

UC Riverside

UC Riverside Electronic Theses and Dissertations

Title

Unifying Behavior Based Control Design and Hybrid Stability Theory for AUV Application

Permalink

<https://escholarship.org/uc/item/9pb793qd>

Author

Djapic, Vladimir

Publication Date

2009

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Unifying Behavior Based Control Design and Hybrid Stability Theory for AUV
Application

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Vladimir Djapic

March 2009

Dissertation Committee:

Professor Jay A. Farrell, Chairperson
Professor Miroslav Krstic
Professor Matthew J. Barth

Copyright by
Vladimir Djapic
2009

The Dissertation of Vladimir Djapic is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

This research was carried out in the Unmanned Maritime Vehicle (UMV) Laboratory at SPAWAR Systems Center Pacific and Control Research Lab at University of California Riverside during the years 2004 – 2008.

Many people helped in this work. Professor Jay A. Farrell is the best research and dissertation advisor anyone might hope for. He shows an unlimited patience with each of his students. He unconditionally passes his in-depth knowledge in the area to his students. I thank him deeply for his guidance throughout my research years and during the writing of this dissertation. Many times he led me to answers to bigger problems by making me perform various simple exercises. His insight, support and enthusiasm are boundless. I truly admire his great analytical skills and his excellency in finding errors, small as large, in my papers and dissertation. In particular, I would like to express my gratitude for allowing me to join his research team, and for his scientific advices in the area of research, career development and life in general. Under his guidance, I have widened my understanding in many different areas which have been pivotal in writing this dissertation.

I am very thankful to Paul Miller who implemented the majority of AUV simulation and vehicle software. I also thank my committee, Prof. Miroslav Krstic, Prof. Jie Chen, Prof. Matthew Barth, and Prof. Gerardo Beni. I would like to thank Dr. Chuck Hicks and Dr. Roger Boss for believing in this research and providing three years of financial support through Office of Naval Research's (ONR) In-House Laboratory Independent Research Program (ILIR). Testing support for this research was provided in part by Rob Simmons of Program Management Office for Explosive Ordinance Disposal (PMS EOD) and Explosive

Ordnance Disposal Technical Division (EODTECHDIV). It was truly a great privilege to work in such an intelligent and stimulating environment as UMV Lab and UCR. I would like to acknowledge the students in Dr. Farrell's lab, especially Wenjie Dong and Yuanyuan Zhao. It would not have been possible to produce the research results without a lot of help from many people and it is a great pleasure to be given this opportunity to express my gratitude to: Steve Koepenick, Bob Watts, Rich Arrieta, Brian Granger, Jim Walton, Andrew Drieling, Mike Vermette, Todd Webber, Todd Pickering, Chuck Adams, Bernadette Mailey, the rest of the members of UMV Lab, and, finally, all of the members of SSC Pacific Dive Locker.

Thanks also to family and friends for the continuous help, encouragement, patience and support even without actually knowing what I was doing.

To my parents, Ljubica and Branko, and my brother Nenad

ABSTRACT OF THE DISSERTATION

Unifying Behavior Based Control Design and Hybrid Stability Theory for AUV
Application

by

Vladimir Djapic

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, March 2009
Professor Jay A. Farrell, Chairperson

Autonomous Underwater Vehicles (AUVs) are extensively being used by the scientific, oil and gas, and military communities. Many of the missions require the vehicle to function in complex, cluttered environments; to react to changing environmental parameters; and, to find a collision-free path through a workspace containing a significant number of obstacles. Many of the AUV missions currently involve high risk for human lives and excessive costs. State-of-art vehicles are not maneuverable enough to successfully accomplish most of the desired tasks. Desirable vehicle control capabilities include the ability to drive at very low, controllable speeds, the ability to maintain a set distance and attitude (pitch and roll) relative to some surface for optimal sensor (both sonar and video) effectiveness, and the ability for the operator to intervene to change the mission activities. Moreover, a vehicle capable of rotating in place or having a fraction of a meter turning radius is needed to conduct desired missions. Novel controllers to implement these specific behaviors are expected to be nonlinear due, for example, to the fact that the vehicle is maneuvering

at nonzero attitude while translating parallel to the surface. A specific mission that this research addresses is ship hull inspection. This dissertation works through the details of a method to control the vehicle's attitude and translation relative to a surface. The surface of interest for example being a ship hull.

This dissertation describes the derivation, design, simulation, and implementation of a Behavior Based control system. Each behavior is designed using a command filtered backstepping (CFBS) approach. Each behavior and the switching among behaviors is provably stable in the sense of Lyapunov. We use the results from Hybrid System Control in order to prove stability during behavior switching, and thus the overall control system stability. This dissertation presents the simulation and in-water testing results of our control design applied to an AUV.

Contents

List of Figures	xii
1 Introduction	1
1.1 Behavior Based Control	2
1.1.1 Single Controller Approach	2
1.1.2 Multi Controller Approach - Behavior Based Control	3
1.1.3 Two simple behaviors example	8
1.1.4 Criticism of Behavior Based Approaches	8
1.2 Command Filtered Backstepping Design	9
1.2.1 Example of CFBS: Yaw control of nonholonomic land vehicle	13
1.3 Behavior Switching Using Hybrid Control	21
1.4 Dissertation Overview	24
2 Theoretical Approach	26
2.1 Control Signal Derivation: Behavior 1	27
2.2 Control Signal Derivation: Behavior 2	29
2.3 Stability Analysis of Behavior Based Control Design	30
2.4 Switching Analysis of Behavior Based Control Design	34
2.4.1 Behavior Stability	34
2.4.2 Zeno Phenomenon	35
2.4.3 Switching Stability	35
2.5 Simulation Results	39
3 AUV Dynamics, Modeling, Navigation, Simulation, and Applications	43
3.1 AUV kinematics and dynamics	44
3.2 AUV modeling and system identification	47
3.3 AUV Navigation	54
3.3.1 Sensors	56
3.3.2 Error state Kalman Filter	57
3.4 AUV Simulator	58
3.5 Relevant AUV applications	61

3.5.1	Ship Hull Search	63
4	Behavior Based Controller for AUV	72
4.1	Control Signal Implementation	76
4.1.1	Behavior 1: 3D Trajectory following controller	76
4.1.2	Behavior 2: Depth, Attitude and Speed Controller	80
4.1.3	Behavior 3: Surface following controller	81
4.2	Stability of Each Behavior and Behavior Switching	81
4.2.1	Lyapunov Stability of Each Behavior	82
4.2.2	Behavior Switching AUV	82
4.3	Control Signal Derivation: Behavior 1	84
4.3.1	3D Trajectory Following	84
4.3.2	Attitude Control	93
4.3.3	Speed and Angular Rate Control	95
4.3.4	Thruster Command Selection	98
4.3.5	Stability Analysis of Behaviors 1	99
4.4	Control Signal Derivation: Behavior 2	101
4.4.1	Depth Controller	102
4.4.2	Attitude Controller	103
4.4.3	Speed and Angular Rate CFBS	105
4.4.4	Stability Analysis of Behavior 2	105
5	Results	107
5.1	Behavior 2 Simulation Results	109
5.1.1	Simulation Results	109
5.2	Behavior 1 Results	115
5.2.1	Simulation Results	115
5.2.2	Field Test Results	124
5.3	Behavior 3 Results	132
5.3.1	Simulation Results	132
5.3.2	Field Test Results	136
5.4	Switching Performance	139
5.4.1	Simulation Results	139
6	Conclusion	141
6.1	Contribution	143
6.2	Future Work	145
6.3	Publication from the Dissertation by Chapter	146
6.3.1	Journal Article	146
6.3.2	Conference Articles	147
6.3.3	Technical Reports	148
	Bibliography	149
	A Command filter	154

B	Derivation of CFBS Terms for Mode 1 3D translation	157
B.1	Derivation of Eqn. (4.34)	157
B.1.1	\bar{u}_n Error Analysis	158
B.1.2	\bar{u}_e Error Analysis	160
B.1.3	\bar{u}_d Error Analysis	162
B.1.4	A, B, C, and $\mathbf{g}(\bar{\psi})$ Definitions	162
C	Lyapunov Equation	163

List of Figures

1.1	Objective - Mission Scenario	9
1.2	Lyapunov Functions of two behaviors vs. time	24
2.1	States x_1 and x_2 vs. time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command. . .	41
2.2	Top - Error signals \bar{x}_1 and ν_1 vs. time. Bottom - Signal ξ_1 vs. time.	41
2.3	Lyapunov Function and $ [\bar{x}_1, \bar{x}_2]^T $ vs. time.	42
3.1	CETUS II AUV	46
3.2	Thruster Distribution	48
3.3	Comprehensive Vehicle Simulation - Value	62
3.4	Comprehensive Vehicle Simulation - Targets and Sonar	63
3.5	TRANSDEC Pool	71
4.1	Yaw Command Selection	88
5.1	Depth and Attitude vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command. 112	112
5.2	Velocities vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command.	112
5.3	Angular Rates vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command. . .	113
5.4	Thruster signals (%) vs. time.	113
5.5	Top - Depth. Second - Vertical Velocity. Third - Error signals \bar{d} and ν_d vs. time. Bottom - Signal ξ_d vs. time.	114
5.6	Lyapunov Function and $ [\bar{x}]^T $ vs. time.	114
5.7	2 D Position vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command. . .	117
5.8	North, East, Depth Position vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command.	117

5.9	Attitude vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command.	118
5.10	Velocities vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command.	118
5.11	Angular Rates vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command. . .	119
5.12	Thruster signals (%) vs. time.	119
5.13	North, East, Depth Position vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command.	122
5.14	Attitude vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command.	122
5.15	Top - Roll. Second - Roll rate. Third - Error signals $\bar{\phi}$ and ν_ϕ vs. time. Bottom - Signal ξ_ϕ vs. time.	123
5.16	Lyapunov Function and $ [\bar{x}]^\top $ vs. time.	123
5.17	2 D Position vs. Time: Blue line is the actual vehicle trajectory, black line is the command, and the red line is filtered command.	126
5.18	North, East, Depth, and Altitude Position vs. Time: Blue line is the actual vehicle trajectory, black line is the command, and the red line is filtered command.	126
5.19	Attitude vs. Time: Blue line is the actual vehicle attitude, green line is the command, and the red line is filtered command.	127
5.20	North, East, Depth Position vs. Time: Blue line is the actual vehicle trajectory, green line is the command, and the red line is filtered command. . . .	130
5.21	Thruster signals (%) vs. time.	130
5.22	Top - Pitch. Second - Pitch rate. Third - Error signals $\bar{\theta}$ and ν_θ vs. time. Bottom - Signal ξ_θ vs. time.	131
5.23	Attitude vs. Time: Blue line is the actual vehicle attitude, green line is the command, and the red line is filtered command.	134
5.24	North, East, Depth, and Altitude Position vs. Time: Blue line is the actual vehicle trajectory, green line is the command, and the red line is filtered command.	134
5.25	North, East, Depth, and Altitude Position vs. Time: Blue line is the actual vehicle trajectory, green line is the command, and the red line is filtered command.	135
5.26	North, East, Depth, and Altitude Position vs. Time: Blue line is the actual vehicle trajectory, green line is the command, and the red line is filtered command.	137
5.27	Attitude vs. Time: Blue line is the actual vehicle attitude, green line is the command, and the red line is filtered command.	137
5.28	Thruster signals (%) vs. time.	138
5.29	North, East, and Depth Position vs. Time: Blue line is the actual vehicle trajectory, green line is the command, and the red line is filtered command. . . .	140
5.30	Lyapunov Function and $ [\bar{x}]^\top $ vs. time.	140

A.1 Command Filter	156
------------------------------	-----

Chapter 1

Introduction

The goal of this chapter is to introduce our concepts, relevant background through previous research performed, and goals of this dissertation. The three main ideas and concepts of this dissertation are:

1. Behavior Based Control;
2. Command-Filtered Backstepping controller design; and,
3. Behavior switching using results from Hybrid System Theory.

We use Behavior Based Control to show that missions performed by autonomous vehicles can be defined by multiple behaviors and logic for switching among them, Command-Filtered Backstepping technique to design stable behaviors in sense of Lyapunov, and proofs from Hybrid Systems Theory to ensure that behavior switching does not lead to instability. We conclude the introduction chapter presenting the overview of this dissertation.

1.1 Behavior Based Control

One of the objectives of this research is to present a Behavior Based Control strategy as means to control an autonomous vehicle. We argue that this strategy is an efficient means to navigate an autonomous vehicle in complex and uncertain environment. The purpose of this section is to give an overview of Behavior Based Control and to show why we believe that this approach can be used for solving some complex missions the autonomous vehicle are being considered for. Although our approach is theoretically applicable to land, air, surface, and underwater vehicles, this dissertation focuses on AUV platforms and particular missions.

Advances in robotic sensor technology enable advances in vehicle maneuverability and planning strategies to allow scientific and military operators to efficiently test and utilize new sensors in missions of interest. For some searching applications it is a requirement for a vehicle to accurately follow a specific trajectory, make accurate turns and continue to follow the next specified trajectory. Many of these missions require the vehicle to function in complex, cluttered environments, to react to changing environmental parameters, and find a collision-free path through a workspace containing a significant number of obstacles. For successful accomplishment of such a mission, at least two approaches are possible: a single analytic controller or a number of controllers each accomplishing a specific behavior.

1.1.1 Single Controller Approach

The single controller approach works well for many applications; however, it might be challenged when multiple control objectives are desired. In that case, a single controller

must be optimized to achieve all objectives. There might be a performance trade-off among multiple objectives using a single feedback function. If different controllers are designed and optimized for different objectives, and the switching logic is designed properly, the performance might be better. According to the argument in [47], the performance using a hybrid controller should be no worse than what would be by using any single feedback function without switching. In many cases, the performance using a hybrid controller is substantially better.

1.1.2 Multi Controller Approach - Behavior Based Control

In traditional Mobile Robot Control, a complete task is broken into “horizontal” components, such as perception, modeling, planning, task execution, motor control, and each is executed in turn after the other. The right interfaces are defined between subcomponents. Each subcomponent from perception to motor control must succeed.

Another solution is Behavior Based control, wherein each behavior has a single well-defined task. One of the first to introduce this approach was Brooks [7]. According to him, robots should be simple in nature but flexible in behavior, capable of acting autonomously over long periods of time in uncertain, noisy, realistic, and changing worlds, etc. Brooks argues that a better approach to traditional Mobile Robot Control is to divide competence up into levels of abstractness (e.g. reason about object behaviors, plan changes to world, identify objects, monitor changes, build maps, explore, avoid objects, etc.). The behaviors are coordinated in a subsumption architecture. In the subsumption architecture, vehicle functionality is accomplished incrementally via task-achieving behaviors. Each be-

havior is achieved separately and then all of them are tied together to form the robot's control system. Thus, control system is built in such a way that the layers of control system are designed so the robot operates at increasing levels of competence. Higher levels impose control over lower ones. Lower levels continue to function as higher levels are added. As explained in [53] the design of a Behavior Based system includes two significant steps. First, the designer must formulate each reactive behavior quantitatively and implement the behavior as an algorithm. Second, the designer must define and implement a methodology for coordinating the possibly conflicting commands from the different behaviors to achieve good mission performance. As summarized in [6] the advantages of Behavior Based approach are: 1) concurrent execution of multiple behaviors; 2) multiple goals are achieved at the same time; 3) extensibility: new behaviors can be added onto existing behaviors; and 4) robustness: underlying behaviors maintain core competency beneath overlaying behaviors.

The Behavior Based Control approach has been used in a large variety of applications some of them being mobile robots and land vehicles [1, 55], and marine vehicles [6, 56, 64]. Since this research focuses on control of underwater vehicle we will concentrate on those applications.

Behavior Based Design for AUVs

State-of-the-art AUV systems are equipped with tools for generating and executing a priori mission plans. Prior to placing the vehicle in the water, the user constructs a list of mission activities. Once the vehicle is placed in the water and the mission is initiated, the mission planner sequentially completes each activity on the mission list while recording

the sensor data. With such systems, the acquired sensor data does not affect the execution of the mission. After the mission is completed and the vehicle is recovered, the operator analyzes the sensor data for items of interest. At this point, either a human or an AUV with a fixed mission plan may be sent to investigate the areas that contained interesting data. Not only is this approach slow, but when the environment is time-varying, the interesting phenomenon may no longer be at the location that it was previously detected.

An alternative approach is to develop data reactive mission plans. In such an approach, the mission plan is developed onboard the vehicle in response to the sensor data. The mission plan is defined by behaviors and logic for switching between the behaviors. An AUV with a data reactive mission planner can acquire data more quickly and reliably than a fixed mission plan since the reactive planner can keep the AUV in the vicinity of, or maneuver relative to, the interesting sensor data. For example, in the anti-swimmer defense application, the AUV might start the mission in a search behavior that patrols a region until the sonar detects a moving object meeting certain sonar characteristics. After such a detection, the AUV might switch to a surface behavior to alert security personnel, to a tracking behavior, or do both sequentially. In either case, the AUV mission trajectory is directly determined based on sensor data.

The researchers at the Mobile Robotics Research Group at Oxford University, the Computer Science and Artificial Intelligence Laboratory (CSAIL) and Department of Mechanical Engineering at MIT, and the Naval Undersea Warfare Center in Newport Rhode Island (NUWC-NPT) have developed the MOOS-IvP Autonomy Architecture [52, 4]. MOOS stands for “Mission Oriented Operating Suite”, and IvP stands for “Interval Programming”.

This architecture consists of an open-source distributed autonomy architecture and an approach to behavior based control of autonomous vehicles using multiple objective functions that allows reactive control in complex environments with multiple constraints. While low-level control tasks such as navigation, depth keeping and vehicle safety are assigned to the AUV main vehicle computer, all high-level control inputs are derived from a separate vehicle payload computer running the MIT MOOS-IvP system. This autonomy architecture was developed to support adaptive sampling using Unmanned Surface Craft for adaptive and cooperative control of the autonomous sensor platforms in such a network [15] and to control an AUV while towing a 100-meter vector sensor array [5]. There are other AUV applications of Behavior Based Control such as for surveys of coral reefs keeping the vehicle altitude from the bottom constant for optimal video and sonar coverage [56].

Another example where Behavior Based Control is successfully applied to an AUV is presented in [29, 53]. A data reactive mission planner has been successfully demonstrated in the chemical plume tracing application. This planner reacted to chemical and flow sensor data to cause the AUV to track the chemical plume to its source, declare the source location, and perform maneuvers relative to that declared source location. In the latter step, the AUV acquired sidescan sonar data to verify the source and its location. A set of behaviors to achieve a task and a switching logic coordinating the behaviors was utilized. The idea was to decompose the mission into Plume Search Components. During this research, an Adaptive Mission Planner (AMP) drove an AUV based on inputs from environmental sensors that detected a target chemical and successfully located and tracked the chemical plume over large distances to the chemical source. The following behaviors were used: Proceed from

a home location to a region of operation, Search for a chemical plume, Track a chemical plume in a turbulent flow to its source, Re-acquire Plume, Declare the source location, and Return home. The AMP successfully tracked chemical plumes over distances of 975 m and declared source locations with 13 m accuracy relative to ground truth.

In our research effort we used a Behavior Based Control approach for the vehicle that has five thrusters; three in the vertical direction and two directed horizontally, giving it five independent degrees of freedom that enable the vehicle to maneuver in any direction except for purely lateral motion. Thus, we attempt to control an underactuated AUV. The primary mission that we targeted in this research was ship-hull inspection mission. One of the ultimate goals for this mission is to control the vehicle's attitude and translation relative to a surface. We started by building a complete AUV control system that achieves low level vehicle control, such as velocity, depth and angular rate control. This behavior was debugged in simulation and in-water tested thoroughly. Next we built a another control behavior: velocity, depth, and attitude. With this behavior AUV achieved higher level of competence. At this stage we used the AUV and its control system to estimate vehicle parameters (perform system identification) and characterize the sensors. Then we continued with trajectory following with zero attitude, trajectory following with nonzero attitude, and finally accomplished surface following behavior. With such a design scheme we had a working control system for the AUV very early in the development with building the first behavior. Additional behaviors were added later, and the initial working system need never had to be changed. Behavior Based Control Design provided us a way to incrementally build and test a complex AUV control system.

1.1.3 Two simple behaviors example

As a specific and simple example for autonomous vehicle Behavior Based Control, consider the objective to track a trajectory consisting of intersecting curves, see Figure 1.1. A control challenge is the requirement to have an accurate transition between curves which intersect at sharp angles. The goal is to track the trajectory γ_1 until the point P_1 is reached. Next the vehicle is supposed to turn to match the direction (tangent) of the next curve, γ_2 , at the point P_1 . Finally, the vehicle will track the curve γ_2 . The process can be repeated for the subsequent curves, for $i = 1, \dots, n - 1$ to transition from γ_i to γ_{i+1} . This objective can be achieved with two controllers: track curve and turn. This is a very useful process for robotic vehicles in security applications. For interior security, the vehicle may be required to follow a hallway center line while for outdoor security, the vehicle may be required to drive around a fence. In both cases collisions with either a wall or a fence must be avoided, cutting corners are not acceptable since a wall or fence may be there and overshooting the corners may also be unacceptable. We will use the methods from the hybrid systems control literature in order to prove the stability of our Behavior Based Control approach.

1.1.4 Criticism of Behavior Based Approaches

Behavior Based Control algorithms are criticized due to the lack of rigorous stability analysis [46]. However, when each behavior is implemented as a nonlinear controller with a rigorous stability analysis the main remaining issue is the design of behavior switching. This issue is addressed herein from a hybrid systems perspective. Our control architecture combines the advantages of Behavior Based Control and Lyapunov stability since at

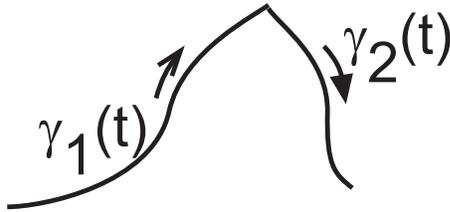


Figure 1.1: Objective - Mission Scenario

the same time allows for a theoretical study of the stability conditions for the system as used in hybrid control. Both the behavior definitions and the behavior switching scheme are discussed. Section 1.2 discusses the approaches for control of autonomous vehicles and presents the high level of Command Filtered Backstepping Design which we used to create stable behaviors for AUV. Section 1.3 shows our approach to create stable behavior switching logic.

1.2 Command Filtered Backstepping Design

The purpose of this section is to show that each behavior would be implemented by a single controller, possibly via multiple backstepping [42] iterations and to show that this can be done using a novel approach to backstepping nonlinear control technique called Command Filtered Backstepping (CFBS). This controller design can be applied to control AUV.

The most common configuration for marine vehicles allows it to move in the horizontal plane having only the surge and yaw axis directly actuated. Current underwater robotic platforms generally fall into one of two categories:

1. Torpedo shaped, single propeller: vehicle hull is long and cylindrical with a single thruster at one end that propels the vehicle forward. Vectored thrust and control surfaces (fins, rudder) are used for control.
2. Rectangular (but smooth, hydrodynamically) shaped, multi thruster: vehicle hull is shaped like a cube or rectangular box with several thrusters. There is at least one thruster (differential thrust) per axis of control, with no active control surfaces. These AUVs look like Remotely Operated Vehicles (ROVs).

These systems contain two major flaws. Torpedo systems lack maneuverability on any axis not along the length of the vehicle and so are not practical in tasks requiring strafing and fine motion control. Box shaped vehicles are generally fairly maneuverable but waste space and power by having each motor manipulate only a single axis. Modern vehicle control algorithms separate horizontal and vertical plane navigation. Our approach is to control horizontal and vertical velocity jointly.

Different approaches for motion control of autonomous vehicles (land, air, surface, and underwater robots) have been analyzed in recent past [9], pp. 121-156 and [59]. The literature, generally, distinguishes among two different motion control problems:

1. path following - where the robot is required to converge to and follow a path where only spatial convergence is necessary without any temporal requirement, and
2. trajectory tracking - where the robot is required to track a time parameterized reference trajectory with temporal requirement.

In this dissertation, we focus on the trajectory tracking problem for autonomous

vehicles. This problem is solved for fully actuated systems and the solutions can be found in the nonlinear control textbooks (pages 540-544 in [41]). The trajectory tracking control for nonholonomic systems is an active area of research interest. An example of such vehicles are wheeled mobile robots. In the earlier work [17] researchers solved the problem of stabilization of a nonholonomic system by linearization around the desired trajectory. This approach gives an explicit control law which locally exponentially stabilizes the system to the desired trajectory assuming that the tracking error due to initial condition or disturbance is not too large. Another approach is to apply feedback linearization techniques. Papers [8, 34] proposed tracking controllers for wheeled mobile robots based on a feedback linearization method, where singular points during the control were avoided by suitably defining the desired trajectory and properly choosing the control parameters. Noting that the dynamics of a wheeled mobile robot is differentially flat [32], dynamic tracking controllers were proposed in [31] with the aid of the differential flatness concept. Backstepping techniques are considered in solving tracking control problem for wheeled mobile robots [30, 38, 39]. Fierro and Lewis propose the dynamical extension which provides a rigorous method of taking into account the specific vehicle dynamics to convert a steering system (kinematic model) command into control inputs for the actual vehicle [30] with the aid of backstepping. In [38], tracking controllers for a wheeled mobile robot were proposed with the aid of backstepping techniques. While in [39], tracking controllers were proposed for a general canonical form which includes the kinematics of wheeled mobile robots based on the backstepping design. Encarnacao and Pascoal combine the trajectory tracking and path following problems and develop a control scheme that can yield good tracking performance

while keeping some of the desired properties of path following [18]. Godhavn considers a full nonlinear model of both the dynamics and the kinematics, and allows large variation in both forward and lateral velocities. He used splines in order to make smooth trajectories and backstepping to stabilize them [36]. It is well known, by Brockett's necessary conditions for stability [63], pages 181-191, that nonholonomic systems cannot be stabilized to a point using smooth-static state feedback. We are interested in trajectory tracking problems where the objective is to force the system position output $z(t) \in \mathfrak{R}^m$ to track a desired ideal output $z_d(t) \in \mathfrak{R}^m$ where $\|\dot{z}_d\| \geq \epsilon > 0$; therefore, we are stabilizing such a system about a trajectory, which is feasible as proven in [36] and [17]. In this dissertation, we focus on trajectory tracking, so $z_d(t)$ represents a position moving along a desired user-defined trajectory. Based on the vehicle kinematics and dynamics, new tracking controllers are proposed with the aid of backstepping techniques. This research presents and analyzes a novel CFBS feedback control implementation approach. In practical applications, implementation of the backstepping approach becomes increasingly complex as the state order increases. The main complicating factor is computation of the command derivatives. This research presents a filtering approach that significantly simplifies the backstepping implementation, analyzes the effect of the command filtering, and derives a compensated tracking error that retains the standard stability properties of backstepping approaches. To reduce the effort of calculating derivatives of signals, command filtered techniques are applied to the proposed tracking controllers. We prove that a compensated tracking error is exponentially decreasing to zero. The tracking controllers with command filters can make the tracking errors converge to a small ball. The radius of the small ball can be reduced by increasing a

parameter in the command filter.

Several places in this dissertation refer to filtering of a signal x_c^o to produce a signal x_c and its derivative \dot{x}_c . This will be referred to as command filtering. The motivation of command filtering is to determine the signals $x_c(t)$ and $\dot{x}_c(t)$ with the error $|x_c^o(t) - x_c(t)|$ being small with magnitude determined by design parameters. Command filtering avoids analytic or numeric differentiation of x_c^o . The filter design is shown in Appendix A. An example for understanding the connection between BS and CFBS is shown in the following section.

1.2.1 Example of CFBS: Yaw control of nonholonomic land vehicle

The objective of this section is to show an example of stabilizing the yaw and yaw rate dynamics for nonholonomic land robot first using conventional BS and then using CFBS. The goal is to use this simple dynamics to show clear transition from BS to CFBS.

The kinematic and dynamic equations for a land vehicle are described as

$$\dot{x} = u \cos(\psi) \tag{1.1}$$

$$\dot{y} = u \sin(\psi) \tag{1.2}$$

$$\dot{\psi} = r \tag{1.3}$$

$$\dot{u} = g(u, r) + F \tag{1.4}$$

$$\dot{r} = f(u, r) + \tau \tag{1.5}$$

where x and y are the earth relative position, ψ is the yaw, u is the speed in body frame, r is the yaw rate in body frame, F is the body-frame control force, τ is the body-frame

control moment, $g(u, r)$ and $f(u, r)$ are friction and other forces acting on the robot. We also assume that the position and speed are measured at the center of the horizontal axle. Therefore, in this model the lateral speed v is zero and has been dropped from the model.

The full control of a land vehicle requires stabilizing eqns. (1.1–1.5). The position dynamics were discussed in [12, 14]. For this section we focus on stabilizing eqns. (1.3) and (1.5) to show the relationship between BS and CFBS.

Conventional Backstepping

This section derives the BS trajectory tracking control law for yaw and yaw rate states. In this section, we use ψ_c^o , r_c^o and their derivatives $\dot{\psi}_c^o$, \dot{r}_c^o in to derive a conventional BS control law.

Given this notation, the tracking error variables are defined as

$$\begin{aligned}\tilde{\psi} &= \psi - \psi_c^o \\ \tilde{r} &= r - r_c^o.\end{aligned}\tag{1.6}$$

For ψ tracking control, the input is the yaw command $\psi_c^o(t)$ and its derivative $\dot{\psi}_c^o(t)$ while for r tracking control, the input is the yaw rate command $r_c^o(t)$ and its derivative $\dot{r}_c^o(t)$. The ideal desired value for ψ , ψ_c^o , is computed by some external controller such as mission planner.

For yaw control, based on eqn. (1.3), we define the signal

$$r_c^o = -K_\psi \tilde{\psi} + \dot{\psi}_c^o\tag{1.7}$$

where K_ψ is a positive constant. Using this definition, the closed-loop tracking error corre-

sponding to eqn. (1.3) is

$$\begin{aligned}
\dot{\psi} &= r_c^o + (r - r_c^o) \\
&= -K_\psi \tilde{\psi} + \dot{\psi}_c^o + \tilde{r} \\
\dot{\tilde{\psi}} &= -K_\psi \tilde{\psi} + \tilde{r}.
\end{aligned} \tag{1.8}$$

Choosing the Lyapunov function as

$$V_\psi = \frac{1}{2} \tilde{\psi}^2,$$

its derivative is

$$\dot{V}_\psi = -K_\psi \tilde{\psi}^2 + \tilde{\psi} \tilde{r}. \tag{1.9}$$

This equation will be used in the stability analysis at the end of this section.

For tracking control using eqn. (1.5) we select the control torque as

$$\tau = -f(u, r) - K_r \tilde{r} - K_r^i e_r + \dot{r}_c^o - r_{bs}, \tag{1.10}$$

where K_r and K_r^i are positive constants. The backstepping term, r_{bs} , will be defined in the stability analysis at the end of this section. With this choice of the control signal and because $\dot{e}_r = \tilde{r}$ and $\ddot{e}_r = \dot{\tilde{r}}$ the dynamics of the r tracking errors

$$\dot{\tilde{r}} = -K_r \tilde{r} - K_r^i e_r - r_{bs} \tag{1.11}$$

can be rewritten as

$$\ddot{e}_r + K_r \dot{e}_r + K_r^i e_r = -r_{bs}. \tag{1.12}$$

We can then define the tracking error vector $q_r = [e_r, \tilde{r}]^\top$ and the tracking error

vector dynamics as

$$\dot{q}_r = \begin{bmatrix} 0 & 1 \\ -K_r^i & -K_r \end{bmatrix} q_r - \begin{bmatrix} 0 \\ 1 \end{bmatrix} r_{bs}, \quad (1.13)$$

where $A_r = \begin{bmatrix} 0 & 1 \\ -K_r^i & -K_r \end{bmatrix}$, and $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

Choosing the Lyapunov function as

$$V_r = \frac{1}{2}(q_r^\top P_r q_r),$$

where P_r is positive definite diagonal matrix defined as in Appendix C.

The appendix also analyzes the terms $Q_r = \begin{bmatrix} 0 & 0 \\ 0 & 2K_r p_{2r} \end{bmatrix}$ and $P_r B = \begin{bmatrix} 0 \\ p_{2r} \end{bmatrix}$,

which are used in the following derivation.

Then the Lyapunov function derivative is

$$\begin{aligned} \dot{V}_r &= \frac{1}{2}(\dot{q}_r^\top P_r q_r + q_r^\top P_r \dot{q}_r) \\ &= \frac{1}{2}[q_r^\top (A_r^\top P_r + P_r A_r) q_r] - q_r^\top P_r B r_{bs} \\ &= -K_r p_{2r} \tilde{r}^2 - r_{bs} p_{2r} \tilde{r}. \end{aligned} \quad (1.14)$$

Define the overall Lyapunov function

$$V = V_\psi + V_r.$$

The time derivative of V along is

$$\begin{aligned} \dot{V} &= \dot{V}_\psi + \dot{V}_r \\ &= -K_\psi \tilde{\psi}^2 - K_r p_{2r} \tilde{r}^2 + \tilde{\psi} \tilde{r} - r_{bs} p_{2r} \tilde{r} \end{aligned}$$

which is the sum of eqns. (1.9) and (1.14). To remove the sign indefinite terms on line two of the above result, we define the backstepping term as

$$r_{bs} = \frac{\tilde{\psi}}{p_{2r}}. \quad (1.15)$$

With these definitions, the derivative of $V(t)$ satisfies

$$\dot{V} \leq -K_\psi \tilde{\psi}^2 - K_r p_{2r} \tilde{r}^2. \quad (1.16)$$

Since the error state is $[\tilde{\psi}, \tilde{r}, e_r]$, the derivative of $V(t)$ is negative semidefinite. This fact proves that the error state is stable and, in particular, e_r is bounded for all $t \geq 0$. LaSalle's invariance theorem (page 128 in [41]), proves that the error state subvector $[\tilde{\psi}, \tilde{r}]$ converges to zero asymptotically.

Remark 1 *Using LaSalle's invariance theorem, eqn. (4.51), as written, would result in e_r converging to zero asymptotically. In reality, the nonlinear moments acting on the vehicle will be distinct from their models $f(u, r)$ and e_r would converge to nonzero values necessary to compensate the model error.*

CFBS

At the beginning of this Section 1.2.1 we defined the BS control law which solved the trajectory tracking problem for the system defined by eqns. (1.3) and (1.5). The conventional BS controller is expressed by eqns. (1.7) and (1.10). Implementation requires that the signal ψ_c^o and its first two derivatives, $\dot{\psi}_c^o$ and $\ddot{\psi}_c^o$, be available to the controller. The second derivative appears in the control of r by selection of τ in eqn. (1.10). The

command filtered backstepping approach avoids the analytic derivation of these expressions by the use of filters. The filter implementation is discussed in detail in Appendix A.

In case of CFBS we define the tracking error variable as

$$\begin{aligned}\bar{\psi} &= \psi - \psi_c \\ \bar{r} &= r - r_c.\end{aligned}\tag{1.17}$$

The methodology for determining the CFBS control law from the BS control law is described in [26, 27]. The resulting CFBS control signals is expressed by eqns.

$$r_c^o = -K_\psi \bar{\psi} + \dot{\psi}_c\tag{1.18}$$

$$\tau = -f(u, r) - K_r \bar{r} - K_r^i \bar{e}_r + \dot{r}_c - \bar{r}_{bs}.\tag{1.19}$$

These equations are derived from the BS control law of eqn. (1.7) and (1.10) by

1. command filtering the signals to generate ψ_c , r_c , $\dot{\psi}_c$, and \dot{r}_c ;
2. replacing $(\tilde{\psi}, \tilde{r}, \text{ and } e_r)$ with $(\bar{\psi}, \bar{r}, \text{ and } \bar{e}_r)$;
3. replacing $(\dot{\psi}_c^o \text{ and } \dot{r}_c^o)$ with $(\dot{\psi}_c \text{ and } \dot{r}_c)$;
4. replacing r_{bs} of eqns. (1.15) by $\bar{r}_{bs} = \frac{\nu_\psi}{p_{2r}}$; and
5. replacing $q_r = [e_r, \tilde{r}]^\top$ with $\bar{q}_r = [\bar{e}_r, \bar{r}]^\top$

where the compensated tracking errors are defined as

$$\nu_\psi = \bar{\psi} - \xi_\psi\tag{1.20}$$

$$\nu_r = \bar{r} - \xi_r\tag{1.21}$$

and

$$\dot{\xi}_\psi = -K_\psi \xi_\psi + (r_c - r_c^o) + \xi_r \quad (1.22)$$

$$\dot{\xi}_r = 0 \quad (1.23)$$

with $\xi_\psi(0) = 0$, and $\xi_r(0) = 0$.

Differentiating the tracking error $\bar{\psi}$ and following a similar procedure to that described above, but using r_c^o as defined in eqn. (1.18), we obtain the $\bar{\psi}$ tracking error dynamics

$$\begin{aligned} \dot{\bar{\psi}} &= \dot{\psi} - \dot{\psi}_c \\ &= r - \dot{\psi}_c = r_c^o + (r - r_c^o) - \dot{\psi}_c \\ \dot{\bar{\psi}} &= -K_\psi \bar{\psi} + (r_c - r_c^o) + \bar{r}. \end{aligned} \quad (1.24)$$

If we then subtract eqn. (1.22) from eqn. (1.24) we get the dynamics of the ψ compensated tracking error

$$\dot{\nu}_\psi = -K_\psi \nu_\psi + \nu_r. \quad (1.25)$$

Differentiating tracking error \bar{r} and following a similar procedure, now using τ defined in eqn. (1.19), we obtain the \bar{r} tracking errors dynamics

$$\begin{aligned} \dot{\bar{r}} &= \dot{r} - \dot{r}_c \\ &= f(u, r) + \tau - \dot{r}_c \\ \dot{\bar{r}} &= -K_r \bar{r} - K_r^i e_r - \bar{r}_{bs}. \end{aligned} \quad (1.26)$$

Subtracting eqn. (1.23) from eqn. (1.26) the dynamics of compensated tracking errors for

r is

$$\dot{\nu}_r = -K_r \nu_r - K_r^i e_r - \frac{\nu_\psi}{p_{2r}}, \quad (1.27)$$

where we have also substituted in the definitions of \bar{r}_{bs} .

From the definitions of the tracking error signals, the definitions of the compensated error signals and (1.22) and (1.23), it follows that $\dot{\bar{e}}_r = \bar{r} = \nu_r$ and $\dot{\bar{q}}_r = A_r \bar{q}_r - B_r \bar{r}_{bs}$. Therefore, the approach from Appendix C can also be used in Lyapunov stability analysis in case of CFBS.

If we define the overall Lyapunov function for CFBS approach as

$$\bar{V} = \bar{V}_\psi + \bar{V}_r$$

and

$$\bar{V}_r = \frac{1}{2}(\bar{q}_r^\top P_r \bar{q}_r),$$

then the Lyapunov function derivative is

$$\dot{\bar{V}} \leq -K_\psi \nu_\psi^2 - K_r p_{2r} \nu_r^2.$$

LaSalle's theorem proves that that the compensated error state subvector $[\nu_\psi, \nu_r]$ converges to zero asymptotically and \bar{e}_r is bounded with values as explained by Remark 1.

For all the nonlinear controllers which implement behaviors that we designed during this research the two theorems proven in [26], [27] apply. Theorem 1 proves that compensated tracking errors converge to zero exponentially. Theorem 2 shows that by increasing the command filter natural frequency ω_n , the solution to the command filtered backstepping closed-loop system can be made arbitrarily close to the backstepping solution that relies on analytic derivatives.

As illustrated above, our strategy will be to design a library of vehicle behaviors, where each behavior uses CFBS control to specify implement a stabilizing controller to enforce the behavior given a suitable trajectory as specified by a user or mission planner. The remaining issue is to use results from hybrid stability to ensure that the behavioral switching does not destabilize the vehicle. Issues related to switching are discussed in Section 1.3.

1.3 Behavior Switching Using Hybrid Control

The purpose of this section is to show that the results from the Hybrid Control Theory can be used to prove the stability in switching when Behavior Based Control is used to achieve the overall mission task.

Many researchers argue that the hybrid control approach using a set of controllers stored in a bank and switching logic between them in order to acquire the desired performance is preferable to a single controller design. Malmborg [45, 16] and McClamroch [47] point out that hybrid control systems can outperform single controller systems and that they can solve problems that can not be dealt with by conventional control, such as when there are multiple design goals that cannot be met by a single controller and can be achieved by using several controllers. There is a set of two or more controllers to choose from in a hybrid control system. Examples in which this strategy is effective includes flight control, air traffic control, missile guidance, process control, robotics etc. Malmborg [45, 16] argues that, for instance, helicopters present dynamical systems with several behaviors of operation. One controller is needed for hovering, one for slow motions and one for fast motions

due to major differences between the models appropriate for these two regimes.

There is a large published literature on advantages and stability analysis for Hybrid Systems [47, 35, 51, 50, 10]. Stability of hybrid systems was analyzed using multiple Lyapunov function approach in [54, 51, 50, 10]. The stability of the overall system is ensured if the Lyapunov function is non-increasing for each behavior and if the stability is maintained during switching instances. The stability of the overall system can be guaranteed if each behavior is designed to be stable in the Lyapunov sense - Lyapunov function is nonincreasing, while it is in use, and if the following conditions are imposed on the switching: Lyapunov function at the initial occurrence of one behavior is equal or less than the Lyapunov function at the initial time of the last occurrence of the same behavior. Fierro [54], Davrazos [35], and Branicky [51, 50] use multiple Lyapunov functions to formulate a theorem for proving asymptotic stability. According to this theorem, stability is guaranteed if a Lyapunov function for each behavior is nonincreasing (meaning that each behavior is stable) and by requiring that the Lyapunov function at the initial occurrence of one behavior is equal or less than the Lyapunov function at the initial time of the last occurrence of the same behavior. This can be graphically shown as in Figure 1.2. In this Figure two behaviors were shown but this can be extended to multiple behaviors. The requirement is that:

$$V_i(x_i(t_{i,k})) \leq V_i(x_i(t_{i,k-1})),$$

where $i = 1, 2, \dots$ represents the active behavior and $k = 1, 2, \dots$ represent the specific time instances. The Figure shows that the value of the Lyapunov function V_2 corresponding to Behavior 2 at time $t_{2,2}$ (second time Behavior 2 is active) is less or equal to its value at

time $t_{2,1}$ (first time Behavior 2 is active) which proves that Behavior 2 is stable in sense of Lyapunov. On the other hand, the Figure shows that the value of the Lyapunov function V_1 corresponding to Behavior 1 at time $t_{1,3}$ (third time Behavior 1 is active) is greater or equal to its value at time $t_{1,2}$ (second time Behavior 1 is active) and at time $t_{1,1}$ (first time Behavior 1 is active) which shows that Behavior 1 cannot be proved stable in sense of Lyapunov.

CFBS method allows us to specify the initial value of the filtered command signals for each state variable that we want to control and the initial values of ξ filter. Since the yaw state compensated error is defined as, for example, in eqn. (4.76) we can set the initial value of filtered yaw command to the value of current vehicle yaw. In addition we can set $\xi_{\psi} = 0$. Since the compensated tracking error is defined as in eqn. (4.76), by selecting those values we ensure that the compensated tracking error is zero at the start of each behavior. Since the compensated tracking error, ν_{ψ} , asymptotically decreases to nonincreasing during the time that the behavior is active, the Lyapunov function remains zero during the entire mission duration.

Via proper design of trajectory generator portion of the Mission Planner we only allow a finite number of switches of the behaviors in any finite amount of time. CFBS design allows us to limit the rate of change of our command signals ensuring sufficiently long duration of each behavior. This is discussed in [28] and will not be discussed further herein. This way we prevent the Zeno effect, a situation where the solution of the system makes an infinite number of discrete transitions between behaviors in a finite amount of time.

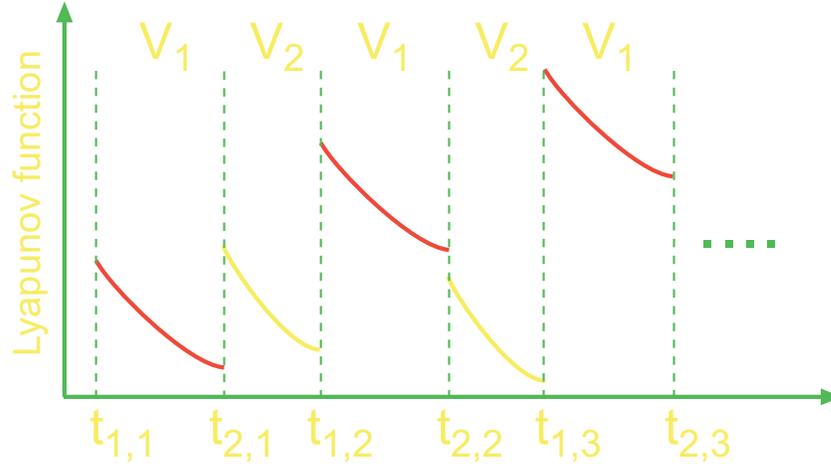


Figure 1.2: Lyapunov Functions of two behaviors vs. time

1.4 Dissertation Overview

This PhD dissertation describes the derivation, simulation, the implementation of the advanced control algorithms onboard the SPAWAR Systems Center’s ship hull inspection unmanned underwater vehicle (AUV) platform (former CETUS AUV). This includes the development of nonlinear controllers using Command Filtered Vector Backstepping approach; defining behaviors described by nonlinear controllers; stability analysis of each controller, behaviors, and overall control system; the development of control algorithms as computer simulation; and the development and field test of the control strategy onboard of AUV.

This Chapter 1 is the introduction of the dissertation. The three main ideas and concepts of this dissertation are presented. Our goal is to use Behavior Based Control design and apply it to design a control system for an AUV. Each behavior will be designed using stable Command-Filtered Backstepping controller design. The stability of the overall

control system will be maintained during behavior switching using results from Hybrid System Theory. The AUV control problem is addressed and related literature is reviewed. Then the proposed behavior based algorithms are given. Chapter 2 presents the specific aspects of each behavior and the logic for switching between the behaviors are described using a simple second order system as an example. This example is used for the analogy of its equations with the kinematic and dynamic equations of motion of robotic vehicles. Chapter 3 presents the AUV dynamics, modeling, navigation, simulation, and related applications. Chapter 4 follows the theoretical approach of Chapter 2 using AUV as a plant for which the control system is designed. Chapter 5 provides the field test results and simulation results. Chapter 6 provides the conclusions and contributions of the Ph.D. research work. In addition, it provides the publications resulting from this PhD work by chapters. Open questions and future research are also discussed in this chapter.

Chapter 2

Theoretical Approach

This chapter describes a Behavior Based Command Filtered Backstepping (CFBS) control design using a second order system as a simple example. The model of the second order system is described by eqns. (2.1–2.2). The reason that we use the second order system is the analogy of its equations with the kinematic and dynamic equations of motion of robotic vehicles. The variable x_1 is analogous to position while the x_2 variable is analogous to the velocity of robotic vehicles. In this chapter we consider two behaviors:

1. Behavior 1: a controller which controls both states, x_1 and x_2 , that defines x_{2c}^o and u to achieve tracking of x_{1c} ; and,
2. Behavior 2: a controller which controls the second system state, x_2 , through u to achieve tracking of x_{2c} .

Making the comparison with the mobile robot control, the Behavior 1 uses the velocity to control position while the Behavior 2 controls the velocity without regard to the position.

In this chapter, we derive and provide proofs for the two theorems that ensure the stability of each behavior, as well as, the stability of the overall hybrid Behavior Based system. The purpose of this chapter is to clearly demonstrate our methodology on a simple system. Once demonstrated, this methodology is applied to a thruster controlled Autonomous Underwater Vehicle (AUV) in subsequent chapters.

The chapter is organized as follows. Section 2.1 and Section 2.2 show the second order system dynamics and derive the control law signals for Behavior 1 and Behavior 2, respectively. Section 2.3 follows the method of [28, 26, 27]. It presents and proves the theorems that guarantee the stability of each behavior designed. Section 2.4 describes our method for stable switching among behaviors. Finally, the performance of the control system proposed is illustrated in simulation in Section 2.5.

2.1 Control Signal Derivation: Behavior 1

The objective of Behavior 1 is to force x_1 to track x_{1c} where x_{1c} and x_{1c}^o are known exogenous signals, specified by an operator or Mission Planner. The x_1 state is controlled by using x_2 as an auxiliary control signal. The state x_2 is controlled by the control signal u .

The dynamic equation for the second order system is described as

$$\dot{x}_1 = f_1(x_1) + x_2 \tag{2.1}$$

$$\dot{x}_2 = f_2(x_1, x_2) + u \tag{2.2}$$

We define tracking and integral errors as

$$\bar{x}_1 = x_1 - x_{1c} \quad (2.3)$$

$$\bar{x}_2 = x_2 - x_{2c} \quad (2.4)$$

$$\bar{e} = \int \bar{x}_2 dt. \quad (2.5)$$

The signal x_{2c} and its derivatives \dot{x}_{2c} are generated from the signal x_{2c}^o using the command filters as explained in Appendix A. Define x_{2c}^o as

$$x_{2c}^o = -f_1(x_1) - K_1 \bar{x}_1 + \dot{x}_{1c} \quad (2.6)$$

Differentiating the tracking error \bar{x}_1 defined in eqn. (2.3) we obtain the \bar{x}_1 tracking error dynamics

$$\begin{aligned} \dot{\bar{x}}_1 &= \dot{x}_1 - \dot{x}_{1c} \\ &= f_1(x_1) + x_2 - \dot{x}_{1c} = f_1(x_1) + x_{2c}^o + (x_2 - x_{2c}^o) - \dot{x}_{1c} \\ \dot{\bar{x}}_1 &= -K_1 \bar{x}_1 + (x_{2c} - x_{2c}^o) + \bar{x}_2. \end{aligned} \quad (2.7)$$

We define the compensated tracking errors as

$$\nu_1 = \bar{x}_1 - \xi_1 \quad (2.8)$$

$$\nu_2 = \bar{x}_2 - \xi_2 \quad (2.9)$$

where

$$\dot{\xi}_1 = -K_1 \xi_1 + (x_{2c} - x_{2c}^o) + \xi_2 \quad (2.10)$$

$$\dot{\xi}_2 = 0 \quad (2.11)$$

with $\xi_1(0) = 0$ and $\xi_2(0) = 0$. If we then subtract eqn. (2.10) from eqn. (2.7) we get the dynamics of the x_1 compensated tracking error

$$\dot{\nu}_1 = -K_1\nu_1 + \nu_2. \quad (2.12)$$

Differentiating tracking error \bar{x}_2 defined in eqn. (2.4) and following a similar procedure, now using u defined as

$$u = -f_2(x_1, x_2) - K_2\bar{x}_2 - K_i\bar{e} + \dot{x}_{2c} - \bar{x}_2^{bs}, \quad (2.13)$$

where the backstepping term, \bar{x}_2^{bs} , will be defined in eqn. (2.23) of the stability analysis of Section 2.3, we obtain the \bar{x}_2 tracking error dynamics

$$\begin{aligned} \dot{\bar{x}}_2 &= \dot{x}_2 - \dot{x}_{2c} \\ &= f_2(x_1, x_2) + u - \dot{x}_{2c} \\ \dot{\bar{x}}_2 &= -K_2\bar{x}_2 - K_i\bar{e} - \bar{x}_2^{bs}. \end{aligned} \quad (2.14)$$

Subtracting eqn. (2.11) from eqn. (2.14) the dynamics of compensated tracking errors for x_2 are

$$\dot{\nu}_2 = -K_2\nu_2 - K_i\bar{e} - \bar{x}_2^{bs}. \quad (2.15)$$

The stability analysis in Section 2.3 will use eqns. (2.12) and (2.15).

2.2 Control Signal Derivation: Behavior 2

The objective of Behavior 2 is to control x_2 . This is implemented by design of the control signal u . While Behavior 2 is active, the state x_1 is not controlled.

Consider only the second state dynamics

$$\dot{x}_2 = f_2(x_2) + u. \quad (2.16)$$

The control signal u can be defined as

$$u = -f_2(x_2) - K_2\bar{x}_2 - K_i\bar{e} + \dot{x}_{2c}, \quad (2.17)$$

where \bar{x}_2 is the tracking error defined in eqn. (2.4). Since, the signal $\xi_2 = 0$, thus $\bar{x}_2 = \nu_2$, by substitution of eqn. (2.17) into the eqn. (2.16), the compensated tracking error dynamics for this controller is

$$\dot{\nu}_2 = -K_2\nu_2 - K_i\bar{e}. \quad (2.18)$$

The stability analysis in Section 2.3 will use eqn. (2.18).

2.3 Stability Analysis of Behavior Based Control Design

This purpose of this section is to show that each of the behaviors implements a stable controller. Two theorems stated and proved in [26] hold. They can be applied to our control design to ensure that it creates closed loop behavior implementations in sense of Lyapunov. For clarity, we will state the theorem for each of the behaviors separately.

Theorem 1 *For the system described by eqns. (2.1–2.2):*

- B1. The feedback control law defined in eqns. (2.6), (2.10–2.11), and (4.2), and the x_{2c} command filter (using the design in Appendix A) provides asymptotic stability for ν_1 , ν_2 and boundedness of ξ_1 , ξ_2 , and \bar{e} .*

B2. The feedback control law defined by eqn. (2.17), with $\xi_2 = 0$ provides asymptotic stability for ν_2 and boundedness of \bar{e} .

△

This theorem will be proved in this section.

It is possible to derive standard backstepping (BS) controllers for Behavior 1 and Behavior 2. Such approach would require analytic computation of the derivative of each pseudo control signal: x_{1c}^o, x_{2c}^o . This would be straight forward in this simple example, but can be quite complicated for systems with higher state order. A motivation for the CFBS approach is that analytic computation of pseudo-command derivatives is not required. Theorem 2 in [26] shows that the difference between the BS tracking errors denoted by \tilde{x} and CFBS tracking errors denoted by \bar{x} (i.e. $|\bar{x}_1 - \tilde{x}_1|$ and $|\bar{x}_2 - \tilde{x}_2|$) are $O\left(\frac{1}{\omega_n}\right)$, which shows that the solution of the BS and CFBS implementations can be made arbitrarily close by choice of the command filter natural frequency ω_n .

Proof: B1

Choose the Lyapunov function for ν_1 as

$$V_{x_1} = \frac{1}{2}\nu_1^2.$$

Its derivative along solutions of eqn. (2.12) is

$$\dot{V}_{x_1} = -K_1\nu_1^2 + \nu_1\nu_2. \tag{2.19}$$

Combining eqns. (2.5) and (2.14) yields

$$\ddot{\bar{e}} + K_2\dot{\bar{e}} + K_i\bar{e} = \bar{x}_2^{bs}. \tag{2.20}$$

We define the tracking error vector as $\bar{\mathbf{q}} = [\bar{e}, \nu_2]^\top$. The tracking error vector dynamic equation is

$$\dot{\bar{\mathbf{q}}} = \begin{bmatrix} 0 & 1 \\ -K_i & -K_2 \end{bmatrix} \bar{\mathbf{q}} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \bar{x}_2^{bs}, \quad (2.21)$$

where $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -K_i & -K_2 \end{bmatrix}$, and $\mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

We choose the Lyapunov function for x_2 as

$$V_{x_2} = \frac{1}{2}(\bar{\mathbf{q}}^\top \mathbf{P} \bar{\mathbf{q}}),$$

where $\mathbf{P} = \text{diag}(p_1, p_2)$ is a positive definite diagonal matrix. By the analysis in Appendix

C

$$\bar{\mathbf{q}}^\top (\mathbf{A}^\top \mathbf{P} + \mathbf{P} \mathbf{A}) \bar{\mathbf{q}} = -K_2 p_2 < 0.$$

The derivative of Lyapunov function along solutions of eqn. (2.21) is

$$\begin{aligned} \dot{V}_{x_2} &= \frac{1}{2}(\dot{\bar{\mathbf{q}}}^\top \mathbf{P} \bar{\mathbf{q}} + \bar{\mathbf{q}}^\top \mathbf{P} \dot{\bar{\mathbf{q}}}) \\ &= \frac{1}{2}[\bar{\mathbf{q}}^\top (\mathbf{A}^\top \mathbf{P} + \mathbf{P} \mathbf{A}) \bar{\mathbf{q}}] - \bar{\mathbf{q}}^\top \mathbf{P} \mathbf{B} \bar{x}_2^{bs} \\ &= -K_2 p_2 \bar{x}_2^2 - \bar{x}_2^{bs} p_2 \bar{x}_2 \\ &= -K_2 p_2 \nu_2^2 - \bar{x}_2^{bs} p_2 \nu_2. \end{aligned} \quad (2.22)$$

Define the overall Lyapunov function for Behavior 1 as

$$V_{b_1} = V_{x_1} + V_{x_2}.$$

The time derivative of V_{b_1} along solutions of eqns. (2.12) and (2.21) is

$$\dot{V}_{b_1} = -K_1 \nu_1^2 + \nu_1 \nu_2 - K_2 p_2 \nu_2^2 - \bar{x}_2^{bs} p_2 \nu_2.$$

which is the sum of eqns. (2.19) and (2.22). To remove the sign indefinite terms of the above result, we define the backstepping term as

$$\bar{x}_2^{bs} = \frac{\nu_1}{p_2}. \quad (2.23)$$

With these definitions, the derivative of $V_{b_1}(t)$ satisfies

$$\dot{V}_{b_1} \leq -K_1\nu_1^2 - K_2p_2\nu_2^2. \quad (2.24)$$

Since the error state is $[\nu_1, \nu_2, \bar{e}]$, the derivative of $V_{b_1}(t)$ is negative semi-definite. This fact proves that the error state is stable and, in particular, \bar{e} is bounded for all $t \geq 0$. LaSalle's invariance theorem (page 128 in [41]), proves that the error state subvector $[\nu_1, \nu_2]$ converges to zero asymptotically.

Proof: *B2*

Similarly, choosing the Lyapunov function for Behavior 2 as

$$V_{b_2} = V_{x_2} = \frac{1}{2}(\bar{\mathbf{q}}^\top \mathbf{P}\bar{\mathbf{q}}),$$

by analysis similar to that above, the time derivative of V_{b_2} along solutions of eqn. (2.21), with the backstepping terms \bar{x}_2^{bs} set to zero, satisfies

$$\dot{V}_{b_2} \leq -K_2p_2\nu_2^2. \quad (2.25)$$

Therefore, each element of $\bar{\mathbf{q}}$ is bounded and ν_2 approaches zero asymptotically.

Remark 2 *The purpose of the integrator is to compensate for model error in the dynamics of the x_2 state. If, for example, the actual system is $\dot{x}_2 = f_2 + A + u$ where eqn. (2.16) has been modified by a model error term A , then a straight forward analysis shows that*

$\bar{e}(t)$ converges to (A/K_i) . We do not discuss model error further herein. See for reference Section 3.2.

2.4 Switching Analysis of Behavior Based Control Design

The purpose of this Section is to show that the switching among behaviors does not lead to instability. The main idea is that for overall stability of our Behavior Based Control design we must ensure three things:

1. Maintain stability during the time each behavior is active.
2. Prevent Zeno Phenomenon.
3. Maintain stability at switching times.

We will show that the CFBS approach allows us to ensure that the Lyapunov function, defined in terms of compensated tracking errors, of the overall switched system is nonincreasing at all times by appropriate choice of the command filter's initial conditions made by a Mission Planner when perfect modeling of the plant is assumed.

2.4.1 Behavior Stability

In Section 2.3 we stated the stability properties for each behavior. Using CFBS control design we ensure that during each time interval for which a behavior is active, its compensated tracking errors converge to zero asymptotically. We use a simple notation useful for two behaviors, but the idea generalizes to a larger number of behaviors. Using Figure 1.2 in Chapter 1 we can visualize the notation for behavior switching. Let $i =$

$0, 1, 2, \dots$ denote switching time. Note that i can be expressed as a function of $j = 0, 1, 2, \dots$. For this Figure and the discussion that follows we assume that Behavior 1 is active initially at $i = 0$, i.e. $t = t_0$. Behavior 1 is active on $t \in [t_{2j}, t_{2j+1}]$ while Behavior 2 is active on $t \in [t_{2j+1}, t_{2j+2}]$. We have already shown that

1. $\dot{V}_{b_1}(t) \leq 0$ and discussed stability properties for the time interval $t \in [t_{2j}, t_{2j+1}]$, and
2. $\dot{V}_{b_2}(t) \leq 0$ and discussed stability properties for the time interval $t \in [t_{2j+1}, t_{2j+2}]$.

2.4.2 Zeno Phenomenon

We ensure that the Zeno effect, a situation where the solution of the system makes an infinite number of discrete transitions between behaviors in a finite amount of time, is precluded. We also preclude the occurrence of the Zeno effects by the design of the mission planner. This means that we must only allow a finite number of switches of the behaviors in any finite amount of time. This is accomplished at the mission planning level by ensuring sufficiently long duration of each behavior, for instance, by limiting the rate of change of our command signals. This is discussed in [28] and will not be discussed further herein.

2.4.3 Switching Stability

We use the results from hybrid systems theory [51, 50, 10] in order to prove the stability during the time instances of switching among behaviors. As explained in [51, 50, 10], we can ensure stability of the switched system if we can show that the Lyapunov functions for each behavior do not increase at subsequent switching instants.

Given a second order system described by eqns. (2.1) and (2.2). We define the following two behaviors:

B1. The control law for the first behavior is

$$\begin{aligned}x_{2c}^o &= -f_1 - K_1 \bar{x}_1 + \dot{x}_{1c}, \\ \ddot{x}_{2c} &= \zeta_2 \omega_{n_2} \dot{x}_{2c} - \omega_{n_2} (x_{2c} - x_{2c}^o), \\ \dot{\xi}_1 &= -K_1 \xi_1 + (x_{2c} - x_{2c}^o), \\ u &= -f_2 - K_2 \bar{x}_2 - K_i \bar{e} + \dot{x}_{2c} - \bar{x}_2^{bs}.\end{aligned}$$

Mission Planner or operator defines x_{1c}^o and x_{1c} .

B2. The control law for the second behavior is

$$\begin{aligned}\ddot{x}_{2c} &= \zeta_2 \omega_{n_2} \dot{x}_{2c} - \omega_{n_2} (x_{2c} - x_{2c}^o), \\ u &= -f_2 - K_2 \bar{x}_2 - K_i \bar{e} + \dot{x}_{2c}.\end{aligned}$$

Mission Planner or operator defines x_{2c}^o and x_{2c} .

The variables $\bar{x}_1 = x_1 - x_{1c}$ and $\bar{x}_2 = x_2 - x_{2c}$ are the state tracking errors, while ζ_2 , ω_{n_2} , K_1 , K_2 , K_i are positive design parameters.

The concern for proving stability of the switched system is to show that $V_{b_1}(t_{2j}) \geq V_{b_1}(t_{2j+2})$ and $V_{b_2}(t_{2j+1}) \geq V_{b_2}(t_{2j+3})$. We have now set the notation framework for stating and proving Theorem 2 which ensures that the stability properties are maintained at behavior switching time instances.

Theorem 2

B1. If, at each start of Behavior 1 ($t = t_{2j}$), the Mission Planner selects the initial values of the filtered command signals to be equal to the current state values, $x_{1c}(t_{2j}) = x_1(t_{2j})$ and $x_{2c}(t_{2j}) = x_2(t_{2j})$, the signals $\xi_1(t_{2j}) = \xi_2(t_{2j}) = 0$, and the integral error to maintain its value (i.e. $\bar{e}(t_{2j}^+) = \bar{e}(t_{2j})$), then the Lyapunov function, V_{b_1} , at each start of each Behavior 1 is a nonincreasing sequence, $V_{b_1}(t_{2j+2}) \leq V_{b_1}(t_{2j})$.

B2. If, at each start of Behavior 2 ($t = t_{2j+1}$), the Mission Planner selects the initial values of the filtered command signal to be equal to the current state value, $x_{2c}(t_{2j+1}) = x_2(t_{2j+1})$, and the integral error to maintain its value (i.e. $\bar{e}(t_{2j+1}^+) = \bar{e}(t_{2j+1})$), then the Lyapunov function, V_{b_2} , at each start of each Behavior 2 is a nonincreasing sequence, $V_{b_2}(t_{2j+3}) \leq V_{b_2}(t_{2j+1})$.

△

Proof: Behavior Switching

The Lyapunov functions for each behavior are defined as

$$V_{b_1} = \frac{1}{2}\nu_1^2 + \frac{1}{2}(\bar{q}^\top P\bar{q})$$

and

$$V_{b_2} = \frac{1}{2}(\bar{q}^\top P\bar{q}),$$

where $\nu_1 = x_1 - x_{1c} - \xi_1$ and $\nu_2 = x_2 - x_{2c} - \xi_2$ are the compensated tracking errors, and the tracking error vector $\bar{q} = [\bar{e}, \bar{x}_2]^\top$, with each element defined as $\bar{e} = \int \bar{x}_2$ and $\bar{x}_2 = \nu_2$.

B1. If Mission Planner selects the initial values of the filters at each start of Behavior 1 ($t = t_{2j}$) as stated in Theorem 2, we have $\nu_1(t_{2j}) = \nu_2(t_{2j}) = \xi_1(t_{2j}) = \xi_2(t_{2j}) = 0$. Thus, the Lyapunov function, $V_{b_1}(t_{2j}) = \bar{e}^2(t_{2j})$. Since $\dot{V}_{b_1}(t) \leq 0$ for $t \in [t_{2j}, t_{2j+1}]$, we have that $\bar{e}^2(t_{2j}) \geq \bar{e}^2(t_{2j+1})$.

B2. If Mission Planner selects the initial values of filters at each start of Behavior 2 ($t = t_{2j+1}$) as stated in Theorem 2, we have $\nu_2 = \xi_2 = 0$. Thus, the Lyapunov function, $V_{b_2}(t_{2j+1}) = \bar{e}^2(t_{2j+1})$. Since $\dot{V}_{b_2}(t) \leq 0$ for $t \in [t_{2j+1}, t_{2j+2}]$, we have that $\bar{e}^2(t_{2j+1}) \geq \bar{e}^2(t_{2j+2})$.

The hybrid system stability requirement states that the Lyapunov function at the initial occurrence of one subsystem is equal or less than the Lyapunov function at the initial time of the last occurrence of the same subsystem [51, 50, 10]. Figure 1.2 in Chapter 1 shows this requirement graphically. Following our notation for Behavior Based hybrid system, the Lyapunov function at the time when one behavior is switched in is equal or greater than the Lyapunov function at the time when the same behavior is switched in next. Since we showed that

$$V_{b_1}(t_{2j}) = \bar{e}^2(t_{2j}) \geq \bar{e}^2(t_{2j+2}) = V_{b_1}(t_{2j+2})$$

for Behavior 1 and

$$V_{b_2}(t_{2j+1}) = \bar{e}^2(t_{2j+1}) \geq \bar{e}^2(t_{2j+3}) = V_{b_2}(t_{2j+3})$$

for Behavior 2 the hybrid stability requirement is met.

2.5 Simulation Results

The purpose of this section is to show the simulation of our controller for the second order plant. We simulated both behaviors and switching between them. Behavior 1 has the goal to cause x_1 to track a signal defined by the Mission Planner to be $x_{1c}^o = \sin(t)$. During Behavior 2, the goal is to cause x_2 to track a signal defined by the Mission Planner to be $x_{2c}^o = 5$. In Behavior 1, control of the x_1 state generates a command for the x_2 state by eqn. (2.6), while the second state generates control signal u to achieve the x_{2c}^o command according to eqn. (4.2).

Figures 2.1–2.3 present the results of an 60 second simulated mission, during which five switching instances between behaviors occurred. The control law parameters are as follows: $f_1 = f_2 = 1$, $K_1 = 1$, $K_2 = 1$, $K_i = 0.1$, $\zeta_1 = \zeta_2 = 0.9$, $p_2 = 1$ for Behavior 1 and $f_2 = 1$, $K_2 = 0.1$, $K_i = 0.1$, $\zeta_2 = 0.9$, $p_2 = 1$ for Behavior 2.

The plot showing x_1 and x_2 versus time is shown in Figure 2.1. Each of these two plots contains three curves, for example x , x_c^o , and x_c . Note that during the time period that Behavior 2 is active, x_1 state plot in Figure 2.1 is not shown, actually zeroed out, since during that behavior x_1 state is not controlled and x_{1c}^o and x_{1c} are undefined. The same was done for the remaining plots in Figures 2.2–2.3. Note that for each state, x_c converges to x_c^o at the rate determined by $\omega_{n_1} = 10 \frac{rad}{s}$ and $\omega_{n_2} = 100 \frac{rad}{s}$ for Behavior 1, and $\omega_{n_2} = 1 \frac{rad}{s}$ for Behavior 2. The signal x_2 converges to and tracks x_{2c} throughout the simulation. The top graph of Figure 2.2 shows \bar{x}_1 and ν_1 . Due to the selection of the CF initial condition, as discussed in Section 2.4, $\nu_1(t)$ is zero. During the time interval following behavior switching while x_{2c} converges to x_{2c}^o , \bar{x}_1 increases and then converges back toward

zero as predicted by the theory. During such time intervals a bounded transient is clearly evident in ξ_1 as seen in the bottom graph of Figure 2.2.

Figure 2.3 plots $V_{b_1}(t)$ versus time for Behavior 1 and $\left\|[\bar{x}_1, \bar{x}_2]^\top\right\|$. Since the tracking error for x_1 state is not zero until the x_2 state converges to its desired value, x_{2c}^o , the $\left\|[\bar{x}_1, \bar{x}_2]^\top\right\|$ will increase during that convergence time, as expected. The Lyapunov function defined in terms of the compensated tracking error is nonincreasing at all times. This result confirms our theoretical conclusion showing that the Lyapunov function of the CFBS approach starts at the value which is a function of the integral error (\bar{e}) at the beginning of each behavior, decreases during the duration of each behavior, and maintains its value during the instances of switching between behaviors.

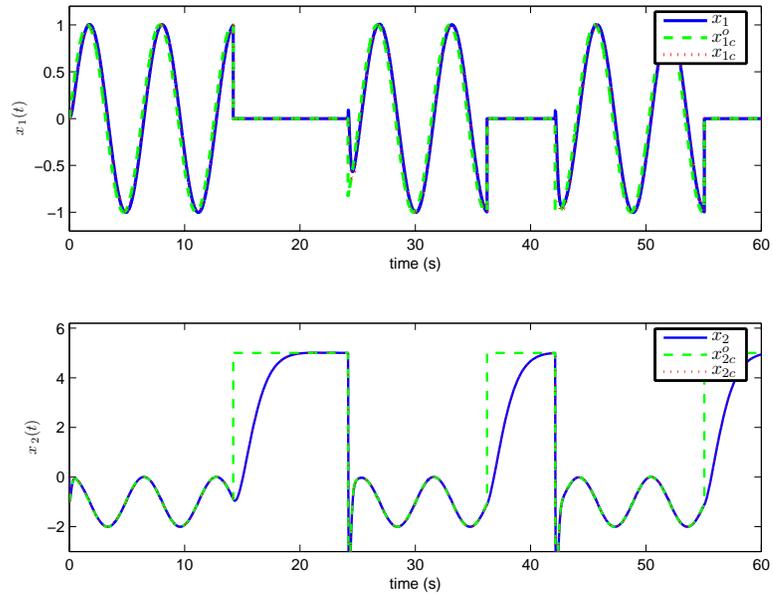


Figure 2.1: States x_1 and x_2 vs. time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command.

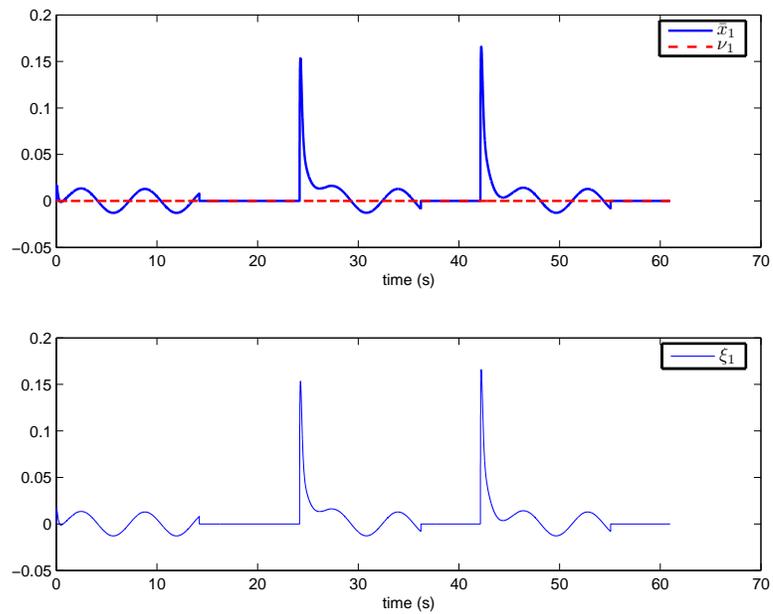


Figure 2.2: Top - Error signals \bar{x}_1 and ν_1 vs. time. Bottom - Signal ξ_1 vs. time.

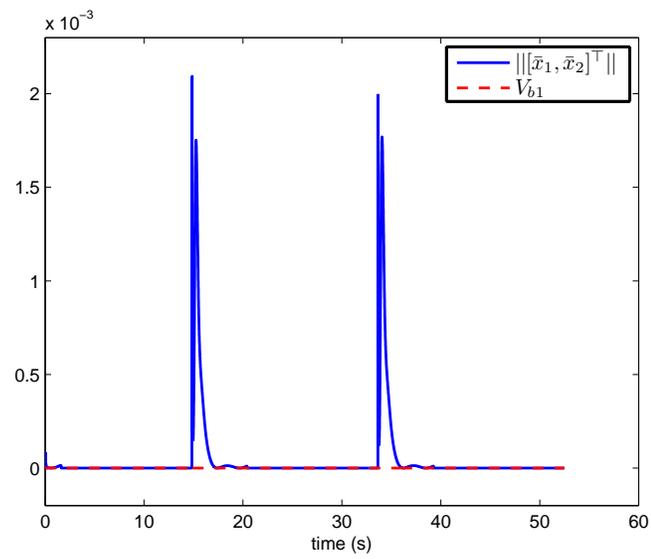


Figure 2.3: Lyapunov Function and $||[\bar{x}_1, \bar{x}_2]^T||$ vs. time.

Chapter 3

AUV Dynamics, Modeling, Navigation, Simulation, and Applications

The objective of this chapter is to introduce various aspects of the AUV that was used for proof of concept for the Behavior Based Control design. The goal of Section 3.1 is to define the equations that characterize the geometrical aspect of vehicle motion through vehicle kinematics equations and describe forces and moments causing the motion through vehicle dynamics equations. The goal of Section 3.2 is to present the simplified model of the vehicle dynamics which was used in our control implementation. In addition, this section describes the methods that we used for system identification and AUV modeling. The goal of Section 3.3 is to describe the vehicle's navigation sensors and briefly describe the design of the navigation system based on an Error state Kalman Filter. The goal of Section 3.4 is

to describe a comprehensive vehicle simulation. The goal of Section 3.5 is to present some of the related AUV applications, focusing on the main application of this research - Ship Hull Inspection.

3.1 AUV kinematics and dynamics

The kinematic model, and forces and torques acting on an AUV are expressed as a set of nonlinear equations as they were derived in Fossen's book [33]. Vehicle motion is fully modeled by six equations of motion that relate force inputs to motion in three translational and three rotational degrees of freedom. The kinematic model relates platform accelerations, velocities, and angular rates to changes in tangent plane position and attitude. The dynamic model is used to study the effect of forces upon these movements.

A reference frame is the perspective from which the motion is described. The Earth-center/Earth-fixed (ECEF) frame is the frame in which the solid Earth is fixed. At a given point on the surface of the Earth a tangent frame might be defined, which is stationary with the respect to the ECEF frame. A moving coordinate frame is fixed to the vehicle and it is called body-fixed reference frame. The origin is usually (and in our case) chosen to coincide with the vehicle's center of gravity (CG). The roll, pitch and yaw angles are a set of Euler angles commonly use in guidance and navigation. We use an Euler attitude representation to describe vehicle orientation, where an Euler three consecutive rotation sequence defines the relationship between tangent and body frames.

The model for an underwater vehicle is described as

$$\dot{\mathbf{p}} = \mathbf{R}_b^t \mathbf{v} \quad (3.1)$$

$$\dot{\Theta} = \mathbf{\Omega} \boldsymbol{\omega} \quad (3.2)$$

$$\dot{\mathbf{v}} = \mathbf{M}^{-1}(\mathbf{F} - \mathbf{F}_n) \quad (3.3)$$

$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1}(\boldsymbol{\tau} - \mathbf{M}_n), \quad (3.4)$$

where $\mathbf{p} = [n, e, d]^\top$ is the earth relative position in meters (m), \mathbf{R}_b^t is the rotation matrix from body to tangent frame, $\mathbf{v} = [u, v, w]^\top$ is the velocity in body frame in $\frac{m}{s}$, $\Theta = [\phi, \theta, \psi]^\top$ is the attitude in rad , $\mathbf{\Omega}$ is a nonlinear (nonsingular except at $\theta = \pm \frac{\pi}{2}$) matrix function of Θ , $\boldsymbol{\omega} = [p, q, r]^\top$ is the inertial rotation rate vector represented in body frame in $\frac{rad}{s}$, \mathbf{F}_n represents the body-frame nonlinear forces in Newtons (N), \mathbf{M}_n represents the body-frame nonlinear moments in Nm , \mathbf{F} is the vector of control forces, $\boldsymbol{\tau}$ is the vector of control moments, \mathbf{M} and \mathbf{J} are mass and inertia matrices, respectively.

The control forces and moments are generated by a set of five thrusters mounted to achieve full angular rate control (i.e., $\boldsymbol{\omega}$), surge control (i.e., u), and heave control (i.e., w). The vehicle is configured (see Fig. 3.1) such that two horizontal thrusters (T_0 and T_2) enable control of u and r while three vertical thrusters enable control of w , p , and q (T_1 , T_3 and T_4).

The vector $\mathbf{T} = [T_0, \dots, T_4]^\top$ of five thrusts is related to the the control forces and moments by a known thrust distribution matrix such that

$$\mathbf{F} = \mathbf{L}_f \mathbf{T}, \quad (3.5)$$



Figure 3.1: CETUS II AUV

and

$$\boldsymbol{\tau} = \mathbf{L}_m \mathbf{T}, \quad (3.6)$$

where $\mathbf{L}_f \in \mathfrak{R}^{2 \times 5}$ and $\mathbf{L}_m \in \mathfrak{R}^{3 \times 5}$, see eqns. (3.5) and (3.6). The AUV propels itself via five thrusters, allowing for a variety of dynamic maneuvers. The vehicle is underactuated since the lateral speed v is not directly affected by the thrusters, but is passively stable. The navigation system and control design will account for v , which as a small effects, but cannot control this variable directly. Effectively, v appears as zero dynamics. This vehicle is available at SSC Pacific and has been used for simulation implementation and in-water testing of our control algorithms. This platform was the CETUS II vehicle originally built at Lockheed Martin. During this research the complete vehicle hardware was rebuilt and software was rewritten.

The rotation matrix, \mathbf{R}_b^t , is defined as

$$\mathbf{R}_b^t = \begin{bmatrix} c\theta c\psi & c\psi s\theta s\phi - c\phi s\psi & c\phi c\psi s\theta + s\phi s\psi \\ c\theta s\psi & c\phi c\psi + s\theta s\phi s\psi & -c\psi s\phi + c\phi s\theta s\psi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}, \quad (3.7)$$

and the angular rate transformation matrix, $\mathbf{\Omega}$, as

$$\mathbf{\Omega} = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix}, \quad (3.8)$$

where the symbols cz , sz , and tz represent $\cos(z)$, $\sin(z)$, and $\tan(z)$.

Note that $\mathbf{\Omega}$ approaches a singularity as $\theta \rightarrow \pm\frac{\pi}{2}$. It is assumed the vehicle will not operate near this singularity. If operation near $\pm\frac{\pi}{2}$ is desired, then an alternative attitude representation, such as quaternions, would remove this singularity.

3.2 AUV modeling and system identification

The objective of this dissertation was not to derive exact equations needed for modeling of an AUV motion, nor to calculate all the hydrodynamic forces acting on the AUV. Both issues have been adequately addressed prior to this research and partially in the earlier phase of this research. Earlier work of this research effort approximated the hydrodynamic coefficients which are included in the control law, such as linear (X_u, Z_w, K_p, M_q, N_r) and quadratic ($X_{uu}, Z_{ww}, K_{pp}, M_{qq}, N_{rr}$) drag coefficients, buoyancy (B) and center of buoyancy ($CB = [x_b, y_b, z_b]^T$) terms, and vehicle's moment of inertia (I_{xx}, I_{yy}, I_{zz}) terms. The values of these terms and other parameters that characterize the AUV, such as mass (m), weight

(W), and lever arm terms ($L_{rx}, L_{ry}, L_{fx}, L_{by}$) are presented in Table 3.1. The mass and inertia matrices are

$$\mathbf{M} = \begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix}, \quad (3.9)$$

and

$$\mathbf{J} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \quad (3.10)$$

L_{rx} is x-distance from the CG to the forward vertical thrusters (T_1 and T_3), L_{ry} is y-distance from CG to forward vertical thrusters (T_1 and T_3), L_{fx} is x-distance from CG to rear vertical thruster (T_4), and L_{by} is y-distance from CG to rear forward thrusters (T_0 and T_2). See Fig. 3.2.

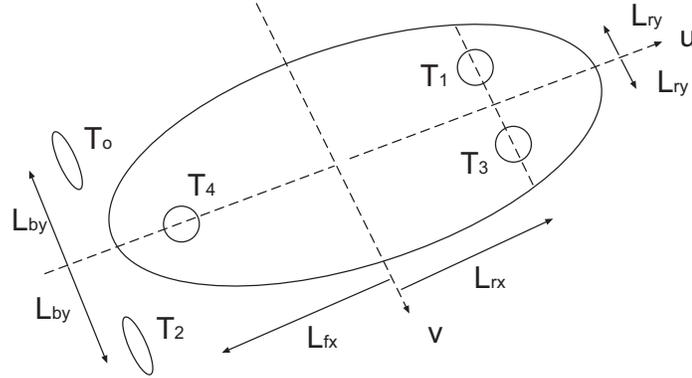


Figure 3.2: Thruster Distribution

Thus, specific to this AUV, lever arm force matrix is

$$\mathbf{L}_f = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \quad (3.11)$$

while lever arm moment matrix is

$$\mathbf{L}_m = \begin{bmatrix} 0 & -L_{ry} & 0 & L_{ry} & 0 \\ 0 & -L_{rx} & 0 & -L_{rx} & L_{fx} \\ L_{by} & 0 & -L_{by} & 0 & 0 \end{bmatrix}. \quad (3.12)$$

Control thruster outputs $\mathbf{T} = [T_0, T_1, T_2, T_3, T_4]^\top$ are expressed as percentage of the full thrust in body frame. The horizontal thrusters are $\frac{1}{2}$ HP while the vertical thrusters are $\frac{1}{4}$ HP; therefore, thrust percent is converted to Newtons using the following conversion for horizontal thrust: $T(\text{N}) = \frac{T(\%)*23*4.45}{100}$, and for vertical thrust: $T(\text{N}) = \frac{T(\%)*12*4.45}{100}$.

The AUV model is approximate, as only drag and buoyancy terms are included in the model. A more detailed model and identification is presented in [33] and [37]. Integral control is introduced within the nonlinear controller to compensate for modeling error.

The following are the reduced dynamic equations for our thruster-powered AUV

$$\dot{u} = -\frac{X_u}{m}u - \frac{X_{uu}}{m}u|u| - \frac{W-B}{m}s\theta + \frac{1}{m}(T_0 + T_2) \quad (3.13)$$

$$\dot{w} = -\frac{Z_w}{m}w - \frac{Z_{ww}}{m}w|w| + \frac{W-B}{m}c\theta c\phi + \frac{1}{m}(T_1 + T_3 + T_4) \quad (3.14)$$

$$\dot{p} = -\frac{K_p}{I_x}p - \frac{K_{pp}}{I_x}p|p| - \frac{y_b B}{I_x}c\theta c\phi + \frac{z_b B}{I_x}c\theta s\phi + \frac{L_{ry}}{I_x}(T_1 - T_3) \quad (3.15)$$

$$\dot{q} = -\frac{M_q}{I_y}q - \frac{M_{qq}}{I_y}q|q| + \frac{z_b B}{I_y}s\theta + \frac{x_b B}{I_y}c\theta c\phi + \frac{L_{fx}}{I_y}T_4 - \frac{L_{rx}}{I_y}(T_1 + T_3) \quad (3.16)$$

$$\dot{r} = -\frac{N_r}{I_z}r - \frac{N_{rr}}{I_z}r|r| - \frac{x_b B}{I_z}c\theta s\phi - \frac{y_b B}{I_z}s\theta + \frac{L_{by}}{I_z}(T_0 - T_2). \quad (3.17)$$

The sixth dynamic eqn. is for v

$$\dot{v} = -\frac{X_v}{m}v - \frac{X_{vv}}{m}v|v| + \frac{W-B}{m}c\theta s\phi. \quad (3.18)$$

Note that v is not directly affected by \mathbf{T} . Also, the coefficients X_v and X_{vv} are positive.

Therefore, the v dynamics is passively stable. The nonlinear control law will not control v ; however, the effect of v on the position kinematics will be accounted for.

Preliminary work focused on estimation of vehicle parameters and characterization of the sensors. The following was the strategy for estimation of unknown parameters.

1. Mass (m) and Weight (W) were determined by direct means.
2. Strategy for X_u and X_{uu} .

First we used the steady-state horizontal velocity, u , condition to approximate X_u and X_{uu} terms. With horizontal acceleration and pitch angle being zero, eqn. (3.18) reduces to

$$T_0 + T_2 = X_u u + X_{uu} u |u|.$$

Different values of $T_0 + T_2$ were applied and u was measured. After performing Least-Squares estimation for limited data sets we estimated X_u and X_{uu} are shown in Table 3.1.

3. Then we used the data set when the control action resulted in vehicle maintaining depth. Therefore, w and \dot{w} were zero. Thus, eqn. (3.14) reduces to

$$T_1 + T_3 + T_4 = (B - W) \cos(\theta) \cos(\phi).$$

For $A = \cos(\theta) \cos(\phi)$ and $x = T_1 + T_3 + T_4$ we obtain the Least-Squares solution for $\Theta = B - W$. Since m and W are already known, we could then easily calculate the buoyancy force.

4. Strategy for Z_w and Z_{ww} .

We performed experiments such that vertical velocity, w , reached steady-state. We

placed the vehicle in the center of 11.6 m deep TRANSDEC pool. With u , roll, pitch, and yaw controllers on and commanded to zero desired value, we commanded the vehicle to follow a triangular pattern between 3 and 9 meters. This pattern results in intervals of time where $T_1 + T_3 + T_4$ and w are constant. Different values of w were effected by changing the period. With vertical acceleration and vehicle's pitch and roll angles being zero, eqn. (3.14) reduces to

$$T_1 + T_3 + T_4 + W - B = Z_w w + Z_{ww} w |w|.$$

This allowed us to estimate Z_w and Z_{ww} using Least-Squares as we approximated X_u and X_{uu} in Step 1. We then turned off the depth control and let the vehicle float to the surface. With thrust values and pitch and roll angles being zero, the eqn. (3.14) reduces to

$$-Z_w w - Z_{ww} w |w| + W - B = m\dot{w}.$$

This allowed us to estimate vehicle's mass and compare it with the measured quantity.

5. Strategy for z_b .

Assuming $y_b = 0$, using data at fixed nonzero commanded roll angles (with $u = \theta = 0$).

With $p = \dot{p} = 0$,

$$z_b B \cos(\theta) \sin(\phi) = -L_{ry}(T_1 - T_3),$$

we easily solved for z_b .

6. Strategy for K_p and K_{pp} .

We performed experiments such that roll rate, p , reaches steady-state. We will command the vehicle to go to 2 m depth. With u , pitch, and yaw controllers on and

commanded to zero, we commanded the vehicle to roll to 30 degrees while maintaining 2 m depth. Immediately after that we commanded a roll of - 30 degrees. We will repeat this experiment several times. This gave us data values in which roll rate was constant. With roll acceleration, pitch, and y_b being zero, eqn. (3.15) reduces to

$$L_{ry}(T_1 - T_3) + z_b B \sin(\phi) = K_p p + K_{pp} p |p|.$$

Because $L_{ry}, z_b, B, T_1, T_3, \phi,$ and p are known this allowed us to estimate K_p and K_{pp} using Least-Squares as we approximated X_u and X_{uu} in Step 1. We then turned off the roll control and let the vehicle roll back from + 30 and - 30 degrees to zero degrees. With thrust values, pitch, and y_b being zero, eqn. (3.15) reduces to

$$-K_p p - K_{pp} p |p| + z_b B \sin(\phi) = I_x \dot{p}.$$

Using parameters estimated in earlier steps, this allowed us to estimate I_x .

7. Strategy for x_b .

Assuming z_b is known from above, we used the data at fixed nonzero commanded pitch angles (with $u = \phi = 0$). With $q = \dot{q} = 0$,

$$z_b B \sin(\theta) - L_{rx}(T_1 + T_3) + L_{fx} T_4 = -x_b B \cos(\theta) \cos(\phi),$$

we easily solved for x_b . This could easily be modified to estimate both x_b and z_b .

8. Strategy for M_q and M_{qq} .

We performed experiments such that pitch rate, q , reached steady-state. We commanded the vehicle to go to 2 m depth. With u , roll, and yaw controllers on and

commanded to zero, we commanded the vehicle to pitch to + 30 degrees. Immediately after that we commanded a pitch of - 30 degrees. We repeated this experiment several times. This gave us data values in which pitch rate was constant. With pitch acceleration and roll being zero, eqn. (3.16) reduces to

$$L_{fx}T_4 - L_{rx}(T_1 + T_3) + z_bB \sin(\theta) + x_bB \cos(\theta) = M_qq + M_{qq}q|q|.$$

This allowed us to estimate M_q and M_{qq} using Least-Squares. We then turned off the pitch control and let the vehicle pitch back from + 30 and - 30 degrees to zero degrees. With thrust values and roll being zero, eqn. (3.16) reduces to

$$-M_qq - M_{qq}q|q| + z_bB \sin(\theta) + x_bB \cos(\theta) = I_y\dot{q}.$$

Using parameters approximated in earlier steps, this allowed us to estimate I_y .

9. Strategy for N_r and N_{rr} .

With the roll, pitch, and u controllers all on and commanded to zero and with the depth controller holding a fixed depth we performed the following. We enabled the yaw controller so we forced the vehicle to spin in circles around its w axis. This resulted in vehicle's yaw rate reaching steady state which will allow us to approximate vehicle's yaw drag parameters, N_r and N_{rr} and inertia term I_z . With yaw acceleration, roll and pitch being zero, the eqn. (3.15) reduces to

$$L_{by}(T_0 - T_2) = N_rr + N_{rr}r|r|.$$

This allowed us to estimate N_r and N_{rr} using Least-Squares. We then turned off the yaw control and let the vehicle settle to some yaw value. With thrust values, pitch

and roll being zero, the eqn. (3.15) reduces to

$$-N_r r - N_{rr} r |r| = I_z \dot{r}.$$

Using parameters approximated in earlier steps, this allowed us to estimate I_z .

The estimated parameters are shown in Table 3.1.

3.3 AUV Navigation

Navigation is one of the primary challenges in AUV research today. It is extremely difficult for an autonomous vehicle to navigate through an unknown environment. Good navigation information is essential for safe operation and recovery of an AUV. With the emergence of inspection-class autonomous underwater vehicles, navigation is becoming increasingly important. As explained in [49, 48], without an operator in the loop, the vehicle itself must use sensors to determine its location, orientation, and motion. The problem is how to effectively use all available sensor inputs to provide a continuous and robust estimate of the vehicle's navigational state. The navigation algorithm tries to estimate the true trajectory given the noisy sensor readings. Navigation estimates the vehicle's true state (position, attitude, velocity, and angular rates).

Section 3.3.1 describes navigation sensors onboard of our AUV while Section provides brief explanation of error state Kalman filter algorithm implemented by navigation software.

Table 3.1: AUV Hydrodynamic Parameters

Parameter	Value	Units	Brief Description
m	117.02	kg	mass
W	1156	N	weight
B	1152.4437	N	bouyancy
$W - B$	-4.4437	N	weight - bouyancy
X_u	0	$\frac{Nms}{rad}$	linear drag in u
X_{uu}	105.16	$\frac{Nms^2}{rad^2}$	quadratic drag in u
Z_w	190.0950	$\frac{Nms}{rad}$	linear drag in w
Z_{ww}	165.2648	$\frac{Nms^2}{rad^2}$	quadratic drag in w
x_b	0.0073	m	center of bouyancy in x
y_b	0	m	center of bouyancy in y
z_b	-0.0062	m	center of bouyancy in z
I_{xx}	3.37	Nms^2	moment of inertia in x
K_p	0.6598	$\frac{Nms}{rad}$	linear drag in p
K_{pp}	0	$\frac{Nms^2}{rad^2}$	quadratic drag in p
I_{yy}	45.4984	Nms^2	moment of inertia in y
M_q	7.3250	$\frac{Nms}{rad}$	linear drag in q
M_{qq}	0	$\frac{Nms^2}{rad^2}$	quadratic drag in q
I_{zz}	30.7391	Nms^2	moment of inertia in z
N_r	0.9525	$\frac{Nms}{rad}$	linear drag in r
N_{rr}	0.0020	$\frac{Nms^2}{rad^2}$	quadratic drag in r
L_{rx}	0.39	m	x-distance from the CG to T_1 and T_3
L_{ry}	0.15	m	y-distance from CG to T_1 and T_3
L_{fx}	0.565	m	x-distance from CG to T_4
L_{by}	0.17	m	y-distance from CG to T_0 and T_2

3.3.1 Sensors

The sensors onboard our AUV include an Inertial Measurement Unit (IMU), Doppler Velocity Log (DVL) / Acoustic Doppler Current Profiler (ADCP), Long-Baseline (LBL transponder), compass, pressure, and altimeters. Most underwater vehicles contain a similar sensor suite. The IMU is the primary high-rate sensor. It measures linear accelerations via accelerometers and angular rates via fiber-optic gyros. We expect this sensor to provide continuous updates without interruption. This sensor is reliable, but due to noise and unknown biases, it alone cannot provide sufficient navigational accuracy. Other sensors provide additional feedback. A DVL provides velocities along four beam directions via acoustic Doppler measurements. The DVL/ADCP measures velocity via the Doppler effect by first emitting encoded acoustic pulses from each of its four transducer heads. These pulses reflect off surfaces, such as the seafloor, and return back to each transducer. The instrument measures the change in frequency between the pulses emitted and those received, which relates to velocities along each beam direction relative to the reflecting object. This sensor can also be used in order to accurately measure the distance from the vehicle to the reflecting surface. If the surface is assumed to be planar, then the vehicle relative attitude to that surface can also be estimated. The three altimeters mounted on top of the vehicle are used to determine the vehicle's distance and attitude to the surface above it. The acoustic LBL system measures round-trip travel time-of-flight of sound waves between a transceiver on the vehicle and four transponder baseline stations at known locations. Because the global position of the four transponders are known the global vehicle position can be determined. An attitude and pressure sensor complete the navigation suite. The attitude sensor provides

orientation measurements, while the pressure sensor provides a sense of vehicle depth. The vehicle's sensor suite is described in detail in [49, 48].

3.3.2 Error state Kalman Filter

One approach to navigation is to treat each sensor independently, each measuring a specific state. A position sensor measures position, a velocity sensor measures velocity, and so forth. This solution, however, is clearly not robust and does not take advantage of the kinematic relationships between states and measurements. We approach this problem using an error state formulation of the Kalman filter to the vehicle navigation problem. The algorithm is explained in detail in [49, 48]. The navigation software is such that, the fast rate sensor is aided with slower rate sensors. Integration of the vehicles high-rate IMUs accelerometers and gyros allows time propagation while other sensors provide measurement corrections. The low-rate aiding sensors include the DVL, acoustic LBL system, pressure sensor, and attitude sensor. Our formulation revolves around the IMU. The augmented system equations model the true system, while the mechanization equations provide the navigation state vector. The difference between the actual and mechanized state is the error state. The error state equations describe the time propagation of the error state. We use the linearized dynamic model of the error state to design the Kalman filter [49, 48]. The Kalman filter time propagation equations propagate the navigation system, the error state, and the error state covariance. The Kalman filter measurement correction equations utilize aiding sensors. Each sensor runs independent of the next, with its own update rate and performance characteristics. Thus, measurement corrections are asynchronous. As a

measurement arrives, it is evaluated and then incorporated into the error state estimate. If a measurement does not arrive, no calculations are necessary. The algorithm does not wait for or expect measurements to arrive in an ordered fashion. The error state and navigation state will propagate, respectively, via the IMU data, with or without measurement corrections. If the measurement lies within three standard deviations of its estimate, the measurement is considered to be valid and it is used to correct the error state estimate. To analyze our filter implementation, we examine the navigation state error, covariance, and measurement residuals. The objective is to drive the navigation state error to zero. Estimation of the unknown accelerometer and gyro bias parameters is a key aspect of our approach. Poor estimation of these parameters leads to inaccuracies in the time propagation of the navigation state vector, and thus poor measurement predictions for the aiding sensors. All of the unknown bias parameters usually converged to reasonable values within the first 500 seconds.

3.4 AUV Simulator

The AUV control development and proof of concept for Behavior Based Control greatly benefited from the comprehensive vehicle simulation. SSC-SD created a vehicle simulation tool to accelerate the integration of vehicle navigation, control, and mission spooling for the SSC-SD CETUS II vehicle. This endeavor proved to be extremely beneficial as it drastically accelerated vehicle development and reduced costly in-water testing requirements. All integration software issues were essentially eliminated prior to any in-water testing. This simulator can simulate any mission from start to end allowing the

operator the chance to check, review, and possibly change the mission plan prior to loading the mission onto the AUV. If the simulation produced successful results, there was an excellent chance that the actual system would obtain similar results. Developers were free to experiment with new algorithms, environments, and situations in faster than real-time simulations, without risk to assets. The graphical display tool allowed developers to visualize performance characteristics quickly. Moreover, the simulator can be used as a great operator training tool through basic operator teleoperation training. The screenshot of the simulation presenting its benefits is shown in Figure 3.3. The Figure is from our team's poster at Autonomous Underwater Vehicle Festival (AUVFest), June 6-15 2007, held in Panama City.

The simulation contains several key components: the vehicle model; sensor models; (actual) vehicle navigation, control, and mission planning code; and the graphical display. The vehicle model represents the true vehicle and attempts to approximate the true vehicle dynamics. It accounts for vehicle dynamics, hydrodynamics, currents, and thruster forces. Given the current vehicle state and thruster forces, the plant integrates one time step to obtain a new true vehicle state. The sensor models take this true vehicle state and generate sensor data. Each sensor model accounts for its update rate, noise and performance characteristics, biases, and random dropouts. The long baseline sensor, for example, must enforce line-of-sight requirements. The Doppler velocity log must verify angle-of-incidence requirements for each beam. All sensor models corrupt their true measurements with Gaussian white noise, and then feed the corrupted measurements into the actual vehicle code. The actual vehicle code is just that. It has no knowledge of the true system. It executes the same

code that resides on the vehicle itself. Maintaining common code between the simulation and vehicle eliminates unnecessary code management. The vehicle code executes the navigation, control, and mission spooler algorithms, which in turn output new thruster forces to the plant. The control simulator allows for behavior based controller implementation. We can specify a mission plan by selecting behaviors from the library, and specifying logic for switching between those behaviors. Each behavior module is implemented by a nonlinear control algorithm. An Adaptive Mission Planner (AMP) manipulates the mission plan in an attempt to obtain a mission goal. The Mission Spooler executes a series of behavioral commands dictated by the AMP. The plant, sensor model, and vehicle code loop propagate the simulation through time. For analysis and verification purposes, an OpenGL graphical display illustrates the vehicle status.

To add geometry to the simulation, one must create a solid model of the object. This process typically involves using a computer aided drawing package to create an object and then exporting the object as a binary stereolithography (STL) file. The STL file format is an industry standard rapid-prototyping format that most packages support. With such a tool, it is easy to create operating scenarios. Consider the ship-hull inspection scenario. If the vehicle has difficulty with certain hull features, such as a bilge keel or around prop shafts, one can create this geometry and experiment. Developers can experiment with countless scenarios without requiring physical access to assets, or risking their own assets. Operators can check for potential hazards prior to executing a mission. Pier pilings, quay walls, and sea floor terrain maps are all possible. The screenshot of the simulation with limpet mines attached to the hull, pier pilings and forward looking sonar coverage is shown in Figure

3.4. This Figure comes from Paul Miller's presentation to our Program Management Office for Explosive Ordnance Disposal (PMS EOD) and Explosive Ordnance Disposal Technical Division (EODTECHDIV) sponsors in Spring 2008.

3.5 Relevant AUV applications

AUVs are becoming important field assets for difficult scientific, commercial, and military missions. An increasing variety of sensors are becoming available for use onboard autonomous vehicles. Given these enhanced sensing capabilities, scientific and military personnel are interested in exploiting autonomous vehicles for increasingly complex missions. Most of these missions require the vehicle to function in complex, cluttered environments and possibly react to changing environmental parameters in order to successfully navigate. These missions currently involve high risk for human lives and high expenditure. For example, unmanned underwater vehicles (AUV) are being considered for mine-countermeasures (MCM) [3, 61, 40], swimmer/diver defense [40, 62], chemical plume tracing [21, 20, 22, 19] and ship hull search [2, 40]. AUVs equipped with high-level control software have a variety of potential applications for Anti-Terrorism/Force Protection (ATFP) objectives. Desirable vehicle control capabilities include the ability to drive at very low, controllable speeds, the ability to maintain a set distance and attitude (pitch and roll) relative to some surface for optimal sensor (both sonar and video) effectiveness, ability to maneuver relative to an object while maintaining that object within view of a specified onboard imaging sensor, and the ability for the operator to intervene to change the mission activities. Moreover, a vehicle capable of rotating in place or having a fraction of a meter turning radius is needed

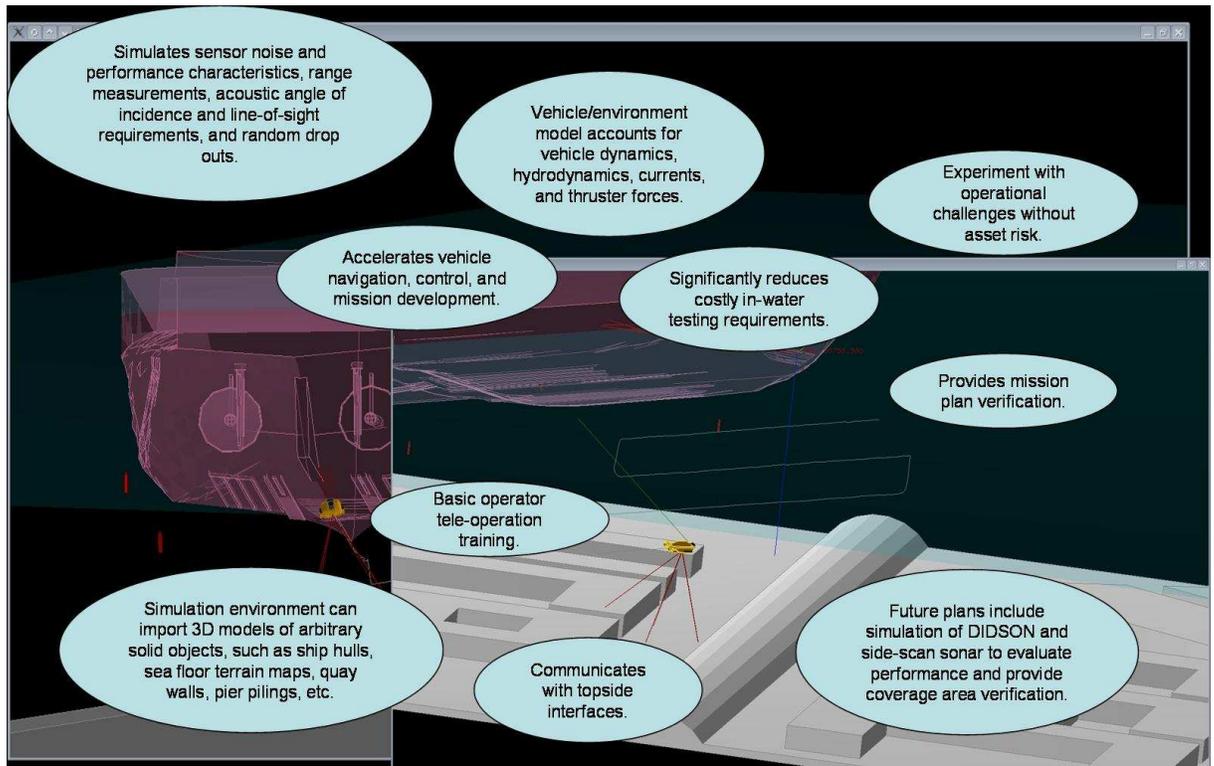


Figure 3.3: Comprehensive Vehicle Simulation - Value

to conduct certain mission. The present state-of-art vehicles are not maneuverable enough to successfully accomplish most of these tasks. The novel controllers are expected to be non-linear due to the fact that the vehicle is translating at nonzero attitude. In addition, such complex missions require on-line adaptive mission planning and control capabilities beyond those demonstrated for the current generation of autonomous vehicles. To be effective for such missions, the AUV must be able to adapt the mission plan in response to sensor data, without human intervention. Furthermore, the utility of new sensors on unmanned vehicles is determined, in part, by the quality and the ease of construction of such data reactive missions. AUVs equipped with this type of software can greatly enhance current under-

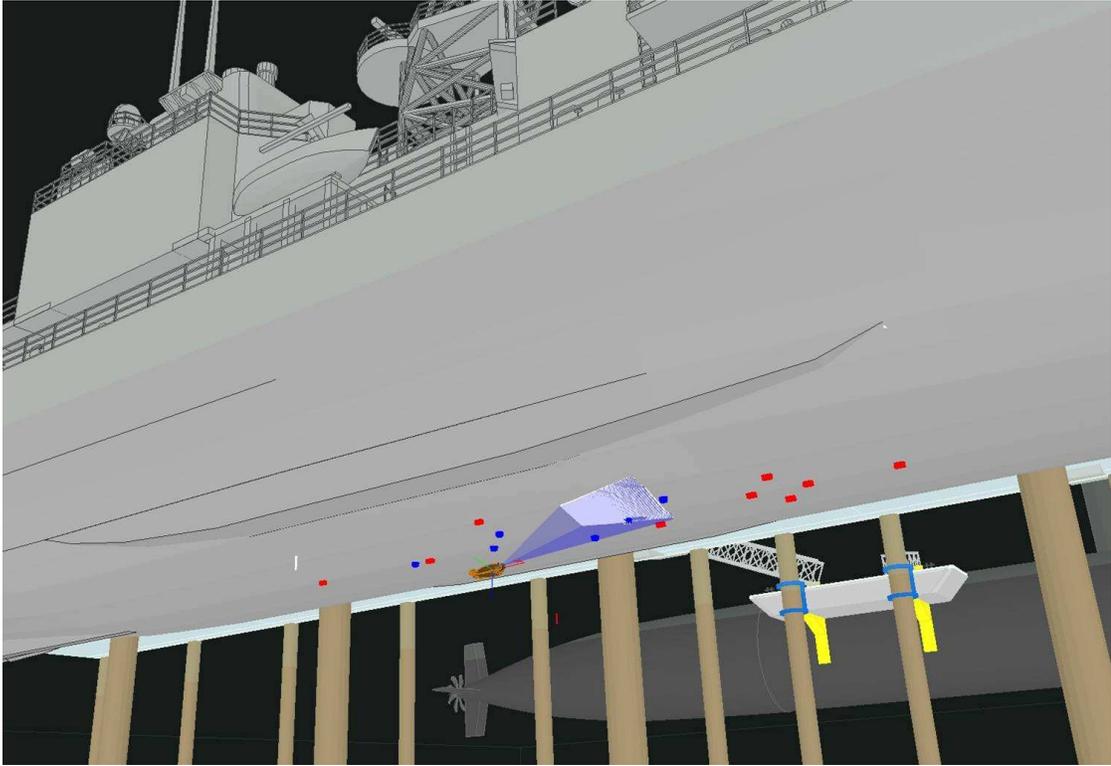


Figure 3.4: Comprehensive Vehicle Simulation - Targets and Sonar

water security capabilities, relieving divers of time-consuming, dangerous tasks, therefore, reducing manpower and mission timeline requirements.

3.5.1 Ship Hull Search

The ship-hull inspection task is such that a “device” must survey the underside of a ship hull and surrounding areas while the ship is at port. At present time, this task is performed by a team of divers. In the future, this inspection device could be a vehicle, either remotely operated or autonomous. We considered select aspects of said inspection vehicle that are essential to the ship-hull inspection task. The particular aspects of interest are

intelligent in nature, and relate to the interaction between the vehicle and operator. This section will step through the evolution of the ship hull search challenge, and ultimately discuss the operational value of implementing Behavior Based Control onto a AUV system to conduct conformal hull searches. Assuming the vehicle has perfect navigation and knows the surrounding environment exactly, controlling the vehicle to avoid contact and explore its environment is complex. By manipulating its control surfaces and/or thrusters on a continuous basis the vehicle must reject ocean currents and surges as it is desirable to maintain a constant standoff distance from the hull. It is difficult to describe “how” a vehicle should survey a ship-hull, especially in complex areas. Typical vehicles exercise predefined search patterns. These search patterns attempt to provide one hundred percent coverage with overlap, but often do not account for complex geometry or the orientation of the sonar. It is not sufficient to pass directly underneath a prop shaft, for example. Crevasses above the shaft are not visible as the shaft obstructs the sonar. The vehicle must attempt to acquire additional aspect angles. Coverage area is difficult to guarantee. Position accuracy is a critical metric for inspection-class vehicles. This information allows operators to localize objects of interest, reacquire contacts, and navigation through complex environments.

Acoustic communications are typically slow, require line-of-sight, and can be unreliable due to the nature of the acoustic environment. It is not possible to transmit real-time sonar data acoustically. For our experiments, the vehicle was equipped with a fiber optic link. Data was sent through the link to ensure vehicle safety. In real a real mission, as the progress is made in the field of acoustic communications, it is likely that the fiber optic

link would be absent. When monitoring a sonar data stream as the vehicle executes its mission, it is often desirable to stop and analyze potential contacts. If the vehicle passed the contact, the operator must reacquire the contact. This task can be challenging. Objects sometimes appear different as the aspect changes. To assist the operator, one possibility is the ability to instruct the vehicle to “rewind” its mission. Here, the vehicle propels itself in the reverse direction along its recent past trajectory. The sonar should continue to point in its original direction. Once the contact comes into view, the operator then instructs the vehicle to maintain station or “pause” while analyzing the contact. When complete, the operator then instructs the vehicle to “play” or continue its original mission. In some situations, however, it may be desirable to instruct the vehicle to navigate around a contact to obtain various aspects. Given the contact location via sonar imagery, operator input, or another data source, it is possible to constrain the vehicle’s movement such that it maintains a constant visual on the contact. This constraint takes partial control of the vehicle, allowing the operator to manipulate the sonar aspect ratio without worrying about losing the contact. The vehicle should also enforce safety standoff distances from its environment at all times. One could automate this inspection process for simple scenarios. The idea of constraining the vehicle’s movements extends beyond contact inspection tasks. At times, it may be desirable for the vehicle to assume partial control while allowing the operator to control other parameters. This ability alleviates the burden of full tele-operation. Consider a station keeping behavior where the objective is to maintain position and attitude. It may be useful to tweak these parameters. The vehicle could hold position as the operator manipulates the desired attitude. Alternatively, the operator could fine-tune the desired position

in any direction as the vehicle maintains its attitude. Consider waypoint navigation. It may be desirable to tweak various parameters, such as desired vehicle speed, depth, standoff, or the waypoints themselves without reloading an entire mission. It is important to note that the objective is to relieve the burden of controlling all vehicle parameters at once, thereby making the vehicle easier to operate.

Military personnel have commenced a series of tests to determine a mechanized approach for conducting underwater searches in navigationally challenging environments which is currently performed by Navy divers. These environments included harbors, ship's berthing areas, berthing piers, and actual ship's hulls. The tools at the Navy team's disposal included preprogram capable AUVs, tow body sonars, and ROVs equipped with a basic video camera. While none of these tools alone proved adequate to conduct such a mission, the team gleaned a comprehensive understanding of the problem and several ideas for an ideal mechanized solution. These concepts and lessons learned have been fed-back to our research team and some have been addressed during our research effort and described in this dissertation. The following sections list the levels of complexity of ship hull inspection mission and basic navigation and control requirements.

Level of complexity

The hull search challenge can be divided into three levels increasing in order of complexity.

1. Side hull search: Search the relatively flat portions of the hull forward of the running gear and aft of the bow dome. Search from the water line to the keel. This effort

is achieved primarily through a precise control system maintaining vehicle relative attitude to and distance from the hull, and precise vehicle navigation (translation along the hull).

2. Complex hull surfaces: bow including sonar domes, and stern areas.
3. Hull appendages: Running gear to include propeller shaft(s), propeller(s), struts, rudder, and other ship's appendages.

A general opinion is that a mechanized solution should focus initially on item one. The second option poses more difficulty and the third option is multiple orders of magnitude more difficult and may be ultimately accomplished more efficiently with the continued use of divers, or with a combination of remotely operated vehicles, and divers. A mechanized solution to accomplish the first item would significantly decrease the dive time and subsequent human-exposure to an extremely high-risk environment.

Navigation requirements

Operators must be certain, with very high level of confidence, that the vehicle has achieved 100 % hull search coverage. This requires the following parameters.

1. Fraction of meter navigational accuracy on a 3 dimensional (x-y-z) coordinate system.
2. Ability for operators to know that the vehicle's sensors have adequately "seen" 100 % of the hull surface.
3. Ability to navigate on both sides of the ship's hull and between the ship and the pier.
4. Ability to navigate in the vicinity of large ferrous objects.

Control System requirements

Desirable vehicle control capabilities include:

1. The ability to drive at very low, controllable speeds (between 0-1 knots) to enable identification of hull objects.
2. The ability to maintain a set distance and attitude (pitch, roll) relative to the ship's hull for optimal sensor (both sonar and video) effectiveness. This also includes not allowing the vehicle to contact the hull surface.
3. The ability to maintain sensor/vehicle position and attitude in order to align the sensors for optimal data capture.
4. The ability for the operator to intervene to change the mission activities or operating parameters.
5. A vehicle capable of rotating in place or having a fraction of a meter turning radius (maximum maneuverability) is needed to conduct the mission.
6. High-frequency sonar and video imagery are still considered the most effective means for AUV/ROVs to identifying Limpet-type mines, and in many cases sonar offers the only identification-quality imagery. The initial goal for an autonomous or semi-autonomous AUV hull-search is to design a control system that can position static sensors in close-proximity to the ship's hull such that they can achieve effective imagery.

Our AUV Ship Hull Search System

This section describes our AUV ship hull inspection system. Operators place one LBL transponder at each corner of a ship and deploy an inspection vehicle. The vehicle executes several passes around and underneath the hull, searching for objects of interest. The AUV has the ability to drive at very low, controllable speeds, the ability to maintain a set distance and attitude (pitch and roll) relative to some surface for optimal sensor (both sonar and video) effectiveness, and the ability for the operator to intervene to change the mission activities. Moreover, the vehicle is capable of rotating in place or having a fraction of a meter turning radius which is needed to conduct the ship-hull inspection mission. Behavior-based, switching, nonlinear controller was implemented. The structure of a controller is hierarchical such that the low level vehicle controller is the same for all mission scenarios while distinct behaviors (upper level controllers) are used based on what mission or part of the mission is desired. The low level controller computes the desired thrust values to achieve the commanded velocities and angular rates and sends the commands to thrusters which create vehicle movement. Behavior controller computes the desired velocities and angular rates to achieve a behavior. On top is the mission planner which selects which behavior will be used. Prior to testing in the ship hull harbor environment we performed testing at the 300 ft by 200 ft by 38 ft deep Transducer Evaluation Center (TRANSDEC) pool at SSC-SD. The pool is bowl shaped so its depth is increasing toward the center of it. It is shown in Figure 3.5. In this pool, the primary mission tested was a mission where the vehicle was using the data from onboard sonar ADCP sensor in order to accurately follow a curved surface below it, the bottom of the pool. The sonar outputs four beams which

measure the distance from the vehicle to the bottom, as well as, the relative vehicle attitude to the bottom. One of the vehicle's behaviors was to maintain the desired stand-off distance from and relative attitude to the curved bottom by adjusting its attitude (pitch and roll) based on the information coming from the sonar sensor. Another task for the vehicle was to track a desired path while maintaining the desired distance (altitude) from the bottom. A similar behavior was utilized to follow a surface above the vehicle (hull of a ship) using the altimeters and at the same time to translate along the hull for the ship hull inspection.



Figure 3.5: TRANSDEC Pool

Chapter 4

Behavior Based Controller for AUV

The objective of this chapter is to describe a Behavior Based Command Filtered Backstepping (CFBS) control design using a thruster powered AUV as an example. We follow the method that we demonstrated on a simple second-order system in Chapter 2. The model of the system is described by eqns. (3.1–3.4). In depth discussion of vehicle kinematics and dynamics is presented in Chapter 3.

In this chapter we consider four behaviors:

1. Behavior 1: 3D Trajectory following controller;
2. Behavior 2: Depth, Attitude, and Speed controller; and,
3. Behavior 3: Surface following controller.

Behavior 1 has the goal to cause the vehicle to track a 3D position (north, east, and

down) signal defined by the Mission Planner and at the same time regulate the attitude (roll and pitch) signal. The inputs to the controller which is used to accomplish this behavior are the arbitrary 3D trajectory and attitude, except $\Theta = \frac{\pi}{2}$. This behavior can be useful in the ship-hull inspection mission as discussed in depth in Chapter 3 and [2, 40]. For example, the vehicle can be commanded to navigate along a straight line alongside the ship and the roll of the vehicle can be adjusted for optimal imaging of a side-scan sonar. Similarly, the vehicle can be commanded to a nonzero value of pitch. This behavior can be useful in the inspection of the complex areas of the ship's hull (rudder, running gear). The forward-looking imaging sonar can then be used to image the complex areas of the hull. Both B1 and B3 require vehicle's nonzero velocity.

During Behavior 2, the goal is to cause the vehicle to track the desired depth, attitude, and speed defined by the Mission Planner. North and east position are not controlled. This behavior is a subset of the 3D trajectory following controller. B2 was designed prior to B1 and served as a building block necessary for the successful design of B1. The inputs to the controller which is used to accomplish this behavior are depth, attitude, except $\phi = \Theta = \frac{\pi}{2}$, and speed. This behavior is used to make accurate turns since this behavior works for zero velocity.

Behavior 3 is a modification of Behavior 1. The roll and pitch commands are externally generated such that the vehicle conforms to the surface above or below it while tracking the desired altitude (distance) from the surface above or below it. The dynamic changing attitude control is a major difference between B1 and B3. B1 served as great exercise and validation of our approach, and preparation for design of B3. B3 is essential

in the ship-hull inspection mission where it is desired that the vehicle attitude conforms to the surface of the hull (above it) and at the same time maintains the desired distance from the hull which places the mounted forward-looking imaging sonar at the optimal grazing angle.

In this chapter, we show Theorems 3 and 4 that ensure the stability of each behavior, as well as, the stability of the overall hybrid Behavior Based system applied to our AUV controller design as shown in Chapter 3. Theorem 3 ensures that each behavior is stable when the appropriate backstepping terms are derived and Theorem 4 ensures that switching among behaviors does not lead to instability by preventing the Zeno phenomenon and by showing that the Lyapunov functions for each behavior do not increase at subsequent switching time instances. Theorem 3 shows that the compensated tracking errors of the CFBS approach have the same properties as the tracking errors of the standard backstepping (BS) approach and it is proved for the AUV system. An additional theorem which proves that the solution to the CFBS closed-loop system can be made arbitrarily close to the BS solution that relies on analytic derivatives is presented and proved in [26, 27]. In this chapter we state Theorem 4 which guarantees stable switching applies, the same it did in the control example of the second-order system of Chapter 2 when certain conditions are met.

Each behavior is designed using CFBS approach. There are two main motivations for CFBS, which is an approximate backstepping approach that

1. eliminates the analytic computation of the derivatives needed in the control design;
- and

2. retains the BS stability properties.

Various approximate BS approaches exist that do not satisfy the second condition [23, 24, 25, 43, 44, 57, 58, 60]. Our design retains the BS stability properties since for a properly designed command filter (unity DC gain to the first output which is the integral of the second output) the closed-loop CFBS will be stable and the compensated tracking error will be $O\left(\frac{1}{\omega_n}\right)$ where ω_n is the bandwidth of the command filter. By increasing ω_n , the solution to the CFBS closed-loop system can be made arbitrarily close to the backstepping solution that relies on analytic derivatives. This holds for any ω_n . Selection of a value for ω_n involves a tradeoff between decreasing the effects of measurement noise (small ω_n) and increasing trajectory tracking accuracy (large ω_n). As ω_n is increased the command filtered variables (e.g., n_c) converge to the ideal desired variables (e.g., n_c^o) more rapidly and track these desired variables more accurately. This, in turn, decreases the magnitude of the signal ζ_n .

To switch among behaviors to achieve complex missions, we build upon Behavior Based control [7], wherein each behavior has a well-defined simple task. In [7], the behaviors are coordinated in a subsumption architecture. An example is presented in [53]. A set of behaviors to achieve a task and a switching logic coordinating the behaviors can be utilized. Behavior Based methods have been criticized due to the lack of rigorous stability analysis [46]. However, when each behavior is implemented as a nonlinear controller with a rigorous stability analysis the main remaining issue is the design of behavior switching. This issue is addressed herein from a hybrid systems perspective.

The chapter is organized as follows. Section 4.1 outlines the control law signals

that need to be implemented for all four behaviors. Section 4.2 follows the method of [28, 26, 27, 11, 13]. It states the theorems that guarantee the stability of each behavior design and describes our method for stable switching among behaviors presented in Chapter 2 of this dissertation. Section 4.3 presents a detailed derivation of 3D trajectory tracking control laws (Behaviors 1 and 3) to deal with vehicle kinematics and dynamics. Section 4.4 presents a derivation of the control laws for Behaviors 2.

4.1 Control Signal Implementation

This section summarizes the control law and the stability properties of the closed loop system for AUV control design. The control law is derived in Section 4.3.

4.1.1 Behavior 1: 3D Trajectory following controller

The inputs to this control loop are: $n_c(t)$, $e_c(t)$, $d_c(t)$, and their derivatives: $\dot{n}_c(t)$, $\dot{e}_c(t)$, $\dot{d}_c(t)$. These signals are generated by the command filtering of the ideal desired values, $n_c^o(t)$, $e_c^o(t)$, $d_c^o(t)$, $\phi_c^o(t)$, $\theta_c^o(t)$ which come from the Mission Planner or the operator. As outputs of command filter, n_c , e_c , and d_c are continuous, bounded, and differentiable as long as n_c^o , e_c^o , and d_c^o , are bounded. In addition, we assume that

Assumption 1

$$\left\| \begin{array}{c} \dot{n}_c \\ \dot{e}_c \end{array} \right\| \geq \epsilon > 0.$$

The following equations describe the control signals for the 3D trajectory following behavior

$$\psi_c^o = \text{atan2}(b, a) \pm \beta \quad (4.1)$$

$$u_c^o = \mathbf{v}_1^\top \left(\mathbf{X} \begin{bmatrix} c\psi_c^o \\ s\psi_c^o \end{bmatrix} + \mathbf{Y} \right) \quad (4.2)$$

$$w_c^o = \mathbf{v}_2^\top \left(\mathbf{X} \begin{bmatrix} c\psi_c^o \\ s\psi_c^o \end{bmatrix} + \mathbf{Y} \right) \quad (4.3)$$

$$\boldsymbol{\omega}_c^o = \boldsymbol{\Omega}^{-1} \left(-\mathbf{K}_\Theta \bar{\boldsymbol{\Theta}} + \dot{\boldsymbol{\Theta}}_c - \boldsymbol{\Theta}_{bs} \right) \quad (4.4)$$

$$\mathbf{F} = \mathbf{M}(\mathbf{F}_n - \mathbf{K}_v \bar{\mathbf{v}} + \dot{\mathbf{v}}_c - \mathbf{v}_{bs}) \quad (4.5)$$

$$\boldsymbol{\tau} = \mathbf{J}(\mathbf{M}_n - \mathbf{K}_\omega \bar{\boldsymbol{\omega}} + \dot{\boldsymbol{\omega}}_c - \boldsymbol{\omega}_{bs}), \quad (4.6)$$

where a , b , β , \mathbf{v}_1 , \mathbf{v}_2 , \mathbf{X} , \mathbf{Y} are defined in Section 4.3.1 while $\bar{\boldsymbol{\Theta}}_{bs}$, \bar{u}_{bs} , \bar{w}_{bs} , and $\bar{\boldsymbol{\omega}}_{bs}$ are defined in Section 4.3.5. The symbols K_{ne} , K_d , \mathbf{K}_Θ , K_u , K_w , \mathbf{K}_ω , K_u^i , K_w^i , \mathbf{K}_ω^i , p_{2u} , p_{2w} , and $p_{2\omega}$ represent positive design parameters. Since v is not directly affected by \mathbf{T} as explained in Section 3.2, for clarity, we use the modified notation for the velocity vector $\mathbf{v} = [u, w]$. As explained in 3.1 since $\boldsymbol{\Omega}$ approaches a singularity as $\theta \rightarrow \pm \frac{\pi}{2}$, this controller is only valid for $\theta \neq \pm \frac{\pi}{2}$ when Euler angles are used for attitude representation. The same constraint is valid for B2 and B3. In addition, the control law implements the signals ξ_n ,

ξ_e , ξ_d , and ξ_Θ using eqns. (4.7) and (4.8)

$$\begin{bmatrix} \dot{\xi}_n \\ \dot{\xi}_e \\ \dot{\xi}_d \end{bmatrix} = - \begin{bmatrix} K_{ne}\xi_n \\ K_{ne}\xi_e \\ K_d\xi_d \end{bmatrix} + \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{Cg}(\bar{\psi}) \end{bmatrix} \begin{bmatrix} \xi_u \\ \xi_w \\ \xi_\psi \end{bmatrix} + \begin{bmatrix} u_{n_c} - u_{n_c}^o \\ u_{e_c} - u_{e_c}^o \\ u_{d_c} - u_{d_c}^o \end{bmatrix} \quad (4.7)$$

$$\dot{\xi}_\Theta = -\mathbf{K}_\Theta \xi_\Theta + \mathbf{\Omega}(\omega_c - \omega_c^o) + \mathbf{\Omega} \xi_w \quad (4.8)$$

$$\dot{\xi}_u = 0 \quad (4.9)$$

$$\dot{\xi}_v = 0 \quad (4.10)$$

$$\dot{\xi}_\omega = \mathbf{0} \quad (4.11)$$

with $\xi_n(0) = 0$, $\xi_e(0) = 0$, $\xi_d(0) = 0$, $\xi_\Theta(0) = \mathbf{0}$, $\xi_u(0) = 0$, $\xi_v(0) = 0$, and $\xi_\omega(0) = \mathbf{0}$, which are derived in Section 4.3.1.

The vehicle model has nine states. Including the six filters defined by eqns. (4.7–4.8), the controller has twenty-seven states: ξ_n , ξ_e , ξ_d , ξ_Θ , Θ_c , $\dot{\Theta}_c$, u_c , \dot{u}_c , w_c , \dot{w}_c , ω_c , $\dot{\omega}_c$, \bar{e}_u , \bar{e}_w , and \bar{e}_ω . Of these,

1. \bar{e}_u , \bar{e}_w , and \bar{e}_ω are five integrator states that are augmented to attain accurate tracking in the presence of model error;
2. Θ_c , $\dot{\Theta}_c$, u_c , \dot{u}_c , w_c , \dot{w}_c , ω_c , and $\dot{\omega}_c$ are the states of the command filters that eliminate the need for analytic command derivatives;
3. ξ_n , ξ_e , ξ_d , and ξ_Θ are the signals defined to compensate the controller for the errors

between the desired and filtered versions of the commands.

Eqns. (4.1–4.6) contain certain subscript and superscript notation. For example, in addition to the variable u , we introduce the variables u_c^o and u_c . The symbol u_c^o represents the ideal desired value for u . The symbol u_c represents a filtered version of u_c^o . The filter, with bandwidth determined by a parameter ω_n , is defined in Appendix A. This notation will also be used similarly to define $n_c^o, n_c, e_c^o, e_c, d_c^o, d_c, \Theta_c^o, \Theta_c, w_c^o, w_c, \omega_c^o$, and ω_c .

Given this notation, the tracking error variables are defined as

$$\begin{aligned} \bar{n} &= n - n_c & \bar{e} &= e - e_c & \bar{d} &= d - d_c & \bar{\Theta} &= \Theta - \Theta_c & \bar{u} &= u - u_c \\ \bar{w} &= w - w_c & \bar{\omega} &= \omega - \omega_c, \end{aligned}$$

and the compensated tracking error variables are

$$\begin{aligned} \nu_n &= \bar{n} - \xi_n & \nu_e &= \bar{e} - \xi_e & \nu_d &= \bar{d} - \xi_d & \nu_\Theta &= \bar{\Theta} - \xi_\Theta \\ \nu_u &= \bar{u} - \xi_u & \nu_w &= \bar{w} - \xi_w & \nu_\omega &= \bar{\omega} - \xi_\omega. \end{aligned}$$

The integral tracking error variables are

$$\bar{e}_u = \int \bar{u} dt \quad \bar{e}_w = \int \bar{w} dt \quad \bar{e}_\omega = \int \bar{\omega} dt.$$

If $\Theta_c = \Theta_c^o$, $\dot{\Theta}_c = \dot{\Theta}_c^o$, $u_c = u_c^o$, $\dot{u}_c = \dot{u}_c^o$, $w_c = w_c^o$, $\dot{w}_c = \dot{w}_c^o$, $\omega_c = \omega_c^o$, and $\dot{\omega}_c = \dot{\omega}_c^o$, then eqns. (4.1–4.6) would implement a conventional backstepping control law. However, the conventional backstepping approach would require analytic expressions for \dot{n}_c^o , \dot{e}_c^o , \dot{d}_c^o , $\dot{\Theta}_c^o$, \dot{u}_c^o , \dot{w}_c^o , and $\dot{\omega}_c^o$, which can be quite complicated. The command filtered backstepping approach avoids the analytic derivation of these expressions by the use of filters. The derivation of the CFBS terms ($\bar{\psi}_{bs}$, \bar{u}_{bs} , \bar{w}_{bs} , and $\bar{\omega}_{bs}$) and the terms that are used to define them (\mathbf{A} , \mathbf{B} , and \mathbf{g}) is explained in detail in Section 4.3.5 and the Appendix B. The approach is designed to maintain the exponential stability properties of the backstepping

approach for a set of the compensated tracking errors denoted by $\nu_n, \nu_e, \boldsymbol{\nu}_\Theta, \nu_u, \nu_w$, and $\boldsymbol{\nu}_\omega$ to ensure that the control signals $\boldsymbol{\Theta}_c, u_c, w_c$, and $\boldsymbol{\omega}_c$ are the same as those of the conventional BS approach within an error proportional to $\frac{1}{\omega_n}$.

4.1.2 Behavior 2: Depth, Attitude and Speed Controller

The inputs to this control loop are: $d_c(t), \phi_c(t), \theta_c(t), \psi_c(t), u_c(t)$, and their derivatives: $\dot{d}_c(t), \dot{\phi}_c(t), \dot{\theta}_c(t), \dot{\psi}_c(t), \dot{u}_c(t)$. These signals are generated by the command filtering of the ideal desired values, $d_c^o(t), \phi_c^o(t), \theta_c^o(t), \psi_c^o(t), u_c^o(t)$, which come from the Mission Planner or the operator.

Depth, attitude, and speed controller is a subset of the 3D trajectory following controller. The control law eqns. (4.12–4.15) describe this controller. The main difference between these two controllers is that the depth (d_c^o), the yaw (ψ_c^o), and the speed (u_c^o) commands are not calculated within the controller, but they are commanded by the operator or the Mission Planning Software. Since this is the case, the BS terms u_{bs} and ψ_{bs} are identically zero.

$$\boldsymbol{w}_c^o = \frac{u \sin(\theta) - \cos(\theta) \sin(\phi)v - K_d \tilde{d} + \dot{d}_c}{\cos(\theta) \cos(\phi)} \quad (4.12)$$

$$\boldsymbol{\omega}_c^o = \boldsymbol{\Omega}^{-1} \left(-\mathbf{K}_\Theta \bar{\boldsymbol{\Theta}} + \dot{\boldsymbol{\Theta}}_c \right) \quad (4.13)$$

$$\mathbf{F} = \mathbf{M}(\mathbf{F}_n - \mathbf{K}_v \bar{\mathbf{v}} + \dot{\mathbf{v}}_c - \mathbf{v}_{bs}) \quad (4.14)$$

$$\boldsymbol{\tau} = \mathbf{J}(\mathbf{M}_n - \mathbf{K}_\omega \bar{\boldsymbol{\omega}} + \dot{\boldsymbol{\omega}}_c - \boldsymbol{\omega}_{bs}). \quad (4.15)$$

In addition, the control law implements the signals ξ_d using eqns. (4.16)

$$\dot{\xi}_d = -K_d \xi_d + (w_c - w_c^o) + \xi_w. \quad (4.16)$$

This controller is not valid for $\theta = \pm\frac{\pi}{2}$ which is the case in all behaviors and it is explained in 4.1.1. In addition to this, this controller is only valid for $\phi \neq \pm\frac{\pi}{2}$. It is assumed that vehicle attitude will not approach $\pm\frac{\pi}{2}$ during this behavior.

4.1.3 Behavior 3: Surface following controller

The inputs to this control loop are: $n_c(t)$, $e_c(t)$, $a_c(t)$, $\phi_c(t)$, $\theta_c(t)$, and their derivatives: $\dot{n}_c(t)$, $\dot{e}_c(t)$, $\dot{a}_c(t)$, $\dot{\phi}_c(t)$, $\dot{\theta}_c(t)$. The $a_c(t)$ command is introduced as it is the desired distance to the surface which is controlled instead of vehicle's depth. These signals are generated by the command filtering of the ideal desired values, $n_c^o(t)$, $e_c^o(t)$, $a_c^o(t)$, which come from the Mission Planner or the operator, while $\phi_c^o(t)$ and $\theta_c^o(t)$ are generated such that the relative roll and pitch angles between the vehicle and surface is zero.

Behavior 3, the surface following controller, has the goal to track vehicle's desired position as in B1 and at the same time track specific altitude, roll and pitch, commands such that the vehicle conforms to the surface above or below it. In this mode, the altimeter is measuring altitude: a , relative roll: ϕ_r , and relative pitch: θ_r . The mission planner is specifying a_c , ϕ_{r_c} , and θ_{r_c} where the last two mean commanded relative angles. At present these were always zero (i.e., AUV attitude matches the hull attitude). The controller constructs roll, ϕ_c^o , and pitch, θ_c^o , commands.

4.2 Stability of Each Behavior and Behavior Switching

This purpose of this section is to show that each of the behaviors implements a stable controller and that the switching among behaviors which happens during an AUV

mission does not cause instability in the control design.

4.2.1 Lyapunov Stability of Each Behavior

In this section we state a theorem which proves that each behavior is stable. For clarity, we will state the theorem for each of the behaviors separately.

Theorem 3 *For the system described by eqns. (4.1–4.6):*

B1, B3, B4. *The feedback control law defined in eqns. (4.1–4.6), and the ψ_c , u_c , and ω_c command filters (using the design in Appendix A) provides asymptotic stability for $\nu_n, \nu_e, \nu_d, \nu_\Theta, \nu_u, \nu_w$, and ν_ω and boundedness of $\xi_n, \xi_e, \xi_d, \xi_\Theta, \xi_u, \xi_w, \xi_\omega, \bar{e}_u, \bar{e}_w$, and \bar{e}_ω .*

B2. *The feedback control law defined by eqn. (4.12–4.15), with $\xi_u = \xi_w = \xi_\omega = 0$ provides asymptotic stability for ν_d, ν_u, ν_w , and ν_ω and boundedness of $\xi_d, \bar{e}_u, \bar{e}_w$, and \bar{e}_ω .*

△

This theorem is proved in Sections 4.3.5 and 4.4.4. Two theorems stated and proved in [26, 27] can be applied to our control design to ensure that it creates closed loop stable behavior implementations in sense of Lyapunov.

4.2.2 Behavior Switching AUV

In this section we reiterate what we explained in Chapter 3 regarding the switching among behaviors. The overall goal of Behavior Based control design is to maintain stability during the time each behavior is active, to prevent Zeno Phenomenon, and maintain stability

at switching times. The AUV control design follows the same approach as the control design for simple second-order system. Again, the CFBS approach allows us to ensure that the Lyapunov function, defined in terms of compensated tracking errors, of the overall switched system is nonincreasing at all times by appropriate choice of the command filter's initial conditions made by a Mission Planner. The following Theorem 4 holds for switching between any two AUV behaviors.

Theorem 4

B1. If, at each start of Behavior 1 ($t = t_{2j}$), the Mission Planner selects the initial values of the filtered command signals to be equal to the current state values, $n_c(t_{2j}) = n(t_{2j})$, $e_c(t_{2j}) = e(t_{2j})$, $d_c(t_{2j}) = d(t_{2j})$, $\Theta_c(t_{2j}) = \Theta(t_{2j})$, $u_c(t_{2j}) = u(t_{2j})$, $w_c(t_{2j}) = w(t_{2j})$, and $\omega_c(t_{2j}) = \omega(t_{2j})$, the signals $\xi_n(t_{2j}) = \xi_e(t_{2j}) = \xi_d(t_{2j}) = \xi_\Theta(t_{2j}) = \xi_u(t_{2j}) = \xi_w(t_{2j}) = \xi_\omega(t_{2j}) = 0$, and the integral error to maintain its value (i.e. $\bar{e}_u(t_{2j}^+) = \bar{e}_u(t_{2j})$), $\bar{e}_w(t_{2j}^+) = \bar{e}_w(t_{2j})$), and $\bar{e}_\omega(t_{2j}^+) = \bar{e}_\omega(t_{2j})$), then the Lyapunov function, V_{b_1} , at each start of each Behavior 1 is a nonincreasing sequence, $V_{b_1}(t_{2j+2}) \leq V_{b_1}(t_{2j})$.

B2. If, at each start of Behavior 2 ($t = t_{2j+1}$), the Mission Planner selects the initial values of the filtered command signal to be equal to the current state value, $d_c(t_{2j}) = d(t_{2j})$, $\Theta_c(t_{2j}) = \Theta(t_{2j})$, $u_c(t_{2j}) = u(t_{2j})$, $w_c(t_{2j}) = w(t_{2j})$, and $\omega_c(t_{2j}) = \omega(t_{2j})$, the signals $\xi_d(t_{2j}) = \xi_\Theta(t_{2j}) = \xi_u(t_{2j}) = \xi_w(t_{2j}) = \xi_\omega(t_{2j}) = 0$, and the integral error to maintain its value (i.e. $\bar{e}_u(t_{2j}^+) = \bar{e}_u(t_{2j})$), $\bar{e}_w(t_{2j}^+) = \bar{e}_w(t_{2j})$), and $\bar{e}_\omega(t_{2j}^+) = \bar{e}_\omega(t_{2j})$), then the Lyapunov function, V_{b_2} , at each start of each Behavior 2 is a nonincreasing sequence, $V_{b_2}(t_{2j+3}) \leq V_{b_2}(t_{2j+1})$.

△

The requirement is that:

$$V_i(x_i(t_{i,k})) \leq V_i(x_i(t_{i,k-1})),$$

where $i = 1, 2, \dots$ represents the active behavior and $k = 1, 2, \dots$ represent the specific time instances. The Figure shows that the value of the Lyapunov function V_2 corresponding to Behavior 2 at time $t_{2,2}$ (second time Behavior 2 is active) is less or equal to its value at time $t_{2,1}$ (first time Behavior 2 is active) which proves that Behavior 2 is stable in sense of Lyapunov. On the other hand, the Figure shows that the value of the Lyapunov function V_1 corresponding to Behavior 1 at time $t_{1,3}$ (third time Behavior 1 is active) is greater or equal to its value at time $t_{1,2}$ (second time Behavior 1 is active) and at time $t_{1,1}$ (first time Behavior 1 is active) which proves that Behavior 1 is unstable in sense of Lyapunov.

4.3 Control Signal Derivation: Behavior 1

This section derives the CFBS 3D trajectory tracking control law for Behavior 1. The same derivation applies to Behavior 3 where the $\phi_c^o(t)$ and $\theta_c^o(t)$ are generated such that the relative roll and pitch angles between the vehicle and surface is zero. In this section, we use $n_c, e_c, d_c, \Theta_c, u_c, w_c$, and ω_c , and their derivatives to derive a CFBS control law.

4.3.1 3D Trajectory Following

The objective of this section is to stabilize the dynamic system of eqns. (3.1) using CFBS. The inputs to this control loop are: $n_c(t), e_c(t), d_c(t)$, and their derivatives: $\dot{n}_c(t)$,

$\dot{e}_c(t), \dot{d}_c(t)$. The control of $[n, e, d]$ will be accomplished by specification of desired values for $[\psi, u, w]$.

Calculation of ψ_c^o

We rewrite the vehicle kinematics of eqn. (3.1) as

$$\mathbf{v} = \mathbf{R}_t^b \dot{\mathbf{p}},$$

where $\mathbf{v} = [u, v, w]$ and $\dot{\mathbf{p}} = [u_n, u_e, u_d]$, our goal is to compute the body frame velocity vector u_c^o, w_c^o , and yaw ψ_c^o . To follow the desired trajectory $[n_c(t), e_c(t), d_c(t)]$ the desired navigation frame velocity is

$$\left. \begin{aligned} u_{n_c}^o &= -K_{ne}\bar{n} + \dot{n}_c \\ u_{e_c}^o &= -K_{ne}\bar{e} + \dot{e}_c \\ u_{d_c}^o &= -K_d\bar{d} + \dot{d}_c. \end{aligned} \right\} \quad (4.17)$$

From the vehicle kinematics

$$\begin{bmatrix} u_c^o \\ v \\ w_c^o \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix} \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} c\psi_c^o & s\psi_c^o & 0 \\ -s\psi_c^o & c\psi_c^o & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{n_c} \\ u_{e_c} \\ u_{d_c} \end{bmatrix}.$$

Pre-multiply both sides with the inverses of the first and second matrices to get

$$\begin{bmatrix} c\theta & s\phi s\theta & c\phi s\theta \\ 0 & c\phi & -s\phi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \begin{bmatrix} u_c^o \\ v \\ w_c^o \end{bmatrix} = \begin{bmatrix} u_{n_c} & u_{e_c} \\ u_{e_c} & -u_{n_c} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} c\psi_c^o \\ s\psi_c^o \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ u_{d_c} \end{bmatrix}. \quad (4.18)$$

If we define, $\mathbf{v}_1 = \begin{bmatrix} c\theta \\ 0 \\ -s\theta \end{bmatrix}$, $\mathbf{v}_2 = \begin{bmatrix} c\phi s\theta \\ -s\phi \\ c\phi c\theta \end{bmatrix}$, $\mathbf{X} = \begin{bmatrix} u_{n_c} & u_{e_c} \\ u_{e_c} & -u_{n_c} \\ 0 & 0 \end{bmatrix}$, and $\mathbf{Y} = - \begin{bmatrix} s\phi s\theta \\ c\phi \\ s\phi c\theta \end{bmatrix}$ $v+$

$\begin{bmatrix} 0 \\ 0 \\ u_{d_c} \end{bmatrix}$, we can rearrange eqn. (4.18) to get

$$\mathbf{v}_1 u_c^o + \mathbf{v}_2 w_c^o = \mathbf{X} \begin{bmatrix} c\psi_c^o \\ s\psi_c^o \end{bmatrix} + \mathbf{Y}. \quad (4.19)$$

We define $\mathbf{v}_3 = \mathbf{v}_1 \times \mathbf{v}_2$. Vectors \mathbf{v}_1 , \mathbf{v}_2 , and \mathbf{v}_3 are an orthonormal basis for \mathfrak{R}^3 . In fact,

$\mathbf{v}_3 = \begin{bmatrix} s\psi s\theta \\ c\phi \\ s\phi c\theta \end{bmatrix}$. A solution $[\psi_c^o, u_c^o, w_c^o]$ exists if and only if $\mathbf{X} \begin{bmatrix} c\psi_c^o \\ s\psi_c^o \end{bmatrix} + \mathbf{Y}$ is in the range

space of \mathbf{v}_1 and \mathbf{v}_2 , therefore, we first attempt to select ψ_c^o such that

$$\mathbf{v}_3^\top (\mathbf{X} \begin{bmatrix} c\psi_c^o \\ s\psi_c^o \end{bmatrix} + \mathbf{Y}) = \mathbf{0}.$$

If no solution exists, then the command trajectory is not feasible. In this case, we minimize

$$\left| \mathbf{v}_3^\top (\mathbf{X} \begin{bmatrix} c\psi_c^o \\ s\psi_c^o \end{bmatrix} + \mathbf{Y}) \right| \text{ with respect to } \psi_c^o.$$

When it exists, the solution for ψ_c^o is:

$$\mathbf{v}_3^\top \mathbf{X} \begin{bmatrix} c\psi_c^o \\ s\psi_c^o \end{bmatrix} = -\mathbf{v}_3^\top \mathbf{Y} \quad (4.20)$$

If we define the components of the product $\mathbf{v}_3^\top \mathbf{X} = [a, b]$ and $-\mathbf{v}_3^\top \mathbf{Y} = c$, eqn.

(4.20) becomes

$$[a, b] \begin{bmatrix} c\psi_c^o \\ s\psi_c^o \end{bmatrix} = c.$$

From the definitions of \mathbf{v}_3 , \mathbf{X} , and \mathbf{Y} :

$$a = s\phi s\theta u_{n_c} + c\phi u_{e_c},$$

$$b = s\phi s\theta u_{e_c} - c\phi u_{n_c},$$

and

$$c = v - s\phi c\theta u_{d_c}.$$

The solution is possible only if and only if

$$|c| \leq \left\| \begin{bmatrix} a \\ b \end{bmatrix} \right\|.$$

From Figure 4.1 we see that

$$\beta = \arccos \frac{c}{\left\| \begin{bmatrix} a \\ b \end{bmatrix} \right\|}$$

and

$$\psi_c^o = \text{atan2}(b, a) \pm \beta. \quad (4.21)$$

Therefore, we see that when a solution does exist, then, in fact, two solutions are possible. For one the AUV has positive forward speed while for the other it has negative forward speed. When a solution exists and a direction of the motion is selected, it is straightforward to select ψ_c^o .

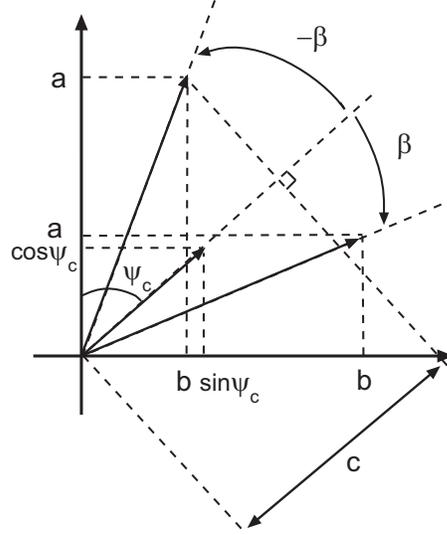


Figure 4.1: Yaw Command Selection

Calculation of u_c^o and w_c^o

Given ψ_c^o , from eqn. (4.21) we can solve eqn. (4.19) for

$$u_c^o = \mathbf{v}_1^\top \left(\mathbf{X} \begin{bmatrix} c\psi_c^o \\ s\psi_c^o \end{bmatrix} + \mathbf{Y} \right) \quad (4.22)$$

and

$$w_c^o = \mathbf{v}_2^\top \left(\mathbf{X} \begin{bmatrix} c\psi_c^o \\ s\psi_c^o \end{bmatrix} + \mathbf{Y} \right). \quad (4.23)$$

The objective of this section was to derive three control signals u_c^o , w_c^o , and ψ_c^o

such that the closed loop position dynamics are

$$\begin{bmatrix} \dot{n} \\ \dot{e} \\ \dot{d} \end{bmatrix} = - \begin{bmatrix} K_{ne}\bar{n} \\ K_{ne}\bar{e} \\ K_d\bar{d} \end{bmatrix} + \begin{bmatrix} \dot{n}_c \\ \dot{e}_c \\ \dot{d}_c \end{bmatrix}. \quad (4.24)$$

Our next step is to derive the backstepping terms, u_{bs} , w_{bs} , and ψ_{bs} , which are necessary so the closed loop position dynamics can be written in a stable form as in eqn. (4.24).

Notation Definition

For clarity, we rewrite the vehicle kinematics of eqn. (3.1)

$$\dot{\mathbf{p}} = \begin{bmatrix} \dot{n} \\ \dot{e} \\ \dot{d} \end{bmatrix} = \begin{bmatrix} u_n \\ u_e \\ u_d \end{bmatrix}$$

where

$$\left. \begin{aligned} u_n &= \mathbf{T}_n \mathbf{v} \\ u_e &= \mathbf{T}_e \mathbf{v} \\ u_d &= \mathbf{T}_d \mathbf{v}, \end{aligned} \right\} \quad (4.25)$$

where

$$\mathbf{T}_n(\phi, \theta, \psi) = \begin{bmatrix} c\theta c\psi & -c\phi s\psi + s\phi s\theta c\psi & s\phi s\psi + c\phi s\theta c\psi \end{bmatrix}, \quad (4.26)$$

$$\mathbf{T}_e(\phi, \theta, \psi) = \begin{bmatrix} c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & -s\phi c\psi + c\phi s\theta s\psi \end{bmatrix}, \quad (4.27)$$

$$\mathbf{T}_d(\phi, \theta, \psi) = \begin{bmatrix} -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}, \quad (4.28)$$

represent the velocity transformation functions from body to navigation frame. Similarly, we can define the transformation functions $\mathbf{T}_{n_c}^o(\phi_c^o, \theta_c^o, \psi_c^o)$, $\mathbf{T}_{e_c}^o(\phi_c^o, \theta_c^o, \psi_c^o)$, and $\mathbf{T}_{d_c}^o(\phi_c^o, \theta_c^o, \psi_c^o)$; and $\mathbf{T}_{n_c}(\phi_c, \theta_c, \psi_c)$, $\mathbf{T}_{e_c}(\phi_c, \theta_c, \psi_c)$, and $\mathbf{T}_{d_c}(\phi_c, \theta_c, \psi_c)$. Therefore, we can define the similar function as in eqn. (4.25) for the commanded variables:

$$\left. \begin{aligned} u_{n_c}^o &= \mathbf{T}_{n_c}^o \mathbf{v}_c^o \\ u_{e_c}^o &= \mathbf{T}_{e_c}^o \mathbf{v}_c^o \\ u_{d_c}^o &= \mathbf{T}_{d_c}^o \mathbf{v}_c^o, \end{aligned} \right\} \quad (4.29)$$

and command filtered variables:

$$\left. \begin{aligned} u_{n_c} &= \mathbf{T}_{n_c} \mathbf{v}_c \\ u_{e_c} &= \mathbf{T}_{e_c} \mathbf{v}_c \\ u_{d_c} &= \mathbf{T}_{d_c} \mathbf{v}_c, \end{aligned} \right\} \quad (4.30)$$

where $\mathbf{v}_c^o = [u_c^o, v, w_c^o]^\top$ and $\mathbf{v}_c = [u_c, v, w_c]^\top$. The error signals

$$\left. \begin{aligned} \bar{u}_n &= u_n - u_{n_c} \\ \bar{u}_e &= u_e - u_{e_c} \\ \bar{u}_d &= u_d - u_{d_c}, \end{aligned} \right\} \quad (4.31)$$

will be important in the subsequent analysis.

Control Design and Error Analysis

The dynamic equation for n, e and d can be written as

$$\begin{bmatrix} \dot{n} \\ \dot{e} \\ \dot{d} \end{bmatrix} = \begin{bmatrix} u_n \\ u_e \\ u_d \end{bmatrix} = \begin{bmatrix} u_{n_c}^o \\ u_{e_c}^o \\ u_{d_c}^o \end{bmatrix} + \begin{bmatrix} \bar{u}_n \\ \bar{u}_e \\ \bar{u}_d \end{bmatrix} + \begin{bmatrix} u_{n_c} - u_{n_c}^o \\ u_{e_c} - u_{e_c}^o \\ u_{d_c} - u_{d_c}^o \end{bmatrix}. \quad (4.32)$$

In the Appendix B, the \bar{u}_n , \bar{u}_e and \bar{u}_d terms on the right hand side of eqn. (4.32) are manipulated into the form

$$\begin{bmatrix} \bar{u}_n \\ \bar{u}_e \\ \bar{u}_d \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{C}\mathbf{g}(\bar{\psi}) \end{bmatrix} \begin{bmatrix} \bar{u} \\ \bar{w} \\ \bar{\psi} \end{bmatrix}. \quad (4.33)$$

Thus, the using eqn. (4.17) position error dynamics can be expressed as

$$\begin{bmatrix} \dot{\bar{n}} \\ \dot{\bar{e}} \\ \dot{\bar{d}} \end{bmatrix} = - \begin{bmatrix} K_{ne}\bar{n} \\ K_{ne}\bar{e} \\ K_d\bar{d} \end{bmatrix} + \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{C}\mathbf{g}(\bar{\psi}) \end{bmatrix} \begin{bmatrix} \bar{u} \\ \bar{w} \\ \bar{\psi} \end{bmatrix} + \begin{bmatrix} u_{n_c} - u_{n_c}^o \\ u_{e_c} - u_{e_c}^o \\ u_{d_c} - u_{d_c}^o \end{bmatrix}, \quad (4.34)$$

where the terms \mathbf{A} , \mathbf{B} , \mathbf{C} , and $\mathbf{g}(\bar{\psi})$ are derived in the Appendix B.

Command Filters

Define the compensated tracking error signals ν_n , ν_e and ν_d as

$$\begin{bmatrix} \nu_n \\ \nu_e \\ \nu_d \end{bmatrix} = \begin{bmatrix} \bar{n} - \xi_n \\ \bar{e} - \xi_e \\ \bar{d} - \xi_d \end{bmatrix}, \quad (4.35)$$

where ξ_n , ξ_e and ξ_d are defined as

$$\begin{aligned} \begin{bmatrix} \dot{\xi}_n \\ \dot{\xi}_e \\ \dot{\xi}_d \end{bmatrix} &= - \begin{bmatrix} K_{ne}\xi_n \\ K_{ne}\xi_e \\ K_d\xi_d \end{bmatrix} + \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{Cg}(\bar{\psi}) \end{bmatrix} \begin{bmatrix} \xi_u \\ \xi_w \\ \xi_\psi \end{bmatrix} \\ &+ \begin{bmatrix} u_{nc} - u_{nc}^o \\ u_{ec} - u_{ec}^o \\ u_{dc} - u_{dc}^o \end{bmatrix}, \end{aligned} \quad (4.36)$$

with $\xi_n(0) = 0$, $\xi_e(0) = 0$, and $\xi_d(0) = 0$. With these definitions, the dynamics of the compensated tracking errors are

$$\begin{bmatrix} \dot{\nu}_n \\ \dot{\nu}_e \\ \dot{\nu}_d \end{bmatrix} = -K_{ned} \begin{bmatrix} \nu_n \\ \nu_e \\ \nu_d \end{bmatrix} + \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{Cg}(\bar{\psi}) \end{bmatrix} \begin{bmatrix} \nu_u \\ \nu_w \\ \nu_\psi \end{bmatrix} \quad (4.37)$$

where ν_u , ν_w and ν_ψ are defined as

$$\begin{bmatrix} \nu_u \\ \nu_w \\ \nu_\psi \end{bmatrix} = \begin{bmatrix} \bar{u} - \xi_u \\ \bar{w} - \xi_w \\ \bar{\psi} - \xi_\psi \end{bmatrix}. \quad (4.38)$$

Lyapunov Analysis

Consider the candidate Lyapunov function

$$V_{ned} = \frac{1}{2} (\nu_n^2 + \nu_e^2 + \nu_d^2). \quad (4.39)$$

The derivative of V_{ned} is

$$\begin{aligned}
\dot{V}_{ned} &= -K_{ne}(\nu_e^2 + \nu_e^2) - K_d\nu_d^2 + \nu_n\bar{u}_n + \nu_e\bar{u}_e + \nu_d\bar{u}_d \\
&= -K_{ne}(\nu_n^2 + \nu_e^2) - K_d\nu_d^2 \\
&\quad + \begin{bmatrix} \nu_n & \nu_e & \nu_d \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{C}\mathbf{g}(\bar{\psi}) \end{bmatrix} \begin{bmatrix} v_u \\ v_w \\ v_\psi \end{bmatrix} \\
&= -K_{ne}(\nu_n^2 + \nu_e^2) - K_d\nu_d^2 + \\
&\quad \mathbf{A}^T \begin{bmatrix} \nu_n \\ \nu_e \\ \nu_d \end{bmatrix} v_u + \mathbf{B}^T \begin{bmatrix} \nu_n \\ \nu_e \\ \nu_d \end{bmatrix} v_w + \mathbf{g}(\bar{\psi})^T \mathbf{C}^T \begin{bmatrix} \nu_n \\ \nu_e \\ \nu_d \end{bmatrix} v_\psi. \tag{4.40}
\end{aligned}$$

Eqn. (4.40) will be used in the stability analysis of Section 4.3.5.

4.3.2 Attitude Control

The objective of Section 4.3 is to stabilize the dynamic system of eqns. (3.1–3.4) using CFBS. Because the position dynamics were already discussed in Section 4.3.1, this section focuses on selection of ω_c^o to stabilize the Θ dynamics. The inputs to this control loop are $\phi_c(t)$, $\theta_c(t)$, $\dot{\phi}_c(t)$, $\dot{\theta}_c(t)$ from the Mission Planner, and $\psi_c^o(t)$ defined by the trajectory tracking as in eqn. (4.21), which is command filtered to generate $\psi_c(t)$ and $\dot{\psi}_c(t)$.

For attitude control, based on eqn. (3.2), we define the signal

$$\omega_c^o = \Omega^{-1} \left(-\mathbf{K}_\Theta \bar{\Theta} + \dot{\Theta}_c - \Theta_{bs} \right), \tag{4.41}$$

where $\mathbf{K}_\Theta = \begin{bmatrix} K_\phi & 0 & 0 \\ 0 & K_\theta & 0 \\ 0 & 0 & K_\psi \end{bmatrix}$ is a positive definite matrix. The backstepping term, Θ_{bs} ,

will be defined in the stability analysis of Section 4.3.5. The nonlinear matrix function Ω

is defined in eqn. (3.8). Note that Ω^{-1} exists for all values of ϕ , θ , and ψ except $\theta = \frac{\pi}{2}$.

Using this definition, the closed-loop tracking error corresponding to eqn. (3.2) is

$$\dot{\Theta} = \Omega \omega_c^o + \Omega(\omega - \omega_c) + \Omega(\omega_c - \omega_c^o) \quad (4.42)$$

$$= -\mathbf{K}_\Theta \bar{\Theta} + \dot{\Theta}_c - \Theta_{bs} + \Omega \bar{\omega} + \Omega(\omega_c - \omega_c^o)$$

$$\dot{\bar{\Theta}} = -\mathbf{K}_\Theta \bar{\Theta} - \Theta_{bs} + \Omega \bar{\omega} + \Omega(\omega_c - \omega_c^o). \quad (4.43)$$

The compensated tracking error signal for the Θ dynamics is defined as

$$\nu_\Theta = \bar{\Theta} - \xi_\Theta. \quad (4.44)$$

The signal ξ_Θ is defined as

$$\dot{\xi}_\Theta = -\mathbf{K}_\Theta \xi_\Theta + \Omega(\omega_c - \omega_c^o) + \Omega \xi_\omega \quad (4.45)$$

with $\xi_\Theta(0) = \mathbf{0}$. With these definitions, the dynamic equation of ν_Θ is

$$\begin{aligned} \dot{\nu}_\Theta &= \dot{\bar{\Theta}} - \dot{\xi}_\Theta \\ &= -\mathbf{K}_\Theta \nu_\Theta - \Theta_{bs} + \Omega \nu_\omega, \end{aligned} \quad (4.46)$$

where ν_ω is defined in Section 4.1.1 with other compensated error variables.

Choosing the candidate Lyapunov function as

$$V_\Theta = \frac{1}{2} \nu_\Theta^\top \nu_\Theta, \quad (4.47)$$

its derivative is

$$\dot{V}_\Theta = -\boldsymbol{\nu}_\Theta^\top \mathbf{K}_\Theta \boldsymbol{\nu}_\Theta + \boldsymbol{\nu}_\Theta^\top \boldsymbol{\Omega} \boldsymbol{\nu}_\omega - \boldsymbol{\Theta}_{bs} \boldsymbol{\nu}_\Theta. \quad (4.48)$$

Equation (4.48) will be used in the stability analysis of Section 4.3.5.

4.3.3 Speed and Angular Rate Control

The objective of this section is to stabilize the dynamic system of eqns. (3.1–3.4) using backstepping. This section focuses on selection of \mathbf{F} and $\boldsymbol{\tau}$ to stabilize the v and ω tracking error dynamics. For \mathbf{v} and $\boldsymbol{\omega}$ tracking control, the inputs are the horizontal and vertical speed and angular rate commands $\mathbf{v}_c(t) = [u_c(t), w_c(t)]^\top$ and $\boldsymbol{\omega}_c(t) = [p_c(t), q_c(t), r_c(t)]^\top$ and their derivatives $\dot{\mathbf{v}}_c(t) = [\dot{u}_c(t), \dot{w}_c(t)]^\top$ and $\dot{\boldsymbol{\omega}}_c(t) = [\dot{p}_c(t), \dot{q}_c(t), \dot{r}_c(t)]^\top$. The ideal desired value for \mathbf{v} , denoted as $\mathbf{v}_c^o = [u_c^o, w_c^o]^\top$, is computed by eqns. (4.22) and (4.23), while the ideal desired value for $\boldsymbol{\omega}$, denoted as $\boldsymbol{\omega}_c^o = [p_c^o, q_c^o, r_c^o]^\top$, is computed by eqn. (4.41).

For tracking control using eqns. (3.3-3.4) we select the control force and the control torque as

$$\mathbf{F} = \mathbf{M} (\mathbf{F}_n - \mathbf{K}_v \bar{\mathbf{v}} - \mathbf{K}_v^i \mathbf{e}_v + \dot{\mathbf{v}}_c^o - \mathbf{v}_{bs}) \quad (4.49)$$

$$\boldsymbol{\tau} = \mathbf{J} (\boldsymbol{\tau}_n - \mathbf{K}_\omega \bar{\boldsymbol{\omega}} - \mathbf{K}_\omega^i \mathbf{e}_\omega + \dot{\boldsymbol{\omega}}_c^o - \boldsymbol{\omega}_{bs}), \quad (4.50)$$

with the thrust vector, \mathbf{F} and $\boldsymbol{\tau}$ defined as in Section 4.3.4.

The backstepping terms, $\mathbf{v}_{bs} = [u_{bs}, w_{bs}]^\top$ and $\boldsymbol{\omega}_{bs} = [p_{bs}, q_{bs}, r_{bs}]^\top$, will be defined in the stability analysis of Section 4.3.5. With this choice of the control signal and because $\dot{e}_u = \bar{u}$, $\dot{e}_w = \bar{w}$, $\dot{\mathbf{e}}_\omega = \bar{\boldsymbol{\omega}}$, we obtain $\ddot{e}_u = \dot{\bar{u}}$, $\ddot{e}_w = \dot{\bar{w}}$, $\ddot{\mathbf{e}}_\omega = \dot{\bar{\boldsymbol{\omega}}}$, and the dynamics of the u , w

and ω tracking errors

$$\left. \begin{aligned} \dot{\bar{u}} &= -K_u \bar{u} - K_u^i e_u - u_{bs} \\ \dot{\bar{w}} &= -K_w \bar{w} - K_w^i e_w - w_{bs} \\ \dot{\bar{\omega}} &= -\mathbf{K}_\omega \bar{\omega} - \mathbf{K}_\omega^i \mathbf{e}_\omega - \omega_{bs}, \end{aligned} \right\} \quad (4.51)$$

can be rewritten as

$$\left. \begin{aligned} \ddot{e}_u + K_u \dot{e}_u + K_u^i e_u &= -u_{bs} \\ \ddot{e}_w + K_w \dot{e}_w + K_w^i e_w &= -w_{bs} \\ \ddot{\mathbf{e}}_\omega + \mathbf{K}_\omega \dot{\mathbf{e}}_\omega + \mathbf{K}_\omega^i \mathbf{e}_\omega &= -\omega_{bs}. \end{aligned} \right\} \quad (4.52)$$

We can then define the tracking error vectors $\mathbf{q}_u = [e_u, \bar{u}]^\top$, $\mathbf{q}_w = [e_w, \bar{w}]^\top$ and $\mathbf{q}_\omega = [\mathbf{q}_p, \mathbf{q}_q, \mathbf{q}_r]^\top = [\mathbf{e}_\omega, \bar{\omega}]^\top = [e_p, \bar{p}, e_q, \bar{q}, e_r, \bar{r}]^\top$ and the tracking error vectors dynamics as

$$\dot{\mathbf{q}}_u = \mathbf{D}_u \mathbf{q}_u - \mathbf{E} u_{bs} \quad (4.53)$$

$$\dot{\mathbf{q}}_w = \mathbf{D}_w \mathbf{q}_w - \mathbf{E} w_{bs} \quad (4.54)$$

$$\dot{\mathbf{q}}_\omega = \begin{bmatrix} \mathbf{D}_p & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & \mathbf{D}_q & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} & \mathbf{D}_r \end{bmatrix} \mathbf{q}_\omega - \mathbf{F} \omega_{bs} \quad (4.55)$$

$$\text{where } \mathbf{D}_u = \begin{bmatrix} 0 & 1 \\ -K_u^i & -K_u \end{bmatrix}, \mathbf{D}_w = \begin{bmatrix} 0 & 1 \\ -K_w^i & -K_w \end{bmatrix}, \mathbf{D}_p = \begin{bmatrix} 0 & 1 \\ -K_p^i & -K_p \end{bmatrix}, \mathbf{D}_q =$$

$$\begin{bmatrix} 0 & 1 \\ -K_q^i & -K_q \end{bmatrix}, \mathbf{D}_r = \begin{bmatrix} 0 & 1 \\ -K_r^i & -K_r \end{bmatrix}, \mathbf{E} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \text{ and } \mathbf{F} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Choosing the candidate Lyapunov function as

$$V_i = \frac{1}{2}(\mathbf{q}_u^\top \mathbf{P}_u \mathbf{q}_u + \mathbf{q}_w^\top \mathbf{P}_w \mathbf{q}_w + \mathbf{q}_\omega^\top \mathbf{P}_\omega \mathbf{q}_\omega), \quad (4.56)$$

where \mathbf{P}_u , \mathbf{P}_w and \mathbf{P}_ω are positive definite diagonal matrices defined in Appendix C. \mathbf{P}_ω

is a 6×6 block-symmetric matrix consisting of three matrices, \mathbf{P}_p , \mathbf{P}_q , and \mathbf{P}_r . The

Appendix C also analyzes the terms $\mathbf{Q}_u = \begin{bmatrix} 0 & 0 \\ 0 & 2K_u p_{2u} \end{bmatrix}$, $\mathbf{Q}_w = \begin{bmatrix} 0 & 0 \\ 0 & 2K_w p_{2w} \end{bmatrix}$, $\mathbf{Q}_p =$

$\begin{bmatrix} 0 & 0 \\ 0 & 2K_p p_{2p} \end{bmatrix}$, $\mathbf{Q}_q = \begin{bmatrix} 0 & 0 \\ 0 & 2K_q p_{2q} \end{bmatrix}$, $\mathbf{Q}_r = \begin{bmatrix} 0 & 0 \\ 0 & 2K_r p_{2r} \end{bmatrix}$, $\mathbf{P}_u \mathbf{E} = \begin{bmatrix} 0 \\ p_{2u} \end{bmatrix}$, $\mathbf{P}_w \mathbf{E} =$

$\begin{bmatrix} 0 \\ p_{2w} \end{bmatrix}$, and $\mathbf{P}_\omega \mathbf{F} = \begin{bmatrix} 0 \\ p_{2p} \\ 0 \\ p_{2q} \\ 0 \\ p_{2r} \end{bmatrix}$, which are used in the following derivation.

Then the Lyapunov function derivative is

$$\begin{aligned}
\dot{V}_i &= \frac{1}{2}(\dot{\mathbf{q}}_u^\top \mathbf{P}_u \mathbf{q}_u + \mathbf{q}_u^\top \mathbf{P}_u \dot{\mathbf{q}}_u + \dot{\mathbf{q}}_w^\top \mathbf{P}_w \mathbf{q}_w + \mathbf{q}_w^\top \mathbf{P}_w \dot{\mathbf{q}}_w \\
&\quad \dot{\mathbf{q}}_\omega^\top \mathbf{P}_\omega \mathbf{q}_\omega + \mathbf{q}_\omega^\top \mathbf{P}_\omega \dot{\mathbf{q}}_\omega) \\
&= \frac{1}{2}[\mathbf{q}_u^\top (\mathbf{D}_u^\top \mathbf{P}_u + \mathbf{P}_u \mathbf{D}_u) \mathbf{q}_u + \mathbf{q}_w^\top (\mathbf{D}_w^\top \mathbf{P}_w + \mathbf{P}_w \mathbf{D}_w) \mathbf{q}_w \\
&\quad + \mathbf{q}_\omega^\top (\mathbf{D}_\omega^\top \mathbf{P}_\omega + \mathbf{P}_\omega \mathbf{D}_\omega) \mathbf{q}_\omega] \\
&\quad - \mathbf{q}_u^\top \mathbf{P}_u \mathbf{E} u_{bs} - \mathbf{q}_w^\top \mathbf{P}_w \mathbf{E} w_{bs} - \mathbf{q}_\omega^\top \mathbf{P}_\omega \mathbf{F} \omega_{bs} \\
&= -K_u p_{2u} \bar{u}^2 - K_w p_{2w} \bar{w}^2 - \bar{\omega}^\top \mathbf{K}_\omega p_{2\omega} \bar{\omega} \\
&\quad - p_{2u} \bar{u} u_{bs} - p_{2w} \bar{w} w_{bs} - \bar{\omega}^\top p_{2\omega} \omega_{bs} \\
&= -K_u p_{2u} \nu_u^2 - K_w p_{2w} \nu_w^2 - \nu_\omega^\top \mathbf{K}_\omega p_{2\omega} \nu_\omega \\
&\quad - p_{2u} \nu_u u_{bs} - p_{2w} \nu_w w_{bs} - \nu_\omega^\top p_{2\omega} \omega_{bs}, \tag{4.57}
\end{aligned}$$

where we used $\mathbf{D}_\omega = \begin{bmatrix} \mathbf{D}_p & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & \mathbf{D}_q & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} & \mathbf{D}_r \end{bmatrix}$ and $\mathbf{K}_\omega = \begin{bmatrix} K_p & 0 & 0 \\ 0 & K_q & 0 \\ 0 & 0 & K_r \end{bmatrix}$. Equation (4.57)

will be used in the stability analysis of Section 4.3.5.

4.3.4 Thruster Command Selection

As introduced in Section 3.1 the vector $\mathbf{T} = [T_0, \dots, T_4]^\top$ of five thrusts is related to the the control forces and moments by a known thrust distribution matrix such that $\mathbf{F} = \mathbf{L}_f \mathbf{T}$ and $\boldsymbol{\tau} = \mathbf{L}_m \mathbf{T}$ where $\mathbf{L}_f \in \mathfrak{R}^{2 \times 5}$ and $\mathbf{L}_m \in \mathfrak{R}^{3 \times 5}$.

Combining eqns. (3.5), (3.6), (4.49) and (4.50), we have five constraint equations

for the five thrust commands. Therefore, the equation

$$\begin{bmatrix} \mathbf{L}_f \\ \mathbf{L}_m \end{bmatrix} \mathbf{T} = \begin{bmatrix} \mathbf{M} (\mathbf{F}_n - \mathbf{K}_v \bar{\mathbf{v}} - \mathbf{K}_v^i \mathbf{e}_v + \dot{\mathbf{v}}_c^o - \mathbf{v}_{bs}) \\ \mathbf{J} (\boldsymbol{\tau}_n - \mathbf{K}_\omega \bar{\boldsymbol{\omega}} - \mathbf{K}_\omega^i \mathbf{e}_\omega + \dot{\boldsymbol{\omega}}_c^o - \boldsymbol{\omega}_{bs}) \end{bmatrix} \quad (4.58)$$

must be solved for \mathbf{T} . Matrices \mathbf{M} , \mathbf{J} , \mathbf{L}_f , and \mathbf{L}_m are defined in eqns. (3.9), (3.10), (3.11),

and (3.12), respectively. Since the thrusters are placed such that the matrix $\mathbf{B} = \begin{bmatrix} \mathbf{L}_f \\ \mathbf{L}_m \end{bmatrix}$

is invertible, a unique solution exists for \mathbf{T} .

4.3.5 Stability Analysis of Behaviors 1

Given Assumption 1 related to the reference trajectory, this section proves that the CFBS control law described by eqns. (4.1–4.6) yields exponential stability of the compensated tracking errors defined in Section 4.1.1 for the system defined by eqns. (3.1–3.4). In Sections 4.3.1, 4.3.2, and 4.3.3, we derived the position, attitude, and speed and angular rate controllers, respectively, which are used to implement Behavior 1. In this section we will use the definitions of candidate Lyapunov functions and their derivatives written in eqns. (4.39), (4.47), (4.56), (4.40), (4.48), and (4.57).

Proof: Combining eqns. (4.39), (4.47), and (4.56) we define the overall candidate Lyapunov function for Behavior 1 as

$$V_{b_1} = V_{ned} + V_\Theta + V_i.$$

Combining eqns. (4.40), (4.48), and (4.57) the time derivative of V_{b_1} is

$$\begin{aligned}
\dot{V}_{b_1} = & -K_{ne}(\nu_n^2 + \nu_e^2) - K_d\nu_d^2 + \mathbf{A}^T \begin{bmatrix} \nu_n \\ \nu_e \\ \nu_d \end{bmatrix} \nu_u + \mathbf{B}^T \begin{bmatrix} \nu_n \\ \nu_e \\ \nu_d \end{bmatrix} \nu_w + \mathbf{g}(\bar{\psi})^T \mathbf{C}^T \begin{bmatrix} \nu_n \\ \nu_e \\ \nu_d \end{bmatrix} \nu_\psi \\
& - \nu_\Theta^T \mathbf{K}_\Theta \nu_\Theta + \nu_\Theta \boldsymbol{\Omega} \nu_\omega - \psi_{bs} \nu_\psi \\
& - K_u p_{2u} \nu_u^2 - K_w p_{2w} \nu_w^2 - \nu_\omega^T \mathbf{K}_\omega p_{2\omega} \nu_\omega - p_{2u} \nu_u u_{bs} - p_{2w} \nu_w w_{bs} - \nu_\omega^T p_{2\omega} \boldsymbol{\omega}_{bs}.
\end{aligned} \tag{4.59}$$

To remove the sign indefinite terms of the above result, we define the backstepping terms

as

$$\psi_{bs} = \mathbf{g}(\bar{\psi})^T \mathbf{C}^T \begin{bmatrix} \nu_n \\ \nu_e \\ \nu_d \end{bmatrix} \tag{4.60}$$

$$u_{bs} = \frac{\mathbf{A}^T \begin{bmatrix} \nu_n \\ \nu_e \\ \nu_d \end{bmatrix}}{p_{2u}} \tag{4.61}$$

$$w_{bs} = \frac{\mathbf{B}^T \begin{bmatrix} \nu_n \\ \nu_e \\ \nu_d \end{bmatrix}}{p_{2w}} \tag{4.62}$$

$$\boldsymbol{\omega}_{bs} = \frac{\boldsymbol{\Omega}^T \nu_\Theta}{p_{2\omega}}. \tag{4.63}$$

With these definitions, the derivative of $V_{b_1}(t)$ satisfies

$$\dot{V}_{b_1} \leq -K_{ne}(\nu_n^2 + \nu_e^2) - K_d\nu_d^2 - \boldsymbol{\nu}_\Theta^\top \mathbf{K}_\Theta \boldsymbol{\nu}_\Theta - K_u p_{2u} \nu_u^2 - K_w p_{2w} \nu_w^2 - \boldsymbol{\nu}_\omega^\top \mathbf{K}_\omega p_{2\omega} \boldsymbol{\nu}_\omega. \quad (4.64)$$

Since the error state is $\mathbf{e} = [\nu_n, \nu_e, \nu_d, \boldsymbol{\nu}_\Theta, \nu_u, e_u, \nu_w, e_w, \boldsymbol{\nu}_\omega, \mathbf{e}_\omega]$, the derivative of $V_{b_1}(t)$ is negative semidefinite. This fact proves that the error state is stable and, in particular, e_u, e_w , and \mathbf{e}_ω are bounded for all $t \geq 0$. LaSalle's invariance theorem (page 128 in [41]), proves that the error state subvector $\mathbf{e}_\nu = [\nu_n, \nu_e, \nu_d, \boldsymbol{\nu}_\Theta, \nu_u, \nu_w, \boldsymbol{\nu}_\omega]$ converges to zero asymptotically. As explained in Remark 2 in Section 2.3, using LaSalle's invariance theorem, eqn. (4.51), as written, would result in e_u, e_w and \mathbf{e}_ω converging to zero asymptotically. In reality, the nonlinear forces and moments acting on the vehicle will be distinct from their models \mathbf{F}_n and $\boldsymbol{\tau}_n$, and e_u, e_w and \mathbf{e}_ω would converge to the nonzero values necessary to compensate the model error.

In this section we proved the stability of Behavior 1. Behavior 3 differs from Behavior 1 only in the fact of how the inputs to the controllers are generated. The attitude commands in Behavior 3 are generated externally with the goal that AUV's attitude matches the attitude of the surface which is followed. Therefore, these two behaviors, B1 and B3, only differ at the mission planning level, external to the controller and the stability analysis for B1 holds for B3.

4.4 Control Signal Derivation: Behavior 2

This section derives the CFBS 3D trajectory tracking control law for Behavior 2. The inputs to this control loop are: $d_c(t), \phi_c(t), \theta_c(t), \psi_c(t), u_c(t)$, and their derivatives:

$\dot{d}_c(t)$, $\dot{\phi}_c(t)$, $\dot{\theta}_c(t)$, $\dot{\psi}_c(t)$, $\dot{u}_c(t)$. These signals are generated by the command filtering of the ideal desired values, $d_c^o(t)$, $\phi_c^o(t)$, $\theta_c^o(t)$, $\psi_c^o(t)$, $u_c^o(t)$, which come from the Mission Planner or the operator.

4.4.1 Depth Controller

The objective of Section 4.4.1 is to stabilize the dynamic system described by eqns. (3.2–3.4) and depth kinematics defined as in eqn. (4.65) using CFBS. This section focuses on selection of $d_c(t)$ and $\dot{d}_c(t)$ to stabilize the depth dynamics. For $d(t)$ tracking control, the input is the depth command $d_c(t)$ and its derivative $\dot{d}_c(t)$.

The kinematics of d are

$$\dot{d} = -s\theta u + c\theta s\phi v + c\theta c\phi w. \quad (4.65)$$

Since desired values for u , ϕ , and θ are already specified and v is not controllable, assuming that $\theta \neq 90^\circ$ and $\phi \neq 90^\circ$, for depth control, based on eqn. (4.65), we define the signal

$$w_c^o = \frac{s\theta u - c\theta s\phi v - K_d \bar{d} + \dot{d}_c^o}{c\theta c\phi} \quad (4.66)$$

for w to control d . This yields the closed loop depth error dynamic equation

$$\begin{aligned} \dot{d} &= -s\theta u + c\theta s\phi v + c\theta c\phi w_c^o + c\theta c\phi(w - w_c^o) \\ &= -K_d \bar{d} + \dot{d}_c + c\theta c\phi \bar{w} + c\theta c\phi(w - w_c^o) \\ \dot{\bar{d}} &= -K_d \bar{d} + c\theta c\phi \bar{w} + c\theta c\phi(w - w_c^o). \end{aligned} \quad (4.67)$$

The compensated tracking error signal for the d dynamics is defined as

$$\nu_d = \bar{d} - \xi_d. \quad (4.68)$$

The signal ξ_d is defined as

$$\dot{\xi}_d = -K_d \xi_d + c\theta c\phi \xi_w + c\theta c\phi(w - w_c^o) \quad (4.69)$$

with $\xi_d(0) = 0$. With these definitions, the dynamic equation of ν_d is

$$\begin{aligned} \dot{\nu}_d &= \dot{d} - \dot{\xi}_d \\ &= -K_d \nu_d + c\theta c\phi \nu_w, \end{aligned} \quad (4.70)$$

where ν_w is defined in Section 4.1.1 with other compensated error variables.

Choosing the candidate Lyapunov function as

$$V_d = \frac{1}{2} \nu_d^2, \quad (4.71)$$

its derivative is

$$\dot{V}_d = -K_d \nu_d^2 + c\theta c\phi \nu_d \nu_w. \quad (4.72)$$

Equation (4.72) will be used in the stability analysis of Section 4.4.4.

4.4.2 Attitude Controller

This controller is very similar as for Behaviors 1, 3, and 4. The difference is that all three attitude commands: ϕ_c^o , θ_c^o , ψ_c^o commands are externally generated by the Mission Planner or operator. That means that Θ_{bs} can be selected to be identically zero. The objective of this section is to stabilize the dynamic system of eqns. (3.1–3.4) using backstepping. Because the position dynamics were already discussed, this section focuses on selection of ω_c^o to stabilize the Θ dynamics. For Θ tracking control, the inputs are roll,

pitch, and yaw commands, $\Theta_c = [\phi_c(t), \theta_c(t), \psi_c(t)]$ and the derivatives of these signals, which are produced by a command filter with input Θ_c^o as discussed in Appendix A.

For attitude control, based on eqn. (3.2), we define the signal

$$\omega_c^o = \Omega^{-1} \left(-\mathbf{K}_\Theta \bar{\Theta} + \dot{\Theta}_c \right), \quad (4.73)$$

where $\mathbf{K}_\Theta = \begin{bmatrix} K_\phi & 0 & 0 \\ 0 & K_\theta & 0 \\ 0 & 0 & K_\psi \end{bmatrix}$ is a positive definite matrix. Using this definition, the closed-loop tracking error corresponding to eqn. (3.2) is

$$\dot{\Theta} = \Omega \omega_c^o + \Omega(\omega - \omega_c) + \Omega(\omega_c - \omega_c^o) \quad (4.74)$$

$$= -\mathbf{K}_\Theta \bar{\Theta} + \dot{\Theta}_c + \Omega \bar{\omega} + \Omega(\omega_c - \omega_c^o)$$

$$\dot{\Theta} = -\mathbf{K}_\Theta \bar{\Theta} + \Omega \bar{\omega} + \Omega(\omega_c - \omega_c^o). \quad (4.75)$$

The compensated tracking error signal for the Θ dynamics is defined as

$$\nu_\Theta = \bar{\Theta} - \xi_\Theta. \quad (4.76)$$

The signal ξ_Θ is defined as

$$\dot{\xi}_\Theta = -\mathbf{K}_\Theta \xi_\Theta + \Omega \xi_\omega + \Omega(\omega_c - \omega_c^o) \quad (4.77)$$

with $\xi_\Theta(0) = \mathbf{0}$. With these definitions, the dynamic equation of ν_Θ is

$$\begin{aligned} \dot{\nu}_\Theta &= \dot{\bar{\Theta}} - \dot{\xi}_\Theta \\ &= -\mathbf{K}_\Theta \nu_\Theta + \Omega \nu_\omega, \end{aligned} \quad (4.78)$$

where ν_ω is defined in Section 4.1.1 with other compensated error variables.

Choosing the candidate Lyapunov function as

$$V_{\Theta} = \frac{1}{2} \boldsymbol{\nu}_{\Theta}^{\top} \boldsymbol{\nu}_{\Theta}, \quad (4.79)$$

its derivative is

$$\dot{V}_{\Theta} = -\boldsymbol{\nu}_{\Theta}^{\top} \mathbf{K}_{\Theta} \boldsymbol{\nu}_{\Theta} + \boldsymbol{\nu}_{\Theta}^{\top} \boldsymbol{\Omega} \boldsymbol{\nu}_{\omega}. \quad (4.80)$$

Equation (4.80) will be used in the stability analysis of Section 4.4.4.

4.4.3 Speed and Angular Rate CFBS

This controller is the same as for Behaviors 1, 3, and 4 except for the controller inputs. The vertical velocity command, w_c^o , is generated by the depth controller with the eqn. (4.66) and horizontal velocity, u_c^o , and yaw, ψ_c^o , commands are externally generated not calculated by the 3D trajectory tracking controller as in Behavior 1. Since u_c^o is externally generated $u_{bs} = 0$.

4.4.4 Stability Analysis of Behavior 2

This section proves that the CFBS control law described by eqns. (4.12–4.15) yields exponential stability of the compensated tracking errors defined in Section 4.1.2 for the system defined by eqns. (4.65) and (3.2–3.4). In Sections 4.4.1, 4.4.2, and 4.4.3, we derived the depth, attitude, and speed and angular rate controllers, respectively, which are used to implement Behavior 2. In this section we will use the definitions of candidate Lyapunov functions and their derivatives written in eqns. (4.71), (4.79), (4.56), (4.72), (4.80), and (4.57).

Proof: Combining eqns. (4.71), (4.79), and (4.56) we define the overall candidate Lyapunov function for Behavior 2 as

$$V_{b_2} = V_d + V_\Theta + V_i.$$

Combining eqns. (4.72), (4.80), and (4.57) the time derivative of V_{b_2} is

$$\begin{aligned} \dot{V}_{b_2} = & -K_d \nu_d^2 + c\theta c\phi \nu_d \nu_w - \boldsymbol{\nu}_\Theta^\top \mathbf{K}_\Theta \boldsymbol{\nu}_\Theta + \boldsymbol{\nu}_\Theta \boldsymbol{\Omega} \boldsymbol{\nu}_\omega \\ & - K_u p_{2u} \nu_u^2 - K_w p_{2w} \nu_w^2 - \boldsymbol{\nu}_\omega^\top \mathbf{K}_\omega p_{2\omega} \boldsymbol{\nu}_\omega - p_{2w} \nu_w w_{bs} - \boldsymbol{\nu}_\omega^\top p_{2\omega} \boldsymbol{\omega}_{bs}. \end{aligned} \quad (4.81)$$

To remove the sign indefinite terms of the above result, we define the backstepping terms as

$$w_{bs} = \frac{c\theta c\phi \nu_d}{p_{2w}} \quad (4.82)$$

$$\boldsymbol{\omega}_{bs} = \frac{\boldsymbol{\Omega}^\top \boldsymbol{\nu}_\Theta}{p_{2\omega}}. \quad (4.83)$$

With these definitions, the derivative of $V_{b_2}(t)$ satisfies

$$\dot{V}_{b_2} \leq -K_d \nu_d^2 - \boldsymbol{\nu}_\Theta^\top \mathbf{K}_\Theta \boldsymbol{\nu}_\Theta - K_u p_{2u} \nu_u^2 - K_w p_{2w} \nu_w^2 - \boldsymbol{\nu}_\omega^\top \mathbf{K}_\omega p_{2\omega} \boldsymbol{\nu}_\omega. \quad (4.84)$$

Chapter 5

Results

The purpose of this section is to illustrate the performance of our behavior based method in simulation and in-water using the AUV. We simulated and tested all three behaviors and switching among them.

Behavior 1 has the goal to cause vehicle's position in 3D, $\mathbf{p} = [n, e, d]$, to track the position command signals defined by the Mission Planner, $[n_c^o, e_c^o, d_c^o]$. These signals are first filtered to generate command filtered signals, $[n_c, e_c, d_c]$, as explained in Appendix A. In Behavior B1, control of the position state generates the yaw command, ψ_c^o , by eqn. (4.21) and a command for the velocity state $\mathbf{v} = [u_c^o, w_c^o]$, by eqns. (4.22) and (4.23), while the velocity control generates the thrust signals F to achieve the u_c^o and w_c^o commands according to eqn. (4.49). At the same time B1 regulates the attitude. The yaw and attitude controller generates a command for the angular rate state, $\boldsymbol{\omega} = [p_c^o, q_c^o, r_c^o]$, by eqn. (4.41). The angular rate control generates control signal, τ , to achieve the $[p_c^o, q_c^o, r_c^o]$ commands according to eqn. (4.50).

Behavior B2 does not control the 3D position. It has the goal is to cause d to track a signal defined by the Mission Planner to be d_c^o . The depth controller uses the filtered depth command d_c and its derivative \dot{d}_c to generate the vertical velocity command w_c^o by eqn. (4.66). The velocity control generates control signal, F , to achieve the u_c^o and w_c^o commands according to eqns. (4.49). The signal u_c^o is defined by the Mission Planner. The attitude and angular rate controllers are the same as in B1 except that ψ_c^o is also defined by the Mission Planner.

Behavior 3 has the goal to track vehicle's desired position as in B1 and at the same time track specific altitude, roll and pitch, commands such that the vehicle conforms to the surface above or below it. In this mode, the altimeter is measuring altitude: a , relative roll: ϕ_r , and relative pitch: θ_r . The mission planner is specifying a_c , ϕ_{r_c} , and θ_{r_c} where the last two mean commanded relative angles. At present these were always zero (i.e., AUV attitude matches the hull attitude). The controller constructs roll, ϕ_c^o , and pitch, θ_c^o , commands.

During all the missions tested the control law parameters were as follows: $K_n = K_e = 0.2$, $K_d = K_a = K_\phi = K_\theta = K_\psi = 1$, $K_u = K_w = K_p = 4.8$, $K_q = 5$, $K_r = 4$, $K_{iu} = K_{iw} = 9$, $K_{ip} = K_{iq} = 1$, $K_{ir} = 16$, $\zeta_n = \zeta_e = \zeta_d = \zeta_a = 0.8507$, $\zeta_\phi = \zeta_\theta = \zeta_\psi = \zeta_u = \zeta_w = \zeta_p = \zeta_q = \zeta_r = 0.85$, $\omega_n^n = \omega_n^e = 0.6 \frac{rad}{s}$, $\omega_n^d = \omega_n^a = 1 \frac{rad}{s}$, $\omega_n^\phi = \omega_n^\theta = \omega_n^\psi = 2\pi \frac{rad}{s}$, $\omega_n^u = \omega_n^w = \omega_n^p = \omega_n^q = \omega_n^r = 12 \frac{rad}{s}$.

As indicated in Section 3.3.2, all of the unknown bias parameters in the navigation system usually converged to reasonable values during the first 500 seconds of the mission. To show our control performance for each behavior we allow some time for navigation performance to become satisfactory by convergence of the unknown bias parameters.

In Chapter 5 we present the relevant simulated and in-water missions to demonstrate each behavior; its performance; and its relevance to desired AUV missions. We also show the control performance during behavior switching. Section 5.1 focuses on the performance of Behavior 2, Section 5.2 on performance of Behavior 1, Section 5.3 on performance of Behavior 3, while Section 5.4 on performance during behavior switching.

5.1 Behavior 2 Simulation Results

The objective of this section is to show the performance of Behavior 2. This was the first behavior that we designed and it served as a building-block for two other behaviors. By designing B2, we were able to control the vehicle's depth, the speed, and attitude, except at $\phi = \theta = \frac{\pi}{2}$.

5.1.1 Simulation Results

Figures 5.1–5.4 present the results of a portion (120 seconds) of a simulated mission, during which the vehicle goes to a desired depth of 2 m and then performs change of attitude maneuvers, roll and pitch of $\pm 45^\circ$ and yaw of $\pm 135^\circ$. Throughout this mission $u_c^o = 0$. The vehicle is initially at 1 m depth and zero attitude.

The plot showing d , ϕ , θ , and ψ versus time is shown in Figure 5.1. Each of these plots contains three curves, for example d , d_c^o , and d_c . Note that for each state, x_c converges to x_c^o at the rate determined by $\omega_n^d = 1 \frac{rad}{s}$ and $\omega_n^\phi = \omega_n^\theta = \omega_n^\psi = 2\pi \frac{rad}{s}$. Rapid convergence to and maintenance of the trajectory is exhibited in Figure 5.1. The signals u , w , p , q , and r converge to and track u_c , w_c , p_c , q_c , and r_c throughout the simulation, as shown in Figures

5.2 and 5.3. As seen in the second graph of Figure 5.2, depth command filter maximum rate is set to $0.3\frac{m}{s}$ and w_c is limited to this value. Figure 5.3 shows that p_c and q_c are limited to $10\frac{rad}{s}$ while r_c are limited to $25\frac{rad}{s}$. This is explained in Remark 3 in Appendix A. Note that the simulation does include sensor noise, which is evident in the u and w graphs of Figure 5.2. Also we can see that the w_c^o changes at around $t = [875]$ and $t = [895]$ in the second graph of Figure 5.2 which can be explained since w is used to maintain depth while tracking ϕ using the same vertical thrusters. The thrusters are not saturated as seen in Figure 5.4.

Figure 5.5 presents selected signals during a magnified portion of the simulated mission, $t = [818, 830]$, where the depth command is achieved. The third graph of Figure 5.5 shows \bar{d} and ν_d . Due to the selection of the CF initial condition, see Theorem 4 in Section 4.2.2, $\nu_d = 0$ to the extend reasonable given measurement noise and navigation error. During the time interval following behavior switching, $t = [818, 825]$, while w_c converges to w_c^o , as seen in the second graph, \bar{d} increases and then converges back toward zero as predicted by the theory. During such time intervals a bounded transient is clearly evident in ξ_d , see the bottom graph of Figure 5.5. Similar analysis can be applied to other state variables and this coincides with the analysis in Section 2.5.

Figure 5.6 plots $V_{b_2}(t)$ and $\|\bar{\mathbf{x}}\|$ versus time, where $\mathbf{x} = [\bar{d}, \bar{\Theta}, \bar{w}, \bar{\omega}]$ is the error state vector for the portion of the mission where AUV changes its pitch, $t = [830, 870]$. The Lyapunov function defined in terms of the compensated tracking error is nonincreasing at all times. This result confirms our theoretical conclusion showing that the Lyapunov function of the CFBS approach starts at the value which is a function of the integral error

(\bar{e}) at the beginning of each behavior, decreases during the duration of each behavior, and maintains its value during the instances of switching between behaviors. On the other hand, at the start of each behavior, x_c is equal to x , not equal to x_c^o . During the time interval (influenced by ω_n) during which x_c converges toward x_c^o , the signal \bar{x} may increase. The distinct nature of the tracking error and compensated tracking error is evident in Figure 5.6.

In Section 5.1 our goal was to demonstrate that AUV can control depth and attitude at zero speed using B2 which is defined using the control law of Section 4.1.2. This was demonstrated with the simulation performance.

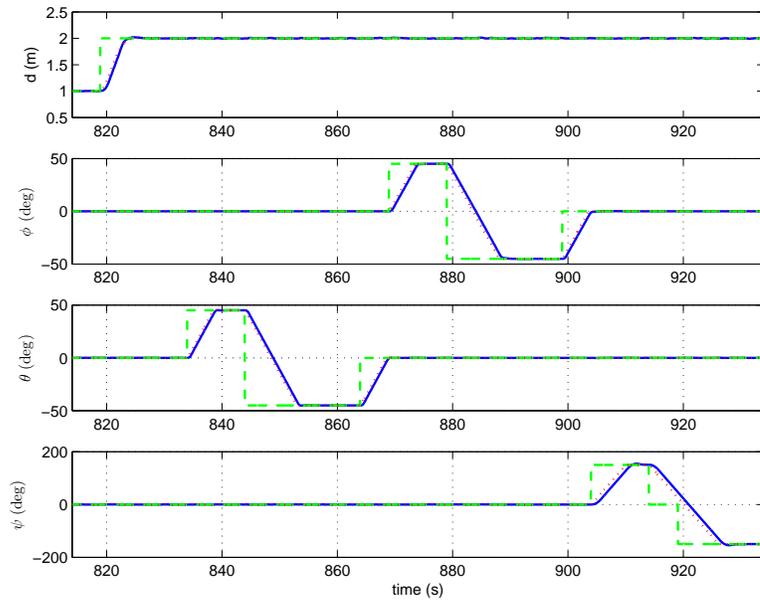


Figure 5.1: Depth and Attitude vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command.

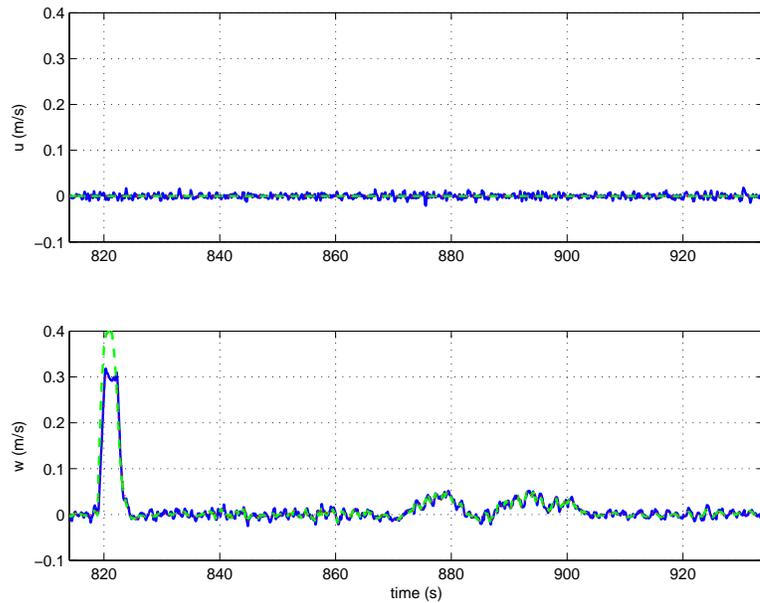


Figure 5.2: Velocities vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command.

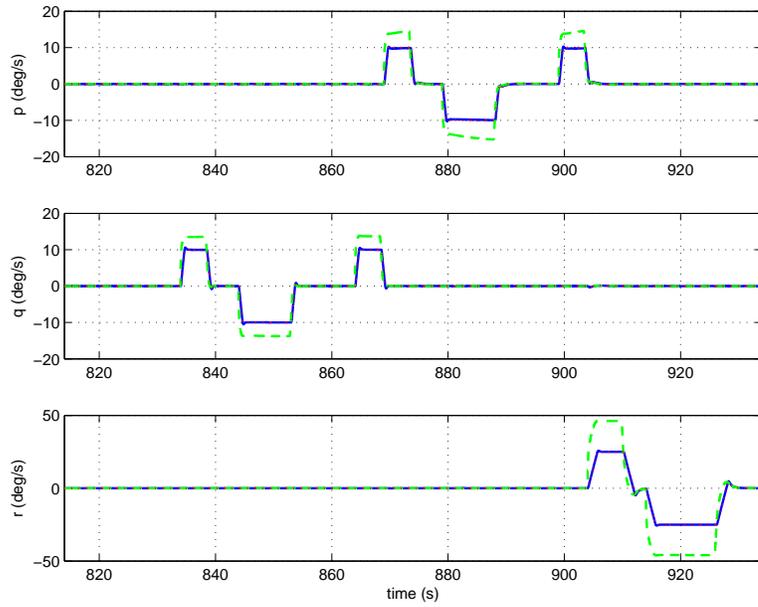


Figure 5.3: Angular Rates vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command.

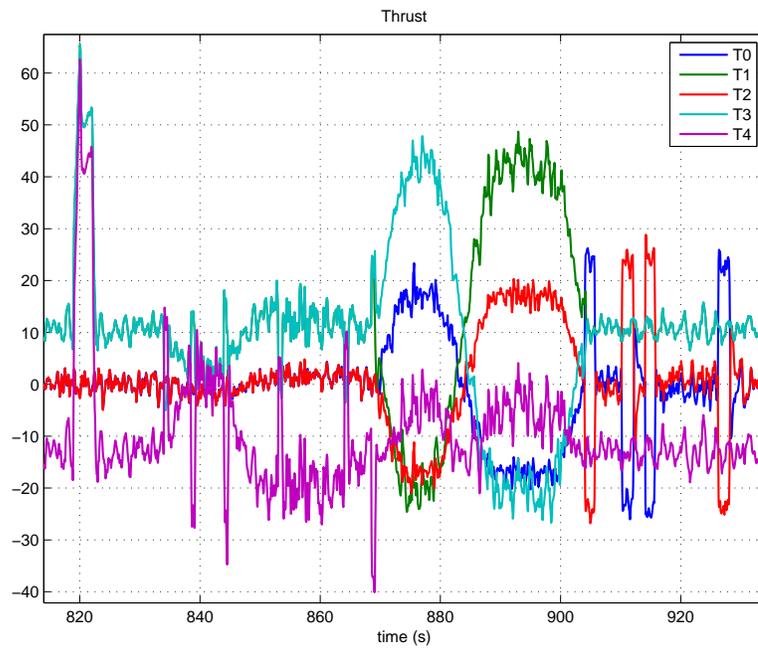


Figure 5.4: Thruster signals (%) vs. time.

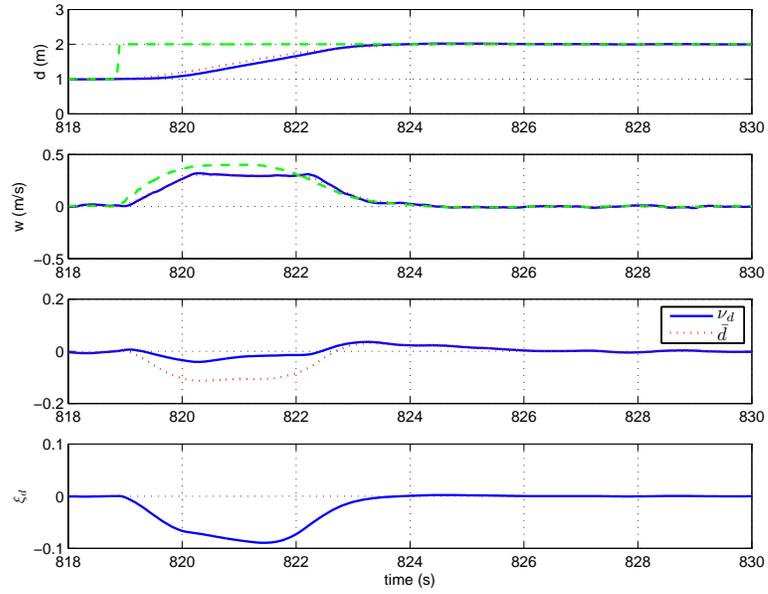


Figure 5.5: Top - Depth. Second - Vertical Velocity. Third - Error signals \bar{d} and ν_d vs. time. Bottom - Signal ξ_d vs. time.

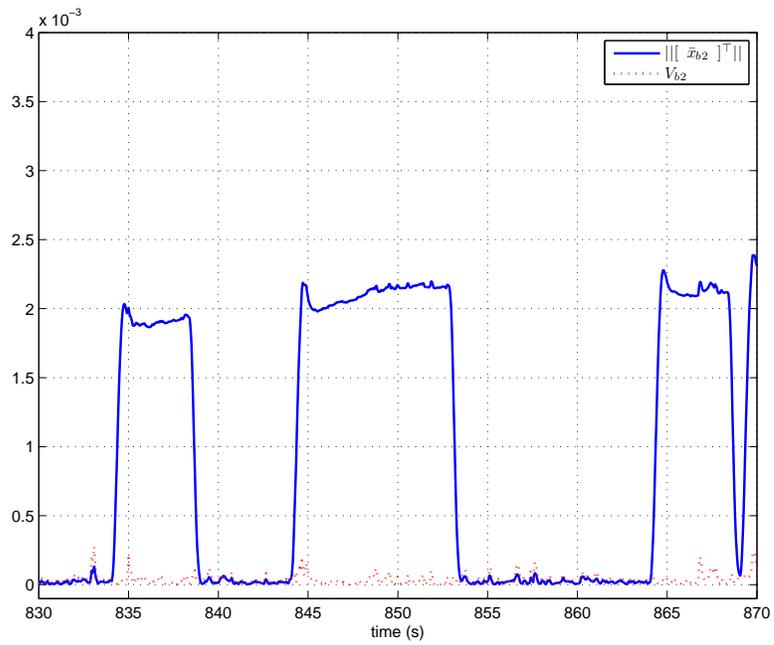


Figure 5.6: Lyapunov Function and $\|\tilde{x}\|^T$ vs. time.

5.2 Behavior 1 Results

The objective of this section is to demonstrate the performance of Behavior 1. The design of Behavior 1 was a major milestone of this research. By designing B1, we were able to control the vehicle's 3D position, speed, angular rate, and attitude, except at $\theta = \frac{\pi}{2}$.

5.2.1 Simulation Results

Trajectory tracking with zero attitude

Figures 5.7–5.12 present the results of an 300 second simulated mission, during which the vehicle navigates around a triangular path inside of simulated TRANSDEC pool. The vehicle is initially at the (0,0) corner and navigates clockwise with speed 0.5 or $0.3\frac{m}{s}$.

The vehicle starts in Behavior 2 and it is commanded to go to 1.5 m depth while maintaining zero attitude. During this behavior north and east position is not being controlled. Following that, Behavior 1 is initiated where the vehicle tracks the desired north and east trajectory while maintaining the desired depth. When the vehicle reaches the desired waypoint with the tolerance of 1 m, Behavior 2 is initiated for 20 sec. During each instance of B2, the AUV depth and yaw are manipulated to the proper initial conditions for the next trajectory segment. Therefore, the AUV navigates in a triangular path stopping at each corner and executing B2. In the last segment shown in this simulated mission the AUV navigates to the (20,0) corner while changing its depth to 4 m.

Figure 5.7 is a 2D position plot which shows the actual position converging to the filtered desired position which converges to the actual desired position. The plot showing n , e and d versus time is shown in Figure 5.8. Each of these two plots (and the following

plots) contains three curves, for example x , x_c^o , and x_c . Note that during the time period that Behavior 2 is active, the n and e state plots in Figure 5.8 are set to be equal (value of the current vehicle position), since during that behavior n and e states are not controlled and n_c^o , e_c^o , n_c and e_c are undefined. Again, w_c and r_c are limited as explained in Remark 3 in Appendix A.

Note that for each state, x_c converges to x_c^o at the rate determined by ω_n^n , ω_n^e , ω_n^d , ω_n^ϕ , ω_n^θ , ω_n^ψ , ω_n^u , ω_n^w , ω_n^p , ω_n^q , and ω_n^r for Behavior 1 (values given in the introduction of Chapter 5). The convergence is evident in plots in Figures 5.8, 5.9, 5.10, and 5.11. The signals x representing the values of each state converge to and track the filtered desired commands x_c throughout the simulation. Figure 5.12 shows that there is not thruster saturation.

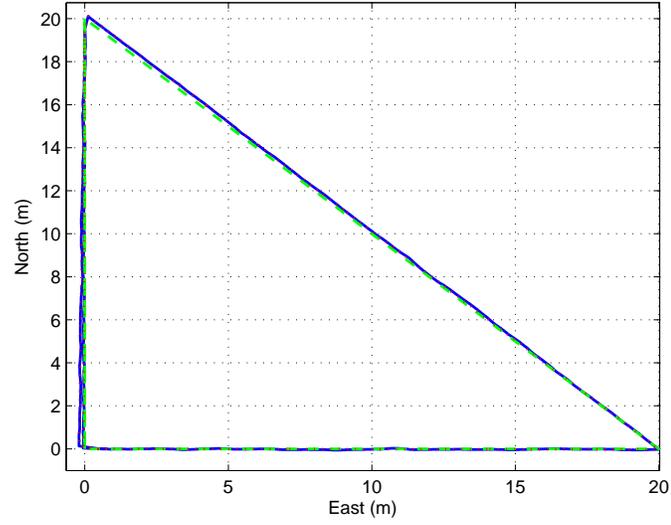


Figure 5.7: 2 D Position vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command.

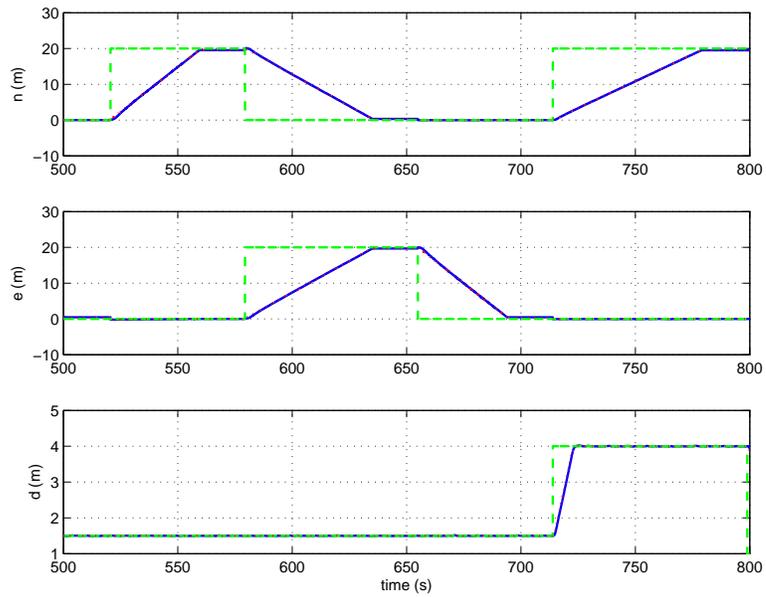


Figure 5.8: North, East, Depth Position vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command.

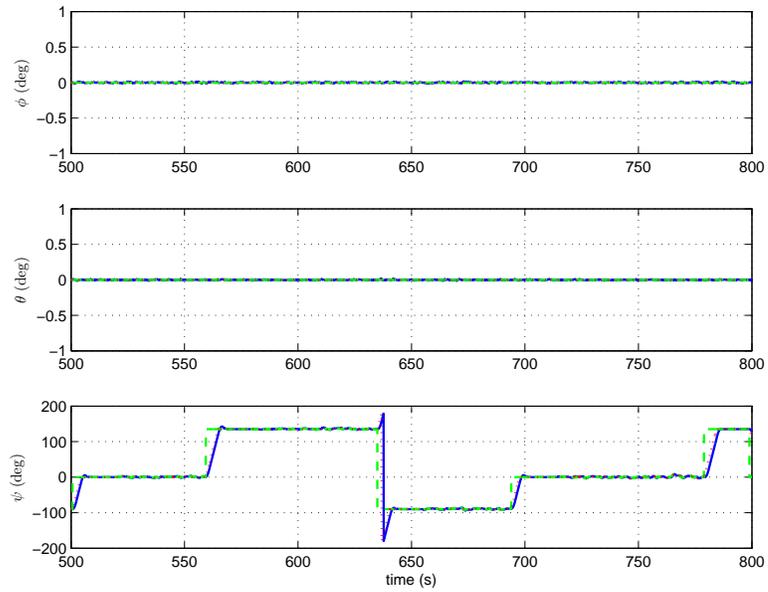


Figure 5.9: Attitude vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command.

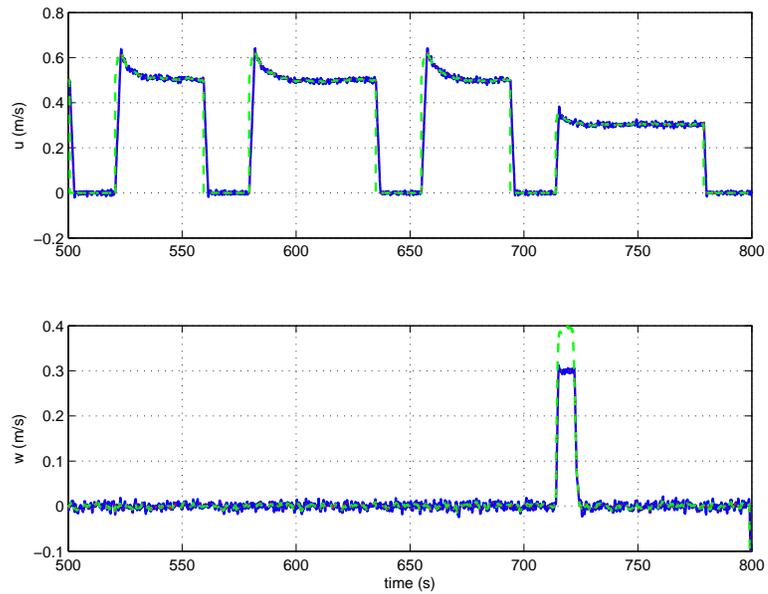


Figure 5.10: Velocities vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command.

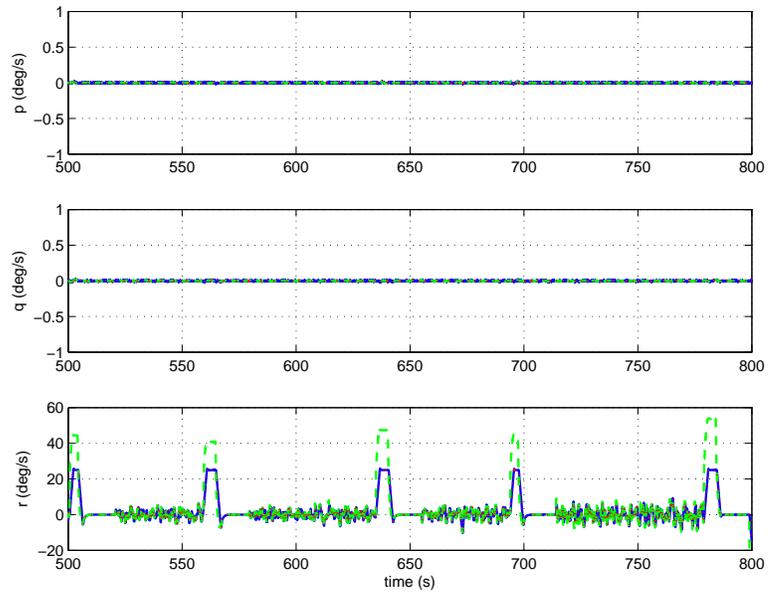


Figure 5.11: Angular Rates vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command.

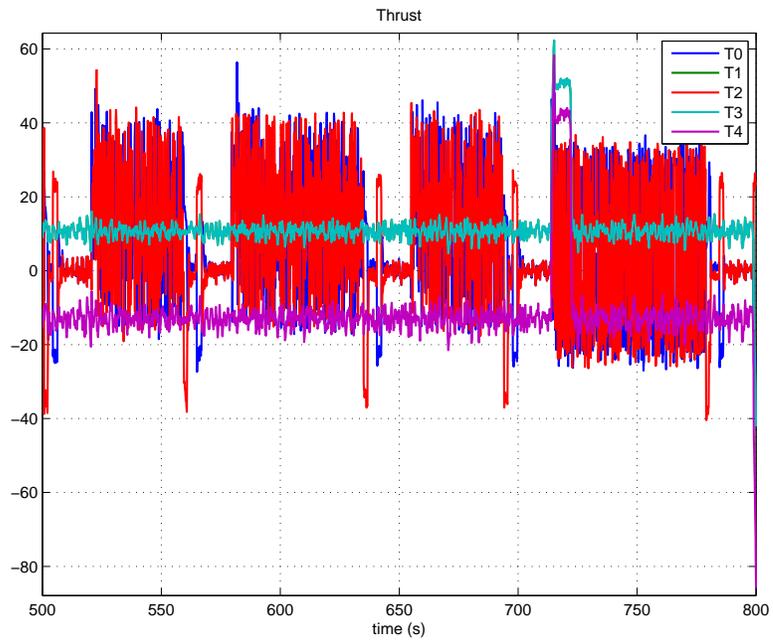


Figure 5.12: Thruster signals (%) vs. time.

Trajectory tracking with nonzero attitude

The mission objective for this simulated mission is the same as for a mission explained above; i.e. to follow a specified trajectory, except that the vehicle is commanded to drive with a pitch of -60° and then to roll of 60° .

Figures 5.13–5.14 present the results of 600 sec. of a simulated mission, during which we tested B1 with nonzero attitude. The vehicle starts in Behavior 2. During B2 north and east position is not being controlled. During this portion of the mission, $t = [1800, 1900]$, Behavior 1 is initiated where the vehicle tracks the desired north and east trajectory and at the same time tracks a time varying depth command (from 4 to 3 m). In addition, the vehicle is commanded a nonzero value of pitch (-60°). When the vehicle reaches the desired waypoint (a corner of a triangle) with the tolerance of 1 m, Behavior 2 is initiated. During the portion of the mission, $t = [2080, 2180]$, Behavior 1 is executed in the same fashion, but the vehicle is commanded a nonzero value of roll (60°). The AUV can achieve 3D navigation while tracking nonzero roll or pitch as shown in Figures 5.13–5.14. B1 with nonzero commanded roll can be useful in the ship-hull inspection of the side of the ship. The vehicle can be commanded to in a straight line alongside the ship and the roll of the vehicle can be adjusted for optimal imaging of a side-scan sonar

Figure 5.15 presents a portion of the simulated mission, $t = [2076_2 096]$, where roll of 60° is achieved. The third graph of Figure 5.15 shows $\bar{\phi}$ and ν_ϕ . Due to the selection of the CF initial condition, see Theorem 4 in Section 4.2.2, $\nu_\phi = 0$ to the extend possible given measurement noise and navigation system estimation error. During the time interval following behavior switching while p_c converges to p_c^o , shown by the second graph. During

this convergence $\bar{\phi}$ increases in magnitude and then converges back toward zero as predicted by the theory. During such time intervals a bounded transient is clearly evident in ξ_ϕ , see the bottom graph of Figure 5.15. As seen in the second graph of Figure 5.15, p_c is limited to $10\frac{rad}{s}$ as explained in Remark 3. Similar analysis can be applied to other state variables and this coincides with the analysis in Sections 2.5 and 5.1.

Figure 5.16 plots $V_{b_1}(t)$ and $\|\bar{\mathbf{x}}\|$ versus time, where $\mathbf{x} = [\bar{\mathbf{p}}, \bar{\boldsymbol{\Theta}}, \bar{\mathbf{v}}, \bar{\boldsymbol{\omega}}]$ is the error state vector. The Lyapunov function defined in terms of the compensated tracking error is nonincreasing at all times. This result confirms our theoretical conclusion showing that the Lyapunov function of the CFBS approach starts at the value which is a function of the integral error (\bar{e}) at the beginning of each behavior, decreases during the duration of each behavior, and maintains its value during the instances of switching between behaviors. The distinct nature of the tracking error and compensated tracking error is evident in Figure 5.16.

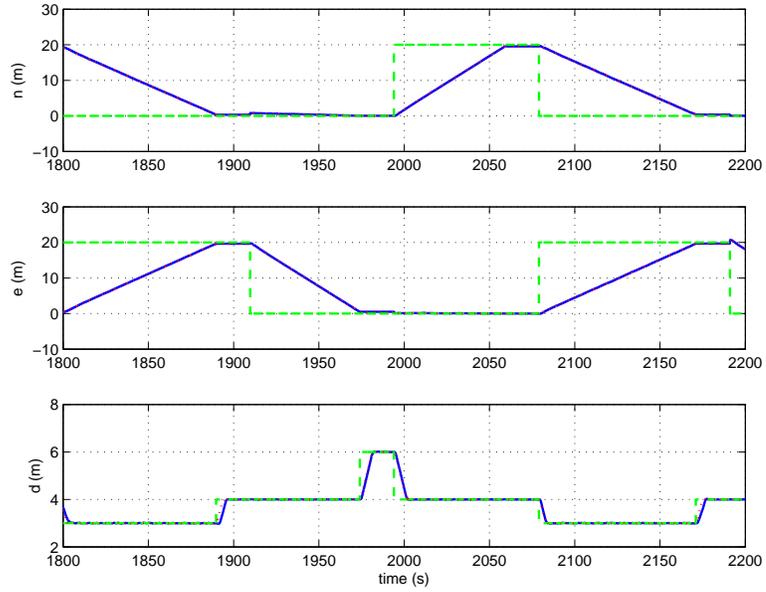


Figure 5.13: North, East, Depth Position vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command.

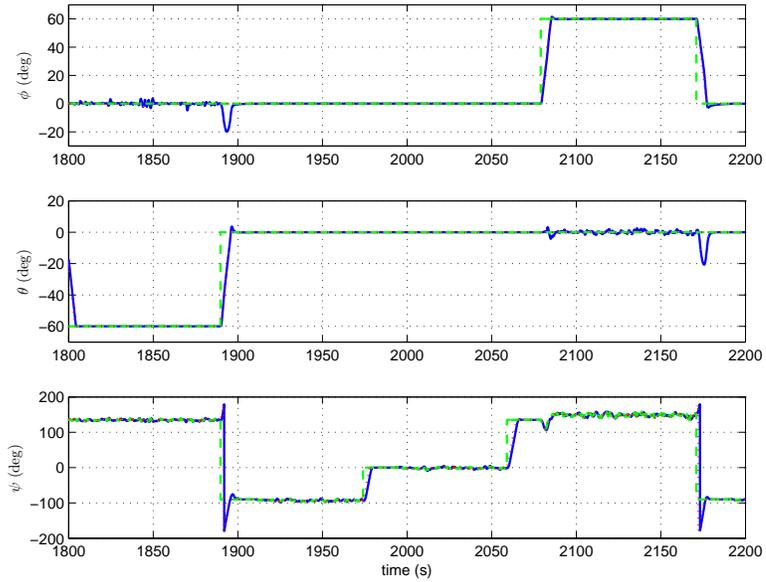


Figure 5.14: Attitude vs. Time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command.

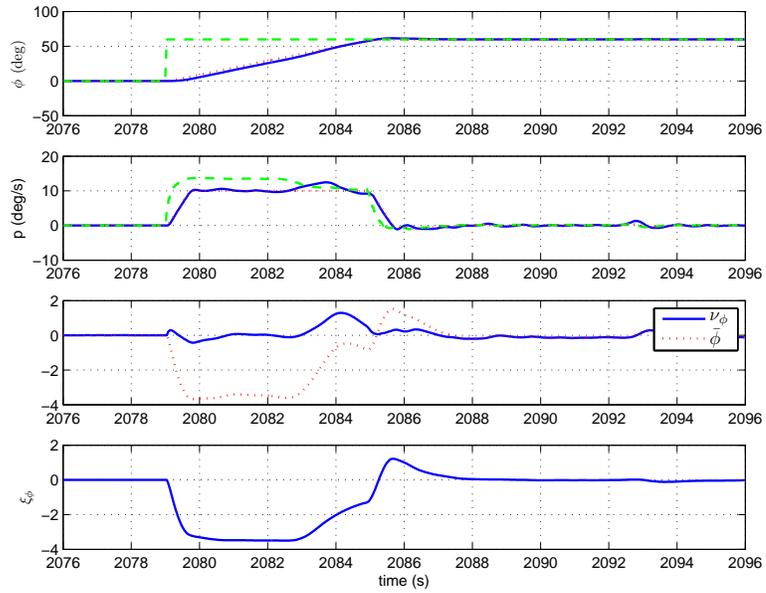


Figure 5.15: Top - Roll. Second - Roll rate. Third - Error signals $\bar{\phi}$ and $\bar{\nu}_\phi$ vs. time. Bottom - Signal ξ_ϕ vs. time.

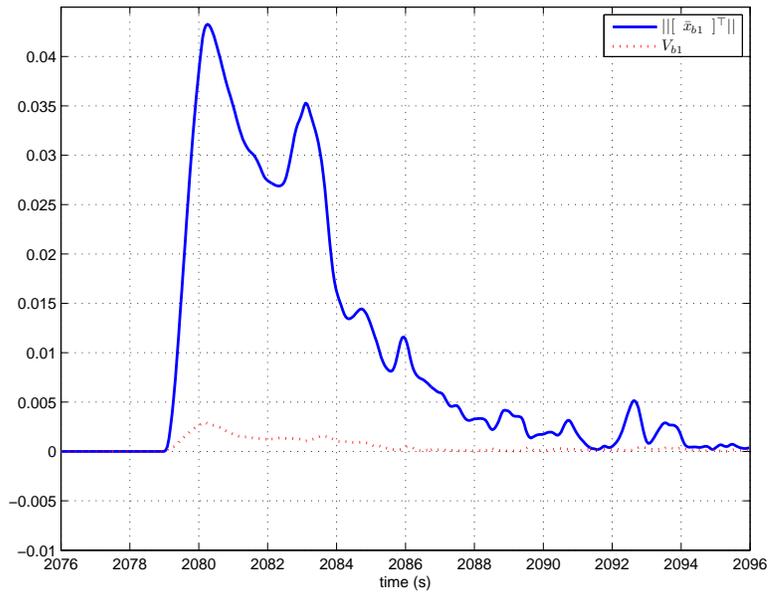


Figure 5.16: Lyapunov Function and $\|[\bar{x}]^T\|$ vs. time.

5.2.2 Field Test Results

The goal of this section is to present data from two AUV experiments. This dissertation emphasizes the comparison between the simulated and actual in-water mission performed by the AUV. The actual results closely resemble the simulation results.

AUV Fest June 2007 Test Results

The following experimental results are from a demonstration at the Autonomous Underwater Vehicle Festival (AUVFest), June 6-15 2007, held in Panama City. The mission plan was to submerge to 3 m in depth for 2 minutes, then execute two sets of three-dimensional waypoints at 1 knot. The first series of waypoints consisted of vertically stacked legs between two waypoints. During this phase, the navigational goal was to estimate the unknown navigation parameters (yaw and biases) before proceeding to the second series of waypoints underneath a barge. With the vehicle transiting along the side of the barge that was supposed to be inspected, at specified distance from the barge and appropriate depths, Marine Sonics Side Looking Sonar (MSTS) was imaging the side of the barge. The second series of waypoints consisted of a lawnmower search pattern in an continuous loop under the barge. Due to severe magnetic interference from the barge, we chose to operate the vehicle without aiding the navigation system with the magnetic compass. During this phase Sound Metrics DIDSON High Definition Imaging Sonar (DIDSON) sonar was used to image the bottom of the hull of the barge while the vehicle was transiting at a predefined depth. The acoustic baseline outlines a $36 \times 9 \text{ m}^2$ box around the second series of waypoints.

During AUV Fest, we accomplished 12 hours of in-water demonstration time.

Hull search AUV behaviors were demonstrated with the following unique capabilities: autonomous mission execution with intervention capability, hull search conducted using side look sonar, sensors parameters optimized by the operator during the mission, real-time top-side display of DIDSON and MSTs, vehicle position and status information embedded in DIDSON sensor data and Joint Architecture for Unmanned Systems (JAUS) communication protocol implemented on AUV. Our team was invited to demonstrate for the Media Day and for the Distinguished Visitors Day.

During AUV Fest Behavior 1 and Behavior 2 were tested. Behavior 1 was used to transit along the side of the hull and to execute the legs of the lawnmower search pattern under the hull, while Behavior 2 was used when the vehicle made turns in both phases of the mission. The 2D position plot is shown Figure 5.17. Great trajectory tracking performance can be noticed since the vehicle maintained its track-line even with the presence of side currents as it should because of the term \mathbf{Y} , which accounts for v direction motion, defined below eqn. (4.18) in Section 4.3.1. This is a greatly desired capability for this type of mission since 100 % ship-hull coverage is requirement and “holidays” in the sonar data is not acceptable. The position plot showing north, east, down, and altitude position versus time is shown in Figure 5.18, while the attitude (roll, pitch, yaw) plot is shown in Figure 5.19. Excellent tracking performance can be observed, for instance, maximum depth tracking errors were around 5 cm, maximum roll tracking errors were around 4 degrees, while maximum tracking errors in pitch were around 2 degrees. Due to time limitations Behavior 3 was not tested at AUV Fest.

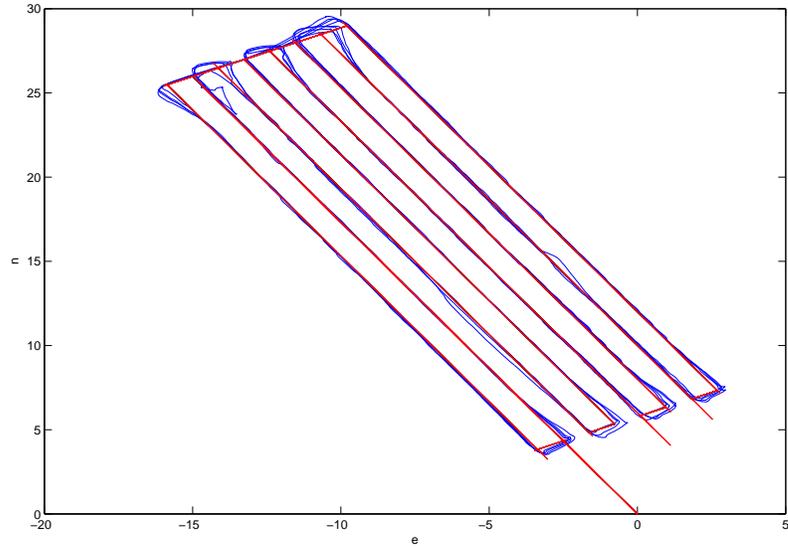


Figure 5.17: 2 D Position vs. Time: Blue line is the actual vehicle trajectory, black line is the command, and the red line is filtered command.

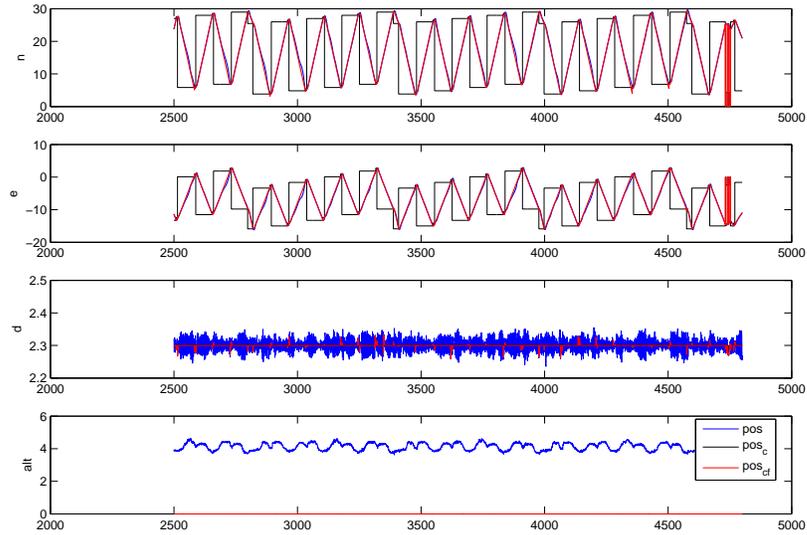


Figure 5.18: North, East, Depth, and Altitude Position vs. Time: Blue line is the actual vehicle trajectory, black line is the command, and the red line is filtered command.

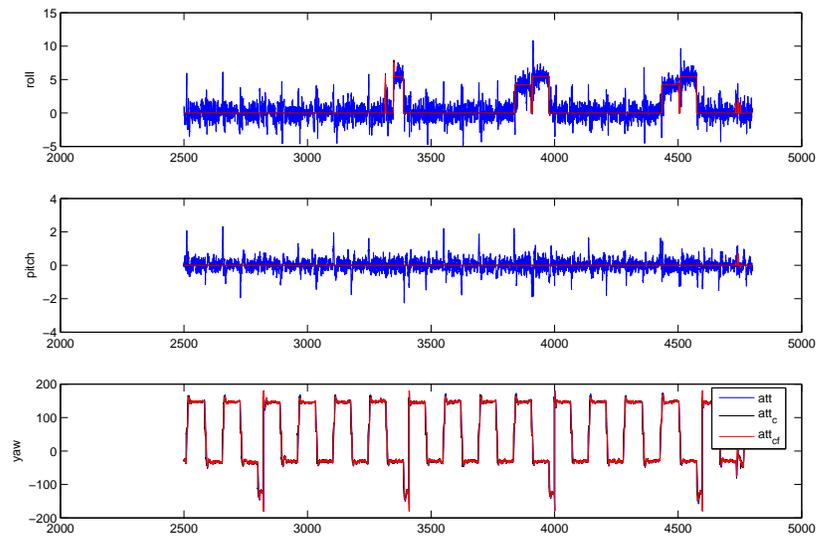


Figure 5.19: Attitude vs. Time: Blue line is the actual vehicle attitude, green line is the command, and the red line is filtered command.

TRANSDEC August 2008 Test Results

The AUV trajectory tracking and surface following was tested in the 300 ft \times 200 \times 38 \times ft deep TRANSDEC pool at SSC-SD. The pool is bowl shaped so its depth is increasing toward its center. It is shown in Figure 3.5. The missions tested were similar to the missions trajectory shown in Figure 5.7. There were different missions, one utilizing B1 and B2 and switching between them and another utilizing B1, B2, and B3 and switching among them. Again, the goal was for a vehicle to transit in 3D along the edges of the triangular path while, first: maintaining zero attitude, second: maintaining commanded nonzero attitudes, and finally transiting while following the surface below it - the bottom of the curved pool.

Figures 5.20–5.22 present the results of a 80 s long actual in-water mission, during which the vehicle navigates around a triangular path in the TRANSDEC pool. As seen in Figure 5.20, north and east position tracking is not great but it should be noted that the vehicle is at the same time changing depth (from 4 to 3 m) and swimming at 60° pitch. This performance can be explained by the fact the vertical T_4 thruster is saturated during the majority of the transit as seen in Figures 5.21. During this testing we limited the thrusters to 80% in order to avoid a catastrophic failure of a power board which occurred at our previous testing. Still, the results very closely resemble the simulated results which validates our approach. For instance, the third graph of Figure 5.22 shows $\bar{\theta}$ and ν_θ . Due to the selection of the CF initial condition, see Theorem 4 in Section 4.2.2, $\nu_\theta = 0$ to the extend possible given measurement noise and navigation system estimation error. During the time interval following behavior switching while q_c converges to q_c^o , as seen in the second

graph, $\bar{\theta}$ increases and then converges back toward zero as predicted by the theory. During such time intervals a bounded transient is clearly evident in ξ_θ , see the bottom graph of Figure 5.22. The fact that θ does not track θ_c in the period around $t = 120s$ and $t = 175s$ can also be explained by thruster saturation since T_4 is used to control both depth, roll, and pitch.

In Section 5.2 our goal was to show that AUV can control its 3D position and attitude while driving at nonzero speed and this was shown with both the simulation and in-water performance. The AUV using the control law of Section 4.1.1 can achieve 3D navigation while tracking nonzero roll or pitch.

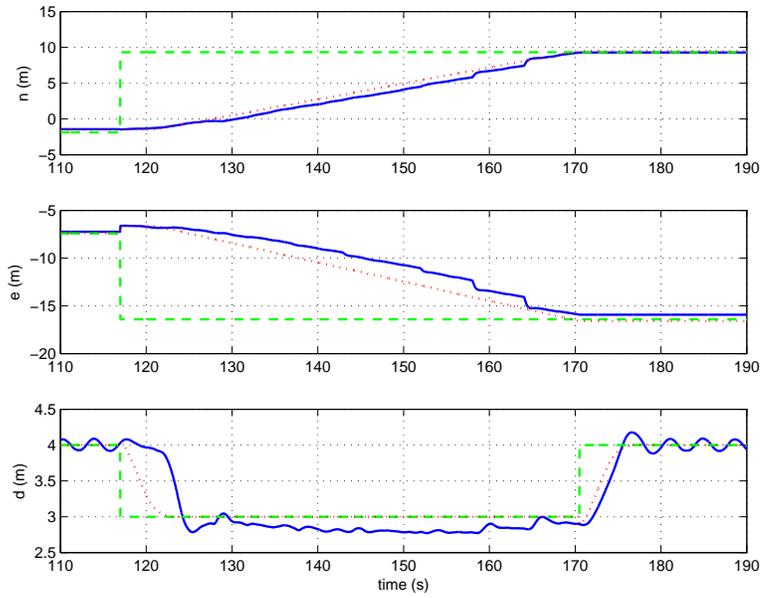


Figure 5.20: North, East, Depth Position vs. Time: Blue line is the actual vehicle trajectory, green line is the command, and the red line is filtered command.

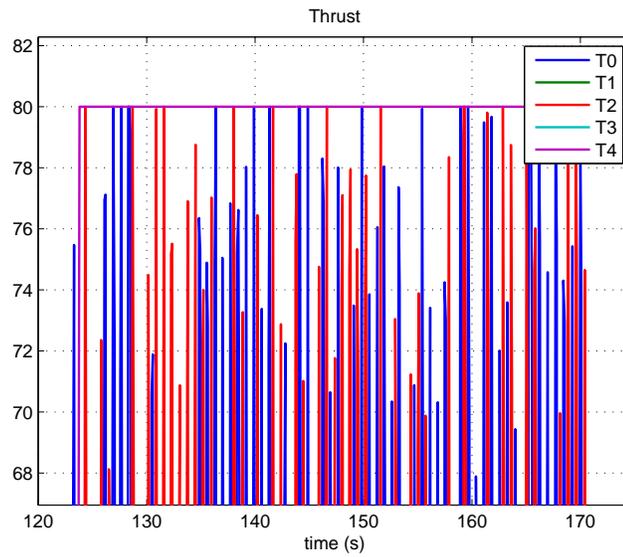


Figure 5.21: Thruster signals (%) vs. time.

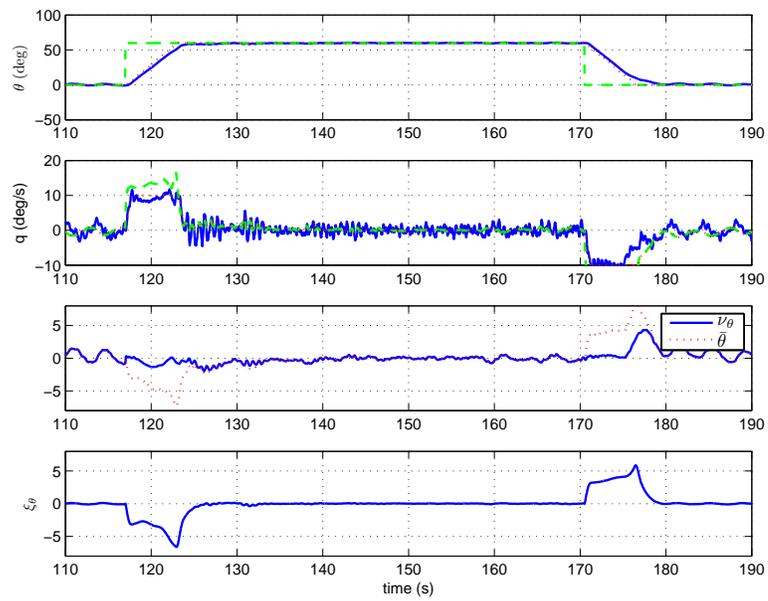


Figure 5.22: Top - Pitch. Second - Pitch rate. Third - Error signals $\bar{\theta}$ and ν_{θ} vs. time. Bottom - Signal ξ_{θ} vs. time.

5.3 Behavior 3 Results

The objective of this section is to show the performance of Behavior 3. The design of Behavior 3 created a capability, surface following, necessary for an AUV to perform inspection of the non-complex portions of the ship hull. By designing B3, we were able to control vehicle's 3D position, transit at varying attitude, except at $\theta = \frac{\pi}{2}$. B3 is needed to follow a surface above or below the AUV.

5.3.1 Simulation Results

Figure 5.23 presents a portion of the simulated mission, $t = [3500, 3670]$, where B3 is executed on two of the segments ($t = [3510, 3580]$ and $t = [3600, 3660]$) while following a triangular path. At each corner B2 is initiated to align the AUV state with the next trajectory portion, as it was done in Section 5.2. The vehicle is maneuvering in three dimensions, constantly adjusting its attitude to maintain a specific attitude and zero attitude relative to surface below it. As seen in Figure 5.23 there are two line segments to be tracked while following surface below the AUV. Two segments require the vehicle to change the pitch with maintaining roll of approximately 0° - going from the edge of the bowl shaped pool to the middle and going from the middle to the edge ($t = [3510, 3580]$ from the (0,20) to the (0,0) corner, and $t = [3600, 3660]$ from the (0,0) to the (20,0) corner). Great tracking of both roll and pitch can be seen in Figure 5.23. As we can see from Figure 5.24, depth is not controlled while B3 is active (third graph) but the altitude from the bottom is controlled (fourth graph).

Figure 5.25 presents a portion of the simulated mission, $t = [3125, 3300]$, where all

three behaviors are tested. As it can be seen from Figure 5.25 the AUV is first in B2, than in B1 ($t = [3135, 3195]$) navigating from the (0,20) to the (0,0) corner while maintaining depth of 4 m, $t = [3045, 3185]$. Then, B2 is initiated for $t = [3195, 3215]$, while the AUV is at rest, maintaining 4 m depth. Finally, for $t = [3220, 3280]$ B3 is initiated as the AUV navigates to the (20,0) corner while following the surface below it. As seen from the third and fourth graph of Figure 5.25, for $t = [3135, 3195]$ B1 controls depth (graph 3) and for $t = [3220, 3280]$ B3 controls altitude from the bottom (i.e., $a = -2m$ is maintained) (graph 4). The smooth parabolic bottom shape is evident from the graph of d vs. t for $t = [3220, 3280]$.

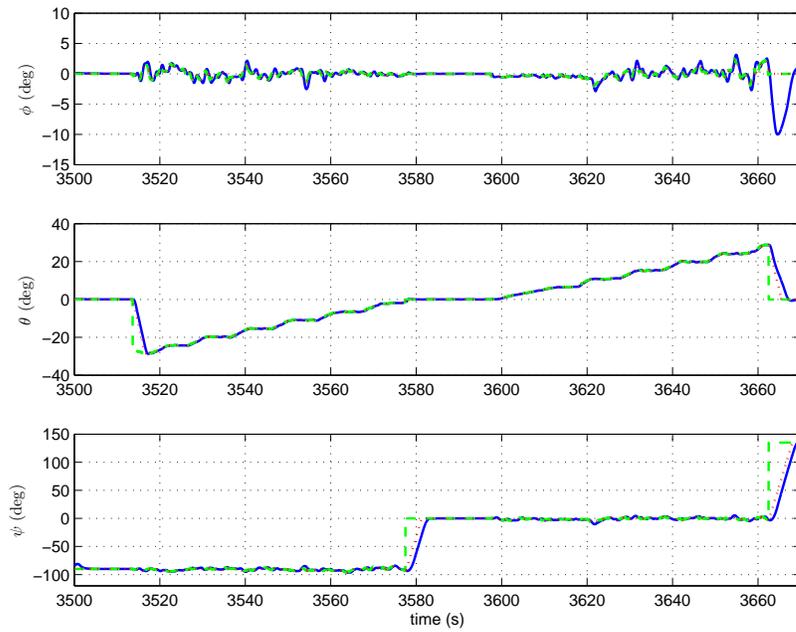


Figure 5.23: Attitude vs. Time: Blue line is the actual vehicle attitude, green line is the command, and the red line is filtered command.

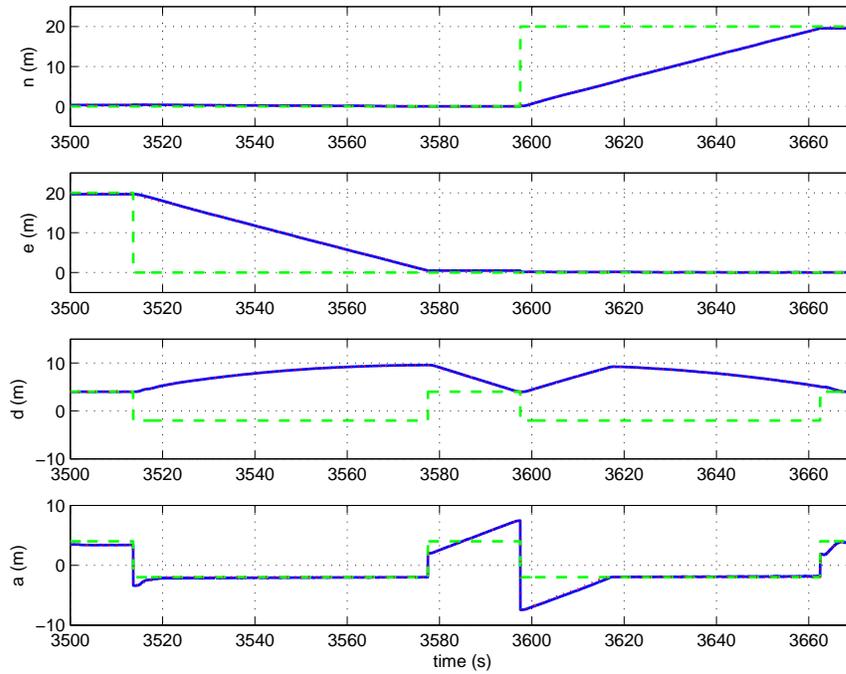


Figure 5.24: North, East, Depth, and Altitude Position vs. Time: Blue line is the actual vehicle trajectory, green line is the command, and the red line is filtered command.

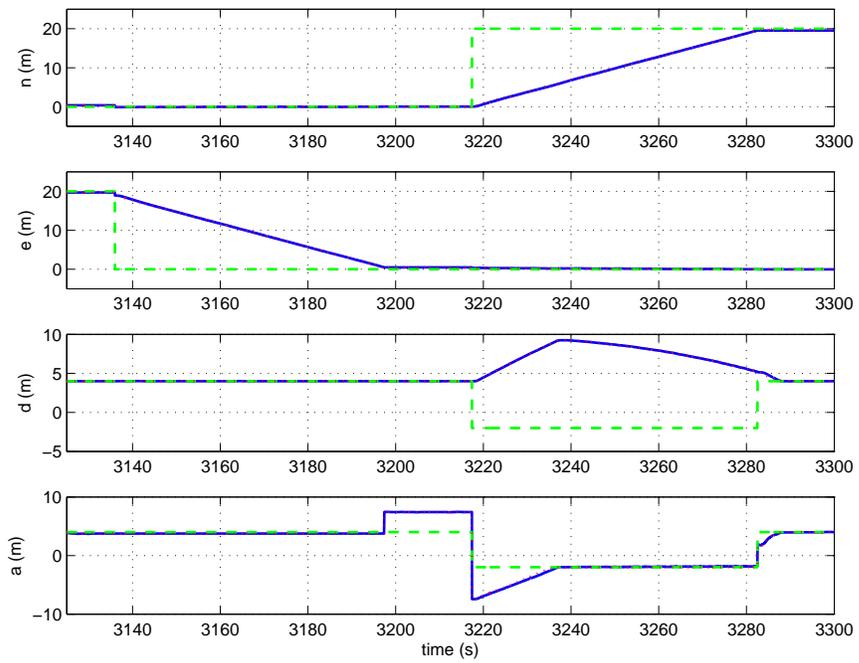


Figure 5.25: North, East, Depth, and Altitude Position vs. Time: Blue line is the actual vehicle trajectory, green line is the command, and the red line is filtered command.

5.3.2 Field Test Results

The vehicle's task was to track a trajectory segment along the desired path while maintaining a specified altitude and zero attitude relative to the bottom based on the information coming from the sonar (ADCP) sensor.

TRANSDEC August 2008 Test Results

Figures 5.26–5.28 present the results of a 130 s actual in-water mission, during which the vehicle navigates around a triangular path in the TRANSDEC pool while maintaining a commanded altitude relative to the bottom. This behavior is essential in the ship-hull inspection as it enables the AUV to conform to and maintains distance to the surface of the hull for optimal imaging sonar positioning. As seen in Figure 5.26, north and east position tracking is not great but it should be noted that the vehicle is at the same time keeping the distance of 2 m away from the bottom. This performance can be explained by the fact the vertical T_4 thruster is saturated during the majority of the transit as seen in Figures 5.28. As indicated above, the thrusters were limited to 80%. The results also very closely resemble the simulated results which validates our approach.

In Section 5.3 our goal was to show that AUV can track a trajectory and adjust its attitude while driving at nonzero speed in order to follow a curved surface above or below it. This capability was demonstrated with both the simulation and in-water performance.

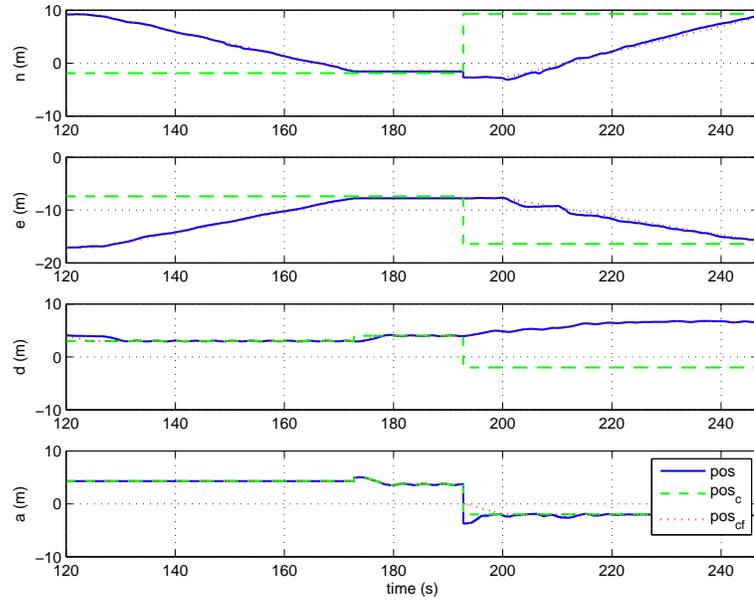


Figure 5.26: North, East, Depth, and Altitude Position vs. Time: Blue line is the actual vehicle trajectory, green line is the command, and the red line is filtered command.

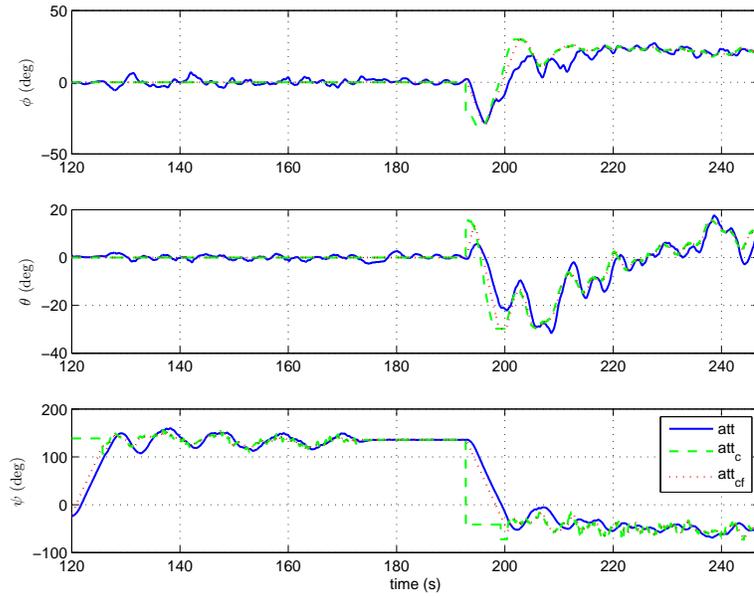


Figure 5.27: Attitude vs. Time: Blue line is the actual vehicle attitude, green line is the command, and the red line is filtered command.

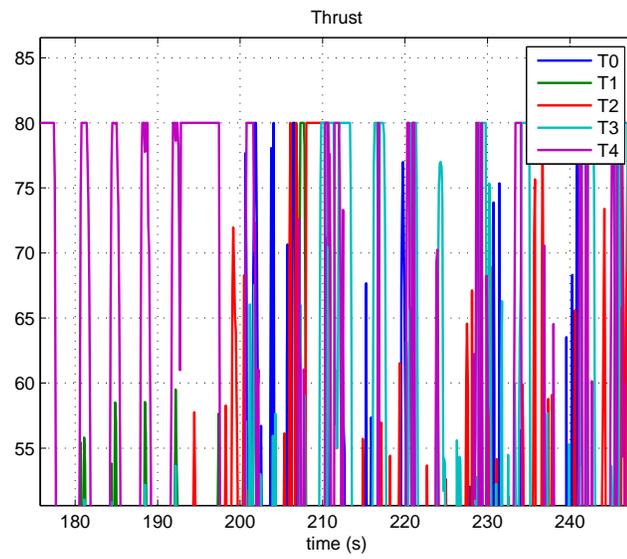


Figure 5.28: Thruster signals (%) vs. time.

5.4 Switching Performance

5.4.1 Simulation Results

Figures 5.29–5.30 present the results of an $t = [1215, 1412]$, second simulated mission, during which two switching instances occurred between behaviors 1 and 2. B2 was active for $t = [1215, 1235]$, then B1 $t = [1235, 1327]$, and then B2 $t = [1327, 1412]$, as seen in Figure 5.30.

Figure 5.30 plots $V_{b_1}(t)$ and $\|\bar{\mathbf{x}}_{b_1}\|$ versus time and $V_{b_2}(t)$ and $\|\bar{\mathbf{x}}_{b_2}\|$, where where $\mathbf{x}_{b_1} = [\bar{\mathbf{p}}, \bar{\boldsymbol{\Theta}}, \bar{\mathbf{v}}, \bar{\boldsymbol{\omega}}]$ and $\mathbf{x}_{b_2} = [\bar{d}, \bar{\boldsymbol{\Theta}}, \bar{w}, \bar{\boldsymbol{\omega}}]$ are the corresponding tracking error state vectors. The Lyapunov function defined in terms of the compensated tracking error is nonincreasing at all times. This result confirms our theoretical conclusion showing that the Lyapunov function of the CFBS approach starts at the value which is a function of the integral error (\bar{e}) at the beginning of each behavior, decreases during the duration of each behavior, and maintains its value during the instances of switching between behaviors. On the other hand, at the start of each behavior, x_c is equal to x , not equal to x_c^o . During the time interval (influenced by ω_n) during which x_c converges toward x_c^o , the signal \bar{x} may increase. The distinct nature of the tracking error and compensated tracking error is evident in Figure 5.30.

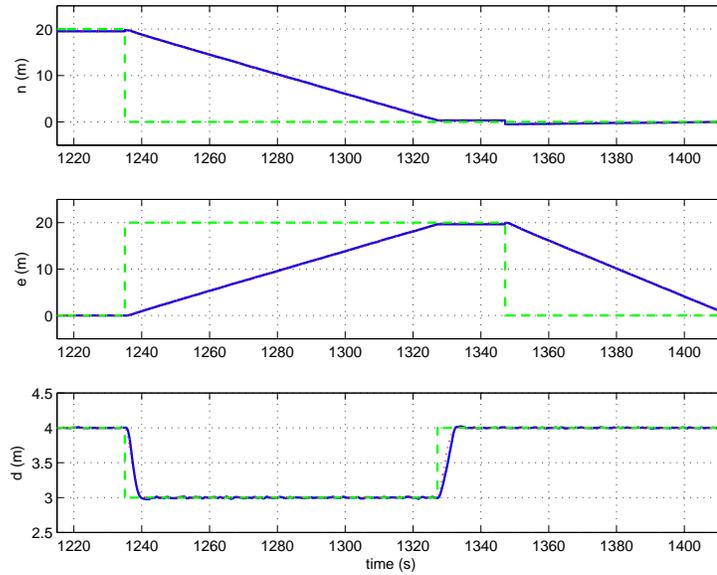


Figure 5.29: North, East, and Depth Position vs. Time: Blue line is the actual vehicle trajectory, green line is the command, and the red line is filtered command.

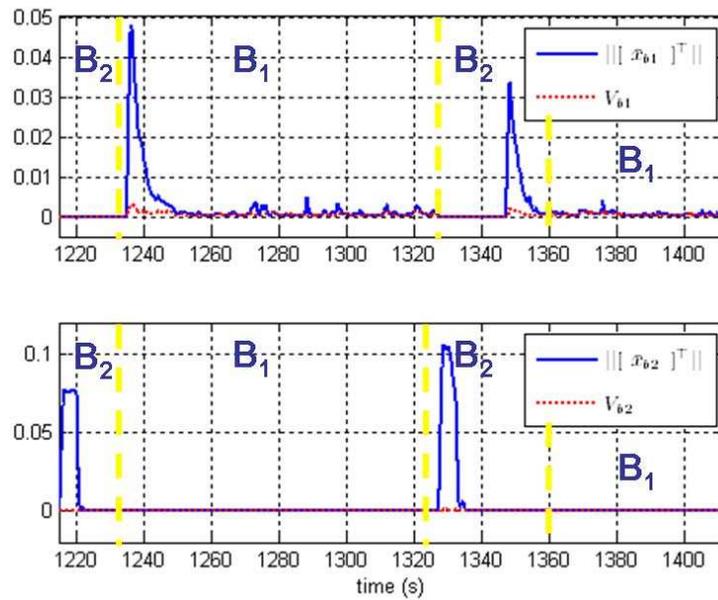


Figure 5.30: Lyapunov Function and $||[\bar{x}]^T||$ vs. time.

Chapter 6

Conclusion

In the introductory Chapter 1 we have indicated that:

1. The behaviors we implement using CFBS are stable in sense of Lyapunov;
2. CFBS method allows us to specify the value of the compensated tracking error at behavior switching times, thus ensures that the Lyapunov function does not increase;
and
3. Zeno Phenomenon can be prevented with Mission Planner design.

The analysis in Section 2.3 showed that the Lyapunov function defined for each behavior is nonincreasing during the time when each of the behaviors is active. The analysis in Section showed that the Lyapunov function defined for each behavior is nonincreasing at behavior switching time instances. Therefore, we can conclude that when perfect modeling of the plant is assumed, the Lyapunov function of each behavior remains zero at all times. Both of the results were verified by the simulation analysis in Chapter 2.

The purpose of Chapter 3 was achieved as follows. Section 3.1 clearly defined the kinematics and dynamics equations for an AUV. Section 3.2 presented the model of the vehicle dynamics which was used in our control implementation and described the methods used to identify reasonable parameters for the vehicle model. Section 3.3 overviewed our navigation algorithm, such that, the fast rate sensor is aided with slower rate sensors in order to accurately navigate in the harsh environment, for example, under the hull of a ship in the harbor. Section 3.4 described the comprehensive vehicle simulation, an essential tool for idea evaluation, debugging, and implementation and testing of complete missions closely resembling the in-water testing. Section 3.5 presented some of the related AUV applications, focusing on the main application of this research - Ship Hull Inspection mission. In addition, it overviews our control approach to the desired mission where we added functionality as needed.

Chapter 4 presents the mathematical derivation for the required AUV nonlinear controllers. The purpose of Chapter 4 was to clearly demonstrate our methodology, which was explained on a simple system in Chapter 2 when applied to a thruster controlled AUV. The purpose was to show that our method can be extended for higher order systems and for vector backstepping. The main benefits of CFBS design are:

1. Decoupling of the design of the controllers for the backstepping iterations.
2. Avoiding the tedious algebra related to computing the command signal derivatives, which becomes especially burdensome for scalar backstepping with $n > 3$ or for vector backstepping.

Even if the designer were to derive exact analytical expressions for the command derivatives relative to the design model, these are still approximations because that model is an approximate representation of the plant. Therefore, the choice is not really between a correct analytic expression for or a filtered estimate of the command derivatives; instead, the choice is between two estimates of the command derivatives.

The main benefits of Behavior Based Control design are:

1. The controller design is defined by control behaviors and a logic for switching among the behaviors.
2. Each behavior has a well-defined simple task.
3. A performance trade-off among multiple objectives using a single feedback function is avoided.
4. This approach can be built on; more mission specific control behaviors can be programmed in order to improve control performance for more complex missions.
5. Stability properties are preserved during the switching among different behaviors.

6.1 Contribution

The main contributions of this research are:

1. Design of a behavior based control design;
2. Derivation of stable nonlinear control laws applied to AUVs; and
3. Design of stable strategy for switching among behaviors.

Theorem 4 shows that behavior switching scheme is stable. The CFBS approach allows us to ensure that the Lyapunov function, defined in terms of compensated tracking errors, of the overall switched system is nonincreasing at all times by appropriate choice of the command filter's initial conditions made by a Mission Planner.

We created control algorithms for precise translational control while following a surface of interest by adjusting attitude. Great accuracy in tracking of desired commanded signal that was accomplished enables close to 100 % coverage of the desired area. We developed the simulation tool which can accelerate control development since we can experiment with operational challenges without asset risk, reduces costly in-water testing requirements as well as provides mission plan verification, and which can be used as a great operator training tool through basic operator tele-operation training. We demonstrated capabilities beyond those demonstrated for the current generation of COTS AUVs.

Our work has shown the ability of AUV to track targets in the field of view of a DIDSON imaging sonar when performing a hull search. During this research project the feedback from the altimeter sensor was used such that the AUV's path conformed to the surface above or below it while tracking the desired altitude (distance) from the surface, thus enabling DIDSON to be positioned at optimal distance and grazing angle. This research led to an AUV capability to inspect non-complex areas of the ship hull. The stable and slow AUV was able to search sides of the hull and bottom of the hull while maintaining stand-off distance and relative attitude from the hull. A mechanized solution to accomplish AUV ship hull inspection significantly decreases the dive time and subsequent human-exposure to an extremely high-risk environment.

6.2 Future Work

An advanced control capability that can be designed in future is scanning an object with the feedback from the imaging sonar such as DIDSON. In many situations it may be desirable to instruct the vehicle to navigate around a contact to obtain various aspects. Given the contact location via sonar imagery, operator input, or another data source, it is possible to constrain the vehicle's movement such that it maintains a constant visual on the contact. This constraint takes partial control of the vehicle, allowing the operator to manipulate the sonar aspect ratio without worrying about losing the contact.

The ship hull and surroundings create complex geometry for the vehicle to navigate. It is not sufficient to pass directly underneath a prop shaft, for example. Crevasses above the shaft are not visible as the shaft obstructs the sonar. The vehicle must attempt to acquire additional aspect angles. Coverage area is difficult to guarantee. Complex hull surfaces are bow including sonar domes, stern areas, hull appendages, running gear (propeller shaft(s), propeller(s), struts, rudder, and other ship's appendages). The AUV search of these areas is multiple orders of magnitude more difficult but presents a challenging objective for future.

This research has direct relevance to port protection (ability of response asset to follow and respond to diver/swimmer or other slow-moving threat). The algorithms are equally applicable to station-keeping relative to fixed contacts, as would be required with MCM disposal systems. In a broader sense, the proposed technology is a key link which will allow more rapid response and neutralization of a number of threats. The applicable threats include bottom and moored mines, IEDs on the sea floor, mines or IEDs placed

on piers, quays, or hulls of ships at anchor or in port, divers or AUVs carrying explosives or other packages. This type of missions involve the AUV patrolling an area searching for swimmers. Example behaviors for the swimmer defense mission would, at a minimum, include search, track, and evade. Detection of a swimmer causes the vehicle to covertly notify friendly security forces and switch to one of several possible other behaviors. For example, the AUV could be designed to track the swimmer at a specified standoff distance, so that the security forces could easily locate the swimmer upon their arrival. Successful tracking might also require evasive maneuvering dependent on the swimmer behavior. Both tracking and evasion could require agile, controlled maneuvering. The search, track, and evade maneuvers would be defined on-board the vehicle based on the real-time sensor data. For these mission, greatly improved onboard data processing is necessary. To process video and sonar data onboard for presentation and analysis will require technological advances and significant effort. The ability to compare this data to stored ordnance imagery for real-time identification; or to compare current hull search imagery against a baseline to identify and react to anomalies real-time will require a very high level of cognizance and significantly improved data processing systems.

6.3 Publication from the Dissertation by Chapter

6.3.1 Journal Article

1. V. Djapic, J. A. Farrell, W. Dong, "Land Vehicle Control Using Command Filtered Backstepping Approach," Submitted to IEEE Transactions on Control Systems

Technology Journal as short paper, Mar. 2009.

This article is based on the work explained in Chapters 1 and 2.

6.3.2 Conference Articles

1. V. Djapic, J. A. Farrell, W. Dong, “Unifying Behavior Based Control Design and Hybrid Stability Theory,” Accepted The 2009 American Control Conference, June 2009, St. Louis, Missouri

2. V. Djapic, J. A. Farrell, W. Dong, “Hybrid Control Design Applied in Land Vehicle Behavior Based Switching Controller,” The 2008 IEEE Multi-conference on Systems and Control, Sep. 2008, San Antonio, TX

3. V. Djapic, J. A. Farrell, W. Dong, “Land Vehicle Control Using Command Filtered Backstepping Approach,” The 2008 American Control Conference, Seattle, WA.

4. Djapic, V., Farrell, J., Miller, P., and R. Arrieta, “AUV Controls in Hull Search Environments,” The 8th Monterey International Symposium on Technology and the Mine Problem, Monterey, CA, May 6-8, 2008.

5. Djapic, V., Farrell, J., Miller, P., and R. Arrieta, “New developments in control and navigation of ship-hull inspection AUV,” 2008 NDIA Joint Undersea Warfare Technology Spring Conference, San Diego, CA, Apr. 28 - May 1, 2008

6. Djapic, V., Farrell, J., Miller, P., and R. Arrieta, “Design and initial in-water testing of advanced non-linear control algorithms onto an Unmanned Underwater Vehicle (UUV),” MTS/IEEE OCEANS 2007, Vancouver, BC, Canada, Sept. 29 - Oct. 4, 2007.

7. Djapic, V., Farrell, J., Miller, P., and R. Arrieta, “Implementation and testing

of advanced control and navigation onboard of ship-hull inspection AUV,” 2007 NDIA Joint Undersea Warfare Technology Fall Conference, Groton, CN, Sept. 10-13, 2007

8. Djapic, V., Farrell, J., Miller, P., and R. Arrieta, “Advanced non-linear control algorithms applied to design highly maneuverable Autonomous Underwater Vehicles (AUVs),” AUSI 15th Int. Symposium on Unmanned Untethered Submersible Technology (UUST), Durham, NH, Aug. 19 - 22, 2007.

The first, second, and third article is based on the work explained in Chapters 1 and 2. The presentation of the third article was recognized with the Best presentation award at the 2008 American Control Conference. The fourth, fifth, sixth, seventh, and eighth article is based on the work explained in Chapters 3 and 4.

6.3.3 Technical Reports

1. Miller, P.A., Farrell, J., Zhao, Y., and V. Djapic, “Autonomous Underwater Vehicle Navigation,” Technical Report 1968, Space and Naval Warfare Systems Center, San Diego, February 2007.

This article is based on the work explained in Chapter 3.

Bibliography

- [1] Ronald C. Arkin. Motor schema based navigation for a mobile robot: An approach to programming by behavior. In *Proceeding of the IEEE Conference on Robotics and Automation*, 1987.
- [2] R. M. Arrieta, B. Granger, and V. Djapic. Highlight the operational value of the adaptive mission planner with respect to searching ship hulls. Technical report, SPAWAR Systems Center, San Diego, 2004.
- [3] F. Baralli, B. Evans, E. Coiras, and A. Bellettini. Auv navigation for mcm operations. Technical report, NATO Undersea Research Centre, 2007.
- [4] M. R. Benjamin, H. Schmidt, and J. Leonard. *A Guide to the IvP Helm for Autonomous Vehicle Control*, December 2007.
- [5] M.R. Benjamin, D. Battle, D. Eickstedt, H. Schmidt, and A. Balasuriya. Autonomous control of an autonomous underwater vehicle towing a vector sensor array. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, 2007.
- [6] A. A. Bennett and J. J. Leonard. A behavior-based approach to adaptive feature detection and following with autonomous underwater vehicles. *IEEE JOURNAL OF OCEANIC ENGINEERING*, 2000.
- [7] R. A. Brooks. A robust layered control system for a mobile robot. *Journal of Robotics and Automation*, 1986.
- [8] B. d'Andrea Novel, G. Campion, and G. Bastin. Control of nonholonomic wheeled mobile robots by state feedback linearization. *Int. J. Robotics Research*, 14:543–559, 1995.
- [9] C. Canudas de Wit, H. Khenouf, C. Samson, and O.J. Sordalen. *Nonlinear control Design for Mobile Robots, Recent Trends in Mobile Robots*. World Scientific, Hackensack, NJ, 1993.
- [10] R. A. DeCarlo, M. S Branicky, S. Pettersson, and B. Lennartson. Perspectives and results on the stability and stabilizability of hybrid systems. In *Proceedings of the IEEE*, volume 88, 2000.

- [11] V. Djapic, J. A. Farrell, and W. Dong. Hybrid control design applied in land vehicle behavior based switching controller. In *Proceedings of the 2008 IEEE Multi-conference on Systems and Control*, 2008.
- [12] V. Djapic, J. A. Farrell, and W. Dong. Land vehicle control using command filtered backstepping approach. In *Proceedings of the 2008 American Control Conference*, 2008.
- [13] V. Djapic, J. A. Farrell, and W. Dong. Land vehicle control using command filtered backstepping approach. In *Proceedings of the American Control Conference*, 2008.
- [14] V. Djapic, J. A. Farrell, and W. Dong. Land vehicle control using command filtered backstepping approach. *Submitted to IEEE Transactions on Control Systems Technology Journal*, (available at <http://www.ee.ucr.edu/farrell/>), 2009.
- [15] D.P. Eickstedt, M.R. Benjamin, and J. Curcio. Behavior based adaptive control for autonomous oceanographic sampling. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, 2007.
- [16] J. Eker and J. Malmberg. Design and implementation of a hybrid control strategy. *IEEE Control Systems Magazine*, 19(4), 1999.
- [17] G. Walsh D. Elbury, S. Sastry, R. Murray, and J. Laumond. Stabilization of trajectories for systems with nonholonomic constraints. *IEEE Trans. Automatic Control*, 39:216–222, 1994.
- [18] P. Encarnacao and A. Pascoal. Combined trajectory tracking and path following for marine craft. In *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001.
- [19] J. A. Farrell, W. Li, S. Pang, and R. Arrieta. Chemical plume tracing experimental results with remus auv. In *Proceedings of the Oceans Conference*, 2003.
- [20] J. A. Farrell and S. Pang. Chemical plume source localization. *IEEE Trans. on Systems, Man, and Cybernetics*, 36, 2006.
- [21] J. A. Farrell, S. Pang, and W. Li. Chemical plume tracing via an autonomous underwater vehicle. *IEEE J. Oceanic Eng.*, 30, 2005.
- [22] J. A. Farrell, S. Pang, W. Li, and R. M. Arrieta. Moth-inspired chemical plume tracing on an autonomous underwater vehicle. *IEEE Trans. on Systems, Man, and Cybernetics*, 22, 2006.
- [23] J. A. Farrell, M. Polycarpou, and M. Sharma. Adaptive backstepping with magnitude, rate, and bandwidth constraints: Aircraft longitude control. In *Proceeding of the IEEE American Control Conference*, 2003.
- [24] J. A. Farrell, M. Polycarpou, and M. Sharma. Longitudinal flight path control using on-line function approximation. *AIAA Journal of Guidance, Control and Dynamics*, 28(6):885897, 2003.

- [25] J. A. Farrell, M. Polycarpou, and M. Sharma. On-line approximation based fixed-wing aircraft control. *AIAA Journal of Guidance, Control and Dynamics*, 28(6):10891102, 2005.
- [26] J. A. Farrell, M. Polycarpou, M. Sharma, and W. Dong. Command filtered backstepping. *IEEE Transactions on Automatic Control*, 2007.
- [27] J. A. Farrell, M. Polycarpou, M. Sharma, and W. Dong. Command filtered backstepping. In *Proceeding of the IEEE American Control Conference*, 2008.
- [28] J. A. Farrell and M. M. Polycarpou. *Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches*. John Wiley, Hoboken, NJ, 2006.
- [29] Jay A. Farrell, Shuo Pang, and Wei Li. Chemical plume tracing via an autonomous underwater vehicle. *IEEE JOURNAL OF OCEANIC ENGINEERING*, 2005.
- [30] R. Fierro and F. L. Lewis. Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics. In *Proceedings of the 34th IEEE Conference on Decision and Control*, 1995.
- [31] M. Fliess, J. Levine, P. Martin, and P. Rouchon. Design of trajectory stabilizing feedback for driftless flat systems. In *Proceedings of the European Control Conf.*, 1995.
- [32] M. Fliess, J. Levine, Ph. Martin, and P. Rouchon. Flatness and defect of nonlinear systems: introductory theory and examples. *International Journal of Control*, 61:13271361, 1995.
- [33] Thor I. Fossen. *Guidance and Control of Ocean Vehicles*. John Wiley, Chichester, England, 1994.
- [34] E. Freund and R. May. Nonlinear path control in automated vehicle guidance. *IEEE Transactions on Robotics and Automation*, 13:4940, 1997.
- [35] Davrazos G. and Koussoulas N.T. A review of stability results for switched and hybrid systems. In *Proceedings of the 9th Mediterranean Conference on Control and Automation*, 2001.
- [36] J. M. Godhavn. Nonlinear tracking of underactuated surface vessels. In *Proceedings of the 35th IEEE Conference on Decision and Control*, 1996.
- [37] Giovanni Indiveri. *Modelling and Identification of Underwater Robotic Systems*. PhD thesis, University of Genova, Italy, 1998.
- [38] Z. P. Jiang and H. Nijmeijer. Tracking control of mobile robots: a case study in backstepping. *Automatica*, 33:1393–1399, 1997.
- [39] Z. P. Jiang and H. Nijmeijer. A recursive technique for tracking control of nonholonomic systems in chained form. *IEEE Trans. Automatic Control*, 44:256–279, 1999.

- [40] Jacobs Keith. U.s. navy master plan for uuv development. Technical report, Naval Forces, 2004.
- [41] H. K. Khalil. *Nonlinear Systems*. Prentice-Hall, New Jersey, 3 edition, 2002.
- [42] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic. *Nonlinear and Adaptive Control Design*. John Wiley Sons, Inc, New York, 1995.
- [43] A. Levant. Higher order sliding modes, differentiation, and output feedback control. *International Journal of Control*, 76:924941, 2003.
- [44] T. Madani and A. Benallegue. Control of a quadrotor mini-helicopter via full state backstepping technique. In *Proceeding of the 45th Conference on Decision and Control*, 2006.
- [45] J. Malmberg. *Analysis and Design of Hybrid Control Systems*. PhD thesis, Lund Institute of Technology, 1998.
- [46] M. Mataric. A distributed model for mobile robot environment-learning and navigation. Master's thesis, MIT, 1990.
- [47] N. H. McClamroch, N. Harris, and I. Kolmanovski. Performance benefits of hybrid control design for linear and nonlinear systems. In *Proceedings of the IEEE*, 2000.
- [48] P. Miller, Y. Zhao, V. Djapic, and J. A. Farrell. Advanced navigation algorithms applied to autonomous underwater vehicles. In *15th International Symposium on Unmanned Untethered Submersible Technology*, 2007.
- [49] P. A. Miller, J. Farrell, Y. Zhao, and V. Djapic. Autonomous underwater vehicle navigation. Technical Report 1968, SSC San Diego, February 2008.
- [50] Branicky M.S. Stability of switched and hybrid systems. In *Proceedings of the IEEE Conf. Decision and Control*, 1994.
- [51] Branicky M.S. Multiple lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans. on Automatic Control*, 43(4), 1998.
- [52] Paul Newman. Introduction to programming with moos. Technical report, Oxford University, 2007.
- [53] Shuo Pang. *Reactive Planning and On-line Mapping for Chemical Plume Tracing*. PhD thesis, UC Riverside, 2004.
- [54] Fierro R. and Lewis F.L. A framework for hybrid control design. *IEEE Trans. on Systems, Man, and Cybernetics*, 27, 1997.
- [55] Jukka Riekkii. *Reactive Task Execution of a Mobile Robot*. PhD thesis, Oulu University, 1999.

- [56] J. K. Rosenblatt, S. B. Williams, and H. Durrant-Whyte. Behavior-based control for autonomous underwater exploration. *International Journal of Information Sciences*, 2002.
- [57] M. Sharma and A. J. Calise. Adaptive backstepping control for a class of nonlinear systems via multilayered neural networks. In *Proceeding of the IEEE American Control Conference*, 2002.
- [58] D.-H. Shin and Y. Kim. Reconfigurable flight control system design using adaptive neural networks. *IEEE Trans. on Control Systems Technology*, 12(1):87100, 2004.
- [59] D. Soetanto, L. Lapiere, and A. Pascoal. Adaptive, non-singular path-following control of dynamic wheeled robots. In *Proceedings of the 42th IEEE Conference on Decision and Control*, 2003.
- [60] L. Sonneveldt, Q. Chu, and J. Mulder. Constrained adaptive backstepping flight control: Application to a nonlinear f-16/matv model. In *Proceeding of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006.
- [61] C. von Alt. Remus 100 transportable mine countermeasure package. In *Proceedings of the Oceans Conference*, 2003.
- [62] Rick Vosburgh. Reacquire identify and localize swimmers. Technical report, Rekton Research LLC, 2008.
- [63] R. W. Brockett, R. S. Millman, and H. J. Sussman. *Differential Geometric Control Theory*. Birkhauser, Boston, 1983.
- [64] S. B. Williams, P. Newman, G. Dissanayake, J. K. Rosenblatt, and H. Durrant-Whyte. A decoupled, distributed auv control architecture. In *Proceedings of the 31st International Symposium on Robotics*, 2000.

Appendix A

Command filter

The purpose of this appendix is to provide an example and discussion of a command filter. Advanced control approaches often assume the availability of a continuous and bounded desired trajectory $x_c(t)$ and its first r derivatives $x_c^{(r)}(t)$. The first time that this assumption is encountered it may seem unreasonable, since a user will often only specify a command signal $x_c^o(t)$. However, this assumption can always be satisfied by passing the commanded signal $x_c^o(t)$ through a single-input, multi-output prefilter.

The motivation of command filtering is therefore to determine the signals $x_c(t)$ and $\dot{x}_c(t)$ with $|x_c^o(t) - x_c(t)|$ being small, without having to analytically or numerically differentiate x_c^o . The effects of command filtering on the backstepping stability analysis are analyzed in [28, 26, 27]. The summary of that analysis is that for a properly designed command filter (unity DC gain to the first output which is the integral of the second output) the closed-loop command filtered implementation of the backstepping controller will be stable and the tracking error will be $O\left(\frac{1}{\omega_n}\right)$ where ω_n is the bandwidth of the

command filter. Therefore, the effect of command filtering on tracking error can be made arbitrarily small by increasing the parameter ω_n . The choice of ω_n is not dependent on the actuator bandwidth.

The state space implementation of such a filter is

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -2\zeta\omega_n x_2 - \omega_n^2 (x_1 - x_c^o)\end{aligned}$$

where $x_c = x_1$ and $\dot{x}_c = x_2$. Note that if x_c^o is bounded, then x_c and \dot{x}_c are bounded and continuous. The transfer function from x_c^o to x_c is

$$\frac{X_c(s)}{X_c^o(s)} = H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (\text{A.1})$$

which has a unity gain at low frequencies, damping ratio ζ and undamped natural frequency ω_n . The error $|x_c^o(t) - x_c(t)|$ is small if the bandwidth of $x_c^o(s)$ is less than the bandwidth of $H(s)$. If the bandwidth of x_c^o is known and the goal of the filter is to generate x_c and its derivative with $|x_c^o - x_c|$ small, then the designer simply chooses ω_n sufficiently large.

Note that the signal \dot{x}_c is computed by integration, not differentiation. This helps to decrease the effects of measurement noise; nonetheless, noise will impose a tradeoff in how large of a value can be selected for ω_n .

Remark 3 *The command filtering can be used to address the physical limitation of AUV actuators. CFBS approach directly accommodates magnitude and rate constraints on the robot states. Filter that generates the command and command derivative while enforcing magnitude, bandwidth, and rate limit constraints is shown in Figure A.1.*

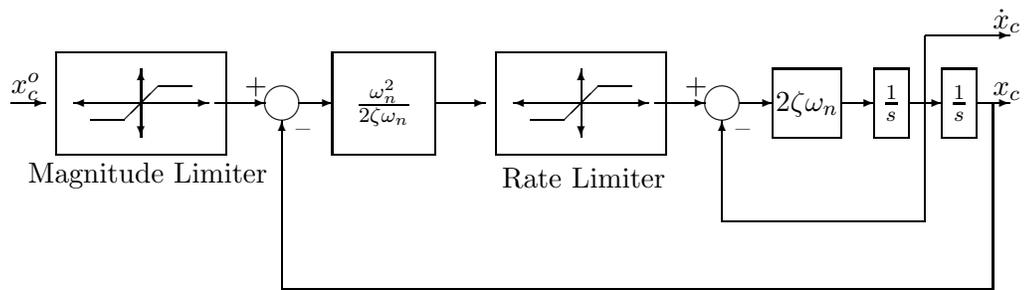


Figure A.1: Command Filter

Appendix B

Derivation of CFBS Terms for Mode 1 3D translation

B.1 Derivation of Eqn. (4.34)

The following equations will be used in the manipulation of the \bar{u}_n , \bar{u}_e and \bar{u}_d terms

1.

$$\sin(x \pm y) = \sin x \cos y \pm \cos x \sin y \quad (\text{B.1})$$

2.

$$\cos(x \pm y) = \cos x \cos y \mp \sin x \sin y \quad (\text{B.2})$$

3.

$$\psi = \bar{\psi} + \psi_c \quad (\text{B.3})$$

B.1.1 \bar{u}_n Error Analysis

We first express the $\bar{u}_n = u - u_{n_c}$ term using the eqns. (4.25), (4.26), (4.30), and (4.31).

$$\begin{aligned}\bar{u}_n = \mathbf{T}_n v - \mathbf{T}_{n_c} v_c &= [c\theta c\psi u - c\theta c\psi_c u_c] + [(c\phi s\psi_c - c\phi s\psi + s\phi s\theta c\psi - s\phi s\theta c\psi_c)v] \\ &+ [s\phi s\psi w - s\phi s\psi_c w_c + c\phi s\theta c\psi w - c\phi s\theta c\psi_c w_c].\end{aligned}\quad (\text{B.4})$$

The first term of eqn. (B.4) can be manipulated by applying the trigonometric identity, eqn. (B.2) combined with eqn. (B.3) with $x = \bar{\psi}$ and $y = \psi_c$, and adding and subtracting the term $c\psi_c u$. The result is

$$\begin{aligned}c\theta(c\psi u - c\psi_c u_c) &= c\theta\{c\psi u - c\psi_c u_c + c\psi_c u - c\psi_c u\} \\ &= c\theta\{c\psi_c u - c\psi_c u_c + c\psi u - c\psi_c u\} \\ &= c\theta\{c\psi_c \bar{u} + u(c\psi - c\psi_c)\} \\ &= c\theta\{c\psi_c \bar{u} + u(c\tilde{\psi}c\psi_c - s\tilde{\psi}s\psi_c - c\psi_c)\} \\ &= c\theta\{c\psi_c \bar{u} + u[c\psi_c(c\tilde{\psi} - 1) - s\tilde{\psi}s\psi_c]\} \\ &= c\theta\left\{c\psi_c \bar{u} + u \begin{bmatrix} c\psi_c & -s\psi_c \end{bmatrix} \begin{bmatrix} \frac{c\tilde{\psi}-1}{\psi} \\ \frac{s\tilde{\psi}}{\psi} \end{bmatrix} \bar{\psi}\right\}.\end{aligned}\quad (\text{B.5})$$

The second term of eqn. (B.4) can be manipulated by applying both trigonometric identities,

eqns. (B.1), (B.2) combined with the eqn. (B.3) with $x = \bar{\psi}$ and $y = \psi_c$. The result is

$$\begin{aligned}
c\phi(s\psi_c - s\psi)v + s\phi s\theta(c\psi - c\psi_c)v &= c\phi(s\psi_c - s\tilde{\psi}c\psi_c - c\tilde{\psi}s\psi_c)v + s\phi s\theta(c\tilde{\psi}c\psi_c - s\tilde{\psi}s\psi_c - c\psi_c)v \\
&= c\phi[-s\psi_c(c\tilde{\psi} - 1) - s\tilde{\psi}c\psi_c]v + s\phi s\theta[c\psi_c(c\tilde{\psi} - 1) - s\tilde{\psi}s\psi_c]v \\
&= c\phi v \begin{bmatrix} -s\psi_c & -c\psi_c \end{bmatrix} \begin{bmatrix} \frac{c\tilde{\psi}-1}{\psi} \\ \frac{s\tilde{\psi}}{\psi} \end{bmatrix} \bar{\psi} \\
+ s\phi s\theta v \begin{bmatrix} c\psi_c & -s\psi_c \end{bmatrix} \begin{bmatrix} \frac{c\tilde{\psi}-1}{\psi} \\ \frac{s\tilde{\psi}}{\psi} \end{bmatrix} \bar{\psi}. & \tag{B.6}
\end{aligned}$$

The third term of eqn. (B.4) can be first split into two terms for easier analysis. The first

out of the two terms can be manipulated by applying the trigonometric identity, eqn. (B.1)

combined with the eqn. (B.3) with $x = \bar{\psi}$ and $y = \psi_c$, and adding and subtracting the term

$s\psi_c w$. The result is

$$\begin{aligned}
s\phi(s\psi w - s\psi_c w_c) &= s\phi\{s\psi w - s\psi_c w_c + s\psi_c w - s\psi_c w\} \\
&= s\phi\{s\psi_c w - s\psi_c w_c + s\psi w - s\psi_c w\} \\
&= s\phi\{s\psi_c \bar{w} + w(s\psi - s\psi_c)\} \\
&= s\phi\{s\psi_c \bar{w} + w(s\tilde{\psi}c\psi_c + c\tilde{\psi}s\psi_c - s\psi_c)\} \\
&= s\phi\{s\psi_c \bar{w} + w[s\psi_c(c\tilde{\psi} - 1) + s\tilde{\psi}c\psi_c]\} \\
&= s\phi \left\{ s\psi_c \bar{w} + w \begin{bmatrix} s\psi_c & c\psi_c \end{bmatrix} \begin{bmatrix} \frac{c\tilde{\psi}-1}{\psi} \\ \frac{s\tilde{\psi}}{\psi} \end{bmatrix} \bar{\psi} \right\}. \tag{B.7}
\end{aligned}$$

The second of the two terms is manipulated similarly to $c\theta(c\psi u - c\psi_c u_c)$ term above. The result is

$$c\phi s\theta(c\psi w - c\psi_c w_c) = c\phi s\theta \left\{ c\psi_c \bar{w} + w \begin{bmatrix} c\psi_c & -s\psi_c \end{bmatrix} \begin{bmatrix} \frac{c\bar{\psi}-1}{\bar{\psi}} \\ \frac{s\bar{\psi}}{\bar{\psi}} \end{bmatrix} \bar{\psi} \right\}. \quad (\text{B.8})$$

Finally, we add the eqns. (B.5), (B.6), (B.7), (B.8) to get

$$\bar{u}_n = c\theta c\psi_c \bar{u} + (s\phi s\psi_c + c\phi s\theta c\psi_c) \bar{w} + \begin{bmatrix} C_{n_1} & C_{n_2} \end{bmatrix} \begin{bmatrix} \frac{c\bar{\psi}-1}{\bar{\psi}} \\ \frac{s\bar{\psi}}{\bar{\psi}} \end{bmatrix} \bar{\psi}, \quad (\text{B.9})$$

where

$$C_{n_1} = (-c\phi v + s\phi w)s\psi_c + (c\theta u + s\phi s\theta v + c\phi s\theta w)c\psi_c = \left. \frac{\partial \dot{e}}{\partial \psi} \right|_{\psi_c} \quad (\text{B.10})$$

$$C_{n_2} = (-c\theta u - s\phi s\theta v - c\phi s\theta w)s\psi_c + (-c\phi v + s\phi w)c\psi_c = \left. \frac{\partial \dot{n}}{\partial \psi} \right|_{\psi_c} \quad (\text{B.11})$$

which will be used in Sections 4.3.5 and 4.4.4 to derive the signals u_{bs} , w_{bs} and ψ_{bs} .

B.1.2 \bar{u}_e Error Analysis

We first express the $\bar{u}_e = u - u_{e_c}$ term using the eqns. (4.25), (4.27), (4.30), and (4.31).

$$\begin{aligned} \bar{u}_e = \mathbf{T}_e v - \mathbf{T}_{e_c} v_c &= [c\theta s\psi u - c\theta s\psi_c u_c] + [(c\phi c\psi - c\phi c\psi_c + s\phi s\theta s\psi - s\phi s\theta s\psi_c)v] \\ &\quad + [s\phi c\psi_c w_c - s\phi c\psi w + c\phi s\theta s\psi w - c\phi s\theta s\psi_c w_c]. \end{aligned} \quad (\text{B.12})$$

The first term of eqn. (B.12) is manipulated similarly to $s\phi(s\psi w - s\psi_c w_c)$ term above. The result is

$$c\theta(s\psi u - s\psi_c u_c) = c\theta \left\{ s\psi_c \bar{u} + u \begin{bmatrix} s\psi_c & c\psi_c \end{bmatrix} \begin{bmatrix} \frac{c\bar{\psi}-1}{\bar{\psi}} \\ \frac{s\bar{\psi}}{\bar{\psi}} \end{bmatrix} \bar{\psi} \right\}. \quad (\text{B.13})$$

The second term of eqn. (B.12) is manipulated similarly to $c\phi(s\psi_c - s\psi)v + s\theta s\phi(c\psi - c\psi_c)v$

term above. The result is

$$\begin{aligned} c\phi(c\psi - c\psi_c)v + s\phi s\theta(s\psi - s\psi_c)v &= c\phi v \begin{bmatrix} c\psi_c & -s\psi_c \end{bmatrix} \begin{bmatrix} \frac{c\tilde{\psi}-1}{\psi} \\ \frac{s\tilde{\psi}}{\psi} \end{bmatrix} \bar{\psi} \\ &+ s\phi s\theta v \begin{bmatrix} s\psi_c & c\psi_c \end{bmatrix} \begin{bmatrix} \frac{c\tilde{\psi}-1}{\psi} \\ \frac{s\tilde{\psi}}{\psi} \end{bmatrix} \bar{\psi}. \end{aligned} \quad (\text{B.14})$$

The third term of eqn. (B.12) is again split is manipulated similarly to $c\phi s\theta(c\psi w - c\psi_c w_c)$

term above. The result is

$$s\phi(c\psi_c w_c - c\psi w) = s\phi \left\{ -c\psi_c \bar{w} + w \begin{bmatrix} -c\psi_c & s\psi_c \end{bmatrix} \begin{bmatrix} \frac{c\tilde{\psi}-1}{\psi} \\ \frac{s\tilde{\psi}}{\psi} \end{bmatrix} \bar{\psi} \right\}. \quad (\text{B.15})$$

The second of the two terms is manipulated similarly to $s\phi(s\psi w - s\psi_c w_c)$ term above. The

result is

$$c\phi s\theta(s\psi w - s\psi_c w_c) = c\phi s\theta \left\{ s\psi_c \bar{w} + w \begin{bmatrix} s\psi_c & c\psi_c \end{bmatrix} \begin{bmatrix} \frac{c\tilde{\psi}-1}{\psi} \\ \frac{s\tilde{\psi}}{\psi} \end{bmatrix} \bar{\psi} \right\}. \quad (\text{B.16})$$

Finally, we add the eqns. (B.13), (B.14), (B.15), (B.16) to get

$$\bar{u}_e = c\theta s\psi_c \bar{u} + (-s\phi c\psi_c + c\phi s\theta s\psi_c) \bar{w} + \begin{bmatrix} C_{e1} & C_{e2} \end{bmatrix} \begin{bmatrix} \frac{c\tilde{\psi}-1}{\psi} \\ \frac{s\tilde{\psi}}{\psi} \end{bmatrix} \bar{\psi}, \quad (\text{B.17})$$

where

$$C_{e1} = (c\theta u + s\phi s\theta v + c\phi s\theta w) s\psi_c + (c\phi v - s\phi w) c\psi_c = - \left. \frac{\partial \dot{n}}{\partial \psi} \right|_{\psi_c} \quad (\text{B.18})$$

$$C_{e2} = (-c\phi v + s\phi w) s\psi_c + (c\theta u + s\phi s\theta v + c\phi s\theta w) c\psi_c = \left. \frac{\partial \dot{e}}{\partial \psi} \right|_{\psi_c} \quad (\text{B.19})$$

which will be used in Sections 4.3.5 and 4.4.4 to derive the signals u_{bs} , w_{bs} and

ψ_{bs} .

B.1.3 \bar{u}_d Error Analysis

This term is expressed as

$$\bar{u}_d = u - u_{d_c} = \mathbf{T}_d v - \mathbf{T}_{d_c} v_c = (s\theta u_c - s\theta u) + (c\phi c\theta w - c\phi c\theta w_c) = -s\theta \bar{u} + c\phi c\theta \bar{w}. \quad (\text{B.20})$$

B.1.4 \mathbf{A} , \mathbf{B} , \mathbf{C} , and $\mathbf{g}(\bar{\psi})$ Definitions

Therefore,

$$\mathbf{A} = \begin{bmatrix} c\theta c\psi_c \\ c\theta s\psi_c \\ -s\theta \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} s\phi s\psi_c + c\phi s\theta c\psi_c \\ -s\phi c\psi_c + c\phi s\theta s\psi_c \\ c\phi c\theta \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} C_{n_1} & C_{n_2} \\ C_{e_1} & C_{e_2} \\ 0 & 0 \end{bmatrix}, \quad \text{and } \mathbf{g}(\bar{\psi}) = \begin{bmatrix} \frac{c\bar{\psi}-1}{\bar{\psi}} \\ \frac{s\bar{\psi}}{\bar{\psi}} \end{bmatrix} \quad (\text{B.21})$$

It is important to note that $\lim_{\psi \rightarrow 0} \mathbf{g}(\bar{\psi}) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

Appendix C

Lyapunov Equation

The purpose of this appendix is to show that by choosing a specific symmetric and positive definite matrix \mathbf{P} we get a symmetric and positive semidefinite matrix \mathbf{Q} that is useful in the proof of eqn. (2.21).

We can write the Lyapunov Equation as

$$\mathbf{A}^\top \mathbf{P} + \mathbf{P} \mathbf{A} = -\mathbf{Q}.$$

Since

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -K_2 & -K_1 \end{bmatrix},$$

and

$$\mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

if we choose

$$\mathbf{P} = \begin{bmatrix} p_1 & 0 \\ 0 & p_2 \end{bmatrix},$$

where $K_1 = K_u$ or K_r , $K_2 = K_u^i$ or K_r^i , p_1 , and p_2 are positive constants, we have

$$-\mathbf{Q} = \begin{bmatrix} 0 & p_1 - K_2 p_2 \\ p_1 - K_2 p_2 & -2K_1 p_2 \end{bmatrix}.$$

We can choose $p_1 = K_2 p_2$, then $p_1 - K_2 p_2 = 0$. Therefore,

$$\mathbf{P} = \begin{bmatrix} K_2 p_2 & 0 \\ 0 & p_2 \end{bmatrix},$$

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 \\ 0 & 2K_1 p_2 \end{bmatrix},$$

and

$$\mathbf{PB} = \begin{bmatrix} 0 \\ p_2 \end{bmatrix},$$

which will be used in the derivation of eqn. (2.22).