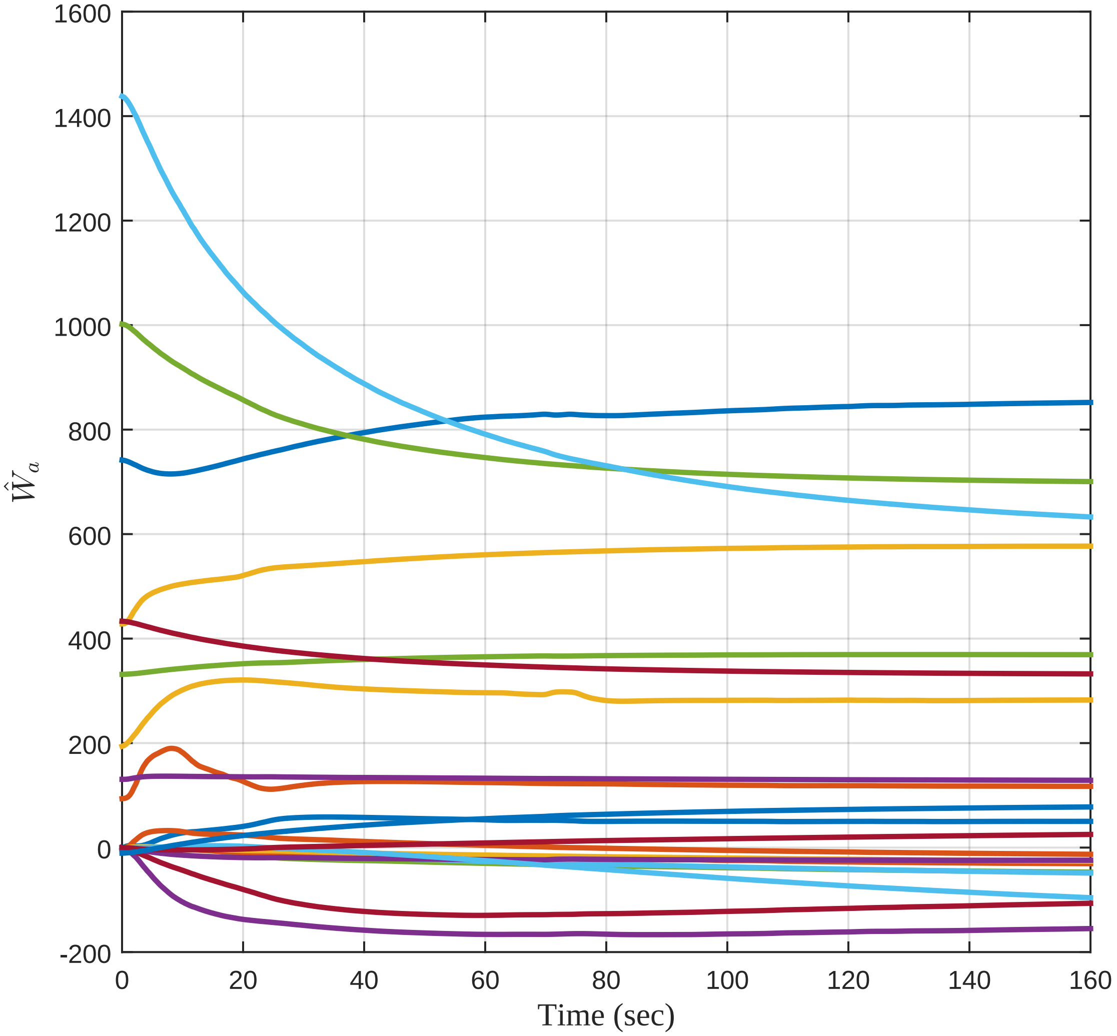

Figure 3-5. Estimated actor weight trajectories generated by developed station keeping method implemented on SubjuGator.

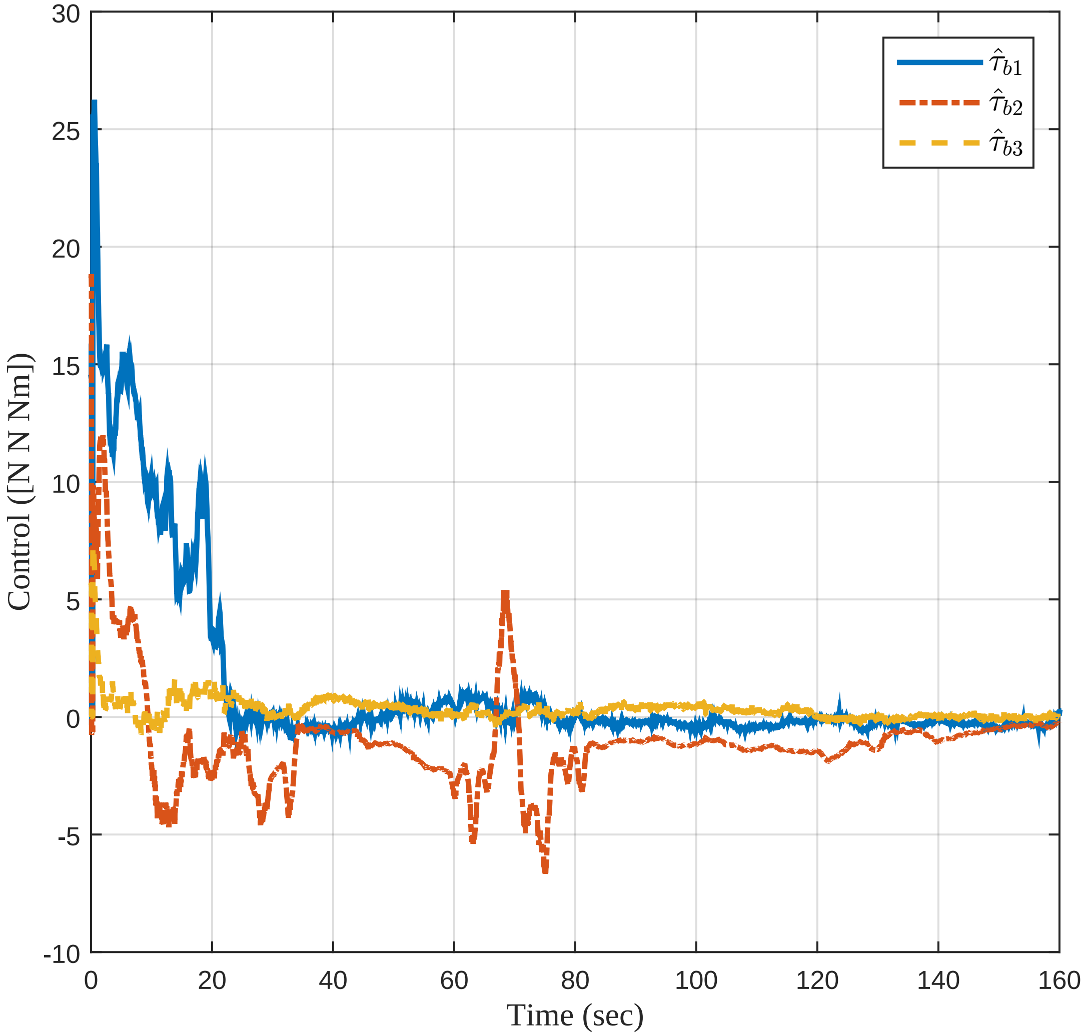

Figure 3-6. Combined control trajectory generated by developed station keeping method implemented on SubjuGator.

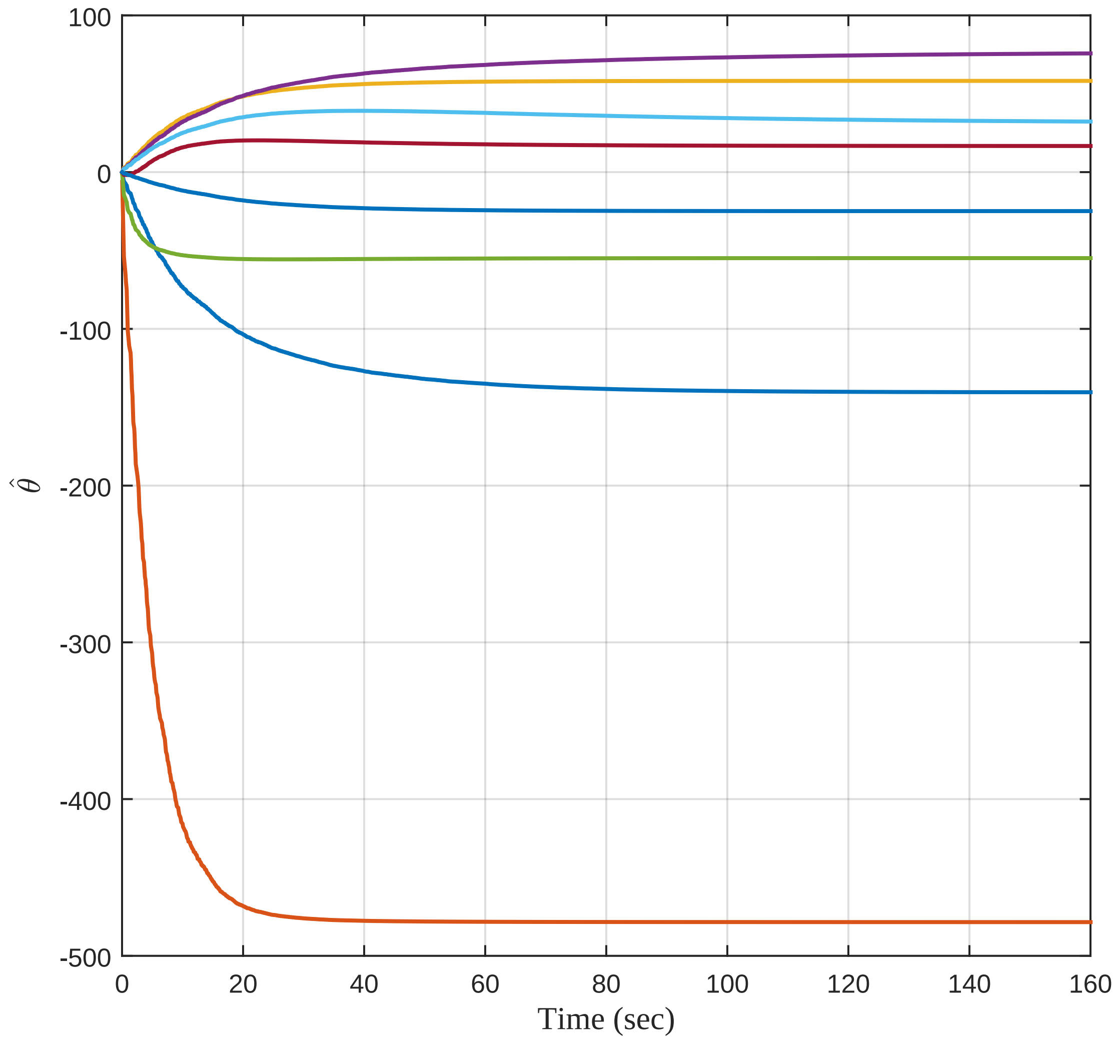

Figure 3-7. Identified system parameters determined for SubjuGator online.

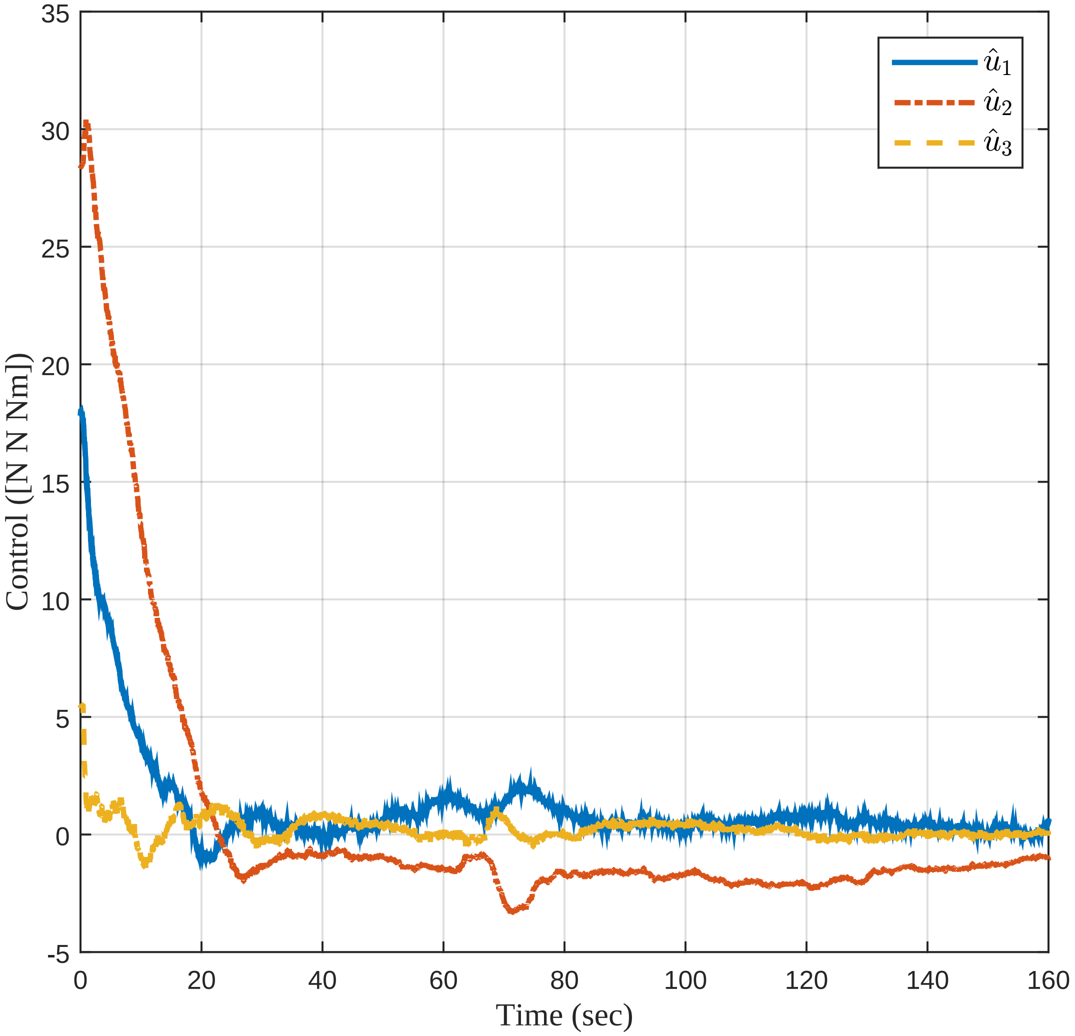

Figure 3-8. Optimal control trajectory generated by developed station keeping method implemented on SubjuGator.

CHAPTER 4
PLANAR PATH FOLLOWING GUIDANCE LAW

The focus of this chapter is to develop a guidance law for approximate optimal path following for an underactuated marine craft. Path following refers to a class of problems where the control objective is to converge to a desired geometric path. The desired path is not necessarily parametrized by time, but by some convenient parameter, e.g., path length. The path following method in this chapter utilizes a virtual target that moves along the desired path. The optimal path following problem is formulated in terms of the HJB equation. An ADP-based controller is developed to approximate the solution to the HJB equation.

4.1 Kinematic Model

The geometry of the path following problem is depicted in Figure 4-1. Let $\mathcal{I}$ denote an inertial frame. Consider the coordinate system i in $\mathcal{I}$ with its origin and the basis vectors $i_1 \in \mathbb{R}^3$ and $i_2 \in \mathbb{R}^3$ in the plane of craft motion. The basis vector i_3 is defined as coming out of the plane. The point $P \in \mathbb{R}^3$ on the desired path represents the location of the virtual target. The location of the virtual target is determined by the path parameter $s_p \in \mathbb{R}$. It is convenient to select the arc length as the path parameter, since the desired speed can be defined as unit length per unit time. Let $\mathcal{F}$ denote a frame fixed to the virtual target with the origin of the coordinate system f fixed in $\mathcal{F}$ at point P. The basis vector $f_1 \in \mathbb{R}^3$ is the unit tangent vector of the path at P, $f_3 \in \mathbb{R}^3$ is defined as coming out of the plane, and $f_2 = f_3 \times f_1$. Let $\mathcal{B}$ denote a frame fixed to the craft with the origin of its coordinate system b at the center of mass $M \in \mathbb{R}^3$. The basis vector $b_1 \in \mathbb{R}^3$ is the unit velocity vector of the craft, $b_3 \in \mathbb{R}^3$ is defined as coming out of the plane, and $b_2 = b_3 \times b_1$. Note, the bases $\{i_1, i_2, i_3\}$, $\{f_1, f_2, f_3\}$, and $\{b_1, b_2, b_3\}$ form standard bases.

Consider the following vector equation from Figure 4-1,

$$r_{M/P} = r_M - r_P,$$

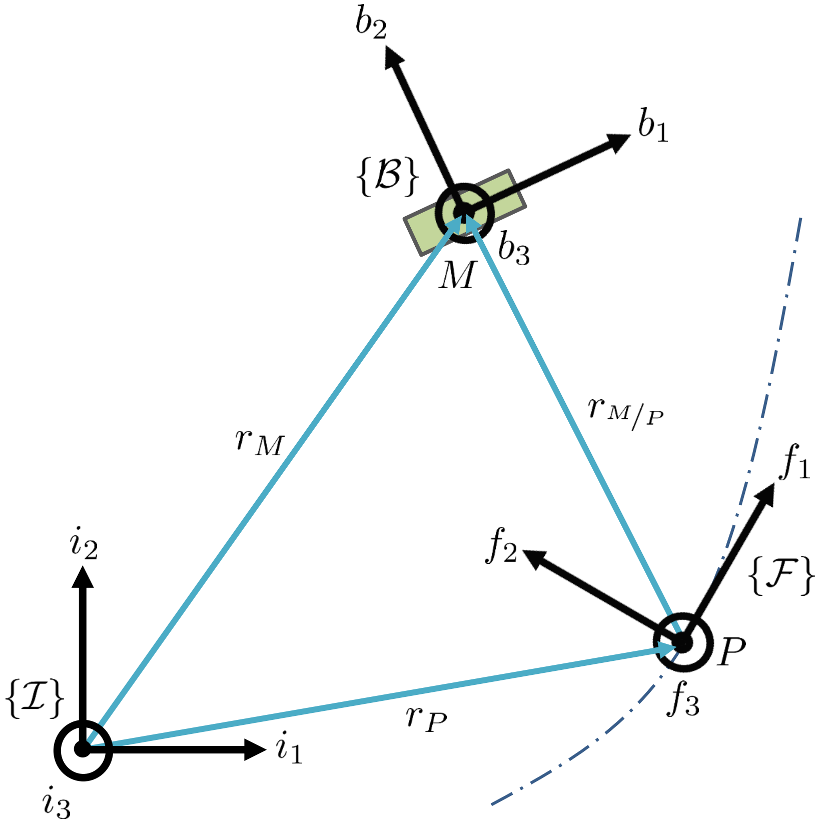

Figure 4-1. Geometric description of path following problem.

where $r_M \in \mathbb{R}^3$ and $r_P \in \mathbb{R}^3$ are the position vectors of points M and P from the origin of the inertial coordinate system, respectively. The rate of change of $r_{M/P}$ as viewed by an observer in $\mathcal{I}$ and expressed in the coordinate system f is given as

$$v^f_{M/P} = v^f_M - v^f_P. \tag{4–1}$$

The velocity of point P as viewed by an observer in $\mathcal{I}$ and expressed in f is given as

$$v^f_P = \begin{bmatrix} \dot{s}_p & 0 & 0 \end{bmatrix},^T \tag{4–2}$$

where $\dot{s}_p \in \mathbb{R}$ is the velocity of the virtual target along the path. The velocity of point M as viewed by an observer in $\mathcal{I}$ and expressed in f may be written as

$$v^f_M = R^f_b v^b_M,$$

where $R^f_b : \mathbb{R} \to \mathbb{R}^{3\times 3}$ is a transformation from b to f, defined as

$$R^f_b \triangleq \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where $\theta \in \mathbb{R}$ is the angle between f_1 and b_1. The velocity of the craft as viewed by an observer in $\mathcal{I}$ expressed in b is $v^b_M = \begin{bmatrix} v & 0 & 0 \end{bmatrix}^T$ where $v \in \mathbb{R}$ is the velocity of the craft. The velocity between points P and M as viewed by an observer in $\mathcal{I}$ and expressed in f is given as

$$v^f_{M/P} = {}^{\mathcal{F}}\tfrac{d}{dt} r^f_{M/P} + {}^{\mathcal{I}}\omega^{\mathcal{F}} \times r^f_{M/P}. \tag{4–3}$$

The angular velocity of $\mathcal{F}$ as viewed by an observer in $\mathcal{I}$ expressed in f is given as ${}^{\mathcal{I}}\omega^{\mathcal{F}} = \begin{bmatrix} 0 & 0 & \kappa\dot{s}_p \end{bmatrix}^T$, where $\kappa \in \mathbb{R}$ is the path curvature, and the relative position of the craft with respect to the virtual target expressed in f is given as $r^f_{M/P} = \begin{bmatrix} x & y & 0 \end{bmatrix}^T$.

Substituting (4–2) and (4–3) into (4–1) yields the planar positional error dynamics

$$\begin{aligned}\dot{x} &= (\kappa y - 1)\,\dot{s}_p + v\cos\theta \\ \dot{y} &= -\kappa x\dot{s}_p + v\sin\theta.\end{aligned}$$

The angular velocity of $\mathcal{B}$ as viewed by an observer in $\mathcal{F}$ is given as

$$^{\mathcal{F}}\omega^{\mathcal{B}} = {}^{\mathcal{F}}\omega^{\mathcal{I}} + {}^{\mathcal{I}}\omega^{\mathcal{B}}. \tag{4–4}$$

From (4–4), the planar rotational error dynamic expressed in f is given as

$$\dot{\theta} = -\kappa\dot{s}_p + w,$$

where $w \in \mathbb{R}$ is the angular velocity of the craft. The full craft error dynamics are given by [12]

$$\begin{aligned}\dot{x} &= \dot{s}_p\,(\kappa y - 1) + v\cos\theta \\ \dot{y} &= -x\kappa\dot{s}_p + v\sin\theta \\ \dot{\theta} &= \omega - \kappa\dot{s}_p.\end{aligned}$$

Assumption 4.1. The desired path is regular and C^2 continuous; hence, the path curvature κ is bounded and continuous.

As described in [12], the location of the virtual target is determined by

$$\dot{s}_p \triangleq v_{des}\cos\theta + k_1 x, \tag{4–6}$$

where $v_{des} \in \mathbb{R}$ is a desired positive, bounded and time-invariant speed profile that does not exceed the maximum speed of the craft, and $k_1 \in \mathbb{R}$ is an adjustable positive gain.

To facilitate the subsequent control development, an auxiliary function $\phi : \mathbb{R} \to (-1, 1)$ is defined as

$$\phi \triangleq \tanh\left(k_2 s_p\right), \tag{4–7}$$

where $k_2 \in \mathbb{R}$ is a positive gain. From (4–6) and (4–7), the time derivative of ϕ is

$$\dot{\phi} = k_2 \left(1 - \phi^2\right) \left(v_{des} \cos\theta + k_1 x\right). \tag{4–8}$$

Note that the path curvature and desired speed profile can be written as functions of ϕ. Based on (4–5) and (4–6), auxiliary control inputs $v_e, w_e \in \mathbb{R}$ are designed as

$$\begin{aligned} v_e &\triangleq v - v_{ss}, \\ w_e &\triangleq w - w_{ss}, \end{aligned}$$

where $w_{ss} \triangleq \kappa v_{des}$ and $v_{ss} \triangleq v_{des}$ are computed based on the control input required to remain on the path. Substituting (4–6) and (4–9) into (4–5), and augmenting the system state with (4–8), the closed-loop system is

$$\begin{aligned} \dot{x} &= \kappa y v_{des} \cos\theta + k_1 \kappa x y - k_1 x + v_e \cos\theta \\ \dot{y} &= v_{des} \sin\theta - \kappa x v_{des} \cos\theta - k_1 \kappa x^2 + v_e \sin\theta \\ \dot{\theta} &= \kappa v_{des} - \kappa \left(v_{des} \cos\theta + k_1 x\right) + w_e \\ \dot{\phi} &= k_2 \left(1 - \phi^2\right) \left(v_{des} \cos\theta + k_1 x\right). \end{aligned}$$

The closed-loop system in (4–10) can be rewritten in the following control affine form

$$\dot{\zeta} = f(\zeta) + g(\zeta) u, \tag{4–11}$$

where $\zeta = \begin{bmatrix} x & y & \theta & \phi \end{bmatrix}^T \in \mathbb{R}^4$ is the state vector, $u = \begin{bmatrix} v_e & w_e \end{bmatrix}^T \in \mathbb{R}^2$ is the control vector, and the locally Lipschitz functions $f : \mathbb{R}^4 \to \mathbb{R}^4$ and $g : \mathbb{R}^4 \to \mathbb{R}^{4\times 2}$ are defined as

$$f(\zeta) \triangleq \begin{bmatrix} \kappa y v_{des} \cos\theta + k_1 \kappa x y - k_1 x \\ v_{des} \sin\theta - \kappa x v_{des} \cos\theta - k_1 \kappa x^2 \\ \kappa v_{des} - \kappa \left(v_{des} \cos\theta + k_1 x\right) \\ k_2 \left(1 - \phi^2\right) \left(v_{des} \cos\theta + k_1 x\right) \end{bmatrix}, \; g(\zeta) \triangleq \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}. \tag{4–12}$$

To facilitate the subsequent stability analysis, a subset of the state denoted by $e \in \mathbb{R}^3$ is defined as $e = \begin{bmatrix} x & y & \theta \end{bmatrix}^T \in \mathbb{R}^3$.

4.2 Problem Formulation

The cost functional for the optimal control problem is defined as

$$J(\zeta, u) \triangleq \int_{t_0}^{\infty} r(\zeta(\tau), u(\tau))\, d\tau, \tag{4–13}$$

where t_0 denotes the initial time, and $r : \mathbb{R}^4 \to [0, \infty)$ is the local cost defined as

$$r(\zeta, u) \triangleq \zeta^T \bar{Q} \zeta + u^T R u.$$

In (4–13), $R \in \mathbb{R}^{2\times 2}$ is a symmetric positive definite matrix, and $\bar{Q} \in \mathbb{R}^{4\times 4}$ is defined as

$$\bar{Q} \triangleq \begin{bmatrix} Q & 0_{3\times 1} \\ 0_{1\times 3} & 0 \end{bmatrix},$$

where $Q \in \mathbb{R}^{3\times 3}$ is a positive definite matrix such that $\underline{q} \left\|\xi_q\right\|^2 \leq \xi_q^T Q \xi_q \leq \overline{q} \left\|\xi_q\right\|^2, \forall \xi_q \in \mathbb{R}^3$ where $\underline{q}$ and $\overline{q}$ are positive constants. The infinite-time scalar value function $V : \mathbb{R}^4 \to [0, \infty)$ is written as

$$V(\zeta) = \min_{u \in \mathcal{U}} \int_{t_0}^{\infty} r(\zeta(\tau), u(\tau))\, d\tau,$$

where $\mathcal{U}$ is the set of admissible control policies. The objective of the optimal control problem is to find the optimal policy $u^* : \mathbb{R}^{2n} \to \mathbb{R}^n$ that minimizes the performance index (4–13) subject to the dynamic constraints in (4–11). The optimal value function is characterized by the HJB equation, which is given as

$$\nabla V(\zeta)\left(f(\zeta) + g(\zeta) u^*(\zeta)\right) + r(\zeta, u^*(\zeta)) = 0 \tag{4–14}$$

with the boundary condition $V(0) = 0$. Provided the HJB equation admits a continuously differentiable solution, the HJB equation constitutes a necessary and sufficient condition for optimality. The optimal control policy can be determined from (4–14) as

$$u^{*}\left(\zeta\right)=-\frac{1}{2}R^{-1}g\left(\zeta\right)^{T}\nabla V\left(\zeta\right)^{T}. \tag{4–15}$$

The analytical expression for the optimal controller in (4–15) requires knowledge of the value function which is the solution to the HJB equation. Given the kinematics in (4–12), it is unclear how to determine an analytical solution to (4–14), as is generally the case since (4–14) is a partial differential equation; hence, the subsequent development focuses on the development of an approximate solution.

4.3 Approximate Solution

The subsequent development is based on a NN approximation of the value function and the optimal policy. Over any compact domain $\chi\subset\mathbb{R}^{4}$, the value function $V:\mathbb{R}^{4}\to[0,\infty)$ can be represented by a single-layer NN with l neurons as

$$V\left(\zeta\right)=W^{T}\sigma\left(\zeta\right)+\epsilon\left(\zeta\right), \tag{4–16}$$

where $W\in\mathbb{R}^{l}$ is the constant ideal weight vector bounded above by a known positive constant, $\sigma:\mathbb{R}^{4}\to\mathbb{R}^{l}$ is a bounded, continuously differentiable activation function, and $\epsilon:\mathbb{R}^{4}\to\mathbb{R}$ is the bounded, continuously differentiable function reconstruction error. From (4–15) and (4–16), the optimal policy can be represented as

$$u^{*}\left(\zeta\right)=-\frac{1}{2}R^{-1}g\left(\zeta\right)^{T}\left(\nabla\sigma\left(\zeta\right)^{T}W+\nabla\epsilon\left(\zeta\right)^{T}\right), \tag{4–17}$$

Based on (4–16) and (4–17), the value function and optimal policy NN approximations are defined as

$$\hat{V}\left(\zeta,\hat{W}_{c}\right)=\hat{W}_{c}^{T}\sigma\left(\zeta\right), \tag{4–18}$$

$$\hat{u}\left(\zeta,\hat{W}_{a}\right)=-\frac{1}{2}R^{-1}g\left(\zeta\right)^{T}\nabla\sigma\left(\zeta\right)^{T}\hat{W}_{a}, \tag{4–19}$$

where $\hat{W}_{c},\ \hat{W}_{a}\in\mathbb{R}^{l}$ are estimates of the ideal weight vector W. The weight estimation errors are defined as $\tilde{W}_{c}\triangleq W-W_{c}$ and $\tilde{W}_{a}\triangleq W-W_{a}$. Substituting (4–18) and (4–19) into (4–14), results in a residual error $\delta:\mathbb{R}^{4}\times\mathbb{R}^{l}\times\mathbb{R}^{l}\to\mathbb{R}$ called the Bellman error given as

$$\delta\left(\zeta,\hat{W}_c,\hat{W}_a\right)=r\left(\zeta,\hat{u}\left(\zeta,\hat{W}_a\right)\right)+\hat{W}_c^T\omega\left(\zeta,\hat{W}_a\right),$$

where $\omega:\mathbb{R}^4\to\mathbb{R}^l$ is given by

$$\omega\left(\zeta,\hat{W}_a\right)=\nabla\sigma\left(\zeta\right)\left[f\left(\zeta\right)+g\left(\zeta\right)\hat{u}\left(\zeta,\hat{W}_a\right)\right].$$

The online implementation of the approximation is presented in Section 2.5, where the parameters $\hat{W}_c$ and $\hat{W}_a$ are updated by (2–7) and (2–9), respectively.

4.4 Stability Analysis

For notational brevity, all function dependencies from previous sections will be henceforth suppressed. To facilitate the subsequent stability analysis, an unmeasurable from of the Bellman error can be written as

$$\delta=-\tilde{W}_c^T\omega-\nabla\epsilon f+\frac{1}{2}\nabla\epsilon G\nabla\sigma^TW+\frac{1}{4}\tilde{W}_a^TG_\sigma\tilde{W}_a+\frac{1}{4}\nabla\epsilon G\nabla\epsilon^T, \tag{4–20}$$

where $G\triangleq gR^{-1}g^T\in\mathbb{R}^{4\times4}$ and $G_\sigma\triangleq\nabla\sigma G\nabla\sigma^T\in\mathbb{R}^{l\times l}$ are symmetric, positive semi-definite matrices. Similarly, at the sampled points the Bellman error can be written as

$$\delta_k=-\tilde{W}_c^T\omega_k+\frac{1}{4}\tilde{W}_a^TG_{\sigma k}\tilde{W}_a+E_k, \tag{4–21}$$

where

$$E_k\triangleq\frac{1}{2}\nabla\epsilon_kG_k\nabla\sigma_k^TW+\frac{1}{4}\nabla\epsilon_kG_k\nabla\epsilon_k^T-\nabla\epsilon_kf_k\in\mathbb{R}$$

is a constant at each data point, and the notation F_k denotes the function $F\left(\zeta,\cdot\right)$ evaluated at a sampled state, i.e., $F_k\left(\cdot\right)=F\left(\zeta_k,\cdot\right)$. The function in (4–12) on any compact set $\chi\subset\mathbb{R}^4$ is Lipschitz continuous, and bounded by

$$\left\|f\right\|\leq L_f\left\|\zeta\right\|,\ \forall\zeta\in\chi,$$

where L is a positive constant.

The augmented equations of motion in (4–10) present a unique challenge with respect to the value function V which is utilized as a Lyapunov function in the stability analysis. To prevent penalizing the craft progression along the path, the path parameter ϕ is removed from the cost function with the introduction of a positive semi-definite state weighting matrix $\bar{Q}$. However, since $\bar{Q}$ is positive semi-definite, efforts are required to ensure the value function is positive definite. To address this challenge, the fact that the value function can be interpreted as a time-invariant map $V : \mathbb{R}^4 \to [0, \infty)$ or a time-varying map $V : \mathbb{R}^3 \times [0, \infty) \to [0, \infty)$ is exploited. Lemma 2 in [24] is used to show that the time-varying map is a positive definite and decrescent function for use as a Lyapunov function. Hence, on any compact set χ the optimal value function $V : \mathbb{R}^3 \times [0, \infty) \to \mathbb{R}$ satisfies the following properties

$$V(0,t) = 0,$$

$$\underline{\upsilon}(\|e\|) \leq V(e,t) \leq \overline{\upsilon}(\|e\|), \tag{4–22}$$

$\forall t \in [0, \infty)$ and $\forall e \subset \chi$ where $\underline{\upsilon} : [0, \infty] \to [0, \infty)$ and $\overline{\upsilon} : [0, \infty] \to [0, \infty)$ are class $\mathcal{K}$ functions. To facilitate the subsequent stability analysis, consider the Lyapunov function candidate $V_L : \mathbb{R}^4 \times \mathbb{R}^l \times \mathbb{R}^l \times [0, \infty) \to [0, \infty)$ given as

$$V_L(Z,t) = V(e,t) + \frac{1}{2}\tilde{W}_c^T \Gamma^{-1} \tilde{W}_c + \frac{1}{2}\tilde{W}_a^T \tilde{W}_a. \tag{4–23}$$

where $Z \triangleq \begin{bmatrix} e^T & \tilde{W}_c^T & \tilde{W}_a^T \end{bmatrix}^T \in \chi \times \mathbb{R}^l \times \mathbb{R}^l$. The function V_L can be bounded by

$$\underline{\upsilon_L}(\|Z\|) \leq V_L(Z) \leq \overline{\upsilon_L}(\|Z\|) \tag{4–24}$$

using Lemma 4.3 of [48] and (4–22), where $\underline{\upsilon_L}, \overline{\upsilon_L} : [0, \infty) \to [0, \infty)$ are class $\mathcal{K}$ functions. Let $\beta_L \subset \chi \times \mathbb{R}^l \times \mathbb{R}^l \times \mathbb{R}^p$ be a compact set, where the notation $\overline{\|(\cdot)\|}$ is defined as $\overline{\|(\cdot)\|} = \sup_{Z \in \beta_L} \|(\cdot)\|$, then

$$\varphi_e \triangleq \underline{q} - \frac{k_{c1}\overline{\|\nabla\epsilon\|}L_f}{2},$$

$$\varphi_c \triangleq \frac{k_{c2}}{N}\underline{c} - \frac{k_a}{2} - \frac{k_{c1}\overline{\|\nabla\epsilon\|}L_f}{2},$$

$$\varphi_a \triangleq \frac{k_a}{2},$$

$$\iota_c \triangleq \overline{\left\| \frac{k_{c2}}{4N}\sum_{k=1}^{N}\tilde{W}_a^T G_{\sigma k}\tilde{W}_a + \frac{k_{c1}}{4}\tilde{W}_a^T G_{\sigma}\tilde{W}_a + \frac{k_{c1}}{2}\nabla\epsilon G\nabla\sigma^T W + \frac{k_{c1}}{4}\nabla\epsilon G\nabla\epsilon^T + \frac{k_{c2}}{N}\sum_{k=1}^{N}E_k + k_{c1}\nabla\epsilon L_f \right\|},$$

$$\iota_a \triangleq \overline{\left\| \frac{1}{2}G_\sigma W + \frac{1}{2}\nabla\sigma G\nabla\epsilon^T \right\|},$$

$$\iota \triangleq \overline{\left\| \frac{1}{4}\nabla\epsilon G\nabla\epsilon^T \right\|}.$$

When Assumption 2.1 and the sufficient gain conditions

$$\underline{q} > \frac{k_{c1}\overline{\|\nabla\epsilon\|}L_f}{2}, \tag{4–25}$$

$$\underline{c} > \frac{Nk_a}{2k_{c2}} + \frac{Nk_{c1}\overline{\|\nabla\epsilon\|}L_f}{2k_{c2}} \tag{4–26}$$

are satisfied, the constant $K \in \mathbb{R}$ defined as

$$K \triangleq \sqrt{\frac{\iota_c^2}{2\alpha\varphi_c} + \frac{\iota_a^2}{2\alpha\varphi_a} + \frac{\iota}{\alpha}}$$

is positive, where $\alpha \triangleq \frac{1}{2}\min\left\{2\varphi_e, \varphi_c, \varphi_a\right\}$.

Theorem 4.1. *Provided Assumptions 2.1 and 4.1 are satisfied along with the sufficient conditions in* (4–25), (4–26), *and*

$$K < \overline{\upsilon_L}^{-1}\left(\underline{\upsilon_L}\left(r_L\right)\right), \tag{4–27}$$

where $r_L \in \mathbb{R}$ is the radius of a selected compact set β_L, then the policy in (4–19) *with the update laws in* (2–7)-(2–9) *guarantee ultimately bounded convergence of the approximate policy to the optimal policy and of the craft to the virtual target.*

Proof. The time derivative of the Lyapunov function candidate in (4–23) is

$$\dot{V}_L = \frac{\partial V}{\partial \zeta} f + \frac{\partial V}{\partial \zeta} g\hat{u} - \tilde{W}_c^T \Gamma^{-1} \dot{\hat{W}}_c - \frac{1}{2} \tilde{W}_c^T \Gamma^{-1} \dot{\Gamma} \Gamma^{-1} \tilde{W}_c - \tilde{W}_a^T \dot{\hat{W}}_a.$$

Substituting (2–7)-(2–9), and (4–14) yields

$$\dot{V}_L = -e^T Q e - u^* R u^* + \frac{\partial V}{\partial \zeta} g\hat{u} - \frac{\partial V}{\partial \zeta} g u^* + \tilde{W}_c^T \left[k_{c1} \frac{\omega_t}{\rho_t} \delta_t + \frac{k_{c2}}{N} \sum_{k=1}^{N} \frac{\omega_k}{p_k} \delta_k \right]$$
$$+ \tilde{W}_a^T k_a \left(\hat{W}_a - \hat{W}_c \right) - \frac{1}{2} \tilde{W}_c^T \Gamma^{-1} \left[\left(\beta \Gamma - k_{c1} \Gamma \frac{\omega_t {\omega_t}^T}{\rho_t} \Gamma \right) \mathbf{1}_{\|\Gamma\| \le \overline{\Gamma}} \right] \Gamma^{-1} \tilde{W}_c.$$

Using Young's inequality, (4–16), (4–17), (4–19), (4–20), and (4–21) yields

$$\dot{V}_L \le -\varphi_e \|e\|^2 - \varphi_c \left\| \tilde{W}_c \right\|^2 - \varphi_a \left\| \tilde{W}_a \right\|^2 + \iota_c \left\| \tilde{W}_c \right\| + \iota_a \left\| \tilde{W}_{a1} \right\| + \iota. \tag{4–28}$$

Completing the squares, (4–28) can be upper bounded by

$$\dot{V}_L \le -\varphi_e \|e\|^2 - \frac{\varphi_c}{2} \left\| \tilde{W}_c \right\|^2 - \frac{\varphi_a}{2} \left\| \tilde{W}_a \right\|^2 + \frac{\iota_c^2}{2\varphi_c} + \frac{\iota_a^2}{2\varphi_a} + \iota,$$

which can be further upper bounded as

$$\dot{V}_L \le -\alpha \|Z\|^2, \forall \|Z\| \ge K > 0. \tag{4–29}$$

Using (4–24), (4–27), and (4–29), Theorem 4.18 in [48] is invoked to conclude that Z is ultimately bounded, in the sense that $\limsup_{t \to \infty} \|Z(t)\| \le \underline{\upsilon_L}^{-1} \left(\overline{\upsilon_L}(K) \right)$.

Based on the definition of Z, and the inequalities in (4–24) and (4–29), $e,\ \tilde{W}_c,\ \tilde{W}_a \in \mathcal{L}_\infty$. Since $\phi \in \mathcal{L}_\infty$ by definition in (4–8), then $\zeta \in \mathcal{L}_\infty$. $\hat{W}_c,\ \hat{W}_a \in \mathcal{L}_\infty$ follows from the definition of W. From (4–18) and (4–19), $\hat{V},\ \hat{u} \in \mathcal{L}_\infty$. From (4–11), $\dot{\zeta} \in \mathcal{L}_\infty$. By the definition in (3–23), $\delta \in \mathcal{L}_\infty$. From (2–7) and (2–9), $\dot{\hat{W}}_a,\ \dot{\hat{W}}_c \in \mathcal{L}_\infty$. □

4.5 Simulation and Experimental Results

To demonstrate the performance of the developed ADP-based guidance law, simulation and experimental results are presented. As a kinematic analog to a underactuated marine craft, the simulation and experimental trials are conducted on a differential steering wheeled mobile robot. Simulations allow the developed method to be compared to other optimal solutions, whereas the experimental results demonstrate the real-time optimal performance. For both, the craft is commanded to follow a figure eight path with a desired speed of $v_{des} = 0.25\,\mathrm{m/s}$. The virtual target is initially placed at the position corresponding to an initial path parameter of $s_p(0) = 0\,\mathrm{m}$, and the initial error state is selected as

$$e(0) = \begin{bmatrix} -0.5\,\mathrm{m} & -0.5\,\mathrm{m} & \pi/2\,\mathrm{rad} \end{bmatrix}^T.$$

Therefore, the initial augmented state is

$$\zeta(0) = \begin{bmatrix} -0.5\,\mathrm{m} & -0.5\,\mathrm{m} & \pi/2\,\mathrm{rad} & 0\,\mathrm{m} \end{bmatrix}^T.$$

The basis for the value function approximation is selected as

$$\sigma = \left[\zeta_1\zeta_2, \zeta_1\zeta_3, \zeta_1\zeta_4, \zeta_2\zeta_2, \zeta_2\zeta_4, \zeta_3\zeta_4, \zeta_1^2, \zeta_2^2, \zeta_3^2, \zeta_4^2\right]^T.$$

The sampled data points are selected on a $5 \times 5 \times 3 \times 3$ grid about the origin. The user defined quadratic cost weighting matrices are selected as $Q = \mathrm{diag}([2, 2, 0.25])$ and $R = I_{2\times 2}$. The learning gains are selected as $k_{c1} = 1.0$, $k_{c2} = 1.0$, $k_a = 1.25$, and $k_\rho = 1$. The least squares gain update law in (2–8) is not implemented in the following results[1] . The resulting least squares gain is a constant and selected as

$$\Gamma = \mathrm{diag}\left(\begin{bmatrix} 1.0 & 2.5 & 2.5 & 1.0 & 0.5 & 1.0 & 0.125 & 2.5 & 7.5 & 0.5 \end{bmatrix}\right).$$

The auxiliary gains in (4–6) and (4–8) are selected as $k_1 = 0.5$ and $k_2 = 0.005$. The policy and value function NN weight estimates are initialized to

$$\hat{W}_a(0) = \begin{bmatrix} 0 & 0 & 0 & 0.5 & 0 & 0 & 0.5 & 0 & 1.0 & 0 \end{bmatrix}^T$$

[1] The stability result in Theorem 4.1 is unaffected by not implementing (2–8).

and

$$\hat{W}_c(0) = \begin{bmatrix} 0 & 0 & 0 & 0.5 & 0 & 0 & 0.5 & 0 & 1.0 & 0 \end{bmatrix}^T.$$

The simulation result utilize the kinematic model in (4–5) as the simulated mobile robot. Since an analytical solution is not feasible for this problem, the simulation results are directly compared to results obtained by an offline optimal solver GPOPS [27]. Figures 4-2 and 4-3 illustrate that the state and control trajectories from the proposed method and the solution found using the offline optimal solver, and Figures 4-4 and 4-5 show the NN weight estimates converge to steady state values[2] . The true values of the ideal NN network weights are unknown. However, after the NN converges to a steady state value, the system trajectories and control values obtained using the developed method correlate with the system trajectories and control value of the offline optimal solver. The overall performance of the controller is demonstrated in the plot of the craft's planar trajectory in Figure 4-6.

Experimental results also demonstrate the ability of the developed controller to perform on real-world hardware. The ADP-based guidance law is implemented on a Turtlebot wheeled mobile robot depicted in Figure 4-7. Computation of the optimal guidance law takes place on the Turtlebot's on-board ASUS Eee PC netbook with 1.8 GHz Intel Atom processor. The Turtlebot is provided velocity commands from the guidance law, where the Turtlebot's existing low-level controller minimizes the velocity tracking error. Figure 4-8 shows convergence of the error state to a ball about the origin. Figures 4-9 and 4-10 show the NN critic and actor weight estimates converge to steady

[2] It takes ~125 seconds for the mobile robot to traverse the desired path. However, all figures with the exception of the craft trajectory are plotted only for 60 seconds to provide clarity on the transient response. The steady-state response remains the same after the initial transient (~20 seconds).

state values that are similar to the simulation result. The ability of the mobile robot to track the desired path is demonstrated in Figure 4-11.

4.6 Summary

An online approximation of an optimal path following guidance law is developed. ADP is used to approximate the solution to the HJB equation without the need for persistence of excitation. A Lyapunov-based stability analysis proves ultimate bounded convergence of the craft to the desired path while maintaining the desired speed profile and of the approximate policy to the optimal policy. Simulation and experimental results demonstrate the performance of the developed controller.

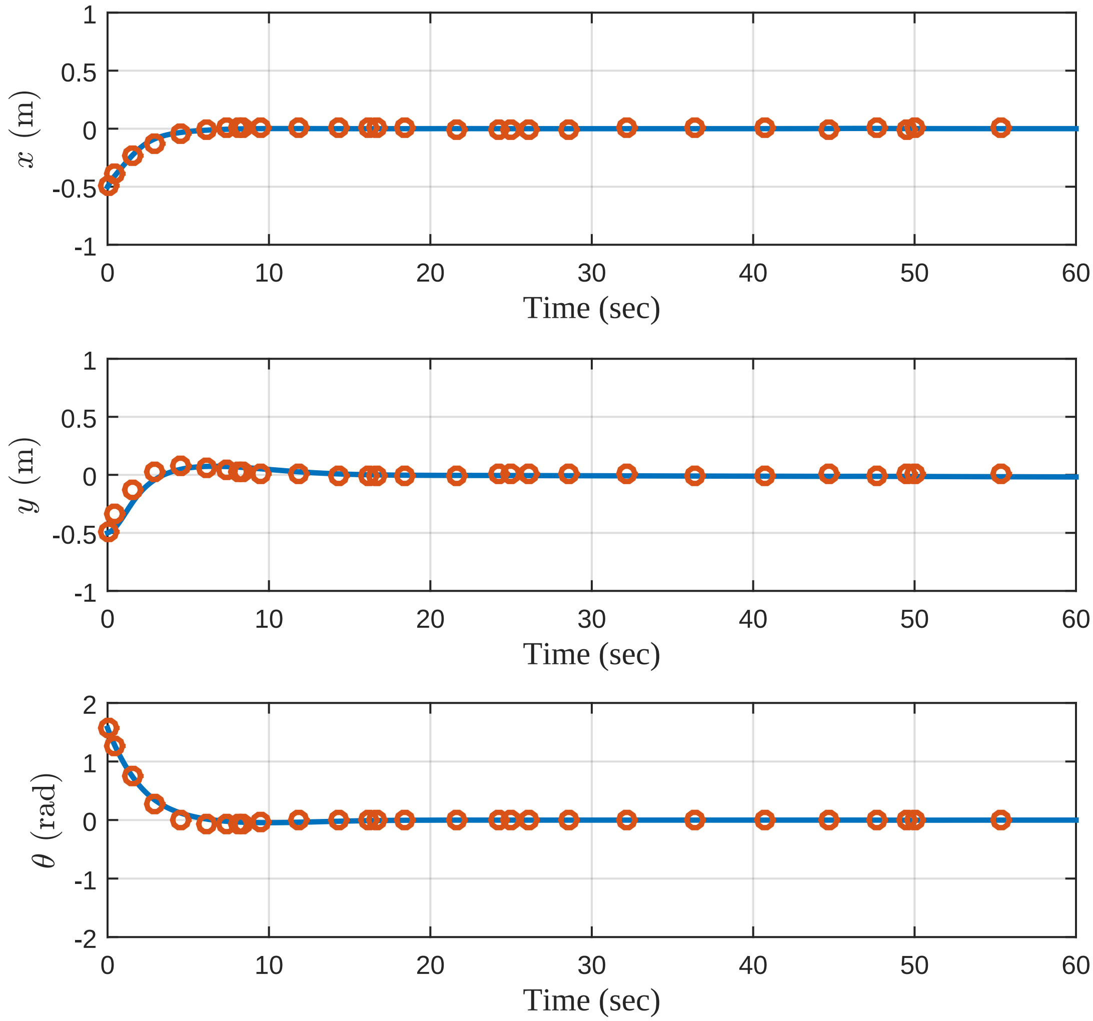

Figure 4-2. Error state trajectory generated by developed path following method as solid lines and by numerical method as markers.

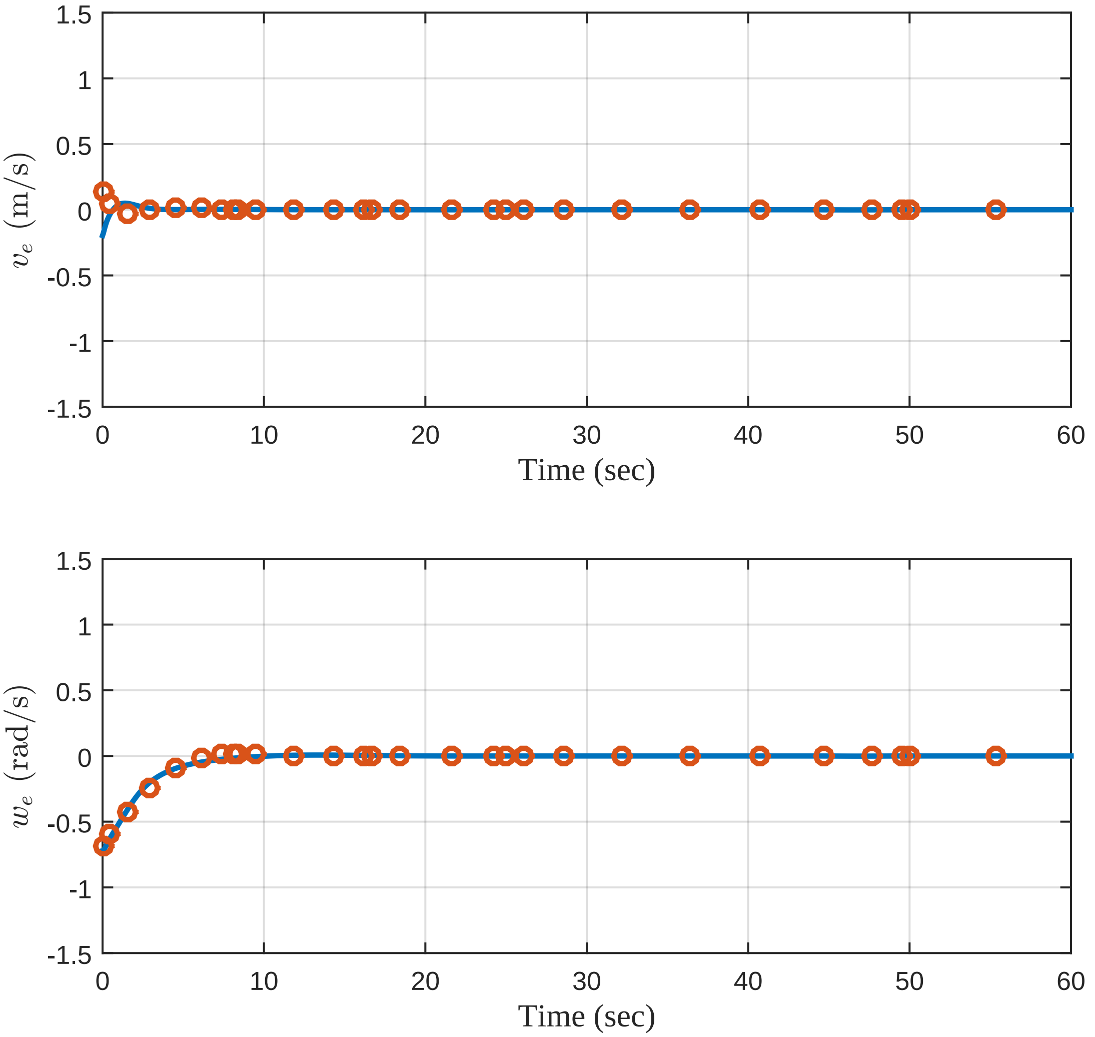

Figure 4-3. Control trajectory generated by developed path following method as solid lines and by numerical method as markers.

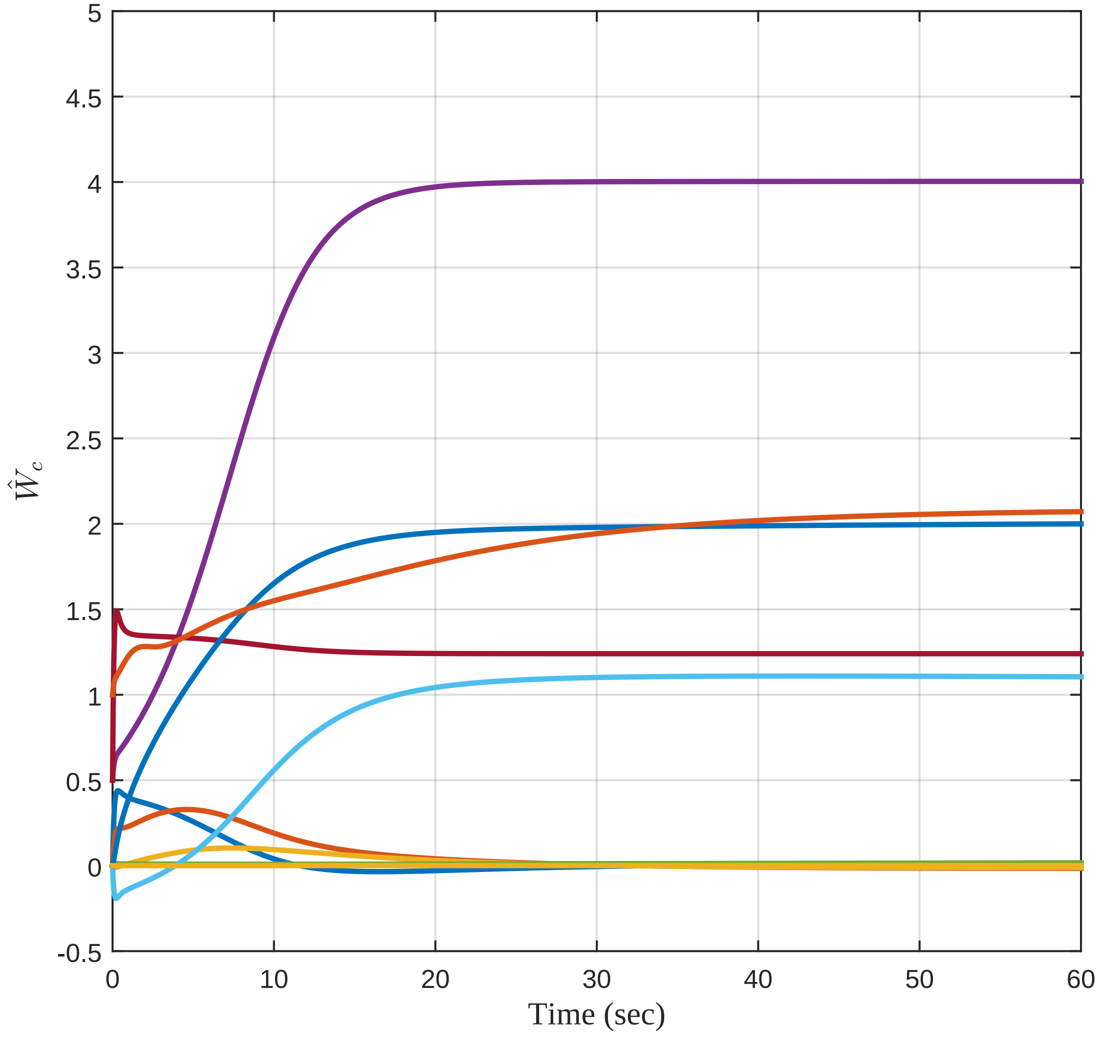

Figure 4-4. Estimated critic weight trajectories generated by developed path following method in simulation.

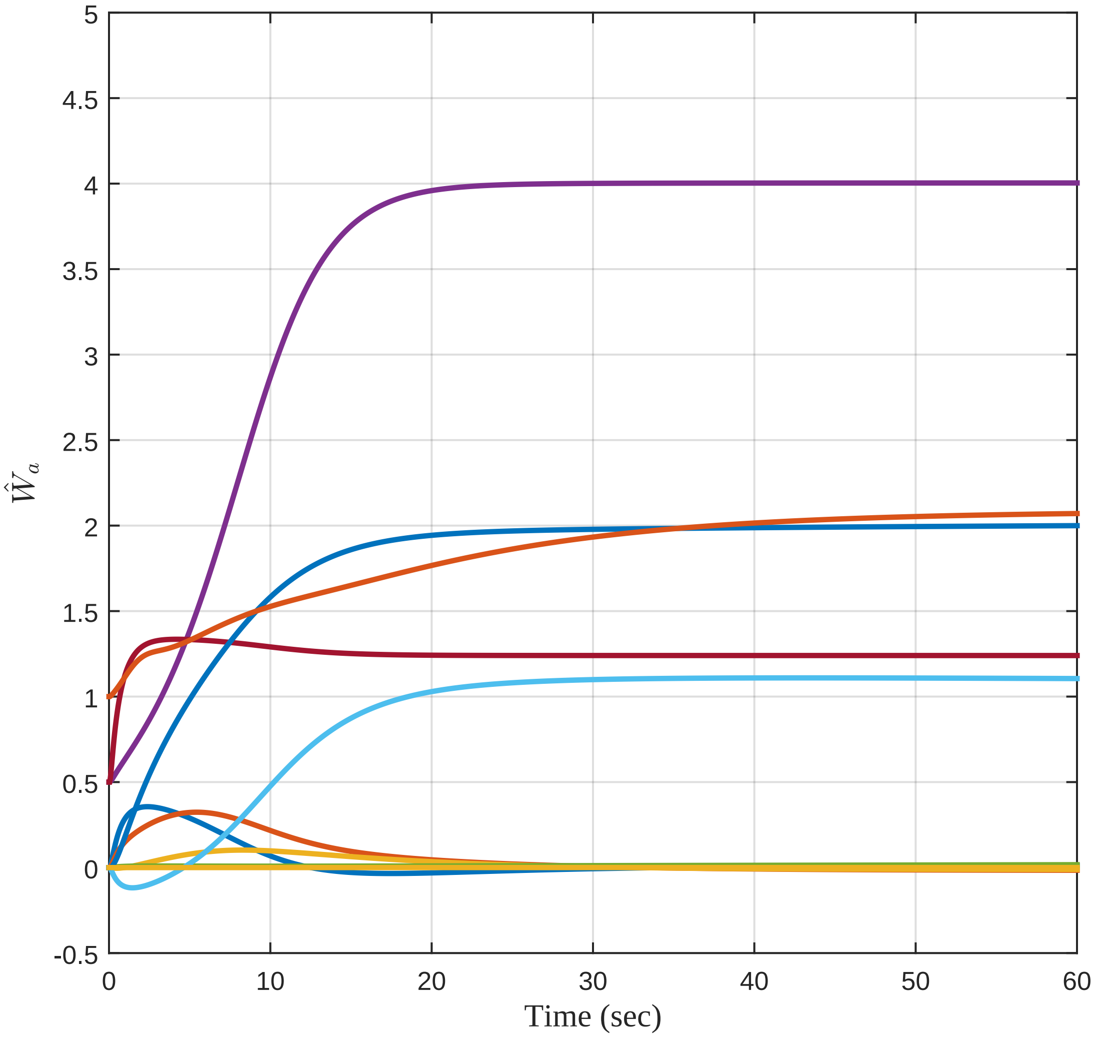

Figure 4-5. Estimated actor weight trajectories generated by developed path following method in simulation.

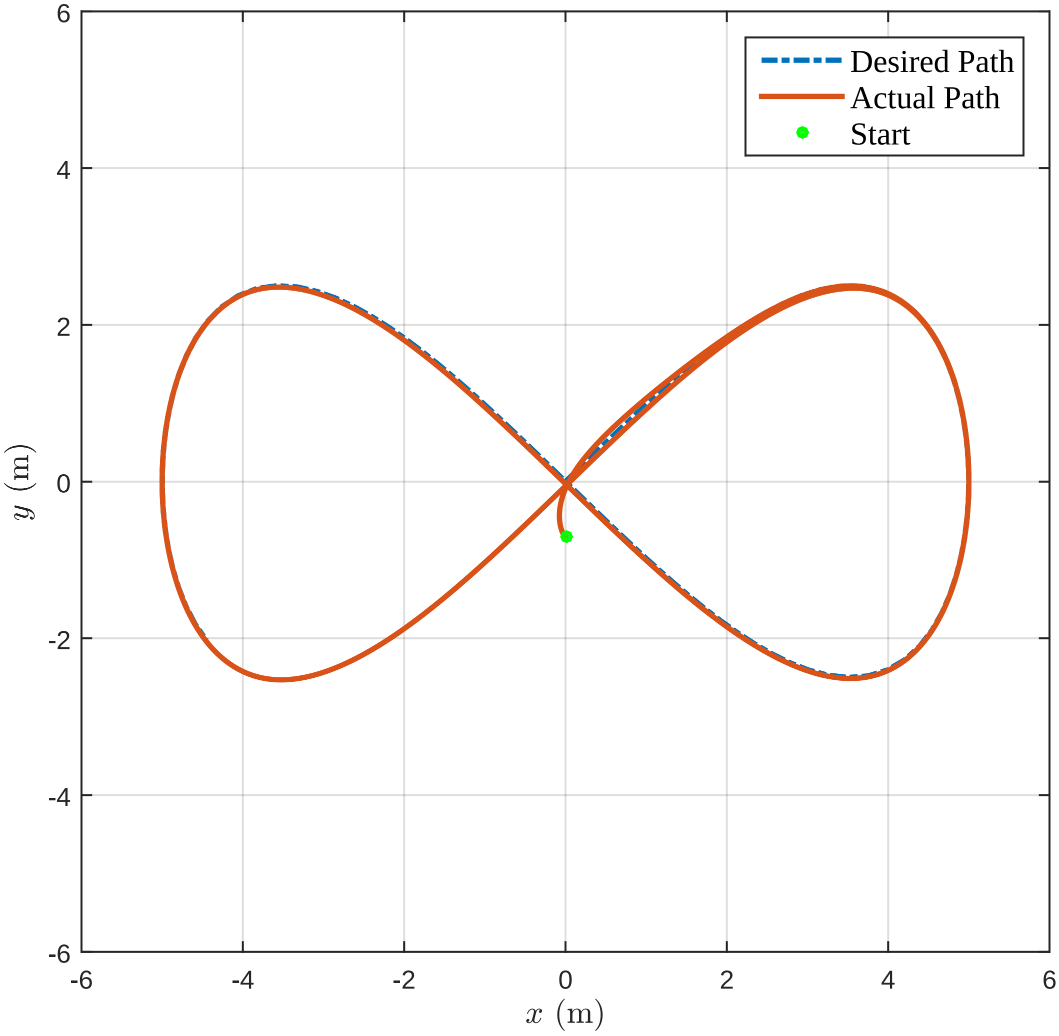

Figure 4-6. Planar trajectory achieved by developed path following method in simulation.

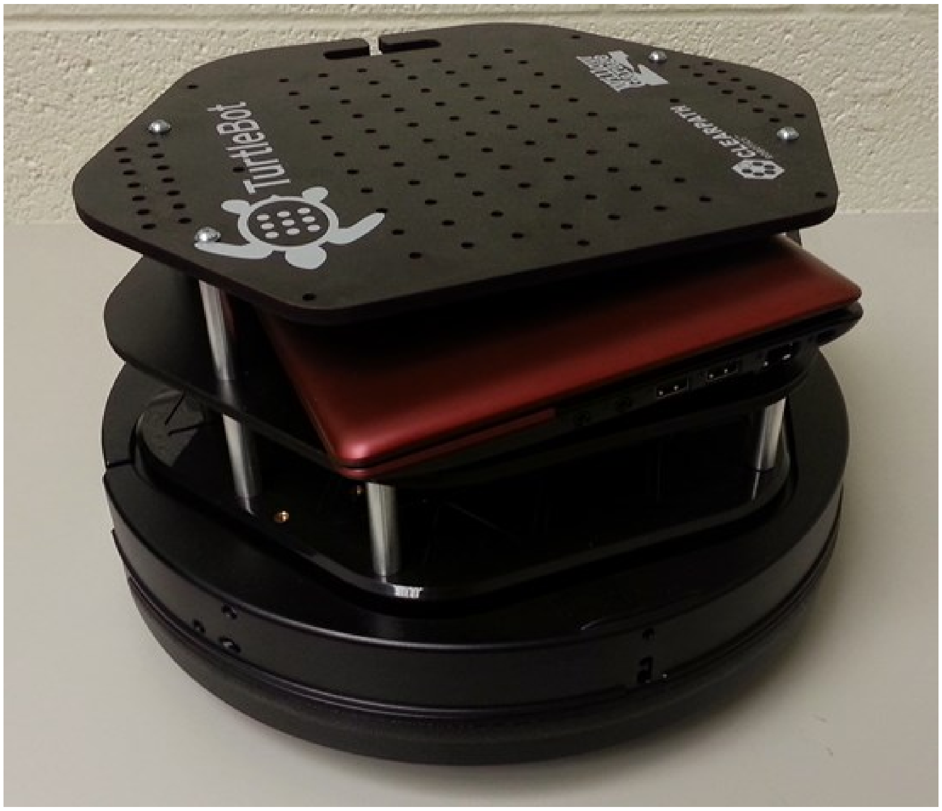

Figure 4-7. Turtlebot wheeled mobile robot. Photo courtesy of author.

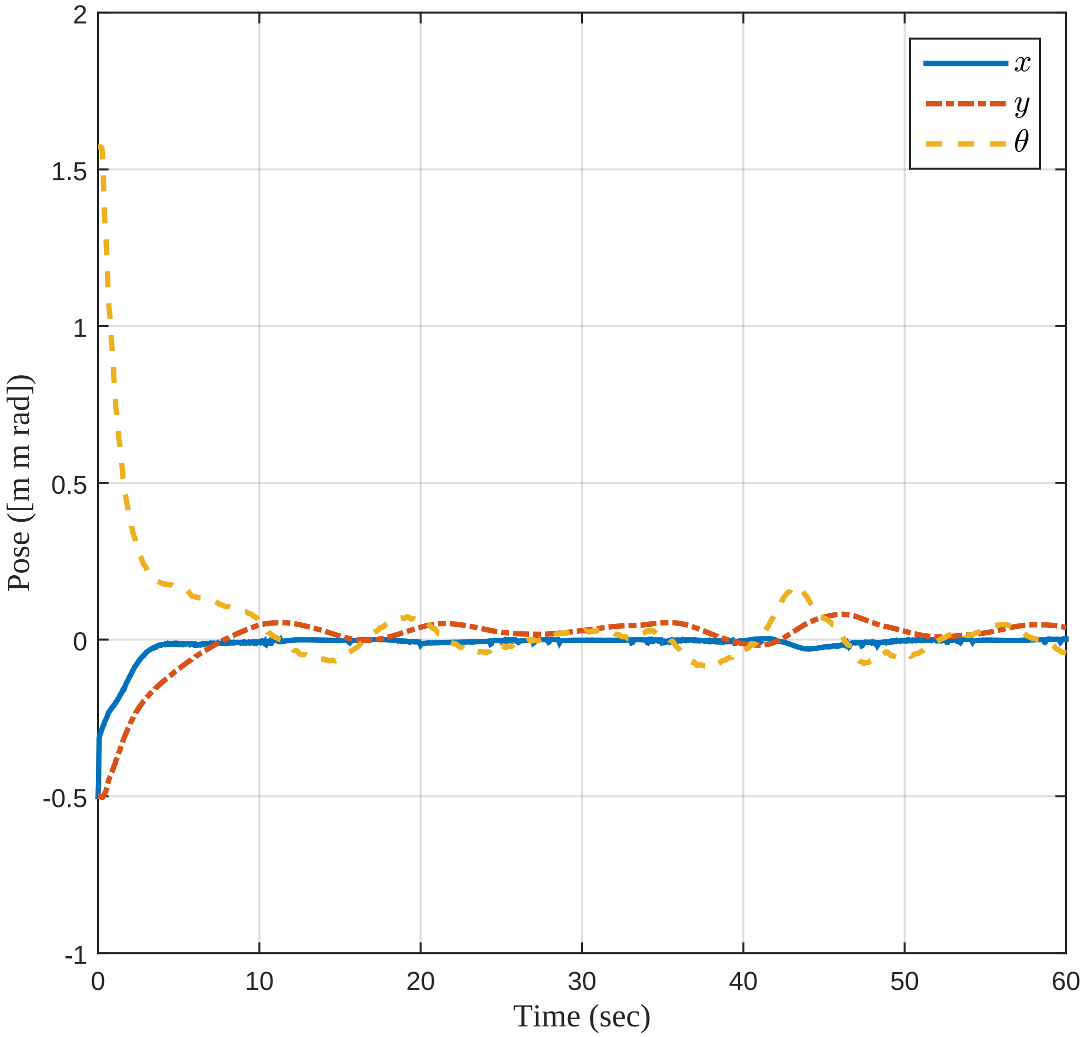

Figure 4-8. Error state trajectory generated by developed path following method implemented on Turtlebot.

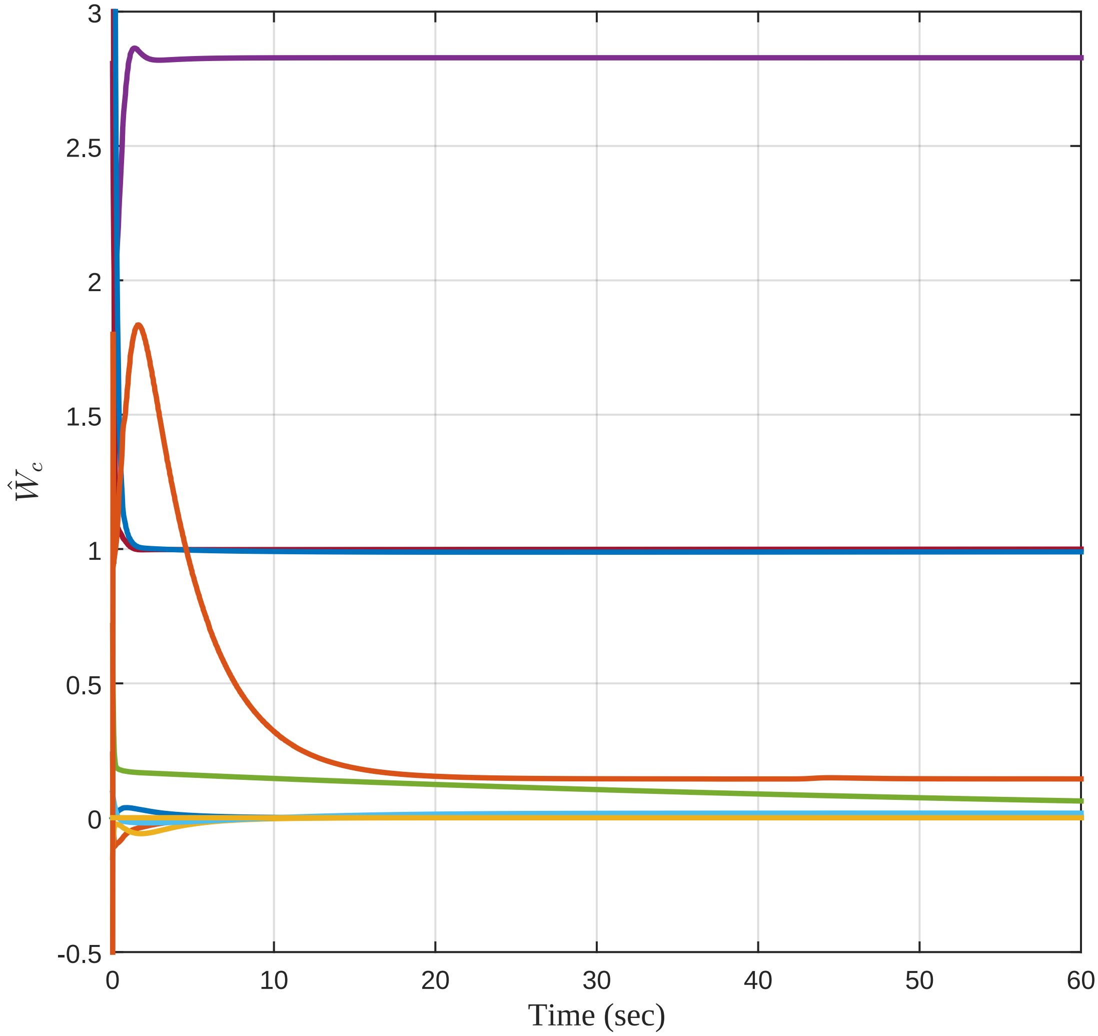

Figure 4-9. Estimated critic weight trajectories generated by developed path following method implemented on Turtlebot.

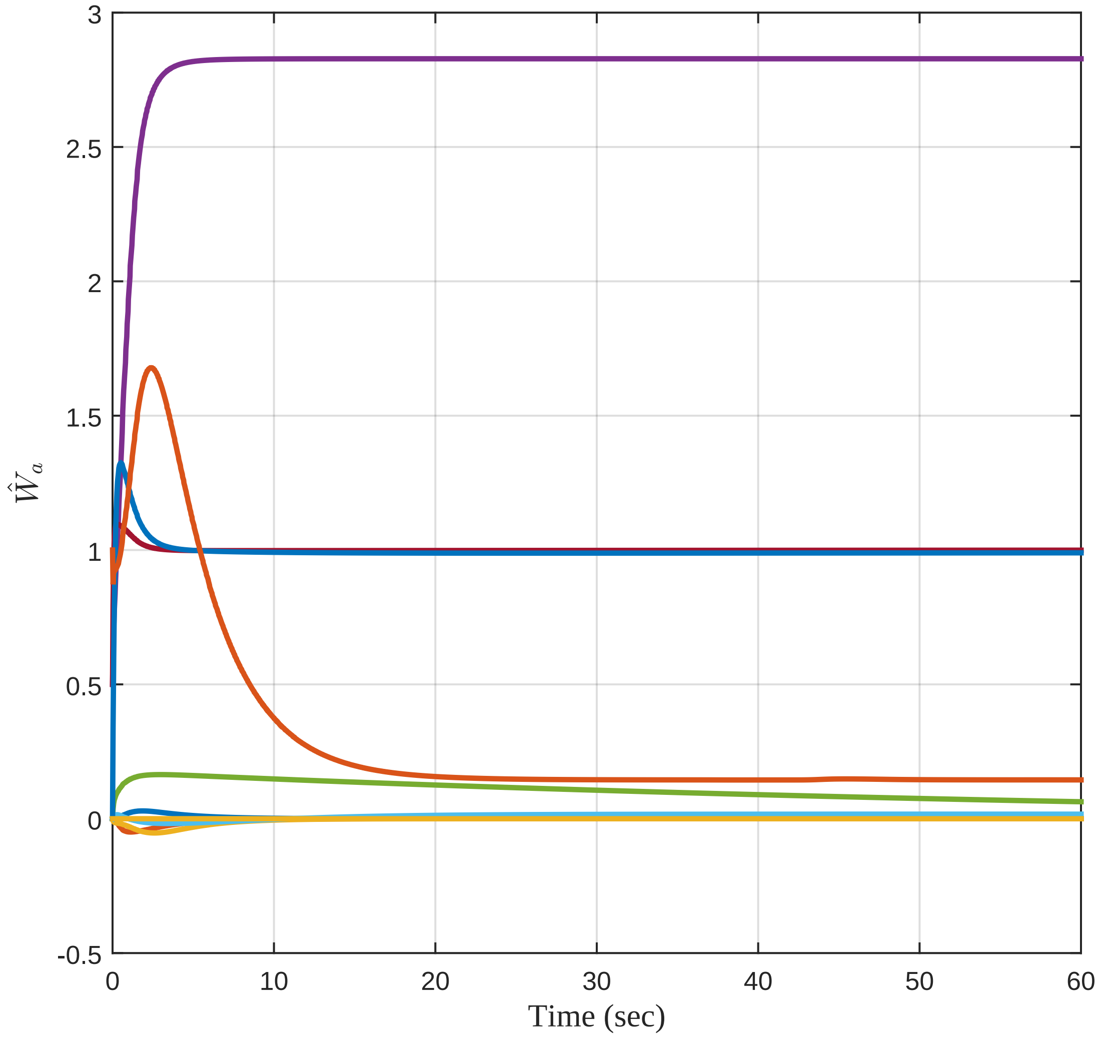

Figure 4-10. Estimated actor weight trajectories generated by developed path following method implemented on Turtlebot.

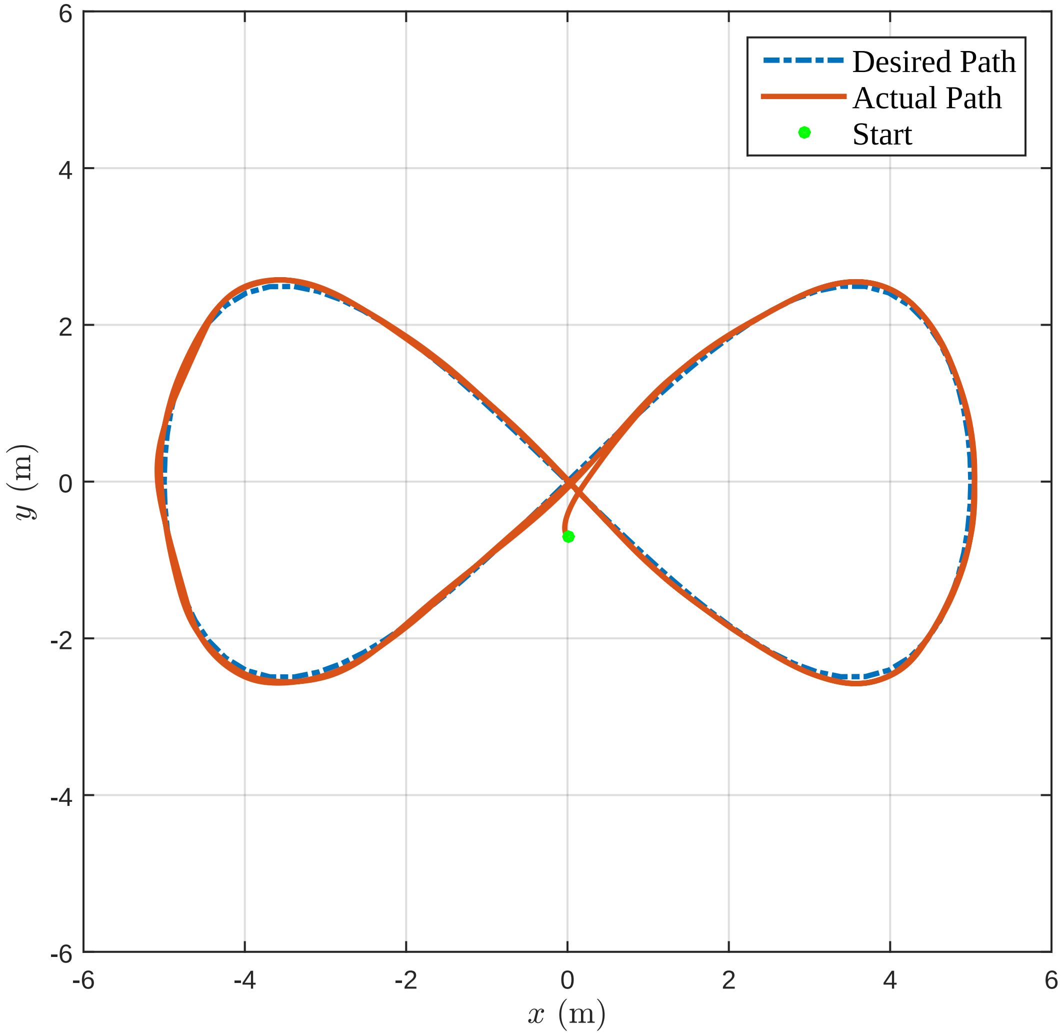

Figure 4-11. Planar trajectory achieved by Turtlebot.

CHAPTER 5
PATH PLANNING WITH STATIC OBSTACLES

The focus of this chapter is to develop a online approximation of the path to optimally navigate to a setpoint while avoiding static obstacles and adhering to a marine craft's actuation constraints. A model-based ADP technique is implemented to locally estimate the unknown value function. By preforming a local approximation, the locations of the static obstacles do not need to be known until the obstacles are within an approximation window. An auxiliary controller is developed to escort the marine craft away from obstacles while the optimal controller is learning.

5.1 Problem Formulation

Consider a nonlinear control affine dynamical system of the form

$$\dot{\zeta} = \overline{f}\left(\zeta\right) + \overline{g}\left(\zeta\right)\overline{u}, \tag{5–1}$$

where $\zeta \in \mathbb{R}^n$ denotes the system state, $\overline{f} : \mathbb{R}^n \to \mathbb{R}^n$ denotes the drift dynamics, $\overline{g} : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ denotes the control effectiveness, and $\overline{u} \in \mathbb{R}^m$ denotes the control input.

From the objective statement, the path planning problem may be posed as a constrained infinite-horizon nonlinear regulation problem, i.e., to design a control signal $\overline{u}$ to minimize a subsequently defined cost function subject to the dynamic constraint in (5–1), while avoiding the obstacles and obeying $\sup_t\left(\overline{u}_i\right) \leq u_{sat}\ \forall i = 1, \ldots, m$, where $\overline{u} = \left[\overline{u}_1, \cdots, \overline{u}_m\right]^T$ and $u_{sat} \in [0, \infty)$ is the control saturation constant.

Static obstacles represent hard constraints that must be taken into account in the development of the approximately optimal path planner. To this end, an auxiliary controller is subsequently developed to assist in navigating a marine craft around obstacles. The auxiliary controller is denoted by $u_s : \mathbb{R}^n \to \mathbb{R}^m$, where $u_s = \left[u_{s_1}, \cdots, u_{s_m}\right]^T$ and $\sup_t\left(u_{s_i}\right) \leq u_{sat}\ \forall i = 1, 2, \ldots, m$. To facilitate the development of u_s, obstacles are augmented with a perimeter that extends from their borders denoting an unsafe region as illustrated in Figure 5-1. A smooth scheduling function $s : \mathbb{R}^n \to [0, a]$, where $a < 1$,

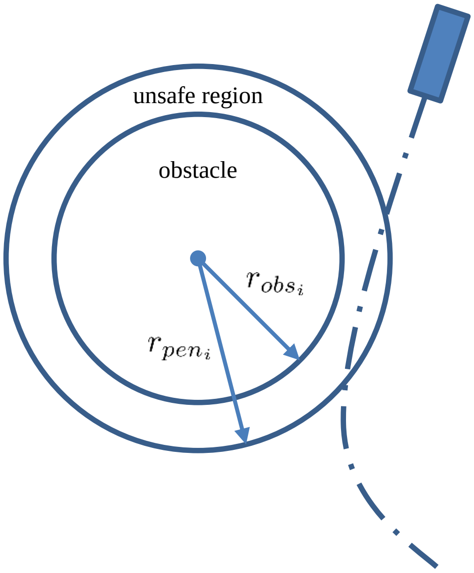

Figure 5-1. Obstacles augmented with unsafe region that extends from its border.

is used to transition between the approximate optimal controller $u : [t_0, \infty) \to \mathbb{R}^m$ and the auxiliary controller u_s without introducing discontinuities to the system dynamics. The scheduling function and the auxiliary controller are designed such that they are functions of the state and that all state trajectories are driven away from a obstacle. In Section 5.4, the auxiliary controller u_s and the scheduling function s are designed for a specific system.

The control input $\overline{u}$ is defined as

$$\overline{u}(\zeta) = s(\zeta)\, u_s(\zeta) + (1 - s(\zeta))\, u, \tag{5–2}$$

where u is the subsequently designed approximate optimal controller. Based on (5–2), the dynamics can be rewritten as

$$\dot{\zeta} = f(\zeta) + g(\zeta)\, u, \tag{5–3}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ denotes the augmented drift dynamics defined as

$$f(\zeta) \triangleq \overline{f}(\zeta) + \overline{g}(\zeta)\, s(\zeta)\, u_s(\zeta),$$

$g:\mathbb{R}^n \to \mathbb{R}^{n\times m}$ denotes the augmented control effectiveness defined as

$$g\left(\zeta\right) \triangleq \overline{g}\left(\zeta\right)\left(1 - s\left(\zeta\right)\right).$$

To account for actuator saturation and the unsafe regions, the cost function is defined as

$$J\left(\zeta, u\right) \triangleq \int\limits_{t_o}^{\infty} r\left(\zeta\left(\tau\right), u\left(\tau\right)\right) d\tau, \tag{5–4}$$

where t_0 denotes the initial time, and $r:\mathbb{R}^n \times \mathbb{R}^m \to [0,\infty)$ is the local cost defined as

$$r\left(\zeta, u\right) \triangleq \zeta^T Q \zeta + P\left(\zeta\right) + U\left(u\right), \tag{5–5}$$

subject to the dynamic constraint in (5–3), where $Q \in \mathbb{R}^{n\times n}$ is a constant, user defined, symmetric positive definite weighting matrix, $P:\mathbb{R}^n \to \mathbb{R}$ is a non-negative continuous function penalizing trajectories that enter the unsafe region, and $U : \mathbb{R}^m \to \mathbb{R}$ is a positive definite function penalizing control effort. The matrix Q has the property $\underline{q}\left\|\xi_q\right\|^2 \leq \xi_q^T Q \xi_q \leq \overline{q}\left\|\xi_q\right\|^2, \forall \xi_q \in \mathbb{R}^n$ where $\underline{q}$ and $\overline{q}$ are positive constants. The positive definite function U in (5–5) is defined as [20, 50]

$$U\left(u\right) \triangleq 2\sum_{i=1}^{m}\left[\int\limits_{0}^{u_i}\left(u_{sat} r_i \tanh^{-1}\left(\frac{\xi_{u_i}}{u_{sat}}\right)\right) d\xi_{u_i}\right] \tag{5–6}$$

where u_i is the i^{th} element of u, ξ_{u_i} is an integration variable, and $R \in \mathbb{R}^{m\times m}$ is a diagonal positive definite, user defined, weighting matrix given as $R = \operatorname{diag}\left(\left[r_1, r_2, \cdots, r_m\right]\right)$.

The infinite-time scalar value function $V : \mathbb{R}^n \to [0,\infty)$ for the optimal solution is written as

$$V\left(\zeta\right) = \min_{u} \int\limits_{t_0}^{\infty} r\left(\zeta\left(\tau\right), u\left(\tau\right)\right) d\tau. \tag{5–7}$$

The objective of the optimal path planner is to find the optimal policy $u^*:\mathbb{R}^n \to \mathbb{R}^m$ that minimizes the performance index (5–4) with the local cost (5–5) subject to the dynamic constraint in (5–3). The optimal value function is characterized by the HJB, which is given as

$$\nabla V\left(\zeta\right)\left(f\left(\zeta\right)+g\left(\zeta\right)u^{*}\left(\zeta\right)\right)+r\left(\zeta,u^{*}\left(\zeta\right)\right)=0 \tag{5–8}$$

with the boundary condition $V\left(0\right)=0$. The optimal control policy can be determined from (5–8) as

$$u^{*}\left(\zeta\right)=-u_{sat}\tanh\left(\frac{1}{2u_{sat}}R^{-1}g^{T}\nabla V\left(\zeta\right)^{T}\right). \tag{5–9}$$

The analytical expression for the optimal path in (5–9) requires knowledge of the value function which is the solution to the HJB equation in (5–8). The HJB equation is a partial differential equation which is generally infeasible to solve analytically; hence, an approximate solution is sought.

5.2 Local Approximation of Solution

The subsequent development is based on an approximation of the value function and optimal policy. Differing from previous ADP literature (e.g., [19–21, 37]) that seeks a global policy, the following development seeks only a local policy. Instead of generating an approximation of the value function over the entire operating region, we aim to approximate a small region about the current state. With the region of approximation limited to a small range about the current state, one only needs to assume that there may exist an obstacle or obstacles outside the local approximation. Once inside the local approximation window, the optimal policy will adapt to avoid the obstacle. Despite the uncertainty of distant obstacles, the following development yields guaranteed stability of the state and convergence to the optimal path.

Leveraging the results of [34], StaF kernels are employed to approximate the local policy on some small compact set $B_{r}\left(\zeta\right)$ (i.e., approximation window) around the state ζ. The StaF representation of the value function and optimal policy are given as

$$V\left(\zeta\right)=W\left(\zeta\right)^{T}\sigma\left(\zeta,c\left(\zeta\right)\right)+\epsilon\left(\zeta\right),$$

$$u^{*}\left(\zeta\right)=-u_{sat}\tanh\left(\frac{1}{2u_{sat}}R^{-1}g\left(\zeta\right)^{T}\left(\nabla\sigma\left(\zeta,c\left(\zeta\right)\right)^{T}W\left(\zeta\right)+\nabla\epsilon\left(\zeta\right)\right)\right), \tag{5–10}$$

respectively, where $W : \mathbb{R}^n \to \mathbb{R}^l$ is the ideal weight vector, $\sigma : \mathbb{R}^n \to \mathbb{R}^l$ is a continuously differentiable kernel function, and $\epsilon : \mathbb{R}^n \to \mathbb{R}$ is the continuously differential function reconstruction error. Note that the centers of the kernel function change as the system state changes; therefore, the ideal weight vector W is a time-varying function. The approximations of the value function and the optimal policy are defined as

$$\hat{V}\left(\zeta, \hat{W}_c\right) \triangleq \hat{W}_c^T \sigma\left(\zeta, c\left(\zeta\right)\right), \tag{5–11}$$

$$\hat{u}\left(\zeta, \hat{W}_a\right) \triangleq -u_{sat} \tanh\left(\frac{1}{2u_{sat}} R^{-1} g\left(\zeta\right)^T \nabla\sigma\left(\zeta, c\left(\zeta\right)\right)^T \hat{W}_a\right), \tag{5–12}$$

where $c\left(\zeta\right) \in B_r\left(\zeta\right)$ is the StaF kernel center, and $\hat{W}_c$, $\hat{W}_a \in \mathbb{R}^l$ are estimates of the ideal weight vector W. Substituting the approximations from (5–11) and (5–12) into (5–8), results in a residual error $\delta : \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^l \to \mathbb{R}$ called the Bellman error, given by

$$\delta\left(\zeta, \hat{W}_c, \hat{W}_a\right) = r\left(\zeta, \hat{u}\left(\zeta, \hat{W}_a\right)\right) + \hat{W}_c^T \omega\left(\zeta, \hat{W}_a\right),$$

where $\omega : \mathbb{R}^n \to \mathbb{R}^l$ is given by

$$\omega\left(\zeta, \hat{W}_a\right) = \nabla\sigma\left(\zeta, c\left(\zeta\right)\right)\left[f\left(\zeta\right) + g\left(\zeta\right)\hat{u}\left(\zeta, \hat{W}_a\right)\right].$$

The online implementation of the approximation is presented in Section 2.5, where the parameters $\hat{W}_c$ and $\hat{W}_a$ are updated by (2–7) and (2–9), respectively.

5.3 Stability Analysis

For notational brevity, all function dependencies from previous sections are henceforth suppressed. Let the notation F_k denote the function $F\left(\zeta, \cdot\right)$ evaluated at the sampled state, i.e., $F_k\left(\cdot\right) = F\left(\zeta_k, \cdot\right)$. An unmeasurable form of the Bellman error can be written as

$$\delta = -\tilde{W}_c^T \omega + W^T \sigma' g\left(\hat{u} - u^*\right) + U\left(\hat{u}\right) - U\left(u^*\right) - \nabla\epsilon\left(f + gu^*\right), \tag{5–13}$$

Similarly, the Bellman error at the extrapolated points can be written as

$$\delta_k = -\tilde{W}_c^T \omega_k + W^T \sigma_k' g_k \left(\hat{u}_k - u_k^*\right) + U\left(\hat{u}_k\right) - U\left(u_k^*\right) - \nabla \epsilon_k \left(f_k + g_k u_k^*\right). \tag{5–14}$$

The following lemma facilitates the stability analysis by establishing an upper bound on the difference $U\left(\hat{u}\right) - U\left(u^*\right)$.

Lemma 5.1. *The function $U\left(\hat{u}\right) - U\left(u^*\right)$ can be bounded by*

$$\left\| U\left(\hat{u}\right) - U\left(u^*\right) \right\| \leq L_{\overline{U}} \left\| \frac{1}{2} R^{-1} g^T \sigma'^T \tilde{W}_a + \frac{1}{2} R^{-1} g^T \epsilon'^T \right\|,$$

for all $\hat{u},\ u^*$ defined in (5–10) *and* (5–12)*, respectively, where $L_{\overline{U}}$ is a positive constant.*

Proof. The change of variables $\xi_{u_i} = -u_{sat} \tanh\left(\xi_{\lambda_i}\right)$ and $d\xi_{u_i} = -u_{sat} \text{sech}^2\left(\xi_{\lambda_i}\right) d\xi_{\lambda_i}$ in (5–6) yields

$$U\left(u\right) = 2 \sum_{i=1}^{m} \left[r_i u_{sat}^2 \int_0^{\tanh^{-1}\left(\frac{u_i}{u_{sat}}\right)} \xi_{\lambda_i} \text{sech}^2\left(\xi_{\lambda_i}\right) d\xi_{\lambda_i} \right].$$

Let $\lambda_i = \tanh^{-1}\left(u_i / u_{sat}\right)$, where $\lambda = \left[\lambda_1, \ldots, \lambda_m\right]$ and let the function $\overline{U} : \mathbb{R}^m \to \mathbb{R}$ be defined as

$$\overline{U}\left(\lambda\right) \triangleq 2 \sum_{i=1}^{m} \left[r_i u_{sat}^2 \int_0^{\lambda_i} \xi_{\lambda_i} \text{sech}^2\left(\xi_{\lambda_i}\right) d\xi_{\lambda_i} \right].$$

Then $\partial \overline{U}\left(\lambda\right) / \partial \lambda_i = r_i u_{sat}^2 \lambda_i \text{sech}^2\left(\lambda_i\right)$ is uniformly bounded for all $\lambda_i \in \mathbb{R}$, $\forall i = 1, \ldots, m$. Hence, $\overline{U}$ is globally Lipschitz continuous, i.e.,

$$\overline{U}\left(\hat{\lambda}\right) - \overline{U}\left(\lambda^*\right) \leq L_{\overline{U}} \left\| \hat{\lambda} - \lambda^* \right\|, \tag{5–15}$$

where $\hat{\lambda} = \tanh^{-1}\left(\hat{u} / u_{sat}\right)$ and $\lambda^* = \tanh^{-1}\left(u^* / u_{sat}\right)$. Using the definitions of $\hat{\lambda}$ and λ^*, (5–10), (5–12), and the fact that $U\left(u\right) = \overline{U}\left(\lambda\right)\ \forall u, \lambda \in \mathbb{R}^m$, (5–15) may be expressed as

$$U\left(\hat{u}\right) - U\left(u^*\right) \leq L_{\overline{U}.} \left\| \frac{1}{2} R^{-1} g^T \sigma'^T \tilde{W}_a + \frac{1}{2} R^{-1} g^T \epsilon'^T \right\|.$$

□

To facilitate the subsequent stability analysis, consider the candidate Lyapunov function $V_L : \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^l \to [0, \infty)$ given as

$$V_L(Z) = V + \frac{1}{2}\tilde{W}_c{}^T \Gamma^{-1} \tilde{W}_c + \frac{1}{2}\tilde{W}_a^T \tilde{W}_a,$$

where $Z \triangleq \begin{bmatrix} \zeta^T & \tilde{W}_c^T & \tilde{W}_a^T \end{bmatrix}^T \in \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^l$. Since the value function V in (5–7) is positive definite [51] using Lemma 4.3 of [48], V_L can be bounded by

$$\underline{\upsilon_L}(\|Z\|) \leq V_L(Z) \leq \overline{\upsilon_L}(\|Z\|), \tag{5–16}$$

where $\underline{\upsilon_L}$, $\overline{\upsilon_L} : [0, \infty) \to [0, \infty)$ are class $\mathcal{K}$ functions. Define the constant $K \in \mathbb{R}$ as

$$K \triangleq \sqrt{\frac{\iota_c^2}{\alpha\left(2\underline{c} - k_a\right)} + \frac{\iota_a^2}{\alpha k_a} + \frac{\iota}{\alpha}}$$

where $\alpha \triangleq \min\left\{\frac{q}{2}, \left(\frac{c}{2} - \frac{k_a}{4}\right), \frac{k_a}{4}\right\}$,

$$\begin{aligned}\iota_c = {} & \frac{k_{c1} L_{\overline{U}}}{2} \overline{\left\| R^{-1} g^T \nabla \sigma^T \tilde{W}_a + R^{-1} g^T \nabla \epsilon^T \right\|} + \frac{k_{c2} L_{\overline{U}}}{2N} \sum_{k=1}^{N} \overline{\left\| R^{-1} g_k^T \nabla \sigma_k^T \tilde{W}_a + R^{-1} g_k^T \nabla \epsilon_k^T \right\|} \\ & + k_{c1} \overline{\| W^T \nabla \sigma g \|}\, \overline{\left\| R^{-1} g^T \nabla \sigma^T \tilde{W}_a + R^{-1} g^T \nabla \epsilon^T \right\|} + \overline{\| \Gamma^{-1} \nabla W f \|} + k_{c1} \overline{\| E \|} + \frac{k_{c2}}{N} \sum_{k=1}^{N} \overline{\| E_k \|} \\ & + \frac{k_{c2}}{N} \sum_{k=1}^{N} \overline{\| W^T \nabla \sigma_k g_k \|}\, \overline{\left\| R^{-1} g_k^T \nabla \sigma_k^T \tilde{W}_a + R^{-1} g_k^T \nabla \epsilon_k^T \right\|} + \frac{1}{2} \overline{\| \Gamma^{-1} \nabla W g \|}\, \overline{\left\| R^{-1} g^T \left(\nabla \sigma^T \hat{W}_a \right) \right\|},\end{aligned}$$

$$\begin{aligned}\iota_a = {} & \overline{\| \nabla W f \|} + \frac{1}{2} \overline{\| \nabla W g \|}\, \overline{\left\| R^{-1} g^T \left(\nabla \sigma^T \hat{W}_a \right) \right\|} + \frac{1}{2} \overline{\| W^T \nabla \sigma g \| \| R^{-1} g^T \nabla \sigma^T \|} \\ & + \frac{1}{2} \overline{\| \nabla \epsilon g \| \| R^{-1} g^T \nabla \sigma^T \|},\end{aligned}$$

$$\iota = \frac{1}{2} \overline{\| W^T \nabla \sigma g \| \| R^{-1} g^T \nabla \sigma \epsilon^T \|} + \frac{1}{2} \overline{\| \nabla \epsilon g \| \| R^{-1} g^T \nabla \epsilon^T \|}.$$

The notation $\overline{\|(\cdot)\|}$ is defined as $\overline{\|(\cdot)\|} \triangleq \sup_{Z \in B_L} \|(\cdot)\|$, and $\beta_L \subset \chi \times \mathbb{R}^l \times \mathbb{R}^l$ is a compact set.

Theorem 5.1. *Provided Assumption 2.1 is satisfied along with the sufficient conditions*

$$\underline{c} > \frac{k_a}{2},$$

$$K < \underline{\upsilon_L}^{-1}\left(\overline{\upsilon_L}\left(r\right)\right), \tag{5–17}$$

where $r \in \mathbb{R}$ is the radius of the compact set β_L, then the policy in (5–12) with the update laws in (2–7)-(2–9) guarantee ultimately bounded regulation of the state ζ and of the approximated policies $\hat{u}$ to the optimal policy u^*.

Proof. The time derivative of the candidate Lyapunov function is

$$\dot{V}_L = \frac{\partial V}{\partial \zeta} f + \frac{\partial V}{\partial \zeta} g\hat{u} - \frac{1}{2}\tilde{W}_c^T \Gamma^{-1}\dot{\Gamma}\Gamma^{-1}\tilde{W}_c - \tilde{W}_c^T\Gamma^{-1}\left(\dot{W} - \dot{\hat{W}}_c\right) - \tilde{W}_a^T\left(\dot{W} - \dot{\hat{W}}_a\right).$$

Using Theorem 2 in [34], the time derivative of the ideal weights can be expressed as

$$\dot{W} = \nabla W\left(f + g\hat{u}\right). \tag{5–18}$$

Substituting (2–7)-(2–9), (5–14), and (5–18) yields

$$\begin{aligned}\dot{V}_L = {} & \frac{\partial V}{\partial \zeta} f + \frac{\partial V}{\partial \zeta} g\hat{u} - \frac{1}{2}\tilde{W}_c^T\Gamma^{-1}\left[\left(\beta\Gamma - k_{c1}\Gamma\frac{\omega_t{\omega_t}^T}{\rho}\Gamma\right)\mathbf{1}_{\|\Gamma\|\leq\overline{\Gamma}}\right]\Gamma^{-1}\tilde{W}_c - \tilde{W}_a^T W'\left(f + g\hat{u}\right) \\ & + \tilde{W}_c^T\left[k_{c1}\frac{\omega_t}{\rho}\delta_t + \frac{k_{c2}}{N}\sum_{j=1}^{N}\frac{\omega_k}{\rho_k}\delta_k\right] - \tilde{W}_c^T\Gamma^{-1}W'\left(f + g\hat{u}\right) + \tilde{W}_a^T k_a\left(\hat{W}_a - \hat{W}_c\right).\end{aligned}$$

.

Using Young's inequality, Lemma 5.1, (5–12), (5–13), and (5–14) the Lyapunov derivative can be upper bounded as

$$\dot{V}_L \leq -\underline{q}\left\|\zeta\right\|^2 - \left(\underline{c} - \frac{k_a}{2}\right)\left\|\tilde{W}_c\right\|^2 - \frac{k_a}{2}\left\|\tilde{W}_a\right\|^2 + \iota_c\left\|\tilde{W}_c\right\| + \iota_a\left\|\tilde{W}_a\right\| + \iota.$$

Completing the squares, the upper bound on the Lyapunov derivative may be written as

$$\dot{V}_L \leq -\frac{\underline{q}}{2}\left\|\zeta\right\|^2 - \left(\frac{\underline{c}}{2} - \frac{k_a}{4}\right)\left\|\tilde{W}_c\right\|^2 - \frac{k_a}{4}\left\|\tilde{W}_a\right\|^2 + \frac{\iota_c^2}{2\underline{c} - k_a} + \frac{\iota_a^2}{2k_a} + \iota,$$

which can be further upper bounded as

$$\dot{V}_L \leq -\alpha \left\| Z \right\|, \, \forall \left\| Z \right\| \geq K > 0. \tag{5–19}$$

Using (5–16), (5–17) and (5–19), Theorem 4.18 in [48] is invoked to conclude that Z is ultimately bounded, in the sense that $\limsup_{t\to\infty} \left\| Z\left(t\right) \right\| \leq \underline{\upsilon_L}^{-1}\left(\overline{\upsilon_L}\left(K\right)\right)$.

Based on the definition of Z and the inequalities in (5–16) and (5–19), $\zeta, \tilde{W}_c, \tilde{W}_a \in \mathcal{L}_\infty$. Since $\zeta \in \mathcal{L}_\infty$ and W is a continuous function of ζ, $W \in \mathcal{L}_\infty$. Hence, $\hat{W}_c$, $\hat{W}_a \in \mathcal{L}_\infty$, which implies $\hat{u} \in \mathcal{L}_\infty$. From the definitions of u_s and s, $\overline{u} \in \mathcal{L}_\infty$. □

5.4 Simulation Results

Simulation results are provided to demonstrate the performance of the developed ADP-based path planner. The simulation is performed for the control affine system given in (5–3), where $\overline{f}\left(\zeta\right) = \mathbf{0}$ and $\overline{g}\left(\zeta\right) = I_{2\times 2}$.

For this particular example, the smooth scheduling function is defined as

$$s\left(\zeta\right) \triangleq \sum_{i=1}^{M} \begin{cases} 0.95, & \left\| \zeta - c_{obs_i} \right\| \leq r_{obs_i} \\ 0.95T\left(\zeta\right), & r_{obs_i} < \left\| \zeta - c_{obs_i} \right\| \leq r_{pen_i} \\ 0, & \text{otherwise} \end{cases}, \tag{5–20}$$

where

$$T\left(\zeta\right) \triangleq \left(\frac{1}{2} + \frac{1}{2}\cos\left(\frac{\pi}{r_{pen_i} - r_{obs_i}} \left\| \zeta - c_{obs_i} \right\| - r_{obs_i}\right)\right),$$

M is the number of obstacles, r_{obs_i} is a positive constant indicating the radius of the i^{th} obstacle, r_{pen_i} is a positive constant indicating the radius corresponding to the unsafe region surrounding the i^{th} obstacle, and $c_{obs_i} \in \mathbb{R}^n$ denotes the center corresponding to the i^{th} obstacle. With this formulation of the smooth scheduling function, it is assumed that the obstacles are selected such that the unsafe regions do not overlap[1] .

The continuous auxiliary controller u_s is defined as

$$u_s\left(\zeta\right) \triangleq \frac{u_{sat}\left(\zeta - c_{obs_i}\right)}{\left\|\zeta - c_{obs_i}\right\|}. \tag{5–21}$$

In Appendix B , the auxiliary controller in (5–21) is shown to prevent the state from entering the interior of an obstacle.

The unsafe region is where the penalty function P begins to effect the cost function. The non-negative function P is given as

$$P\left(\zeta\right) = \sum_{i=1}^{M} \begin{cases} 40 & \left\|\zeta - c_{obs_i}\right\| \leq r_{obs_i} \\ 40T\left(\zeta\right), & r_{obs_i} < \left\|\zeta - c_{obs_i}\right\| \leq r_{pen_i} \\ 0, & \text{otherwise} \end{cases} .$$

In the following, two simulation trials are presented starting at different initial states, $\zeta_0 = \begin{bmatrix} 0.8 & 1.2 \end{bmatrix}^T$ and $\zeta_0 = \begin{bmatrix} 0.8 & -1.0 \end{bmatrix}^T$. The StaF basis for the value function approximation is selected as

$$\sigma = \begin{bmatrix} \zeta^T\left(\zeta + d_1\right) & \zeta^T\left(\zeta + d_2\right) & \zeta^T\left(\zeta + d_3\right) \end{bmatrix}^T,$$

where the centers of the kernels are selected as

$$d_1 = 0.005 \cdot \begin{bmatrix} 0 & 1 \end{bmatrix}^T,$$

$$d_2 = 0.005 \cdot \begin{bmatrix} 0.8660 & -0.5 \end{bmatrix}^T,$$

$$d_3 = 0.005 \cdot \begin{bmatrix} -0.8660 & -0.5 \end{bmatrix}^T.$$

[1] If a group of obstacles are close enough for the unsafe regions to overlap, then the group may be considered as one large obstacle.

The Bellman error is extrapolated to 25 sampled data points that are selected on a uniform 5×5 grid that spans a square of size 0.01, and is centered about the current state. The weighting matrices are selected as $Q = I_{2\times 2}$ and $R = I_{2\times 2}$. The learning gains are selected as $k_{c1} = 0.25$, $k_{c2} = 0.15$, $k_a = 0.5$, $\beta = 0.3$, and $k_\rho = 0.05$. The least squares update law's initial condition is selected as $\Gamma_0 = 300 \cdot I_{3\times 3}$. The policy and value function weight estimates are arbitrarily initialized to

$$\hat{W}_c(0) = \hat{W}_a(0) = \begin{bmatrix} 0.4 & 0.4 & 0.4 \end{bmatrix}^T.$$

Since an analytical solution is not feasible for this problem, the simulation results are directly compared to results obtained using an offline optimal solver GPOPS [27]. The generated path from the two simulation trials are shown in Figures 5-2 and 5-3 for the initial states $\zeta_0 = \begin{bmatrix} 0.8 & 1.2 \end{bmatrix}^T$ and $\zeta_0 = \begin{bmatrix} 1.0 & -0.8 \end{bmatrix}^T$, respectively. Note that the state trajectories in Figures 5-2 and 5-3 briefly enter the unsafe region, where the auxiliary controller successfully escorts the state trajectory away from the obstacle. Figures 5-4 and 5-5 illustrate the state and input trajectories corresponding to the initial state $\zeta_0 = \begin{bmatrix} 0.8 & 1.2 \end{bmatrix}^T$. In Figure 5-4, the lines denote paths generated by the proposed method, and the circular markers denote the solution generated by the offline optimal solver. The system trajectories obtained using the developed method correlate well with the system trajectories of the offline optimal solver.

5.5 Summary

An online approximation of an optimal path planning strategy is developed. The solution to the HJB equation is approximated using adaptive dynamic programming. Locally estimating the unknown value function, the locations of the static obstacles do not need to be known until the obstacles are within an approximation window. The developed feedback policy guarantees ultimately bounded convergence of the approximated path to the optimal path without the requirement of persistence of excitation. The results are validated with simulations.

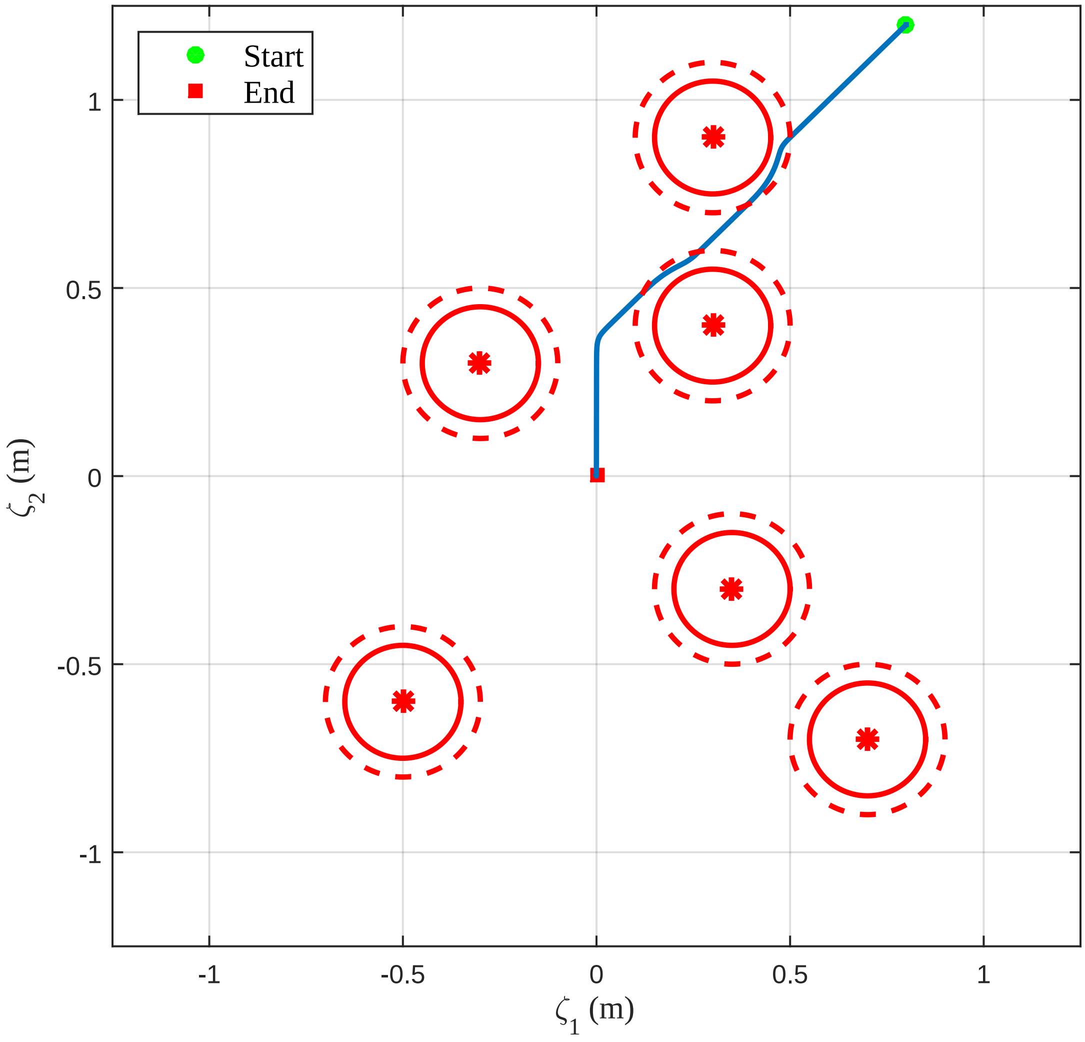

Figure 5-2. Path generated by developed method for first trial where dashed lines denote boundary of unsafe regions and solid lines denote boundary of obstacles.

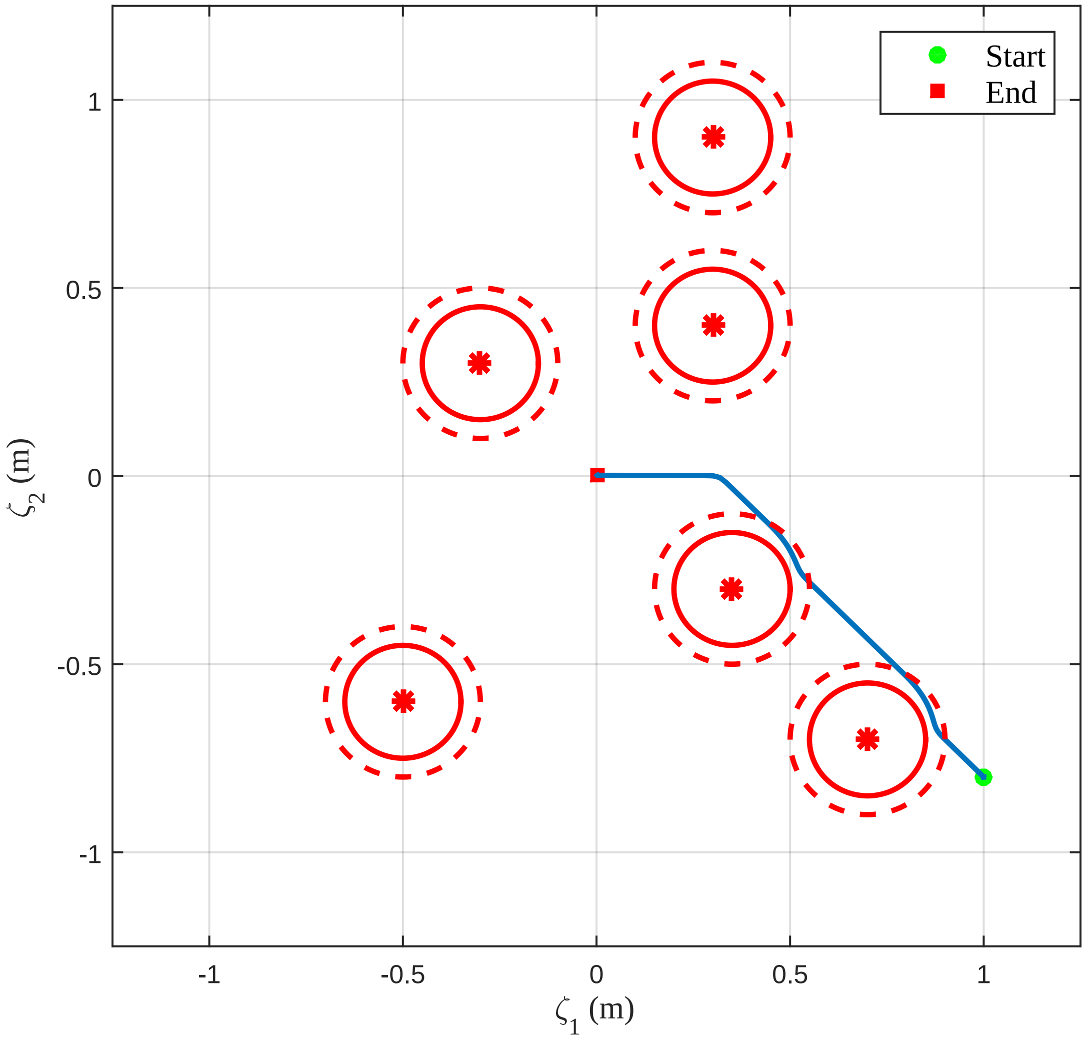

Figure 5-3. Path generated by developed method for second trial where dashed lines denote boundary of unsafe regions and solid lines denote boundary of obstacles.

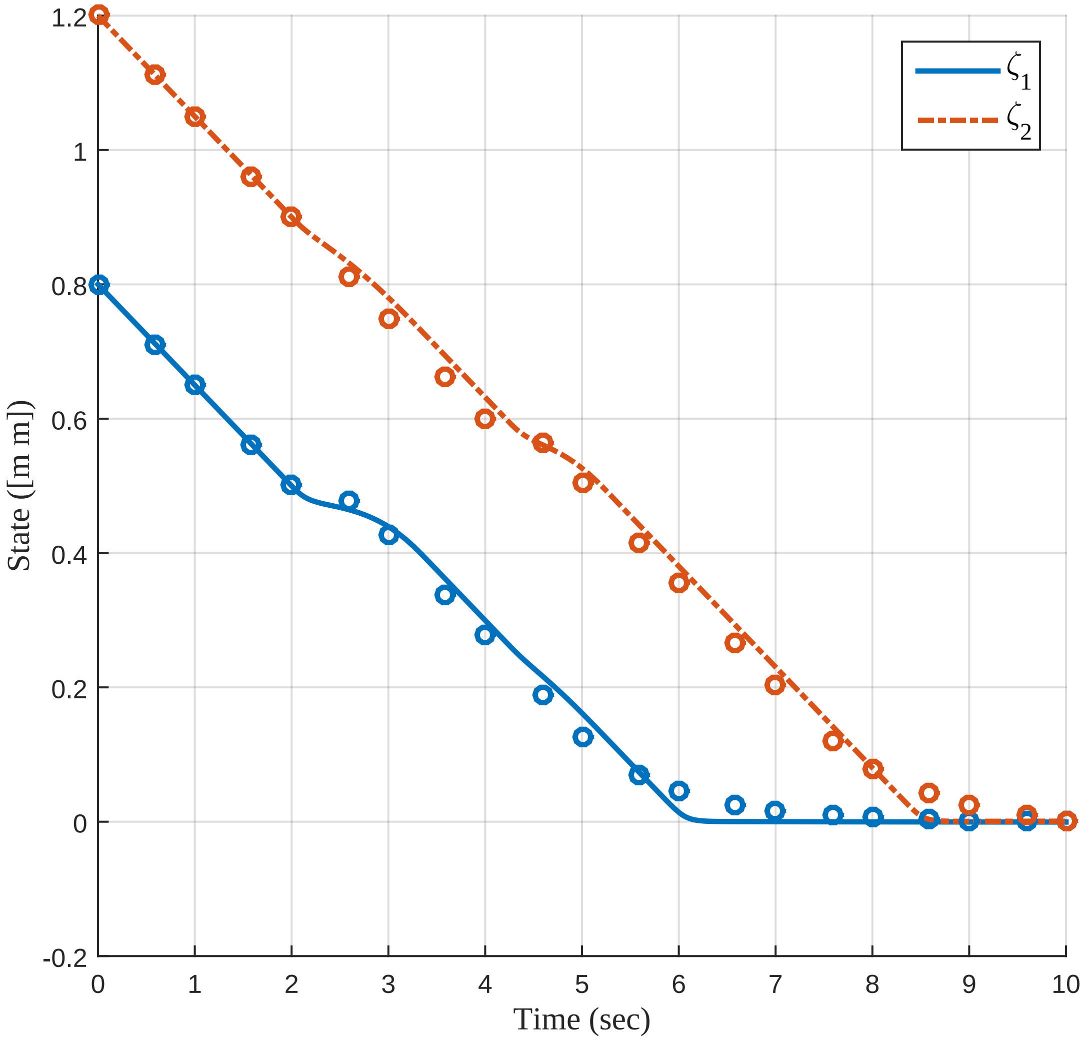

Figure 5-4. Path trajectory generated by developed method as lines and by numerical method as markers.

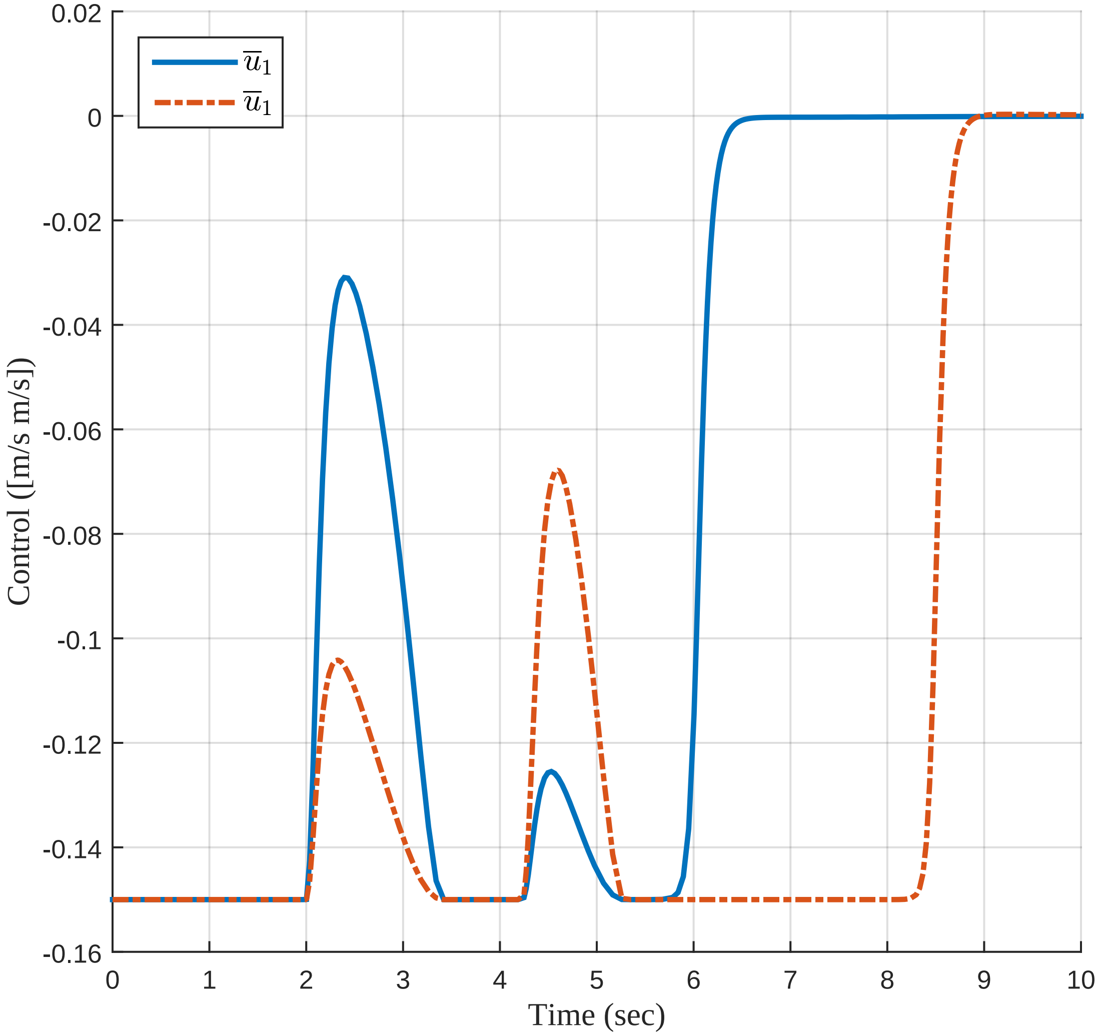

Figure 5-5. Control trajectory generated by developed path planning method.

CHAPTER 6
CONCLUSIONS

Adaptive dynamic programming is a powerful tool for learning optimal policies online. While advances in the theory of ADP have been fruitful, many challenges still exist with the transition of these theoretical contributions to real-world systems. The results of this dissertation begin to address the challenges of implementing these controllers, and show promise to the benefits this family of online learning optimal controllers can bear.

In Chapter 3, an online approximation of a robust optimal control strategy is developed to enable station keeping by an AUV. The solution to the HJB equation is approximated using ADP. The hydrodynamic effects are identified online with a CL-based system identifier. Leveraging the identified model the developed strategy simulates the exploration of the state space to learn the optimal policy without the need of a persistently exciting trajectory. A Lyapunov-based stability analysis concludes ultimately bounded convergence of the states and ultimately bounded convergence of the approximated policies to the optimal polices. Experiments in a central Florida second-magnitude spring demonstrate the ability of the controller to generate and execute an approximate optimal policy in the presence of a time-varying irrotational current.

In Chapter 4, an online approximation of an optimal path following guidance law is developed. ADP is used to approximate the solution to the HJB equation without the need for persistence of excitation. A Lyapunov-based stability analysis proves ultimately bounded convergence of the vehicle to the desired path while maintaining the desired speed profile and ultimately bounded convergence of the approximate policy to the optimal policy. Simulation and experimental results demonstrate the performance of the developed controller.

In Chapter 5, an online approximation of a optimal path planning strategy is developed that takes into consideration input and state constraints. The solution to the HJB equation is approximated using ADP. Locally estimating the unknown value function, the locations of the static obstacles do not need to be known until the obstacles are within an approximation window. The developed feedback policy guarantees ultimately bounded convergence of the approximated path to the optimal path without the requirement of persistence of excitation. An auxiliary controller is developed to ensure the marine craft avoids obstacles while the optimal controller is being identified online. The results are validated with simulations.

The potential of model-based ADP is exciting. Although in Chapters 4 and 5 basic planar kinematic models were utilized, model-based ADP provides a theoretical framework able to develop optimal policies for more complex dynamics. The results in Chapters 4 and 5 could be extended to include full vehicle dynamics. Implementing these extensions on either a surface vehicle or an underwater vehicle would further demonstrate the efficacy of the path following and path planning methods presented in this dissertation for marine craft.

A limitation of the result in Chapter 3 is the reliance on knowledge of the irrotational current's velocity and acceleration. Although the velocity is easily measurable by instruments commonly found on marine craft, the acceleration cannot be directly measured. In this dissertation, the acceleration of the current was numerically computed and filtered for use in the controller. An extension of Chapter 3 would be to develop a controller that did not require acceleration measurements.

In addition to extensions of the existing work in this dissertation, there are countless examples of where approximate optimal control would benefit the motion control systems of marine craft, especially autonomous systems that must act independent of an operator. Being able to adapt to a changing environment, such as coastal environments, can be the difference between success and failure of a mission objective. ADP allows an autonomous system to adapt to its environment since the optimal policy is synthesized in real-time on the vehicle. The ADP methods can be an avenue toward more intelligent motion control systems.

APPENDIX A
EXTENSION TO CONSTANT EARTH-FIXED CURRENT (CH 2)

In the case where the earth-fixed current is constant, the effects of the current may be included in the development of the optimal control problem. The body-relative current velocity $\nu_c(\zeta)$ is state dependent and may be determined from

$$\dot{\eta}_c = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \nu_c,$$

where $\dot{\eta}_c \in \mathbb{R}^n$ is the known constant current velocity in the inertial frame. The functions $Y_{res}\theta$ and $f_{0_{res}}$ in (3–10) can then be redefined as

$$Y_{res}\theta \triangleq \begin{bmatrix} 0 \\ -M^{-1}C_A(-\nu_c)\nu_c - M^{-1}D(-\nu_c)\nu_c \ldots \\ -M^{-1}C_A(\nu_r)\nu_r - M^{-1}D(\nu_r)\nu_r \end{bmatrix},$$

$$f_{0_{res}} \triangleq \begin{bmatrix} J\nu \\ -M^{-1}C_{RB}(\nu)\nu - M^{-1}G(\eta) \end{bmatrix},$$

respectively. The control vector u is

$$u = \tau_b - \tau_c,$$

where $\tau_c(\zeta) \in \mathbb{R}^n$ is the control effort required to keep the vehicle on station given the current and is redefined as

$$\tau_c \triangleq -M_A\dot{\nu}_c - C_A(-\nu_c)\nu_c - D(-\nu_c)\nu_c.$$

APPENDIX B
AUXILIARY CONTROLLER ANALYSIS (CH 5)

Consider a change of coordinates, where $\overline{\zeta} = \zeta - c_{obs_i}$. A positive definite function $V_{obs} : [0, \infty) \to \mathbb{R}^n$ is given as

$$V_{obs} = \overline{\zeta}^T \overline{\zeta}. \tag{B–1}$$

The time derivative of (B–1) is

$$\dot{V}_{obs} = 2\overline{\zeta}^T \dot{\zeta}.$$

Substituting the dynamics (5–1) with the definitions of $\overline{f}$ and $\overline{g}$ provided in Section 5.4, and the controller in (5–2) yields

$$\dot{V}_{obs} = 2\overline{\zeta}^T \left(s\left(\zeta\right) u_s + \left(1 - s\left(\zeta\right)\right) u\left(t\right) \right).$$

Substituting the auxiliary controller defined in (5–21) and the fact that the norm of the optimal controller u is bounded by $\sqrt{2}u_{sat}$, the derivative is lower bounded by

$$\dot{V}_{obs} \geq 2u_{sat} \left\| \zeta \right\| \left(s\left(\zeta\right) - \frac{\sqrt{2}}{1 + \sqrt{2}} \right). \tag{B–2}$$

Let B_{obs_i} denote the local domain of the obstacle centered at c_{obs_i} defined as $B_{obs_i} \triangleq \left\{ \zeta | \overline{\zeta} \leq r_{obs_i} \right\}$. By the definition of the scheduling function in (5–20),

$$\inf_{\zeta \in B_{obs_i}} s\left(\zeta\right) = 0.95.$$

Consider the inequality in (B–2) on the local domain B_{obs_i}, then (B–2) is further bounded by

$$\dot{V}_{obs} \geq 2u_{sat} \left\| \zeta \right\| \left(\sqrt{2} - 1.05 \right).$$

Substituting the function V_{obs}, the derivative may be written as

$$\dot{V}_{obs} \geq 2u_{sat} V_{obs}^{\frac{1}{2}} \left(\sqrt{2} - 1.05 \right).$$

Solving the differential equation using separation of variables, yields

$$V_{obs} \geq \left(\sqrt{2} - 1.05\right)^2 u_{sat}^2 t^2$$

Hence, the obstacle center c_{obs_i} is unstable in the local domain B_{obs_i}. Furthermore, a state trajectory starting outside the local domain B_{obs_i} will not enter the interior of B_{obs_i}.

REFERENCES

[1] G. Griffiths, *Technology and applications of autonomous underwater vehicles*, G. Griffiths, Ed. CRC Press, 2003.

[2] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 2011.

[3] A. J. Sørensen, "A survey of dynamic positioning control systems," *Annual Reviews in Control*, vol. 35, pp. 123–136, 2011.

[4] T. Fossen and A. Grovlen, "Nonlinear output feedback control of dynamically positioned ships using vectorial observer backstepping," vol. 6, pp. 121–128, 1998.

[5] E. Sebastian and M. A. Sotelo, "Adaptive fuzzy sliding mode controller for the kinematic variables of an underwater vehicle," *J. Intell. Robot. Syst.*, vol. 49, no. 2, pp. 189–215, 2007.

[6] E. Tannuri, A. Agostinho, H. Morishita, and L. Moratelli Jr, "Dynamic positioning systems: An experimental analysis of sliding mode control," *Control Engineering Practice*, vol. 18, pp. 1121–1132, 2010.

[7] N. Fischer, D. Hughes, P. Walters, E. Schwartz, and W. E. Dixon, "Nonlinear rise-based control of an autonomous underwater vehicle," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 845–852, Aug. 2014.

[8] R. W. Beard and T. W. Mclain, "Successive galerkin approximation algorithms for nonlinear optimal and robust control." *Int. J. Control*, vol. 71, pp. 717–743, 1998.

[9] Åsmund Våge Fannemel, "Dynamic positioning by nonlinear model predictive control," Master's thesis, Norwegian University of Science and Technology, 2008.

[10] A. Morro, A. Sgorbissa, and R. Zaccaria, "Path following for unicycle robots with an arbitrary path curvature," *IEEE Trans. Robot.*, vol. 27, no. 5, pp. 1016–1023, 2011.

[11] P. Morin and C. Samson, "Motion control of wheeled mobile robots," in *Springer Handbook of Robotics*. Springer Berlin Heidelberg, 2008, pp. 799–826.

[12] L. Lapierre, D. Soetanto, and A. Pascoal, "Non-singular path-following control of a unicycle in the presence of parametric modeling uncertainties," *Int. J. Robust Nonlinear Control*, vol. 16, pp. 485–503, 2003.

[13] D. Dacic, D. Nesic, and P. Kokotovic, "Path-following for nonlinear systems with unstable zero dynamics," *IEEE Trans. Autom. Control*, vol. 52, no. 3, pp. 481–487, 2007.

[14] T. Faulwasser and R. Findeisen, "Nonlinear model predictive path-following control," in *Nonlinear Model Predictive Control*, L. Magni, D. Raimondo, and F. Allgöwer, Eds. Springer, 2009, vol. 384, pp. 335–343.

[15] J. E. da Silva and J. B. de Sousa, “A dynamic programming based path-following controller for autonous vehicles,” *Control and Intell. Syst.*, vol. 39, pp. 245–253, 2011.

[16] P. Sujit, S. Saripalli, and J. Borges Sousa, “Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles,” *IEEE Control Syst. Mag.*, vol. 34, no. 1, pp. 42–59, Feb 2014.

[17] K. Vamvoudakis and F. Lewis, “Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem,” *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[18] S. Bhasin, R. Kamalapurkar, M. Johnson, K. Vamvoudakis, F. L. Lewis, and W. Dixon, “A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems,” *Automatica*, vol. 49, no. 1, pp. 89–92, 2013.

[19] R. Kamalapurkar, P. Walters, and W. E. Dixon, “Concurrent learning-based approximate optimal regulation,” in *Proc. IEEE Conf. Decis. Control*, Florence, IT, Dec. 2013, pp. 6256–6261.

[20] H. Modares, F. Lewis, and M.-B. Naghibi-Sistani, “Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 10, pp. 1513–1525, 2013.

[21] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, “Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems,” *Automatica*, vol. 50, no. 1, pp. 193–202, 2014.

[22] T. Dierks and S. Jagannathan, “Output feedback control of a quadrotor UAV using neural networks,” *IEEE Trans. Neural Netw.*, vol. 21, pp. 50–66, 2010.

[23] H. Zhang, L. Cui, X. Zhang, and Y. Luo, “Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method,” *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2226–2236, 2011.

[24] R. Kamalapurkar, H. Dinh, S. Bhasin, and W. E. Dixon, “Approximate optimal trajectory tracking for continuous-time nonlinear systems,” *Automatica*, vol. 51, pp. 40–48, January 2015.

[25] H. Modares and F. L. Lewis, “Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning,” *Automatica*, vol. 50, no. 7, pp. 1780 – 1792, 2014.

[26] A. Alvarez, A. Caiti, and R. Onken, “Evolutionary path planning for autonomous underwater vehicles in a variable ocean,” vol. 29, pp. 418–429, 2004.

[27] A. V. Rao, D. A. Benson, C. L. Darby, M. A. Patterson, C. Francolin, and G. T. Huntington, “Algorithm 902: GPOPS, A MATLAB software for solving multiple-phase optimal control problems using the Gauss pseudospectral method,” *ACM Trans. Math. Softw.*, vol. 37, no. 2, pp. 1–39, 2010.

[28] K. Yang, S. K. Gan, and S. Sukkarieh, “An efficient path planning and control algorithm for RUAVs in unknown and cluttered environments,” *J. Intell. Robot Syst.*, vol. 57, pp. 101–122, 2010.

[29] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *Int. J. Rob. R*, vol. 30, pp. 846–894, 2011.

[30] S. M. LaValle and P. Konkimalla, “Algorithms for computing numerical optimal feedback motion strategies,” *Int. J. Rob. Res.*, vol. 20, pp. 729–752, 2001.

[31] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[32] A. Shum, K. Morris, and A. Khajepour, “Direction-dependent optimal path planning for autonomous vehicles,” *Robot. and Auton. Syst.*, 2015.

[33] C. Petres, Y. Pailhas, P. Patron, Y. Petillot, J. Evans, and D. Lane, “Path planning for autonomous underwater vehicles,” vol. 23, pp. 331–341, 2007.

[34] R. Kamalapurkar, J. A. Rosenfeld, and W. E. Dixon, “State following (StaF) kernel functions for function approximation part II: Adaptive dynamic programming,” in *Proc. Am. Control Conf.*, 2015, to appear (see also arXiv:1502.02609).

[35] M. Abu-Khalaf, F. L. Lewis, and J. Huang, “Hamilton-Jacobi-Isaacs formulation for constrained input nonlinear systems,” in *Proc. IEEE Conf. Decis. Control*, vol. 5, 2004, pp. 5034–5040.

[36] S. Lyshevski, “Optimal control of nonlinear continuous-time systems: design of bounded controllers via generalized nonquadratic functionals,” in *Proc. Am. Control Conf.*, 1998.

[37] D. Vrabie and F. Lewis, “Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems,” *Neural Netw.*, vol. 22, no. 3, pp. 237 – 246, 2009.

[38] G. V. Chowdhary and E. N. Johnson, “Theory and flight-test validation of a concurrent-learning adaptive controller,” *J. Guid. Control Dynam.*, vol. 34, no. 2, pp. 592–607, March 2011.

[39] G. Chowdhary, T. Yucelen, M. Mühlegg, and E. N. Johnson, “Concurrent learning adaptive control of linear systems with exponentially convergent bounds,” *Int. J. Adapt. Control Signal Process.*, vol. 27, no. 4, pp. 280–301, 2013.

[40] M. Egerstedt, X. Hu, and A. Stotsky, “Control of mobile platforms using a virtual vehicle approach,” *IEEE Trans. Autom. Control*, vol. 46, no. 11, pp. 1777–1782, Nov 2001.

[41] W. E. Dixon, D. M. Dawson, E. Zergeroglu, and A. Behal, *Nonlinear Control of Wheeled Mobile Robots*, ser. Lecture Notes in Control and Information Sciences. Springer-Verlag London Ltd, 2000, vol. 262.

[42] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*, 3rd ed. Wiley, 2012.

[43] R. Kamalapurkar, “Model-based reinforcement learning for online approximate optimal control,” Ph.D. dissertation, University of Florida, 2014.

[44] P. Ioannou and J. Sun, *Robust Adaptive Control*. Prentice Hall, 1996.

[45] W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti, *Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach*. Birkhauser: Boston, 2003.

[46] R. Kamalapurkar, J. Klotz, and W. Dixon, “Concurrent learning-based online approximate feedback Nash equilibrium solution of N -player nonzero-sum differential games,” *IEEE/CAA J. Autom. Sin.*, vol. 1, no. 3, pp. 239–247, July 2014.

[47] S. Sastry and A. Isidori, “Adaptive control of linearizable systems,” *IEEE Trans. Autom. Control*, vol. 34, no. 11, pp. 1123–1131, Nov. 1989.

[48] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2002.

[49] W. Schmidt, “Springs of Florida,” Florida Geological Survey, Bulletin 66, 2004.

[50] H. Zhang, Y. Luo, and D. Liu, “Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints,” *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1490–1503, 2009.

[51] M. Abu-Khalaf and F. Lewis, “Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach,” *Automatica*, vol. 41, no. 5, pp. 779–791, 2005.

BIOGRAPHICAL SKETCH

Patrick Walters received his Bachelor of Science degree in mechanical engineering from the University of Florida, Gainesville, FL, USA. He participated in undergraduate research focused on developing autonomous underwater vehicles as a member of the Machine Intelligence Laboratory. He received his Master of Science degree and his Doctor of Philosophy degree from the Department of Mechanical and Aerospace Engineering at the University of Florida under the supervision of Dr. Warren E. Dixon. His research interests include dynamic programming, optimal control, reinforcement learning, and robust control of uncertain nonlinear systems with a focus on the application to marine craft.