

NONLINEAR CONTROL OF ROBOTIC FISH

By

Maria L. Castaño

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering – Doctor of Philosophy

2021

ABSTRACT

NONLINEAR CONTROL OF ROBOTIC FISH

By

Maria L. Castaño

In the past few decades, robots that propel and maneuver themselves like fish, known as robotic fish, have received substantial attention due to their efficiency, maneuverability, and lifelike features. Their agile locomotion can be partially attributed to their bio-inspired propulsion methods, which range from tail (caudal) and dorsal to paired pectoral fins. While these characteristics make robotic fish an attractive choice for a myriad of aquatic applications, their highly nonlinear, often under-actuated dynamics and actuator constraints present significant challenges in control design. The goal of this dissertation is to develop systematic model-based control approaches that guarantee closed-loop system stability, accommodate input constraints, and are computationally viable for robotic fish.

We first propose a nonlinear model predictive control (NMPC) approach for path-following of a tail-actuated robotic fish, where the control design is based on an averaged dynamic model. The bias and the amplitude of the tail oscillation are treated as physical variables to be manipulated and are related to the control inputs via a nonlinear map. A control projection method is introduced to accommodate the inputs constraints while minimizing the optimization complexity in solving the NMPC problem. Both simulation and experimental results on a tail-actuated robotic fish support the efficacy of the proposed approach and its advantages over alternative approaches.

Although NMPC is a promising candidate for tracking control, its computational complexity poses significant challenges in its implementation on resource-constrained robotic fish. We thus propose a backstepping-based trajectory tracking control scheme that is computationally inexpensive and guarantees closed-loop stability. We demonstrate how the control scheme can be synthesized to handle input constraints and establish via singular perturbation analysis the ultimate boundedness of three tracking errors (2D-position and orientation) despite the under-actuated nature of the robot.

The effectiveness of this approach is supported by both simulation and experimental results on a tail-actuated robotic fish.

We then turn our attention to pectoral fin-actuated robotic fish. Despite its benefits in achieving agile maneuvering at low swimming speeds, the range constraint of pectoral fin movement presents challenges in control. To overcome these challenges, we propose two different backstepping-based control approaches to achieve trajectory tracking and quick-maneuvering control, respectively. We first propose a scaling-based approach to develop a control-affine nonlinear dynamic average model for a pectoral fin-actuated robotic fish, which is validated via both simulation and experiments. The utility of the developed average dynamic model is then demonstrated via the synthesis of a dual-loop backstepping-based trajectory tracking controller.

Cyclic actuation can often limit precise manipulation of the fin movements and the full exploitation of the maneuverability of pectoral fin-actuated robotic fish. To achieve quick velocity maneuvering control, we propose a dual-loop control approach composed of a backstepping-based controller in the outer loop and a fin movement-planning algorithm in the inner loop. Simulation results are presented to demonstrate the performance of the proposed scheme via comparison with a nonlinear model predictive controller.

Copyright by
MARIA L. CASTAÑO
2021

*To my beloved parents, Mary and Luis Javier, for their love, support and encouragement.
To the loving memory of my brother, Wilson Eduardo.*

ACKNOWLEDGEMENTS

Throughout my Ph.D. and academic journey, I have received a great deal of support and assistance from many people.

First, I would like to sincerely thank my advisor Dr. Xiaobo Tan for his continuous support, patience, motivation, and guidance. I feel very honored to have had the opportunity to do my graduate research under his tutelage and will forever be thankful for his patient guidance throughout my Ph.D. journey.

I would also like to thank the rest of my thesis committee: Dr. Hassan Khalil, Dr. Guoming Zhu, Dr. Vaibhav Srivastava, and Dr. Lixin Dong, for kindly joining my advisory committee and for their insightful comments, suggestions, and advice.

I would like to express my gratitude to the technical and administrative staff of the Electrical and Computer Engineering Department for their assistance throughout the years.

My deepest gratitude to John Thon for his fabrication expertise, especially his help in ensuring that the robots were waterproof. I am incredibly grateful for his willingness to go far and beyond to help me during times of urgency and duress.

I am grateful to my friends at Michigan State University and my fellow labmates at Smart Microsystems Lab for their support, friendship, memories, and all the fun we have had in the last few years. Special thanks to Dr. Pratap Bhanu Solanki and Dr. Jason Greenberg. Thank you for the unforgettable memories of late-night study and research sessions.

I also want to acknowledge the funding support from my National Science Foundation Graduate Research Program fellowship (DGE1424871), the Graduate School, and other National Science Foundation grants (IIP 1343413, IIS 1319602, CCF 1331852, ECCS 1446793, IIS 1715714, IIS 1848945) that made this work possible.

I would like to thank all my uncles, aunts, cousins, and close friends for their love and support. A special thanks to my aunts Patricia and Delia and my cousin Jorge Humberto who have supported me during different transitional and challenging periods of my life. Many thanks to my high school

best friend, Shawny Espinoza, for her emotional support, love, and motivation all these years.

I am foremost and forever thankful to my source of inspiration, my parents Mary and Luis Javier, for their unconditional love, emotional and financial support, motivation, and encouragement. It is needless to say that without them, none of my accomplishments would be possible. Words can never fully express my appreciation for everything they have done and sacrificed for me. Thank you for being brave and for taking hard steps to seek better opportunities for our future. I love you both. Finally, I want to praise and thank God, who has granted me countless blessings, knowledge, health, and opportunity throughout my life.

TABLE OF CONTENTS

LIST OF TABLES	xi
LIST OF FIGURES	xii
CHAPTER 1 INTRODUCTION	1
1.1 Control of Robotic Fish	2
1.1.1 Control of Body and Caudal Fin (BCF)-actuated Robotic Fish	2
1.1.2 Control of Pectoral Fin-actuated Robotic Fish	3
1.2 Overview of Contributions	6
1.2.1 Model Predictive Control-based Path Following for Tail-actuated Robotic Fish	6
1.2.2 Backstepping Control-based Trajectory Tracking for Tail-actuated Robotic Fish	7
1.2.3 Averaged Modeling and Trajectory Tracking Control of Pectoral Fin-actuated Robotic Fish	9
1.2.4 Rapid Maneuvering Control of Pectoral Fin-actuated Robotic Fish	10
CHAPTER 2 MODEL PREDICTIVE CONTROL-BASED PATH FOLLOWING FOR TAIL-ACTUATED ROBOTIC FISH	12
2.1 Tail-actuated Robotic Fish Model	12
2.1.1 Dynamic Model	12
2.2 Path-following Control Algorithm	16
2.2.1 Path-following Error Coordinates	16
2.2.2 Path-following Control Design	19
2.2.3 Control Projection	21
2.3 Simulation Results	25
2.4 Experimental Results	29
2.4.1 Model Parameter Identification	32
2.4.2 Experimental Results on Path-Following	35
2.5 Chapter Summary	39
CHAPTER 3 BACKSTEPPING CONTROL-BASED TRAJECTORY TRACKING FOR TAIL-ACTUATED ROBOTIC FISH	40
3.1 System Model	42
3.2 Trajectory Tracking Control Algorithm	44
3.2.1 Trajectory Tracking Error Coordinates	44
3.2.2 Backstepping Control Design	45
3.3 Stability Analysis of the Closed-Loop System	53
3.3.1 Asymptotic Stability of Multi-Time-Scale Systems: The Setup	54
3.3.2 Lyapunov Function Candidates and Two-Time-Scale Analysis	59
3.3.2.1 Asymptotic Stability of the Closed-Loop Perturbed System	61
3.4 Backstepping Control of Robotic Fish	63

3.4.1	Dynamic Model of Robotic Fish	63
3.4.2	Backstepping Control Design	65
3.4.2.1	Control Synthesis	66
3.4.2.2	Control Synthesis Incorporating Input Constraints	68
3.4.3	Stability Analysis of Closed-Loop Robotic Fish System	71
3.5	Simulation Results	72
3.6	Experimental Results	76
3.7	Chapter Summary	78
 CHAPTER 4 AVERAGED MODELING AND TRACKING CONTROL OF PECTORAL FIN-ACTUATED ROBOTIC FISH		
		82
4.1	Dynamic Model For Pectoral Fin-actuated Robotic Fish	84
4.1.1	Rigid Body Dynamics	84
4.1.2	Drag and Lift on the Robot Body	86
4.1.3	Hydrodynamic Forces from Rowing Pectoral Fins	86
4.2	Averaging with Scaled Forcing	89
4.2.1	Averaged Model	89
4.3	Simulation and Experimental Model Validation	91
4.3.1	Experimental Setup	92
4.3.2	Original Dynamics Parameter Identification	92
4.3.3	Identification of Scaling Functions	96
4.3.4	Validation of the Averaged Model	99
4.4	Average Dynamic Model-based Trajectory Tracking Control	101
4.4.1	Trajectory Tracking Problem	101
4.4.2	Trajectory Tracking Control Algorithm	102
4.5	Trajectory Tracking Simulation Results	107
4.6	Conclusion	109
 CHAPTER 5 RAPID MANEUVERING CONTROL OF PECTORAL FIN-ACTUATED ROBOTIC FISH		
		110
5.1	Dynamic Model of Pectoral Fin-actuated Robotic Fish	111
5.2	Dual-Loop Fin Control Scheme	112
5.2.1	Velocity Tracking Problem	112
5.2.2	Velocity Tracking Control Algorithm	113
5.2.3	Outer-loop Tracking Control Synthesis	113
5.2.4	Pectoral Fin Control Algorithm	115
5.3	Simulation Results	119
5.4	Chapter Summary	123
 CHAPTER 6 SUMMARY AND FUTURE WORK		
		124
6.1	Summary	124
6.2	Future Work	126
 APPENDICES		
		128
APPENDIX A	PROOF OF THE ROBOTIC FISH'S INPUT-TO-STATE STABILITY	129

APPENDIX B	PROOF OF STABILITY OF THE PERTURBATION-FREE ROBOTIC FISH CLOSED-LOOP SYSTEM	133
APPENDIX C	PROOF OF STABILITY OF THE ROBOTIC FISH'S PERTURBED CLOSED-LOOP SYSTEM	138
BIBLIOGRAPHY		142

LIST OF TABLES

Table 2.1: Identified parameters for the robotic fish used in this work.	32
Table 2.2: Model validation results: relative model prediction error for turning radius and turning period, when the tail beats at 1.5 Hz.	35
Table 3.1: Comparison functions and inequalities that guarantee the asymptotic stability requirements for the two-time scale system.	58
Table 3.2: Parameters of the robotic fish.	76
Table 4.1: Identified model parameters for the robotic fish.	95
Table 4.2: Model validation results: relative model prediction error for turning radius and turning period.	96
Table 4.3: Model validation results: surge velocities predicted by the original model and measured from experiments, and their relative error.	96
Table 4.4: Averaged model validation results: relative model prediction error between the original and averaged models for turning radius and turning period.	99

LIST OF FIGURES

Figure 1.1: Examples of reported robotic fish prototypes.	1
Figure 1.2: The main types of pectoral fin motion (adapted from [1]). The rotation axes for the rowing, feathering, and flapping motions are vertical, transverse, and longitudinal, respectively.	4
Figure 2.1: Top view of the tail-actuated robotic fish undergoing planar motion.	13
Figure 2.2: Illustration of the path-following problem for robotic fish.	16
Figure 2.3: Illustration of the admissible input control set for robotic fish.	22
Figure 2.4: Illustration of a rectangular section chosen from the admissible control set.	23
Figure 2.5: Boxed section encompassing the admissible control set.	23
Figure 2.6: Simulation: line-tracking results for NMPC with the proposed control projection method.	27
Figure 2.7: Simulation: line-tracking results for NMPC with boxed constraints inside the admissible sector-shaped control region.	27
Figure 2.8: Simulation: line-tracking results for NMPC with direct nonlinear control constraints.	28
Figure 2.9: Simulation: arc-tracking results for NMPC with the proposed control projection method.	28
Figure 2.10: Simulation: arc-tracking results for NMPC with boxed constraints inside the admissible sector-shaped control region.	29
Figure 2.11: Simulation: arc-tracking results for NMPC with nonlinear control constraints.	29
Figure 2.12: The experimental setup.	31
Figure 2.13: Tail-actuated robotic fish develop at Smart Microsystems Lab.	31
Figure 2.14: Experiments: line-tracking results for NMPC with the proposed control projection method.	37

Figure 2.15: Experiments: line-tracking results for NMPC with boxed constraints inside the admissible sector-shaped control region.	37
Figure 2.16: Experiments: computed NMPC physical inputs and their projected values. . . .	37
Figure 2.17: Experiments: arc-tracking results for NMPC with the proposed control projection method.	38
Figure 2.18: Experiments: arc-tracking results for NMPC with boxed constraints inside the admissible sector-shaped control region.	38
Figure 3.1: Top view of an aquatic robotic system undergoing planar motion.	42
Figure 3.2: The <i>bottom-up</i> approach to the analysis of two-time-scale system.	55
Figure 3.3: Simulation: line and circle-tracking trajectory results, respectively, for backstepping-based and PI control.	74
Figure 3.4: Simulation: line-tracking position error for backstepping-based and PI control. “STD” curves represent the standard deviation envelopes for the average error under each controller.	74
Figure 3.5: Simulation: circle-tracking position error for backstepping-based and PI control. “STD” curves represent the standard deviation envelopes for the average error under each controller.	75
Figure 3.6: Simulation: line-tracking angle error for backstepping-based and PI control. “STD” curves represent the standard deviation envelopes for the average error under each controller.	75
Figure 3.7: Simulation: circle-tracking angle error for backstepping-based and PI control. “STD” curves represent the standard deviation envelopes for the average error under each controller.	75
Figure 3.8: Experiments: line and circle-tracking trajectory results, respectively, for backstepping-based and PI control.	78
Figure 3.9: A snapshot of an experimental run for line and circle-tracking using backstepping-based (a) and PI control (b), respectively.	79

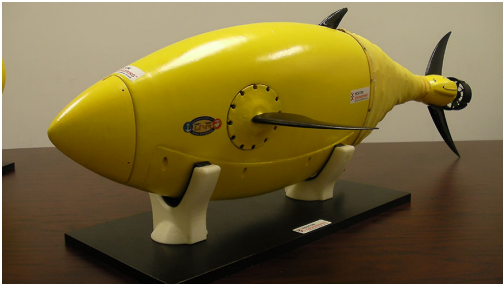
Figure 3.10: Experiments: line-tracking (3.10a) and circle-tracking (3.10b) averaged position error for backstepping-based control and PI control. The solid blue line with circular markers represents the averaged position error obtained for backstepping-based control, while the dashed blue lines represent its standard deviation. Similarly, the solid magenta line represents the averaged position error obtained for PI control, while the dotted-dashed magenta lines represent its standard deviation.	80
Figure 3.11: Experiments: line-tracking (3.11a) and circle-tracking (3.11b) averaged angle error for backstepping-based control and PI control. The solid blue line with circular markers represents the averaged angle error obtained for backstepping-based control, while the dashed blue lines represent its standard deviation. Similarly, the solid magenta line represents the averaged angle error obtained for PI control, while the dotted-dashed magenta lines represent its standard deviation.	81
Figure 4.1: (a) Top view of the pectoral fin-actuated robotic fish undergoing planar motion; (b) side view and blade element of the right pectoral fin with its parameters and variables; (c) top view of the pectoral fin with its associated forces and angles.	84
Figure 4.2: The experimental setup. During experiments the pectoral-fin actuated robotic fish swims within the enclosed area (denoted by the yellow lines) in the tank, and the overhead Logitech camera captures a video of the robot swimming. An image processing algorithm detects the red and blue markers placed on top of the robot to localize it and determine its heading.	93
Figure 4.3: Obtained scaling values $K_{f_x R}$, $K_{f_y R}$, $K_{m R}$ versus the amplitude $\gamma_{A R}$ and bias $\gamma_{0 R}$, for fixed actuation period $T_{P R} = 1$ s and ratio $\zeta = 2$	98
Figure 4.4: Circular trajectories captured in experiments and predicted by the original and average dynamic models. In this case the right fin remains still and the left fin undergoes actuation with the following parameters fixed: $\gamma_{A L} = 22^\circ$, $\gamma_{0 L} = 85^\circ$, $T_{P L} = 1$ and $\zeta = 4$	100
Figure 4.5: Comparison between experimental results and the predictions of the steady-state forward swimming speed based on the average model and the original dynamic model. In this study the left and right fins undergo symmetric actuation with the following parameters fixed: $\gamma_A = 15^\circ$, $\gamma_0 = 95^\circ$ and $\zeta = 4$	101
Figure 4.6: Illustration of the proposed dual-loop pectoral fin-actuated robotic fish control scheme. The green dashed line encompasses the outer loop trajectory tracking controller, while the inner red dotted and dashed line encompasses the fin parameter optimization algorithm.	103

Figure 4.7: Simulation: position, heading angle and velocity tracking for the line-tracking case.	108
Figure 4.8: Simulation: position, heading angle and velocity tracking for the circle-tracking case.	108
Figure 4.9: Simulation: desired input versus achievable inputs for the line tracking case. . .	109
Figure 5.1: Illustration of the proposed dual-loop pectoral fin-actuated robotic fish control scheme. The green dashed line encompasses the outer loop velocity tracking controller, while the inner red dotted and dashed line encompasses the fin movement planning and control algorithm.	112
Figure 5.2: Example of different possible fin acceleration, velocity and position trajectories generated within a planning interval T_P that is divided into three sub-intervals.	116
Figure 5.3: Example of weights for each value of $\ddot{\gamma}_{Ri}$ within a given allowed range. The vertical dotted red lines indicate the maximum and minimum values allowed for $\ddot{\gamma}_{Ri}$ in the i th interval, while the green solid line represents the desired value.	118
Figure 5.4: Simulation: velocity-tracking trajectories results for the proposed scheme (a) and the alternative scheme with NMPC for the inner loop controller (b).	121
Figure 5.5: Simulation: fin position trajectories resulting from the proposed scheme (a) and the alternative scheme with NMPC for the inner loop controller (b). The dashed blue lines depict the fin's position constraints.	122

CHAPTER 1

INTRODUCTION

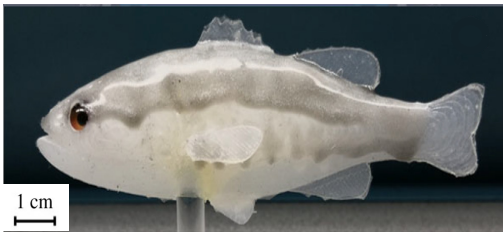
The efficiency, astounding agility, and remarkable feats in swimming of real fish have motivated and inspired scientific interest in developing a class of autonomous underwater vehicles called robotic fish. As illustrated in Figure 1.1, various bio-inspired actuation mechanisms, ranging from oscillating tail (caudal) and paired pectoral fins [2,3], a combination of both movements [4], to undulatory motion of the entire body [5], have been explored in the design of these robots. In addition to being promising platforms in applications such as aquatic environmental monitoring and search and rescue, robotic fish are also useful tools to study robot-animal interactions [6–9]. Their efficiency, maneuverability, and stealth are some of the characteristics that have made robotic fish an attractive choice over traditional propeller-driven underwater vehicles [10–12].



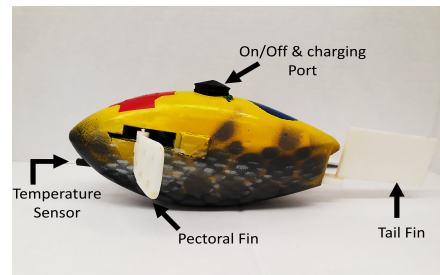
(a) Ghostswimmer, a robotic fish developed by Boston Engineering Corporation [13].



(b) Robotic fish developed by Delft University of Technology [14].



(c) A robotic fish developed at New York University [9].



(d) A robotic fish developed by the Smart Microsystems Lab at Michigan State University.

Figure 1.1: Examples of reported robotic fish prototypes.

1.1 Control of Robotic Fish

To be suitable for aquatic applications and effectively carry out monitoring and sensing tasks, accurate trajectory tracking is essential for robotic fish. For example, trajectory tracking is critical when exploring and sampling specific areas, seeking pollutant sources, and mapping the whole aquatic environment. However, input constraints, and highly nonlinear and often under-actuated dynamics of these robots present significant challenges in trajectory control.

Much of the work done on robotic fish has been focused on robot development, prototyping, [1, 5, 10, 15–31] and high-fidelity modeling [32–38]. There has also been extensive work on motion control of robotic fish, which has mainly been focused on the generation of coordinated movements of the actuation components to produce some fish-like swimming gaits. In the case of swimming gait generation, several kinematics and dynamics-based schemes [39–47], as well as bio-inspired approaches, such as central pattern generators [48–53], have been used to produce fish-like swimming. However, these approaches are typically open-loop in nature. Although some works have examined closed-loop approaches to drive the robot to achieve some desired motion [31, 54, 55], they have mainly been focused on heading or depth control.

1.1.1 Control of Body and Caudal Fin (BCF)-actuated Robotic Fish

There has been additional work done on model-based closed-loop motion control of body and caudal fin-actuated robotic fish to achieve maneuvering, speed and orientation control, path following, point-to-point tracking, and line-of-sight (LOS) control [56–63]. In [56] a point-to-point control of a four-link robotic fish with an oscillating tail was implemented, where a classical proportional-integral-derivative controller combined with a fuzzy logic controller was designed for speed and orientation control. The authors in [57] devised a control strategy using motion primitives to achieve maneuvering of an aquatic vehicle using an oscillating foil. The strategy consists of an optimal off-line primitive planning step and an online feedback control step composed of a cascade of finite-time, time-scalable linear quadratic control, and input-output linearization, in combination

with a sliding mode controller. Furthermore, in [58] a target-tracking and collision-avoidance algorithm for two autonomous robotic fish was implemented via a situated-behavior-based and motion primitive decentralized control approach, in combination with an attractive force toward a target and a repulsive force for collision avoidance. In [59] three simplified linearized models of the decoupled fish dynamics were used for the design of linear quadratic regulators to achieve speed and orientation control and to stabilize the pitch and roll. Furthermore, a line-of-sight guidance scheme was implemented for way-point tracking using linearized controllers. Zou et al. developed a neural-network-based sliding mode control algorithm for cooperative trajectory tracking of multiple robotic fish [60]. The authors in [61] designed a sliding mode controller for swimming, orientating, and LOS way-point tracking of robotic fish in three-dimensional motion. In [62], the authors proposed a target-tracking hybrid controller that consists of an open-loop turning controller and a closed-loop backstepping controller to drive a robotic fish to a specified target location. Finally, in [63], the authors proposed a backstepping controller for tracking control of a nonholonomic fish robot with dynamics that can be expressed in a chained form.

Despite the aforementioned progress in the control of robotic fish, most of the work focused on speed, orientation, and way-point tracking control. Furthermore, it has relied on utilizing either motion primitives, which can be system-specific, or linearized methods, which only provide local results (i.e., stability is only guaranteed in a neighborhood of the selected operating points). A unified, systematic control approach that accommodates input constraints and has stability guarantees is in general lacking for such robots.

1.1.2 Control of Pectoral Fin-actuated Robotic Fish

While caudal fins have proven to be an efficient propulsion mode at higher speeds, pectoral fins are vital in assisting propulsion and achieving agile maneuvering at low swimming speeds [25, 64, 65]. As shown in Figure 1.2, pectoral fin motions can generally be classified into three modes based on the axis of rotation: rowing, feathering, and flapping, where the axes of rotation are vertical, transverse, and longitudinal, respectively. Rowing motion is classified as a “drag-based” swimming

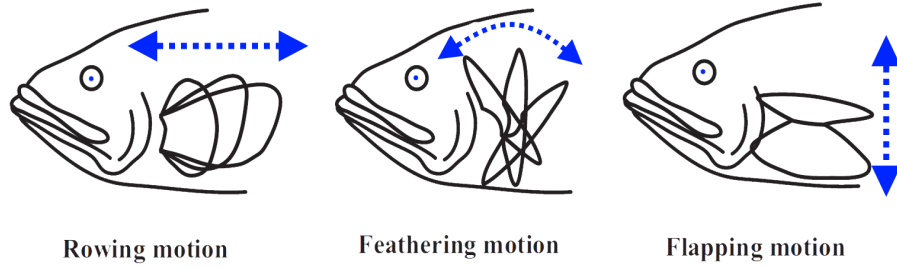


Figure 1.2: The main types of pectoral fin motion (adapted from [1]). The rotation axes for the rowing, feathering, and flapping motions are vertical, transverse, and longitudinal, respectively.

mechanism, where the drag element of fluid dynamics generates the thrust, and it is often regarded as an effective type of motion for achieving a number of in-plane locomotion and maneuvering tasks, such as forward swimming, side-way swimming, and turning [66,67]. The rowing motion of pectoral fins comprises two sub-movements during the fin-beat cycle: a power stroke and a recovery stroke. During the power stroke, the pectoral fin moves backward to produce thrust through induced drag on the pectoral fin surface, while during the recovery stroke, the fin moves toward the front of the body, ideally with minimal loading, to get ready for the next fin-beat cycle.

Although beneficial in maneuvering, utilizing rowing motion for propulsion gives rise to challenges for the control of robotic fish. The challenge lies with the actuation constraints (i.e., angular position, velocity, and acceleration limitations) and the mechanism in which the “drag-based” swimming method is used to generate thrust. For example, forward thrust can only be generated during the power stroke; however, when the pectoral fin reaches its maximum angular position, it has to recover in order to be able to generate forward thrust again. During the recovery phase, the fin will produce a “negative” thrust, thus opposing the objective of producing forward thrust in the first place. While such “backing-up” behavior seems natural to human understanding and is widely used by live fish, it is challenging to incorporate this behavior through systematic, rigorous controller synthesis.

Significant work has been done on the modeling of pectoral fin-actuated robotic fish [3,5,24–30, 68–70]. Some of these works have focused on developing Computational Fluid Dynamics (CFD) models to carry out numerical analysis of the robot’s hydrodynamic characteristics and the force

generation of the fins. Despite being instrumental in studying pectoral fins' propulsive mechanism, CFD-based models are not suitable for control design. Some efforts have also gone into developing analytical models to study the propulsive mechanism and gait analysis. For example, [71] and [72] utilized blade element theory (BET) to evaluate the quasi-static hydrodynamic forces generated by undulating and rowing fins, respectively, and [73] utilized Euler-Lagrange equation methods to develop a dynamic model for batoid swimming robots. Others have focused on developing modeling frameworks for analyzing the effects of different pectoral fin designs and materials on the robot's swimming performance and mechanical efficiency [2,25,26,74,75]; however, these models tend to be complex and are not amenable to controller design. Control-affine models that easily allow studying the effect of the input parameters on the dynamics and fins movement are essential for control design and are lacking for these types of robots.

Limited work has been reported on the control of pectoral fin-actuated robotic fish. Some of the work in this area has focused on open-loop motion control, i.e., the generation of coordinated movements of the actuation components to produce some fish-like swimming gaits [4, 76–78]. In terms of closed-loop control, several authors have proposed sensory-feedback Central Pattern Generators (CPGs) for target tracking or obstacle avoidance control [79, 80]. Similarly, in [81] the authors proposed a control strategy composed of two layers: an upper decision-making layer that uses a finite state machine to determine the proper swimming gait and a layer that uses a CPG to implement the desired gait. Fuzzy rule-based control laws were proposed in [82] to control fin-beat parameters to drive a robotic fish to perform rendezvous and docking in a three-dimensional workspace. The authors in [31] implemented geometric control methods for closed-loop depth control of a robotic fish using pectoral fins undergoing feathering motion. All of the aforementioned work is based on motion primitives, which can be system-specific, or have focused mainly on heading, depth, and velocity tracking for robotic fish using pectoral fins undergoing feathering or lead-lag motion. Systematic trajectory tracking control approaches that accommodate input constraints and have closed-loop stability guarantees have not been proposed for robotic fish with pectoral fins undergoing rowing motion.

The limited work on control of pectoral fin-actuated robotic fish utilized cyclic fin actuation for the lead-lag or feathering motion of the fins while treating the fin-beat parameters or the phase differences between the fins as the control variables for the fish locomotion control. However, such cyclic fin movement limits precise manipulation of the fin movements and the thrust profile that can be generated, which impedes full exploitation of the maneuverability of pectoral fin-actuated robotic fish. Acyclic thrust or moments could be more instrumental for generating a quick maneuvering response, which can be valuable in scenarios like counteracting disturbances or avoiding fast obstacles.

1.2 Overview of Contributions

The contributions of this dissertation reside in the development of systematic model-based control approaches that guarantee closed-loop system stability, accommodate input constraints, and are computationally viable for robotic fish. In particular, we present two nonlinear model-based control approaches for path-following and trajectory tracking of a tail-actuated robotic fish. Furthermore, to exploit the benefits of pectoral fins and overcome their challenges in actuation, we propose two different backstepping-based control approaches to achieve trajectory tracking and quick-maneuvering control, respectively. The contributions of this research are further summarized in the following subsections.

1.2.1 Model Predictive Control-based Path Following for Tail-actuated Robotic Fish

First, we propose a nonlinear model predictive control (NMPC) approach to path-following of a tail-actuated robotic fish that accommodates the nonlinear dynamics and actuation constraints while minimizing the control effort. Although there is extensive work on NMPC for path-following of mobile robots [83–88], little work has been reported on its application to control of robotic fish.

In this controller design, a high-fidelity averaged nonlinear dynamic model developed in [89] is used. The physical control inputs consist of two of the tail-beat parameters, the bias and the amplitude, while the angular frequency of the tail beat is kept constant. We propose a scheme to

address the nonlinear input constraints. Specifically, to maximize the use of the admissible control and handle the nonlinear control constraints in a computationally efficient manner, we employ an analytical projection scheme for the control inputs. We further propose a novel estimation scheme to identify some critical unknown parameters in the robotic fish model. In particular, inspired by the work in [89], we develop a parameter estimation method to empirically identify the hydrodynamic and scaling coefficients of the model instead of utilizing time-consuming CFD simulations, or relying on trial-and-error data-fitting between dynamic simulation and experimental measurement. To implement the controller in real-time, we employ the ACADO Toolkit [90] to solve repeatedly the optimal control problem, and an image processing algorithm using OpenCV to provide feedback.

Some preliminary results of this work were presented at the 2016 ASME Dynamic Systems and Control Conference [91]. More comprehensive results were published in ASME Journal of Dynamic Systems and Control [92], where we proposed a computation-efficient method to deal with the nonlinear constraints, formulated a parameter estimation scheme to identify crucial parameters in the model, and reported experimental results on a tail-actuated robotic fish. The proposed NMPC scheme was compared with two alternative approaches. In one approach, the nonlinear input constraints were directly defined, and in the other boxed constraints were used. Overall, the tracking simulation and experimental results showed that the proposed NMPC scheme with projection resulted in faster convergence to the desired path and smaller path error than the alternative approaches. Chapter 2 describes the details of this work.

1.2.2 Backstepping Control-based Trajectory Tracking for Tail-actuated Robotic Fish

Although NMPC is a promising candidate for tracking control, its computational complexity poses significant challenges in implementing such controllers on resource-constrained robotic fish. Backstepping-based control design presents a practical and promising approach for trajectory tracking with stability guarantees. In particular, it is computationally inexpensive, especially when compared to methods such as NMPC, offers a systematic framework for the design of the feedback control law, guarantees the stability of the system, and allows the accommodation of input

constraints.

Some limited work has been reported on backstepping-based control of robotic fish [62, 63]. In [62], the authors proposed a target-tracking hybrid controller that consists of an open-loop turning controller and a closed-loop backstepping controller to drive a robotic fish to a specified target location. In [63], the authors proposed a backstepping controller for tracking control of a nonholonomic fish robot with dynamics that can be expressed in a chained form. The proposed approach guaranteed asymptotic convergence to a desired trajectory generated by a reference fish robot, but limitations were placed on the desired trajectory since it had to be generated by dynamics in the same chained form. Furthermore, input constraints were not accommodated.

In this work, we address the trajectory tracking control problem for a general class of autonomous under-actuated planar aquatic robots. In particular, we present a systematic method to synthesize a backstepping-based continuous tracking controller to ultimately bound the position tracking errors and the orientation error between the robot's heading and a vector tangential to the reference. Via multi-time-scale analysis of perturbed systems, we prove how the control scheme achieves boundedness and convergence of the tracking errors to a neighborhood of the origin.

The scheme proposed in this work is inspired by the approach in [93]. In particular, we utilize the idea of introducing an additional error coordinate that couples the lateral position error and the heading angle error. In this manner, by stabilizing the longitudinal tracking error along with this new state, we can guarantee that the whole error system is stabilized at a neighborhood of the origin. Furthermore, we establish a systematic method that provides necessary conditions to be considered when designing this augmenting state for a general class of systems.

The proposed control approach is implemented in a tail-actuated robotic fish, where the controller design uses a high-fidelity averaged nonlinear dynamic model [89]. In addition, the physical control inputs consist of the tail amplitude, and bias with the angular frequency kept constant. Furthermore, an auxiliary system is designed to accommodate the input constraints such that the controller compensates for the error due to the difference between the feasible and nominal inputs. Finally, simulation and experimental results demonstrate the effectiveness of the controller.

Preliminary simulation results were presented at the 2019 International Conference on Advanced Intelligent Mechatronics [94]. The details of this work are provided in Chapter 3.

1.2.3 Averaged Modeling and Trajectory Tracking Control of Pectoral Fin-actuated Robotic Fish

Pectoral fins have become a useful actuation mechanism for robotic fish [25, 64, 65]. Utilizing pectoral fins for locomotion improves maneuverability and can allow robotic fish to achieve trajectories that could not otherwise be possible with just a tail fin, such as circular trajectories with smaller radii, and sharp, aggressive turning. However, despite their benefit, pectoral fins give rise to challenges in control.

Given the rhythmic nature of the robotic fish's body and fin movements, averaging has proven to be a useful approach in obtaining control-affine models [95, 96] and studying the effect of the input parameters on its dynamics and fins movement [31, 54, 89]. This makes an averaged model best suited for trajectory planning and tracking control. To the best of our knowledge, the use of averaging has not been explored in developing control-affine models for pectoral fin-actuated robotic fish.

In this work, we present a nonlinear dynamic average model for robotic fish propelled by a pair of rigid pectoral fins undergoing rowing motion. In particular, we consider the robot undergoing planar motion, with its original dynamics incorporating pectoral fin-generated hydrodynamic forces evaluated via the blade element theory. Inspired by the work in [89], which deals with averaged dynamics for tail-actuated robotic fish, we seek scaling factors, as functions of fin-beat parameters, for the original hydrodynamic forces and moment. These scaling factors are determined such that when classical averaging is applied to the resulting modified dynamics, the obtained average model produces locomotion behaviors close to those of the original dynamic model. One fundamental step in identifying the scaling functions is estimating the scaling values for a given fin-beat pattern. [89] used a trial-and-error approach for the tail-actuated robotic fish, which is time-consuming. We propose a novel systematic approach to finding optimal scaling values by formulating a nonlinear

model-predictive control (NMPC) problem, which can be readily solved with NMPC packages instead. Once the scaling values are found for a set of fin-beat patterns, nonlinear regression is used to determine the scaling functions with minimal complexity. Simulation comparison between the averaged model and the original dynamic model, under fin-beat patterns not used in identifying the scaling functions, supports the efficacy of the developed averaged model. Furthermore, we conduct experiments on a pectoral fin-actuated robotic fish and compare the experimental results with simulation predictions when considering the forward swimming motion, where both fins are actuated symmetrically.

Finally, to demonstrate the utility of the dynamic average model, we design a controller for trajectory tracking of a pectoral fin-actuated robotic fish. In particular, the proposed scheme uses a backstepping-based controller that finds the needed inputs for the robot to track the desired trajectory based upon the averaged model. In this design, the physical control inputs involve two of the fin-beat parameters, the bias, and the amplitude, while the other parameters (angular frequency and power/recover stroke ratio) are kept constant. We further use a multi-variable minimization solver to determine the optimal fin-beat parameters such that the achieved inputs are closed to the needed values. Simulation results are presented to demonstrate the effectiveness of the proposed model-based tracking control scheme.

Preliminary experimental results demonstrating the effectiveness of the proposed average model will be presented at the 2021 Modeling, Estimation and Control Conference (MECC) [97]. Chapter 4 provides more detail of this work.

1.2.4 Rapid Maneuvering Control of Pectoral Fin-actuated Robotic Fish

Cyclic fin movement can limit precise manipulation of the fin movements and the thrust profile that can be generated which can be detrimental in tasks like maneuvering. Acyclic actuation could be more instrumental in such scenarios. In this work, we propose a systematic approach to the control of pectoral fins that naturally accommodates the fins' constraints and automatically generates "intelligent" behavior (e.g., "backing-up" when required) to produce acyclic thrust for

quick maneuvering. In particular, we consider the velocity tracking problem as an example to illustrate the challenges and propose a dual-loop control structure to drive the velocity tracking error to a neighborhood of the origin. The outer loop of the proposed scheme is composed of a backstepping-based velocity-tracking controller that finds the needed thrust and moment for the robot's velocity to track the desired velocity profile. The inner loop is composed of a randomized, model-predictive fin planning algorithm, which determines a feasible sequence for the fins' angular accelerations such that the thrust and moments generated are close to desired values. To demonstrate the effectiveness of the proposed scheme, we present simulation results on tracking forward and angular velocities that are abruptly changing. The proposed scheme is compared with an alternative approach, in which the inner loop is implemented with a nonlinear model predictive controller (NMPC).

Preliminary simulation results were presented at the 2021 International Conference on Advanced Intelligent Mechatronics [98]. Further details of this work are presented in Chapter 5.

CHAPTER 2

MODEL PREDICTIVE CONTROL-BASED PATH FOLLOWING FOR TAIL-ACTUATED ROBOTIC FISH

To be suitable for monitoring and sensing aquatic environments, accurate path-following control is vital for robotic fish. In this chapter we propose a nonlinear model predictive control (NMPC) approach for path-following of a tail-actuated robotic fish. In this controller design, a high-fidelity averaged nonlinear dynamic model is used, and the physical control inputs consist of two of the tail-beat parameters, the bias and the amplitude, while the other parameter (angular frequency) is kept constant. To address the nonlinear input constraints and maximize the use of the admissible control in a computationally efficient manner, we employ an analytical projection scheme. Furthermore, we present a novel parameter estimation scheme to empirically identify unknown parameters present in the robotic fish model instead of utilizing time-consuming CFD simulations or relying on trial-and-error data fitting between dynamic simulation and experimental measurement. To implement real-time control, we employ ACADO Toolkit [90] to solve repeatedly the optimal control problem, and an image processing algorithm using OpenCV to provide feedback.

This chapter is organized as follows. In section 2.1, the dynamic and scaled averaging models of the tail-actuated robotic fish, followed by a simplified averaged model, are reviewed. In Section 2.2 the path-following problem formulation is presented, followed by the NMPC design and the proposed control projection scheme. In Section 2.3 simulation results are discussed, and in Section 2.4 the experimental setup, the proposed parameter estimation scheme and the experimental results are presented.

2.1 Tail-actuated Robotic Fish Model

2.1.1 Dynamic Model

As presented in [89], the tail-actuated robotic fish is modeled as a rigid body with a rigid tail that is actuated at its base, and it is assumed that the robot operates in an inviscid, irrotational,

and incompressible fluid within an infinite domain. Let $[\mathbf{X}, \mathbf{Y}, \mathbf{Z}]^T$ and $[\mathbf{x}, \mathbf{y}, \mathbf{z}]^T$ be the inertial coordinate system and the body-fixed coordinate system, respectively, as illustrated in Figure 2.1. The velocity of the center of mass in the body-fixed coordinates is expressed as $\mathbf{V}_c = [V_{c_x}, V_{c_y}, V_{c_z}]$, where V_{c_x} , V_{c_y} , and V_{c_z} denote the surge, sway, and heave velocities, respectively. Furthermore, let β denote the angle of attack, formed by the direction of \mathbf{V}_c with respect to the \mathbf{x} -axis, and ψ denote the heading angle, formed by the \mathbf{x} -axis relative to the \mathbf{X} -axis. The angular velocity expressed in the body-fixed coordinate system is given by $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]$, which is composed of roll (ω_x), pitch (ω_y), and yaw (ω_z). Finally, let α denote the tail deflection angle with respect to the negative \mathbf{x} -axis.

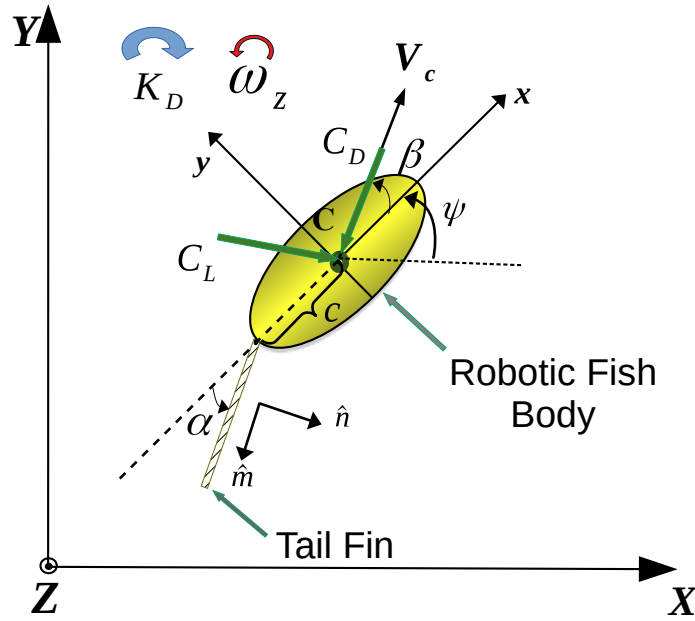


Figure 2.1: Top view of the tail-actuated robotic fish undergoing planar motion.

We only consider the planar motion and further assume that the body is symmetric with respect to the \mathbf{xz} -plane and that the tail moves in the \mathbf{xy} -plane. As a result, the system has three degrees of freedom, surge (V_{c_x}), sway (V_{c_y}), and yaw (ω_z). It is further assumed that the inertial coupling between yaw, sway and surge motions is negligible, which leads to the following equations of planar

motion

$$(m_b - m_{a_x})\dot{V}_{c_x} = (m_b - m_{a_y})V_{c_y}\omega_z + f_x \quad (2.1)$$

$$(m_b - m_{a_y})\dot{V}_{c_y} = -(m_b - m_{a_x})V_{c_x}\omega_z + f_y \quad (2.2)$$

$$(J_{b_z} - J_{a_z})\dot{\omega}_z = (m_{a_y} - m_{a_x})V_{c_x}V_{c_y} + M_z \quad (2.3)$$

where m_b is the mass of the body, J_{b_z} is the inertia of the body about the \mathbf{z} -axis, m_{a_x} and m_{a_y} are the hydrodynamic derivatives that represent the added masses of the robotic fish along the \mathbf{x} and \mathbf{y} directions, respectively, and J_{a_z} represents the added inertia effect of the body about the \mathbf{z} -axis. The hydrodynamic forces and moment due to tail fin actuation and the interaction of the body itself with the fluid are captured by f_x , f_y , and M_z . To evaluate the hydrodynamic forces exerted by the tail, Lighthill's large amplitude elongated body theory is used [89]. The kinematic equations for the robotic fish are given by

$$\dot{X} = V_{c_x} \cos \psi - V_{c_y} \sin \psi \quad (2.4)$$

$$\dot{Y} = V_{c_x} \sin \psi + V_{c_y} \cos \psi \quad (2.5)$$

$$\dot{\psi} = \omega_z \quad (2.6)$$

Given the rhythmic nature of the robotic fish movement and the periodic tail actuation, averaging has proven to be a useful approach in studying the effect of the input parameters on the dynamics of the robotic fish [89]. Furthermore, in practical applications, it is more natural to control the parameters for periodic fin beats than to directly control the fin position at every moment. Therefore, an averaged model is best suited for trajectory planning and tracking control. We next review the averaged model proposed in [89], where the following periodic pattern for the tail deflection angle is considered:

$$\alpha(t) = \alpha_0 + \alpha_a \sin(\omega_\alpha t) \quad (2.7)$$

where α_0 , α_a , and ω_α represent the bias, amplitude, and frequency of the tail beat, respectively. The original hydrodynamic force and moment terms in Eqs. (2.1)-(2.3) are scaled by some functions

dependent on the tail beat parameters, α_0 , α_a , and ω_α , and classical averaging is then conducted over these scaled dynamics. In particular, we define the states $x_1 = V_{cx}$, $x_2 = V_{cy}$ and $x_3 = \omega_z$, so that the averaged dynamics takes the following form

$$\dot{x}_1 = f_1(x_1, x_2, x_3) + K_f \bar{f}_4(\alpha_0, \alpha_a, \omega_\alpha) \quad (2.8)$$

$$\dot{x}_2 = f_2(x_1, x_2, x_3) + K_f \bar{f}_5(\alpha_0, \alpha_a, \omega_\alpha) \quad (2.9)$$

$$\dot{x}_3 = f_3(x_1, x_2, x_3) + K_m \bar{f}_6(\alpha_0, \alpha_a, \omega_\alpha) \quad (2.10)$$

with

$$f_1(x_1, x_2, x_3) = \frac{m_2}{m_1} x_2 x_3 - \frac{c_1}{m_1} x_1 \sqrt{x_1^2 + x_2^2} + \frac{c_2}{m_1} x_2 \sqrt{x_1^2 + x_2^2} \arctan\left(\frac{x_2}{x_1}\right) \quad (2.11)$$

$$f_2(x_1, x_2, x_3) = -\frac{m_1}{m_2} x_1 x_3 - \frac{c_1}{m_2} x_2 \sqrt{x_1^2 + x_2^2} - \frac{c_2}{m_2} x_1 \sqrt{x_1^2 + x_2^2} \arctan\left(\frac{x_2}{x_1}\right) \quad (2.12)$$

$$f_3(x_1, x_2, x_3) = (m_1 - m_2) x_1 x_2 - c_4 \omega_z^2 \text{sgn}(\omega_z) \quad (2.13)$$

$$\bar{f}_4(\alpha_0, \alpha_a, \omega_\alpha) = \frac{m}{12m_1 L^2} \omega_\alpha^2 \alpha_a^2 \left(3 - \frac{3}{2} \alpha_0^2 - \frac{3}{8} \alpha_a^2\right) \quad (2.14)$$

$$\bar{f}_5(\alpha_0, \alpha_a, \omega_\alpha) = \frac{m}{4m_2 L^2} \omega_\alpha^2 \alpha_a^2 \alpha_0 \quad (2.15)$$

$$\bar{f}_6(\alpha_0, \alpha_a, \omega_\alpha) = -\frac{m}{4J_3 L^2} c \omega_\alpha^2 \alpha_a^2 \alpha_0 \quad (2.16)$$

where $m_1 = m_b - m_{ax}$, $m_2 = m_b - m_{ay}$, $J_3 = J_{bz} - J_{az}$, $c_1 = \frac{1}{2} \rho S C_D$, $c_2 = \frac{1}{2} \rho S C_L$, $c_3 = \frac{1}{2} m L^2$, $c_4 = \frac{1}{(J_3)} K_D$, $c_5 = \frac{1}{(2J_3)} L^2 m c$, and $c_6 = \frac{1}{(3J_3)} L^3 m$. Here S denotes the reference surface area for the robot body, C_D , C_L and K_D represent the drag force coefficient, lift coefficient, and drag moment coefficient, respectively, ρ is the density of water, L is the tail length, c is the distance from the body center to the pivot point of the actuated tail and m represents the mass of water displaced by the tail per unit length and is approximated by $\frac{\pi}{4} \rho d^2$ with d denoting the tail depth. K_f is a scaling constant, and $K_m(\alpha_0)$ is a scaling function affine in α_0 . To further facilitate control design, in this work K_m is considered as a constant during the NMPC design. This term is found by taking the average of K_m for a given range of α_0 . The resulting model is called the simplified averaged model in this dissertation.

2.2 Path-following Control Algorithm

It is important to design a controller that is able to meet performance objectives such as minimizing the path-tracking error while accommodating input constraints. We are thus motivated to develop an NMPC scheme for path-following. NMPC is an attractive choice because it allows explicit consideration of state and input constraints, is capable of handling nonlinear models, and can optimize control performance [99, 100].

2.2.1 Path-following Error Coordinates

In contrast to trajectory tracking, in path-following one is interested in following a geometric reference parametrized by some scalar without any specified timing. The kinematic model of the robotic fish is expressed in a Frenet-Serret frame $\{F\}$ that moves along the reference path according to some desired function of time. Figure 2.2 illustrates the path-following problem.

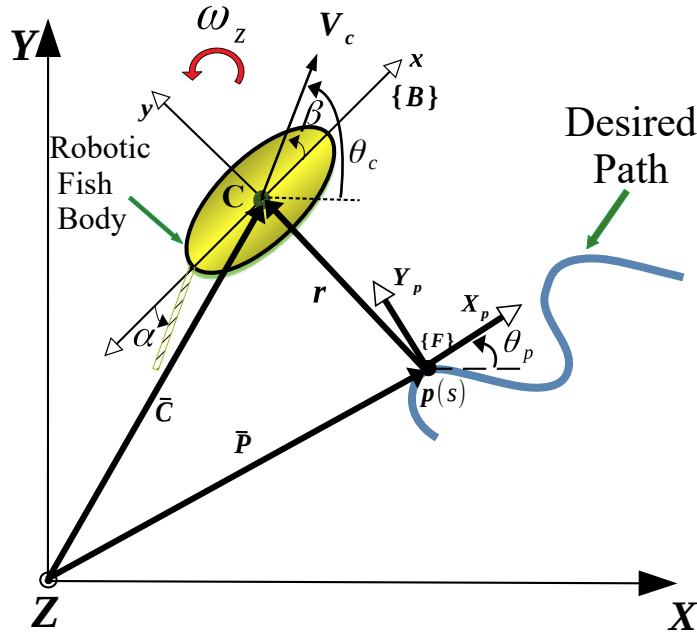


Figure 2.2: Illustration of the path-following problem for robotic fish.

Assume that the reference path is a twice continuously differentiable geometric curve that is

defined as a set of points \mathbf{P} parametrized by the scalar s ,

$$\mathcal{P} = \{\mathbf{P} \in \mathbb{R}^2 | \mathbf{P} = \mathbf{p}(s), \forall s \in [0, l_p]\} \quad (2.17)$$

where l_p denotes the length of the path, and the function $\mathbf{p}: \mathbb{R}^1 \rightarrow \mathbb{R}^2$ is twice differentiable. Let \mathbf{P} denote a point on the path to be followed, θ_p the tangential angle of the path at point \mathbf{P} , and θ_c the angle between the robotic fish velocity vector \mathbf{V}_c and the inertial \mathbf{X} -axis, while the coordinate axes x_p and y_p are directed along the tangential and normal directions at point \mathbf{P} . We let point \mathbf{C} denote the center of the robotic fish, and the vectors $\bar{\mathbf{C}}$ and $\bar{\mathbf{P}}$ describe the positions of \mathbf{C} and \mathbf{P} in the 3-D inertial frame $\{\mathbf{I}\}$. Note that since we are only considering the planar case, the third component of the position vectors is taken as 0. Let $\mathbf{r} = [X_e; Y_e; 0]^T$ denote the position of the robotic fish center \mathbf{C} with respect to the point \mathbf{P} on the path expressed in $\{\mathbf{F}\}$. Let ${}^I\mathbf{R}_F$ denote the rotation matrix from $\{\mathbf{I}\}$ to $\{\mathbf{F}\}$ and ${}^F\mathbf{R}_I$ denote the rotation matrix from $\{\mathbf{F}\}$ to $\{\mathbf{I}\}$, with

$${}^I\mathbf{R}_F = \begin{bmatrix} \cos \theta_p & \sin \theta_p & 0 \\ -\sin \theta_p & \cos \theta_p & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.18)$$

Define $\dot{\theta}_p = C_p(s)\dot{s}$, where $C_p(s)$ is the path's curvature. One can express

$$\bar{\mathbf{C}} = \bar{\mathbf{P}} + {}^F\mathbf{R}_I \mathbf{r} \quad (2.19)$$

The velocity of $\bar{\mathbf{C}}$ in $\{\mathbf{I}\}$ is given by

$$\left(\frac{d\bar{\mathbf{C}}}{dt}\right)_{\{\mathbf{I}\}} = \left(\frac{d\bar{\mathbf{P}}}{dt}\right)_{\{\mathbf{I}\}} + {}^F\mathbf{R}_I \left(\frac{d\mathbf{r}}{dt}\right)_{\{\mathbf{F}\}} + {}^F\mathbf{R}_I (\omega_p \times \mathbf{r}) \quad (2.20)$$

where

$$\omega_p = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_p \end{bmatrix} \quad (2.21)$$

Multiplying the equation above on the left by ${}^I\mathbf{R}_F$ gives the velocity of \mathbf{C} expressed in $\{\mathbf{F}\}$:

$${}^I\mathbf{R}_F \left(\frac{d\bar{\mathbf{C}}}{dt}\right)_{\{\mathbf{I}\}} = \left(\frac{d\bar{\mathbf{P}}}{dt}\right)_{\{\mathbf{F}\}} + \left(\frac{d\mathbf{r}}{dt}\right)_{\{\mathbf{F}\}} + (\omega_p \times \mathbf{r}) \quad (2.22)$$

where

$$\left(\frac{d\bar{\mathbf{C}}}{dt}\right)_{\{I\}} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ 0 \end{bmatrix} \quad (2.23)$$

$$\left(\frac{d\bar{\mathbf{P}}}{dt}\right)_{\{F\}} = \begin{bmatrix} \dot{s} \\ 0 \\ 0 \end{bmatrix} \quad (2.24)$$

$$\left(\frac{d\mathbf{r}}{dt}\right)_{\{F\}} = \begin{bmatrix} \dot{X}_e \\ \dot{Y}_e \\ 0 \end{bmatrix} \quad (2.25)$$

$$\boldsymbol{\omega}_p \times \mathbf{r} = \begin{bmatrix} 0 \\ 0 \\ C_p(s)\dot{s} \end{bmatrix} \times \begin{bmatrix} X_e \\ Y_e \\ 0 \end{bmatrix} = \begin{bmatrix} -C_p(s)\dot{s}Y_e \\ C_p(s)\dot{s}X_e \\ 0 \end{bmatrix} \quad (2.26)$$

After rearranging and solving for \dot{X}_e and \dot{Y}_e from Eq. (2.22), we get the following expression

$$\begin{bmatrix} \dot{X}_e \\ \dot{Y}_e \end{bmatrix} = \begin{bmatrix} \dot{X} \cos \theta_p + \dot{Y} \sin \theta_p - \dot{s} + C_p(s)\dot{s}Y_e \\ -\dot{X} \sin \theta_p + \dot{Y} \cos \theta_p - C_p(s)\dot{s}X_e \end{bmatrix} \quad (2.27)$$

Let $\alpha_e = \psi - \theta_p$, and further expand \dot{X} and \dot{Y} with Eq. (2.4) and Eq. (2.5), respectively, which results in the following error state model

$$\begin{pmatrix} \dot{X}_e \\ \dot{Y}_e \\ \dot{\alpha}_e \end{pmatrix} = \begin{pmatrix} V_c \cos(\alpha_e + \beta) - \dot{s} + C_p(s)\dot{s}Y_e \\ V_c \sin(\alpha_e + \beta) - C_p(s)\dot{s}X_e \\ \omega_z - C_p(s)\dot{s} \end{pmatrix} \quad (2.28)$$

where V_c is the magnitude of the robotic fish's translational velocity. The dynamical model of the robotic fish in the error state is then obtained by augmenting the equations above with the simplified averaged scaled dynamics as seen in Eqs. (2.8)-(2.10). Ideally, one would like the robotic fish to not only converge to a desired path, but also move along the path with some desired surge velocity and desired angular velocity. Let V_{dx} be the desired surge velocity, and let the velocity error states be defined by $\eta_e = V_{cx} - V_{dx}$, $\omega_e = \omega_z - C_p(s)\dot{s}$, and $\zeta_e = \dot{s} - V_c \cos(\alpha_e + \beta)$ so that

$$\begin{pmatrix} \dot{\eta}_e \\ \dot{\omega}_e \\ \dot{\zeta}_e \end{pmatrix} = \begin{pmatrix} \dot{V}_{cx} - \dot{V}_{dx} \\ \dot{\omega}_z - C_p(s)\ddot{s} - g_c(s)\dot{s}^2 \\ \ddot{s} - \dot{V}_c \cos(\alpha_e + \beta) + V_c \sin(\alpha_e + \beta)(\dot{\alpha}_e + \dot{\beta}) \end{pmatrix} \quad (2.29)$$

where $g_c(s) = \frac{dC_p(s)}{ds}$. Since we are interested in steering the robotic fish such that $V_{cx} = V_{dx}$, and $\omega_z = C_p(s)\dot{s}$, by doing a change of variables on Eqs. (2.8)-(2.10) using the above definitions, we can express the robotic fish dynamic equations in terms of the error velocity states. Note that in this work we assume $\dot{V}_{dx} = 0$. The error state vector is then given by

$$\mathbf{\Omega}_e = \begin{pmatrix} X_e \\ Y_e \\ \alpha_e \\ \eta_e \\ \omega_e \\ \zeta_e \end{pmatrix} \quad (2.30)$$

Since we have formulated the problem with respect to the error dynamics, and have shifted the equilibrium point of the dynamic equations, our control objective has become a stabilization problem for the resultant error dynamics.

2.2.2 Path-following Control Design

To steer the robotic fish to the desired path, and drive the error state vector $\mathbf{\Omega}_e$ to zero, we utilize an NMPC scheme using the robot's simplified averaged model. NMPC is an optimization-based method for feedback control of nonlinear systems, where the basic idea is to repeatedly solve a finite-horizon optimization problem subject to state and input constraints. At a given time t , measurements are obtained, and using a model of the process, the controller predicts the behavior of the system over a prediction horizon T_p and then determines over the control horizon T_c the input necessary to maximize the performance objective. The first part of the optimal control obtained is implemented until the next sampling instant, and then a new measurement is obtained and the process repeats [100].

To design the controller, we consider the robot's simplified averaged model in which the control represents functions of the actual control variables, namely, the tail-beat pattern parameters α_0 , α_a , and ω_α . By choosing the control in this manner, we allow the control inputs to appear linearly in the dynamic equation. In particular, we have chosen our control inputs as

$$u_{f_1} = \alpha_a^2 \left(3 - \frac{3}{2} \alpha_0^2 - \frac{3}{8} \alpha_a^2 \right) \quad (2.31)$$

$$u_{f_2} = \alpha_a^2 \alpha_0 \quad (2.32)$$

which are present in functions $f_4(\alpha_0, \alpha_a, \omega_\alpha)$ to $f_6(\alpha_0, \alpha_a, \omega_\alpha)$ in Eqs. (2.8)-(2.10). To simplify discussion, it is assumed that the robotic fish uses a fixed tail-beat frequency ω_α . Furthermore, since the system dynamics are expressed in terms of the velocity errors states, by doing a change of variables we have essentially shifted the equilibrium point of the dynamical system, which means that there is also a shift on the control inputs u_{f_1} and u_{f_2} . Let u_{2ss} and u_{3ss} represent the shifted control values, which are defined as follows

$$u_{2ss} = \frac{6\rho s C_D V_{dx}^2}{K_f m L^2 \omega_\alpha^2} \quad (2.33)$$

$$u_{3ss} = \frac{4K_D C_P(s) V_{dx}^2}{K_m m L^2 c \omega_\alpha^2} \quad (2.34)$$

In order to satisfy the condition that $\dot{\mathbf{\Omega}}_e = 0$ when $\mathbf{\Omega}_e = 0$ and $\mathbf{u}_e = 0$, we define the control inputs as follows

$$\mathbf{u}_e = \begin{pmatrix} u_{e1} \\ u_{e2} \\ u_{e3} \end{pmatrix} = \begin{pmatrix} \ddot{s} - \dot{V}_c \cos(\alpha_e + \beta) + V_c \sin(\alpha_e + \beta)(\dot{\alpha}_e + \dot{\beta}) \\ u_{f_1} - u_{2ss} \\ u_{f_2} - u_{3ss} \end{pmatrix} \quad (2.35)$$

where u_{e1} is essentially $\dot{\zeta}_e$ as seen in (2.29).

Since one is interested in steering the robotic fish to the desired path, we employ a stage cost that is a function of the error state vector $\mathbf{\Omega}_e$. Furthermore, to minimize the control effort a weighting term on the control inputs is introduced. The following quadratic cost is chosen:

$$F(\mathbf{\Omega}_e, \mathbf{u}_e) = (\mathbf{\Omega}_e)^T Q(\mathbf{\Omega}_e) + (\mathbf{u}_e)^T R(\mathbf{u}_e) \quad (2.36)$$

where Q and R are positive definite weighting matrices that penalize deviations from the desired values.

Furthermore, to guarantee closed-loop stability and convergence, we utilize the terminal penalty, and the fictitious terminal control law $\pi(\mathbf{\Omega}_e)$ as proposed in [101], where a polytopic linear differential inclusion (PLDI)-based method is employed to obtain the weighting matrix Q_T for a terminal penalty of the following form

$$E(\mathbf{\Omega}_e(t + T_p)) = (\mathbf{\Omega}_e(t + T_p))^T Q_T (\mathbf{\Omega}_e(t + T_p)) \quad (2.37)$$

The reader is referred to [101] for details on how to obtain this weighting matrix.

By solving the optimal control problem we obtain the optimal control sequence for u_{e1} , u_{e2} , and u_{e3} . From u_{e1} , we can obtain $\dot{\zeta}_e$, and thus the state ζ_e from which we can then solve for \dot{s} . Furthermore, from u_{e2} and u_{e3} , along with Eqs.(2.31)-(2.34), one can solve for the actual robotic fish control variables α_0 and α_a .

2.2.3 Control Projection

Given that the NMPC inputs, u_{e2} and u_{e3} , consist of functions of the actual robotic fish control variables α_0 and α_a , the NMPC input constraints are nonlinear in nature. As an illustration, Figure 2.3 plots the admissible control inputs in terms of u_{f1} and u_{f2} when the tail beat bias and amplitude have the following limits:

$$\alpha_{0_{\min}} = -40^\circ, \quad \alpha_{0_{\max}} = 40^\circ, \quad \alpha_{a_{\min}} = 0^\circ, \quad \alpha_{a_{\max}} = 30^\circ$$

where $\alpha_{0_{\min}}$, $\alpha_{0_{\max}}$, $\alpha_{a_{\min}}$ and $\alpha_{a_{\max}}$ are the physical limits on the tail-beat bias and amplitude, respectively.

Although NMPC is able to handle nonlinear control constraints, defining the constraints in this manner leads to an increase in computational time and complexity which in turn makes it challenging to implement in real time. It is thus desirable to define boxed-constraints since this can reduce significantly the complexity of the optimization problem and thus lower the computational time.

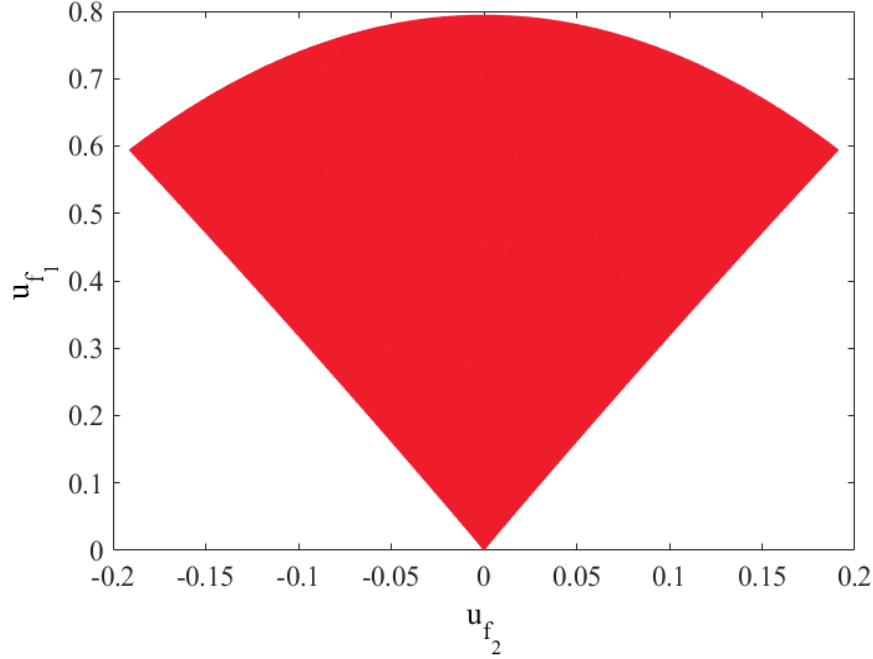


Figure 2.3: Illustration of the admissible input control set for robotic fish.

One way of handling the irregular sector-shaped admissible control region shown in Figure 2.3, is to choose a rectangular area that lies inside this sector-shaped region as depicted by the light gray box in Figure 2.4. However, this deprives one of fully utilizing the admissible control. To overcome this problem, we propose to employ a projection method, where we define the NMPC boxed constraint to be such that it encompasses the admissible control region as depicted in Figure 2.5, and then project the computed values onto the true region depicted by the red sector-shaped section.

Let z denote a control point anywhere in this rectangular region, and let the sector-shaped set be represented by U . We can then project the point z onto the convex set U such that

$$Proj_U(z) \triangleq \arg \min_{u \in U} \|z - u\| \quad (2.38)$$

Given that U is convex, this problem is then well defined and $Proj_U(z)$ is unique. Instead of relying on an iterative optimization algorithm to determine the projected value, one can directly obtain an analytical solution that will simplify the projection and minimize computational complexity, as explained next. By taking advantage of the symmetry of the admissible control set, one can restrict the analysis to the left-half plane. To characterize the boundaries of the admissible control

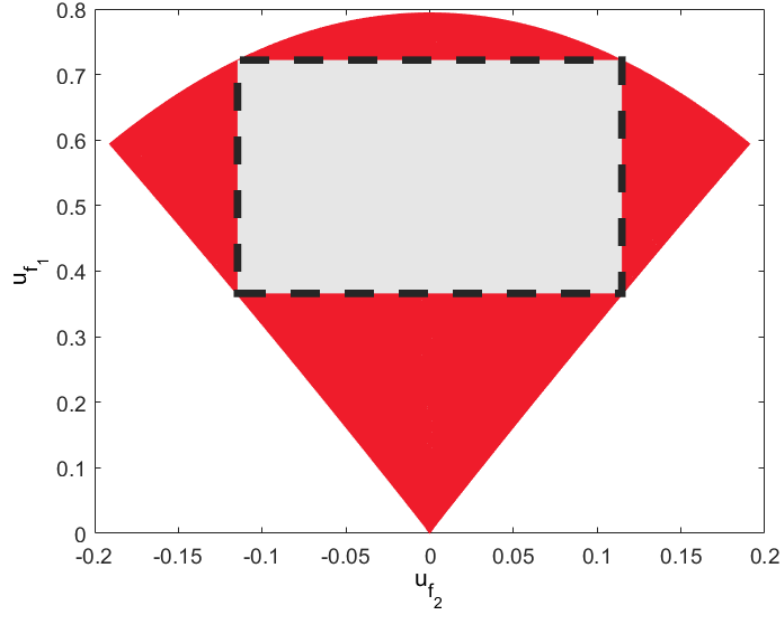


Figure 2.4: Illustration of a rectangular section chosen from the admissible control set.

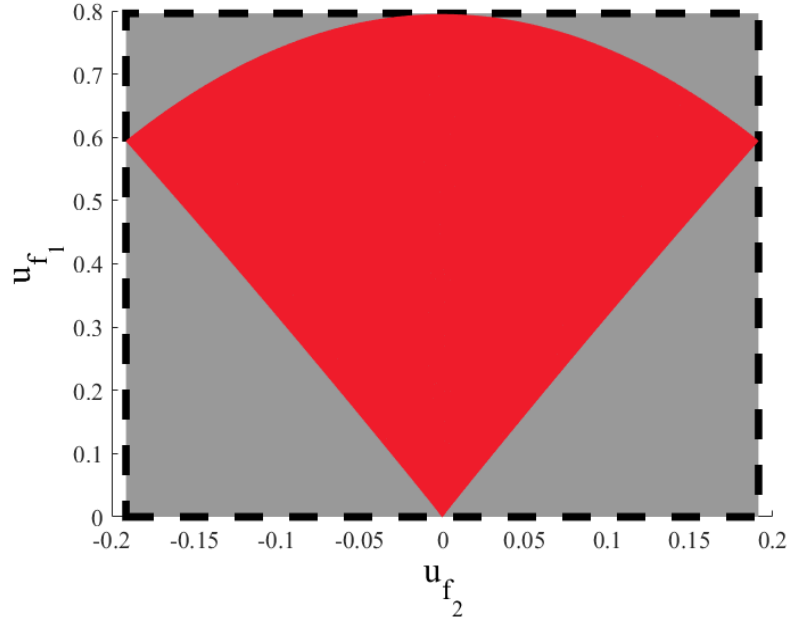


Figure 2.5: Boxed section encompassing the admissible control set.

region, we obtain the relationship between u_{f1} , u_{f2} , and α_0 by solving for α_a from Eq. (2.32) and then substituting that into Eq. (2.31). Similarly, by solving for α_0 from Eq. (2.32), we can obtain an equation that captures the relationship between u_{f1} , u_{f2} , and α_a . These equations are given as

follows

$$\chi_1(u_{f_1}, u_{f_2}, \alpha_0) \triangleq -\alpha_0^2 u_{f_1} + (3 - \frac{3}{2}\alpha_0^2)\alpha_0 u_{f_2} - \frac{3}{8}u_{f_2}^2 = 0 \quad (2.39)$$

$$\chi_2(u_{f_1}, u_{f_2}, \alpha_a) \triangleq -\alpha_a^2 u_{f_1} - \frac{3}{2}u_{f_2}^2 + 3\alpha_a^4 - \frac{3}{8}\alpha_a^2 = 0 \quad (2.40)$$

where Eq. (2.39) represents the left boundary when $\alpha_0 = \alpha_{0_{\min}}$ and Eq. (2.40) represents the arc at the top when $\alpha_a = \alpha_{a_{\max}}$. To implement the projection scheme, the following cases are considered:

$$\left\{ \begin{array}{l} (A) \quad \chi_1(u_{f_1}, u_{f_2}, \alpha_{0_{\min}}) \leq 0 \text{ and } \chi_2(u_{f_1}, u_{f_2}, \alpha_{a_{\max}}) \leq 0 \\ (B) \quad \chi_1(u_{f_1}, u_{f_2}, \alpha_{0_{\min}}) \leq 0 \text{ and } \chi_2(u_{f_1}, u_{f_2}, \alpha_{a_{\max}}) > 0 \\ (C) \quad \chi_1(u_{f_1}, u_{f_2}, \alpha_{0_{\min}}) > 0 \text{ and } \chi_2(u_{f_1}, u_{f_2}, \alpha_{a_{\max}}) \leq 0 \\ (D) \quad \chi_1(u_{f_1}, u_{f_2}, \alpha_{0_{\min}}) > 0 \text{ and } \chi_2(u_{f_1}, u_{f_2}, \alpha_{a_{\max}}) > 0 \end{array} \right.$$

For case (A), the point to be projected is inside or on the boundary of the convex set U and no projection is needed. For case (D), the point to be projected would be outside of the box encompassing the constraint set U and thus does not need to be considered. For case (B), the point to be projected is above the arc, and therefore $Proj_U(z)$ can be found by finding the minimum distance from the point z to the arc described by Eq. (2.40). Let $z = (p, q)$ and $u^* = (u_{f_1}^*, u_{f_2}^*) = Proj_U(z)$. The relationship between $u_{f_1}^*$ and $u_{f_2}^*$ is given by

$$u_{f_1}^* = \frac{1}{\alpha_{a_{\max}}^2} \left(\frac{-3}{2}u_{f_2}^{*2} + 3\alpha_{a_{\max}}^4 - \frac{3}{8}\alpha_{a_{\max}}^6 \right) \quad (2.41)$$

and the distance between z and u^* is then given by

$$\begin{aligned} g_2(u_{f_2}^*) &= (u_{f_2}^* - p)^2 + (u_{f_1}^* - q)^2 \\ &= (u_{f_2}^* - p)^2 + \left(\frac{1}{\alpha_{a_{\max}}^2} \left(\frac{-3}{2}u_{f_2}^{*2} + 3\alpha_{a_{\max}}^4 - \frac{3}{8}\alpha_{a_{\max}}^6 \right) - q \right)^2 \end{aligned} \quad (2.42)$$

By taking the partial derivative of $g_2(u_{f_2}^*)$ with respect to $u_{f_2}^*$, one obtains the following

$$\frac{\partial g(u_{f_2}^*)}{\partial u_{f_2}^*} = 2(u_{f_2}^* - p) + 2 \left(\frac{1}{\alpha_a^2} \left(\frac{3}{2}u_{f_2}^* + 3\alpha_a^4 - \frac{3}{8}\alpha_a^6 \right) - q \right) \left(2 \left(\frac{-3}{2\alpha_a^2} \right) \right) = 0 \quad (2.43)$$

After simplifying Eq. (2.43), we obtain

$$\frac{9u_{f_2}^{*3}}{\alpha_a^4} + (18 - \frac{18\alpha_a^2}{8} - \frac{6q}{\alpha_a^2} + 2)u_{f_2}^* - 2p = 0 \quad (2.44)$$

and by setting Eq (2.44) to zero, we can obtain a unique real root for $u_{f_2}^*$ that would minimize this distance. Finally, $u_{f_1}^*$ is obtained with Eq. (2.41).

For case (C), the point to be projected is below the left boundary. In this case, $u_{f_1}^*$ and $u_{f_2}^*$ are related by

$$u_{f_1}^* = \frac{1}{\alpha_{0\min}^2} \left(-\frac{3\alpha_{0\min}^3}{2} u_{f_2}^* + 3\alpha_{0\min} u_{f_2}^* - \frac{3}{8} u_{f_2}^{*2} \right) \quad (2.45)$$

and the distance between z and u^* can be captured by:

$$\begin{aligned} g_3(u_{f_2}^*) &= (u_{f_2}^* - p)^2 + (u_{f_1}^* - q)^2 \\ &= (u_{f_2}^* - p)^2 + \left(-\frac{3\alpha_{0\min}^2}{2} u_{f_2}^* + \frac{3u_{f_2}^*}{\alpha_{0\min}} - \frac{3}{8} u_{f_2}^{*2} - q \right)^2 \end{aligned} \quad (2.46)$$

By taking the partial derivative of $g_3(u_{f_2}^*)$ with respect to $u_{f_2}^*$ as follows:

$$\frac{\partial g_3(u_{f_2}^*)}{\partial u_{f_2}^*} = 2(u_{f_2}^* - p) + \left(-\frac{6}{2} u_{f_2}^* \alpha_0 + \frac{6}{\alpha_0} u_{f_2}^* - \frac{6}{8\alpha_0^2} u_{f_2}^{*2} - q \right) \left(\frac{-3}{2} + \frac{3}{\alpha_0} - \frac{6u_{f_2}^{*2}}{8\alpha_0^2} \right) = 0 \quad (2.47)$$

and by setting it to zero, we can obtain a unique real root for $u_{f_2}^*$, and consequently $u_{f_1}^*$ with Eq. (2.45).

2.3 Simulation Results

To evaluate the effectiveness of the designed controller, simulation was carried out using ACADO Model Predictive Control Toolkit. The parameters used (Table 2.1) were based on a robotic fish developed by Smart Microsystems Lab at Michigan State University. Furthermore, while the input constraints are the same as those presented in the experiment section, the parameters used to solve the optimization problem and implement the NMPC are as follows:

Length of optimization horizon : $T_c = T_p = 12$ s

Sampling interval : $t_s = 1$ s

Weighting matrix : $Q = \text{diag}(7, 7, 0.3, 1, 1, 7)$

Control weighting matrix : $R = 0.001I_3$

$V_{c \max} = 0.04 \text{ m/sec}$

$\dot{s}_{\max} = 0.04 \text{ m/sec}$

$\alpha_{0\min} = -40^\circ$

$\alpha_{0\max} = 40^\circ$

$\alpha_{a\min} = 0^\circ$

$\alpha_{a\max} = 30^\circ$

where $V_{c \max}$ is the maximum velocity the robotic fish can achieve, \dot{s}_{\max} is the maximum speed the point s can move along the path with, and $\alpha_{0\min}$, $\alpha_{0\max}$, $\alpha_{a\min}$ and $\alpha_{a\max}$ are the physical limits on the tail-beat bias and amplitude, respectively. Note that all of the following simulation was run with the same set of parameters and initial conditions. Furthermore, the terminal penalty weighting matrix was determined as described in Section 2.2.2. Though the controller was designed using the simplified averaged model, the simulation was performed using the original dynamic model. In other words, the model of the process was based on the simplified averaged dynamics as described by Eqs.(2.8)-(2.10), and the inputs obtained from solving the optimization problem were applied to the system described by Eqs. (2.1)-(2.3).

We first considered the following path

$$\begin{aligned} x_p &= s \\ y_p &= 0 \end{aligned} \tag{2.48}$$

where x_p and y_p represent the position of the point P in the $\{I\}$ frame. This path has a curvature of $c_p(s) = 0$, and we chose to require the robotic fish to move with a constant velocity $V_c = 0.03 \text{ m/s}$. In Figures 2.6-2.8 we compare the desired path and the closed-loop trajectory of the robotic fish for three cases. In particular, Figure 2.6 shows the simulation results of the NMPC utilizing

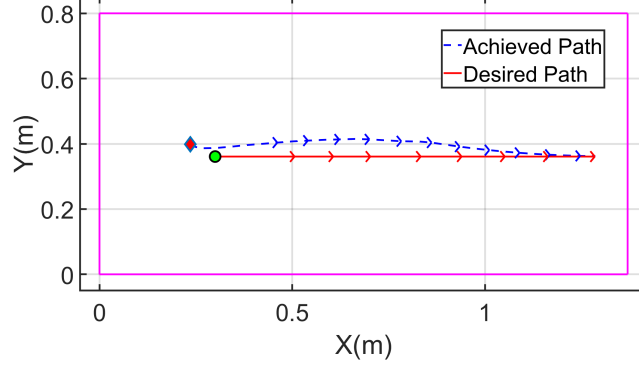


Figure 2.6: Simulation: line-tracking results for NMPC with the proposed control projection method.

the projection scheme, while Figure 2.7 shows the results for the case when no projection was employed and a boxed constraint within U was chosen instead (as shown in Figure 2.4). Finally, Figure 2.8 shows the results when the nonlinear constraints for the set U were directly defined. Note that in this work the blue dashed line represents the closed-loop trajectory of the robotic fish while the solid red line represents the desired path, and the arrowheads point in the direction of progression. Furthermore, the red diamond represents the starting position of the robotic fish, the green dot represents the starting point of the path, and the magenta box represents the imaginary boundaries of the fish tank.

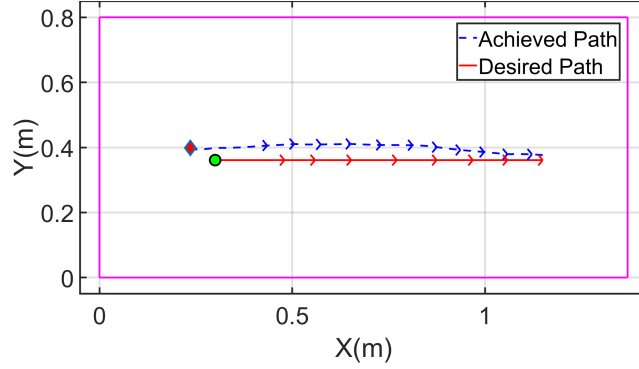


Figure 2.7: Simulation: line-tracking results for NMPC with boxed constraints inside the admissible sector-shaped control region.

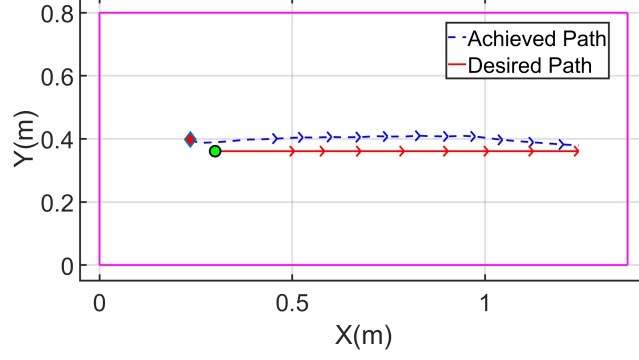


Figure 2.8: Simulation: line-tracking results for NMPC with direct nonlinear control constraints.

Similarly, we considered the following circular path

$$\begin{aligned} x_p &= 0.3 \sin(s) \\ y_p &= 0.3 \cos(s) \end{aligned} \tag{2.49}$$

which has a constant curvature of $c_p(s) = 3.03$. In Figures 2.9-2.11 we compare the desired path trajectory with those obtained by the robotic fish using the three aforementioned control schemes, respectively.

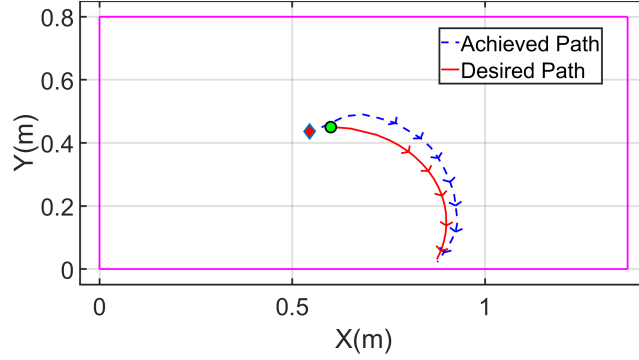


Figure 2.9: Simulation: arc-tracking results for NMPC with the proposed control projection method.

From the simulation results, one can see that the proposed NMPC scheme with projection outperforms the other two schemes in both line-tracking and arc-tracking cases; in particular, it results in smaller tracking error at the steady state. Compared with the case with boxed constraint within the set U , the proposed scheme offers larger control authority. The better tracking results from the proposed scheme compared to the case using direct, nonlinear constraints, however, were

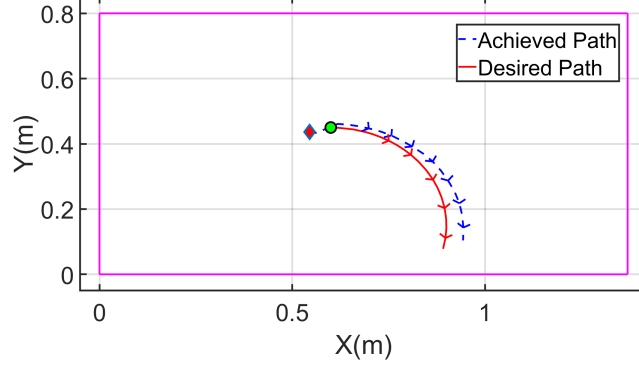


Figure 2.10: Simulation: arc-tracking results for NMPC with boxed constraints inside the admissible sector-shaped control region.

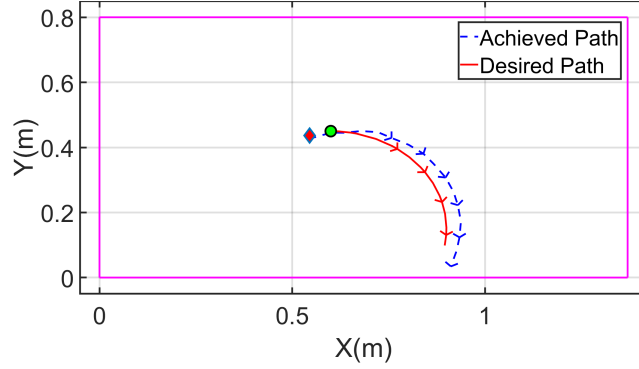


Figure 2.11: Simulation: arc-tracking results for NMPC with nonlinear control constraints.

somewhat surprising. We conjecture that this is because the latter algorithm cannot reach an optimal solution within the allotted computing time. In particular, directly defining the nonlinear constraints requires the optimization algorithm to conduct more iterations in order to find the solution, which also makes it difficult to implement in real time.

2.4 Experimental Results

In order to evaluate the effectiveness of the designed controller, experiments were carried out using the robotic fish depicted in Figure 2.13. The robot consisted of a rigid-shell body and a relatively rigid tail, which were both 3D-printed. The tail was actuated using a Hitec digital micro waterproof servo (HS-5086WP), while a Microchip Digital Signal Processors and Controller (DSPIC30F6014) was used to control the tail actuation. Furthermore, an XBee-PRO module was used for communication with a computer. Two Tenergy Li-Ion rechargeable batteries(7.4V, 3350mAh) were used

to power the robot. For the experiments, the robotic fish was run in an approximately 1.4 m by 0.8 m tank equipped with an overhead Logitech camera as seen in Figure 2.12. Furthermore, to obtain the robotic fish's position and orientation in the tank, two markers were attached to the anterior and posterior of the robotic fish body. We then captured an overhead video of the robotic fish swimming in the tank using the camera, and utilized Visual C++ and the OpenCV library to implement an image processing algorithm. The algorithm detected the positions of the two markers and then used their average to obtain the center position of the robotic fish. The heading angle of the robot was estimated using the positions of the two markers. Additionally, the Kalman filter function in OpenCV was used to estimate the linear and angular velocities of the robot based on the measured position and heading. During every sampling time t_s , the OpenCV algorithm was used to obtain measurements for NMPC, which were then passed to the nonlinear optimization tool ACADO to solve the optimal control problem. In particular, we ran the software on a Surface Pro tablet with an Intel(R) Core(TM) i5 CPU @ 2.50 GHz with 4.0 GB of DDR3 RAM. Once the control inputs were calculated, the bias and amplitude values for the tail beat were obtained and then transmitted to the robotic fish wirelessly, and the process was repeated.

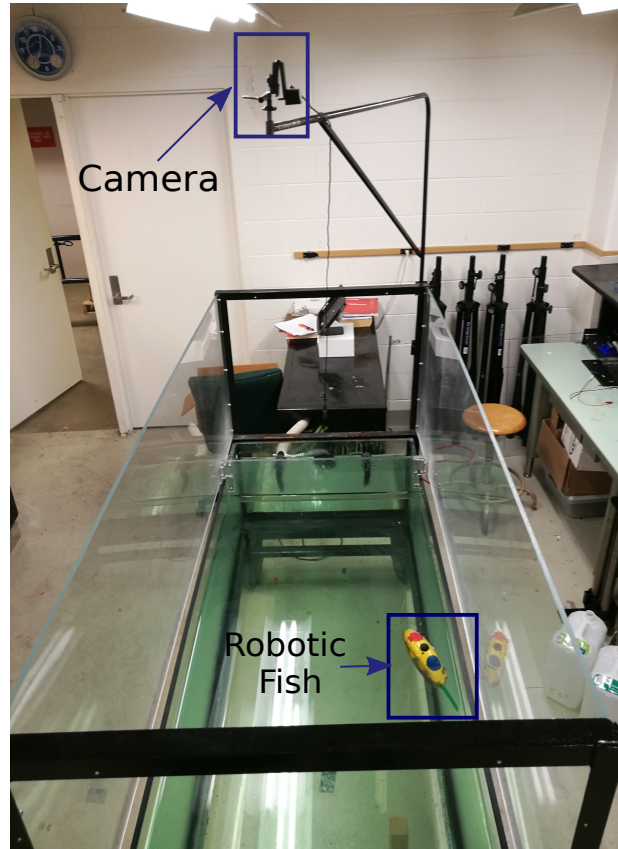


Figure 2.12: The experimental setup.

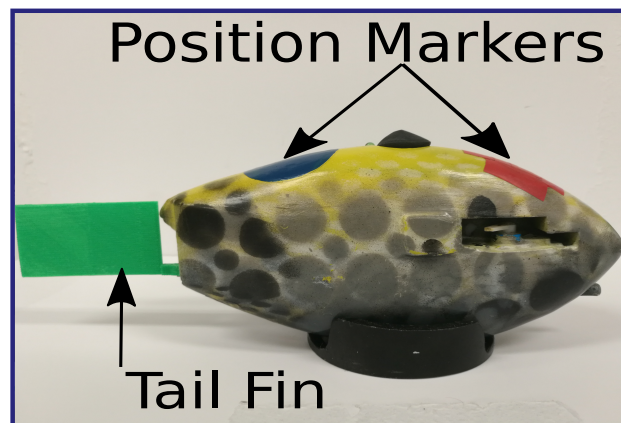


Figure 2.13: Tail-actuated robotic fish develop at Smart Microsystems Lab.

2.4.1 Model Parameter Identification

Table 2.1: Identified parameters for the robotic fish used in this work.

PARAMETER	VALUE
m_b	0.725 kg
m_{ax}	-0.217 kg
m_{ay}	-0.7888 kg
J_{bz}	$2.66 \times 10^{-3} \text{ kg} \cdot \text{m}^2$
J_{az}	$-7.93 \times 10^{-4} \text{ kg} \cdot \text{m}^2$
L	0.071 m
d	0.04 m
c	0.105 m
ρ	1000 kg/m ³
S	0.03 m ²
C_D	0.97
C_L	3.9047
K_D	$4.5 \times 10^{-3} \text{ kg} \cdot \text{m}^2$
K_f	0.7
$K_m(\text{averaged})$	0.45

The robotic fish mass and tail fin dimensions were measured, the values of which are as shown in Table 2.1. Furthermore, the added masses, added inertia, and wetted surface were calculated based on a prolate spheroid approximation of the robotic fish body [102]. Identification of the hydrodynamic parameters (such as C_D , C_L , and K_D) of the robotic fish model (2.8)-(2.16) typically requires extensive effort in fitting dynamic simulation data to experimental data by scanning the parameter space [35]. Furthermore, the determination of the scaling coefficients of K_f and K_m requires scanning the parameter space for multiple sets of tail beat patterns and matching the simulated average model data to the simulated dynamic model [89], which is time-consuming. In this work, we propose an efficient and systematic way to identify the model parameters by exploiting the approximate, analytical relationship between the steady-state turning parameter (turning radius, turning period, etc.) and the model parameters established in [89]. With the assumption $|x_1| \gg |x_2|$, which is reasonable in general, we can obtain the unique equilibrium of the system (2.8)-(2.10),

under a given tail beat pattern, as

$$\bar{x}_1 = \sqrt{\frac{K_f m L^2 \omega_\alpha^2 \alpha_a (3 - \frac{3}{2} \alpha_0^2 - \frac{3}{8} \alpha_a^2)}{6 \rho S C_D}} \quad (2.50)$$

$$\bar{x}_2 = \frac{K_f}{2 \rho S (C_D + C_L)} \sqrt{\frac{6 m \rho S C_D L^2 \omega_\alpha^2 \alpha_a^2 \alpha_0^2}{K_f (3 - \frac{3}{2} \alpha_0^2 - \frac{3}{8} \alpha_a^2)}} + \quad (2.51)$$

$$\frac{2 m_1}{\rho S (C_D + C_L)} \sqrt{\frac{K_m m L^2 c \omega_\alpha^2 \alpha_a^2 \alpha_0}{4 K_D}} \quad (2.52)$$

$$\bar{x}_3 = -\sqrt{\frac{K_m m L^2 c \omega_\alpha^2 \alpha_a^2 \alpha_0}{4 K_D}}$$

And the steady-state turning period T_p (i.e., how long the robot takes to complete a full orbit), turning radius R , and angle of attack β can be expressed as

$$T_p = 2\pi / |\bar{x}_3| \quad (2.53)$$

$$R = \sqrt{(\bar{x}_1^2 + \bar{x}_2^2)} / |\bar{x}_3| \quad (2.54)$$

$$\beta = \arctan(\bar{x}_2 / \bar{x}_1) \quad (2.55)$$

Using Eqs. (2.50)-(2.55), we formulate the following algorithm to obtain the hydrodynamic coefficients C_D , C_L and K_D , as well as the scaling coefficients K_m and K_f for the averaged model. Let $R_1 = \frac{K_f}{C_D}$. By solving for the ratio $\frac{K_f}{C_D}$ from Eq. (2.50), we obtain

$$R_1 = \frac{6 \rho S \bar{x}_1^2}{m L^2 \omega_\alpha^2 \alpha_a (3 - \frac{3}{2} \alpha_0^2 - \frac{3}{8} \alpha_a^2)} \quad (2.56)$$

Using the equation above, one can obtain the numerical value of the ratio R_1 for a given set of tail beat parameters and the corresponding measured \bar{x}_1 . In particular, we found this ratio by averaging the different values obtained for each set of measurements. Furthermore, let $K_m = K_0 + K_1 \alpha_0$. By solving for $\frac{K_m}{K_D}$ from Eq. (2.52), and using the definition $\theta_0 = K_0 / K_D$ and $\theta_1 = K_1 / K_D$, one gets

$$\theta_0 + \alpha_0 \theta_1 = \frac{\bar{x}_3^2}{m L^2 c \omega_\alpha^2 \alpha_a^2 \alpha_0} \quad (2.57)$$

Using Eq. (2.57) one can then estimate the numerical values for θ_0 and θ_1 by utilizing, for example, the constrained linear least squares (*lsqlin*) function in MATLAB, based on the tail beat parameters and the corresponding measurement of \bar{x}_3 for a set of experiments.

By considering R_1 and the ratio $\frac{K_m}{K_D}$, we have reduced the number of parameters to be estimated from 5 (K_f, K_m, C_D, C_L, K_D) to 3 ($R_1, \frac{K_m}{K_D}, C_L$). In order to obtain the particular values for C_D , K_f , K_m , and K_D , and to estimate the remaining parameter C_L , we utilize Eqs. (2.56) and (2.57) along with Eqs. (2.51) and (2.54). By letting $c_0 = C_D + C_L$ and substituting R_1 , θ_0 and θ_1 into (2.54), one obtains

$$R^2|\bar{x}_3|^2 = \bar{x}_1^2 + \left(\frac{C_D}{c_0}d_1 + \frac{d_2}{c_0} \right)^2 \quad (2.58)$$

where

$$d_1 = R_1 \sqrt{\frac{6m\rho SL^2\omega_\alpha^2\alpha_a^2\alpha_0^2}{R_1(3 - \frac{3}{2}\alpha_0^2 - \frac{3}{8}\alpha_a^2)}} \quad (2.59)$$

$$d_2 = \frac{2m_1}{\rho S} \sqrt{\frac{(\theta_1 + \theta_2\alpha_0)mL^2c\omega_\alpha^2\alpha_a^2\alpha_0}{4}} \quad (2.60)$$

Using Eq. (2.58) and letting $c_1 = \frac{1}{c_0}$, we can obtain the following:

$$y = c_1^2 d_2^2 + d_1^2 C_D^2 c_1^2 + 2c_1 d_2 d_1 C_D \quad (2.61)$$

where $y = R^2|\bar{x}_3|^2 - \bar{x}_1^2$. Letting $\phi_1 = c_1^2$, $\phi_2 = C_D^2 c_1^2$, and $\phi_3 = c_1^2 C_D$, one can rewrite the above expression as

$$y = \phi_1 d_2^2 + d_1^2 \phi_2 + 2d_2 d_1 \phi_3 \quad (2.62)$$

With a set of collected data, the parameters ϕ_1 through ϕ_3 can be estimated readily using techniques, such as the constrained linear-least square method. We can then solve for C_D , C_L and K_f using the definitions established above. Since the proposed estimation method only provides the ratio $\frac{K_m}{K_D}$, to obtain the values for K_m and K_D , we run simulations with the original dynamical model and choose K_D such that the angular velocity of the dynamic model matches that of the averaged model.

Table 2.2: Model validation results: relative model prediction error for turning radius and turning period, when the tail beats at 1.5 Hz.

(α_a, α_0)	Turning Radius Error(%)	Turning Period Error(%)
$(15^\circ, 45^\circ)$	11.23	8.80
$(15^\circ, 50^\circ)$	7.57	11.65
$(20^\circ, 45^\circ)$	8.46	3.82
$(20^\circ, 50^\circ)$	8.74	12.55
$(25^\circ, 45^\circ)$	2.90	0.97
$(25^\circ, 50^\circ)$	11.07	15.88

For the implementation of the above parameter estimation scheme, we first ran experiments to obtain the steady-state turning radii and periods for different tail biases (0° , 25° , 40°), and amplitudes (15° , 20° , 25°) while holding the frequency at 1 Hz. The values obtained for the parameters K_f , K_m , C_D , C_L and K_D are listed in Table 2.1. Furthermore, to validate the models we ran experiments with the same set of biases and amplitudes as previously stated while holding the frequency at 1.5 Hz. Table 2.2 lists the errors in turning radius and period between those obtained from experiments and those obtained from simulation using the parameters estimated above. The comparison indicates that the estimated model has acceptable accuracy.

2.4.2 Experimental Results on Path-Following

The parameters used to solve the optimization problem and implement the NMPC were as follows:

Length of optimization horizon : $T_c = T_p = 7s$

Sampling interval: $t_s = 1s$

Weighting matrix: $Q = 0.9I_5$

Control weighting matrix: $R = 0.001I_3$

$V_{c \text{ max}} = 0.04 \text{ m/s}$

$\dot{s}_{\text{max}} = 0.04 \text{ m/s}$

$$\alpha_{0\min} = -40^\circ$$

$$\alpha_{0\max} = 40^\circ$$

$$\alpha_{a\max} = 30^\circ$$

$$\alpha_{a\min} = 0^\circ$$

The following were the inputs constraints used for the case implementing projection:

$$\begin{bmatrix} -1.810 & \leq u_{e1} \leq 0.190 \\ 0 & \leq u_{f1} \leq 0.794 \\ -0.191 & \leq u_{f2} \leq 0.191 \end{bmatrix} \quad (2.63)$$

For the case using boxed constraints (without projection), we considered the following input constraints:

$$\begin{bmatrix} -1.81 \leq u_{e1} \leq 0.1900 \\ 0.366 \leq u_{f1} \leq 0.794 \\ -0.115 \leq u_{f2} \leq 0.115 \end{bmatrix} \quad (2.64)$$

We first considered the following path

$$\begin{aligned} x_p &= s \\ y_p &= 0 \end{aligned} \quad (2.65)$$

where x_p and y_p represent the position of the point P in the {I} frame. The desired velocity for the robotic fish was set to be 0.03 m/s. In Figures 2.14 and 2.15 we compare the desired path and the closed-loop robotic fish trajectory, obtained by using the NMPC with the proposed control projection scheme, and with a boxed constraint inside the nonlinear constraint set U , respectively. We do not report the case of NMPC with nonlinear constraints U directly, because it could not be implemented in real time due to its long computation time. Figure 2.16 shows the computed physical inputs from solving the NMPC with the larger boxed constraint and their final values after the proposed projection. Furthermore, it shows the desired surge velocity V_{dx} versus the robot's linear velocity's V_{cx} .

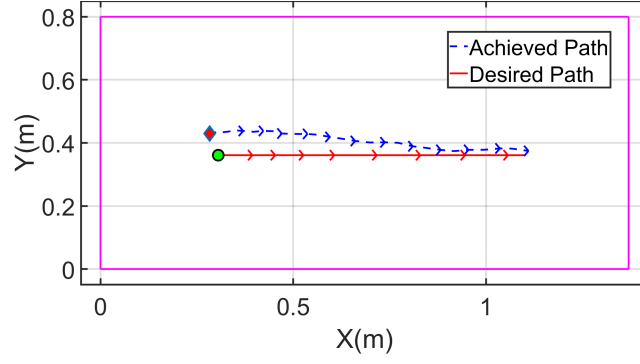


Figure 2.14: Experiments: line-tracking results for NMPC with the proposed control projection method.

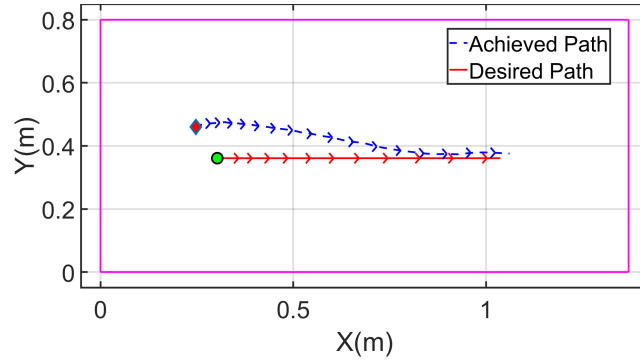


Figure 2.15: Experiments: line-tracking results for NMPC with boxed constraints inside the admissible sector-shaped control region.

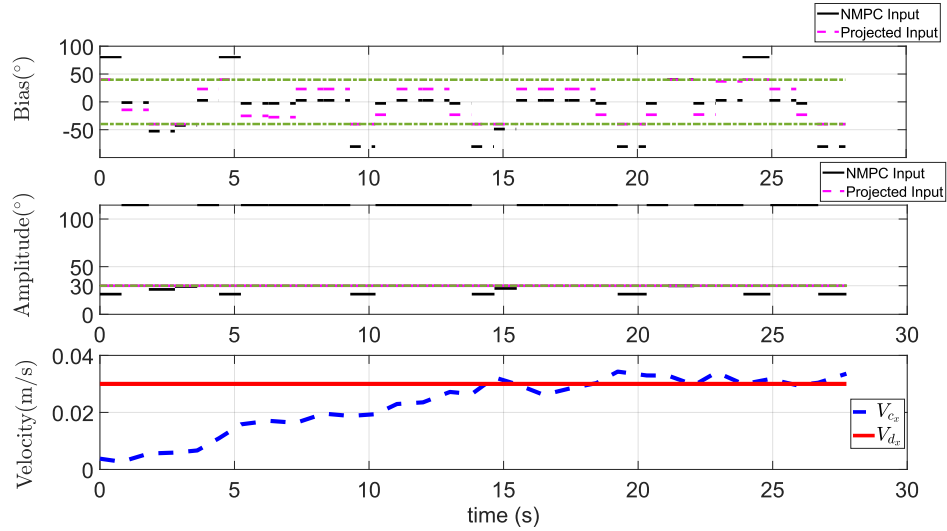


Figure 2.16: Experiments: computed NMPC physical inputs and their projected values.

Similarly, we considered the following circular path

$$\begin{aligned} x_p &= 0.3 \sin(s) \\ y_p &= 0.3 \cos(s) \end{aligned} \tag{2.66}$$

and Figure 2.17 and Figure 2.18 show the path-following results for NMPC with the proposed projection and for NMPC with boxed constraints inside U , respectively.

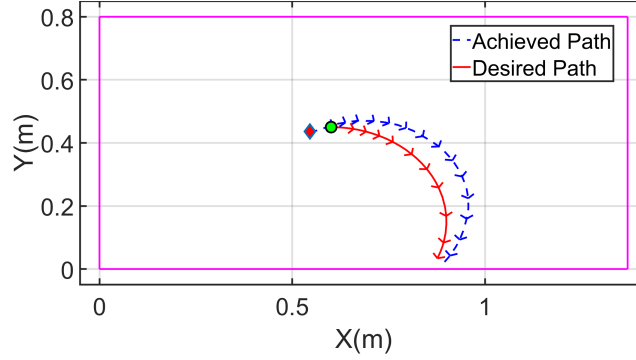


Figure 2.17: Experiments: arc-tracking results for NMPC with the proposed control projection method.

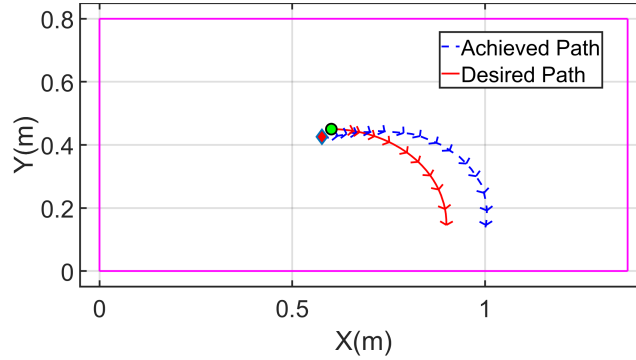


Figure 2.18: Experiments: arc-tracking results for NMPC with boxed constraints inside the admissible sector-shaped control region.

Overall, the tracking results shown in Figures 2.14-2.18, one can see that, consistent with the simulation results, the proposed NMPC scheme with projection resulted in faster convergence to the desired path and smaller path error, due to the availability of larger control authority.

2.5 Chapter Summary

In this chapter, we proposed and implemented in real time a path-following NMPC scheme for a tail-actuated robotic fish. A high-fidelity averaged nonlinear dynamic model was used for controller design. A parameter estimation scheme was employed to empirically identify the hydrodynamic parameters and scaling coefficients of the model. Furthermore, given that the control inputs were functions of two of the tail-beat parameters, specifically the tail bias and tail amplitude, a control projection strategy was implemented to handle these nonlinear input constraints and maximize the use of the admissible control region in a computationally efficient manner. Finally, simulation and experimental results demonstrated the effectiveness of the proposed scheme.

CHAPTER 3

BACKSTEPPING CONTROL-BASED TRAJECTORY TRACKING FOR TAIL-ACTUATED ROBOTIC FISH

Although NMPC offered a systematic framework for control, its computational complexity poses challenges on implementation in resource-constrained robotic fish. Furthermore, the effect of the proposed projection scheme on stability is unclear and challenging to analyze. We are thus inspired to explore alternative control approaches that can handle input constraints and provide stability analysis of the closed-loop system. Backstepping-based control design presents a practical and systematic approach to the trajectory tracking problem as it is computationally inexpensive, especially when compared to methods such as nonlinear model predictive control (NMPC), and provides stability guarantees. In this chapter, we first consider the trajectory tracking control problem for a general class of autonomous under-actuated planar aquatic robots and then implement the proposed scheme on a tail-actuated robotic fish.

In the case of under-actuated marine robotic systems confined to a horizontal plane, several groups have proposed nonlinear control methods to achieve tracking of desired trajectories. However, some of these works have been limited to tracking a restrictive class of trajectories with reference angular velocities bigger than zero [103–112]. Others have only focused on regulating the position error while ignoring the vehicle’s orientation, which in some applications, e.g., in the case of the robotic fish, can often lead to unstable configurations where no proper control can be applied due to the actuator constraints [83, 113–120].

Several authors have designed backstepping and Lyapunov-based controllers that have achieved both position and orientation tracking [121–127]. However, these approaches suffer from the drawback that the position and orientation tracking controllers only allow the vehicles to track pre-specified feasible state-space trajectories that are generated using the model of the vehicle.

In this work, we address the trajectory tracking control problem for autonomous under-actuated planar aquatic robots and vehicles. In particular, we present a systematic method to design a

backstepping-based continuous tracking controller that yields global ultimate boundedness of the position tracking errors and the orientation error. Furthermore, we demonstrate its effectiveness via simulation and experiments on a tail-actuated robotic fish.

The control scheme proposed in this work was inspired by [93] (chapter 5), where the control of an under-actuated ship is considered. In particular, we utilize the idea of introducing an augmented error coordinate to address the under-actuation issue and stabilize the heading, lateral and longitudinal tracking errors. While [93] focuses on the dynamic model of a ship, in this work we consider a general class of nonlinear systems with under-actuated planar dynamics and input coupling where each input could affect multiple velocity states. Furthermore, unlike [93] and other work previously reported in literature we do not impose that the reference trajectory be generated from a nominal dynamics model, i.e., we do not require that the desired body-fixed velocities are generated from a virtual vehicle with the same dynamics, nor do we assume that the reference angular velocity is persistently exciting or decays to zero as in [93]. Another key difference between our work and [93] is that, while [93] uses a particular form for the augmented error state, we provide a systematic approach to the synthesis of this state that guarantees the global boundedness and convergence of the position and heading tracking errors to a neighborhood of the origin. We establish the theoretical results via multi-time-scale analysis of singularly perturbed systems.

The proposed scheme is illustrated and validated via the trajectory tracking control of a tail-actuated robotic fish, as the latter is a good example of highly nonlinear, under-actuated robotic systems that exhibit input coupling [128]. By considering the cyclic nature of tail actuation, an averaged dynamic model for the robotic fish is used, where the bias and amplitude of the tail oscillation are treated as physical variables to be manipulated, while the tail beat frequency is fixed. Experimental results demonstrate the effectiveness of the proposed scheme and show its advantages over a well-tuned PI controller.

The rest of the chapter is organized as follows. We first describe the dynamic model for the class of under-actuated robots considered in this work in Section 3.1. In Section 3.2, the problem

formulation for trajectory tracking is presented, followed by the backstepping control design. The closed-loop system stability analysis is shown in Section 3.3. In Section 3.4 the backstepping controller is designed for a robotic fish. In Section 3.5 simulation results are presented. Finally, in Section 3.6 the experimental set-up and experimental results for the robotic fish are discussed. Finally, we provide some concluding remarks in Section 3.7.

3.1 System Model

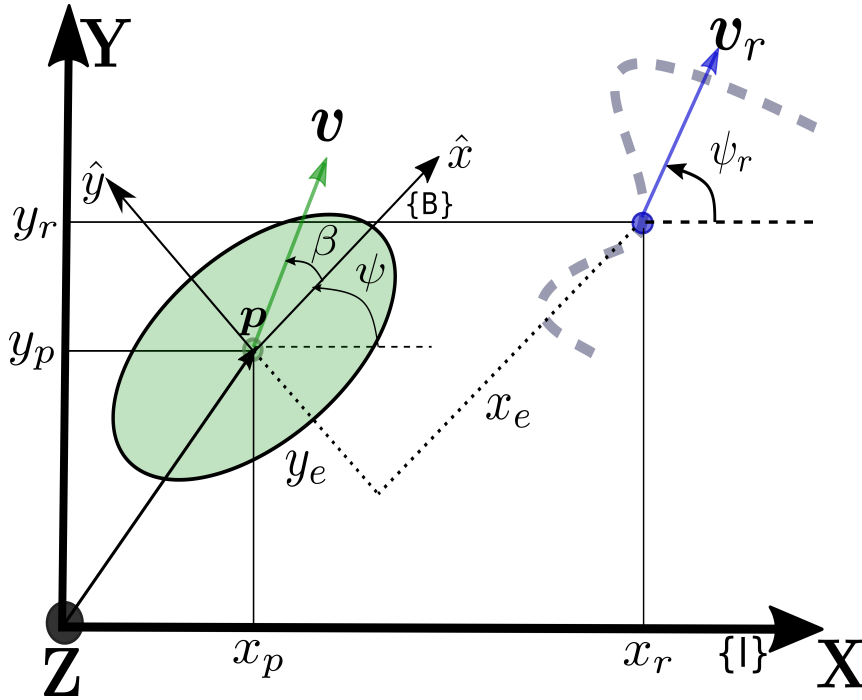


Figure 3.1: Top view of an aquatic robotic system undergoing planar motion.

Consider an under-actuated robotic system modeled as a 3-DOF rigid body with two inputs. Let $\{I\}$ denote the inertial coordinate frame and $\{B\}$ the body-fixed reference frame attached to the center of mass of the robot. In particular, let $[X, Y, Z]^T$ and $[\hat{x}, \hat{y}, \hat{z}]^T$ denote the inertial and the body-fixed coordinate systems, respectively, as illustrated in Figure 3.1. Assume that the robot undergoes planar motion such that its configuration $({}^B R_I, p)$ is an element of the Special Euclidean group $SE(2) := SO(2) \times \mathbb{R}^2$ where ${}^B R_I \in \mathbb{R}^{2 \times 2} : \{{}^B R_I {}^B R_I^T = I_2, \det({}^B R_I) = +1\}$ is a rotation matrix that maps the vehicle's body coordinates into the inertial coordinates, and $p = [x_p, y_p]^T$ is the position of the origin of $\{B\}$ in $\{I\}$. Let $v = [v_1 \ v_2]^T$ denote the body-fixed linear velocity,

where v_1 and v_2 denote the surge and sway velocities, respectively. The angular velocity expressed in the body-fixed coordinates is given by $\omega \in \mathbb{R}$. Furthermore, let β denote the angle of attack, formed by the direction of \mathbf{v} with respect to the body-fixed \hat{x} -axis, and let ψ denote the heading angle, formed by the body-fixed \hat{x} -axis relative to the inertial \mathbf{X} -axis. The robot's kinematics are thus expressed as

$$\dot{\mathbf{p}} = {}^B \mathbf{R}_I \mathbf{v} \quad (3.1a)$$

$${}^B \dot{\mathbf{R}}_I = {}^B \mathbf{R}_I \mathbf{S}(\omega) \quad (3.1b)$$

where

$${}^B \mathbf{R}_I = \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \quad (3.2)$$

and $\mathbf{S}(\cdot)$ is the skew-symmetric matrix defined as

$$\mathbf{S}(\omega) = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} \quad (3.3)$$

Let $\boldsymbol{\eta} = [v_1 \ v_2 \ \omega]$, and consider the class of under-actuated aquatic robot models with the following dynamic equations of motion:

$$\dot{\boldsymbol{\eta}} = \mathbf{H}(\boldsymbol{\eta}) + \mathbf{B}\mathbf{U} \quad (3.4)$$

where

$$\mathbf{H}(\boldsymbol{\eta}) = \begin{bmatrix} h_1(\boldsymbol{\eta}) \\ h_2(\boldsymbol{\eta}) \\ h_3(\boldsymbol{\eta}) \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \\ b_5 & b_6 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

u_1, u_2 represent the inputs, $b_1 - b_6$ are constants, and $h_1(\cdot), h_2(\cdot), h_3(\cdot)$ can be nonlinear functions of the velocity states.

Assumption 1 *The dynamic system described by Eq. (3.4) is input-to-state stable.*

Assumption 2 *The sway velocity is bounded such that $|v_2| \leq \bar{\vartheta}$ for some constant $\bar{\vartheta} > 0$.*

Assumption 3 *The submatrix*

$$\bar{b} = \begin{bmatrix} b_1 & b_2 \\ b_5 & b_6 \end{bmatrix} \quad (3.5)$$

is not singular.

3.2 Trajectory Tracking Control Algorithm

3.2.1 Trajectory Tracking Error Coordinates

The trajectory tracking problem involves the design of a control law that enables the robot to track a desired reference trajectory in the inertial frame, i.e., a geometric path with a specified time law, such that the robot's position and pose coincide with those of the desired trajectory. Figure 3.1 depicts the general idea.

Consider a robot with dynamics given by Eq. (3.1) and (3.4), and let the vectors $\bar{\mathbf{P}}(t)$ and $\bar{\mathbf{T}}(t)$ denote the robot's actual and desired position and pose with respect to the inertial frame $\{\mathbf{I}\}$ at a given time t , respectively, such that

$$\bar{\mathbf{P}} = \begin{bmatrix} x_p \\ y_p \\ \psi \end{bmatrix}, \quad \bar{\mathbf{T}} = \begin{bmatrix} x_r \\ y_r \\ \psi_r \end{bmatrix} \quad (3.6)$$

For ease of presentation, the time-dependence notation is omitted for the remainder of the paper.

Assume that the time evolution of $\bar{\mathbf{T}}$ is given by

$$\frac{d\bar{\mathbf{T}}}{dt} = \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\psi}_r \end{bmatrix} = \begin{bmatrix} v_r \cos \psi_r \\ v_r \sin \psi_r \\ \omega_r \end{bmatrix} \quad (3.7)$$

where v_r and ω_r are some desired surge and angular velocities, respectively. Note that in this work, we are interested in aligning the robot's heading angle ψ with the velocity vector $\mathbf{v}_r = [\dot{x}_r, \dot{y}_r]$, which lies tangent to the reference trajectory, i.e., $\psi_r = \arctan \frac{\dot{y}_r}{\dot{x}_r}$.

Furthermore, let $\mathbf{e} = [x_e \ y_e \ \psi_e]^T$ denote the tracking error vector expressed in the body-fixed frame such that

$$\mathbf{e} = \begin{pmatrix} {}^I\mathbf{R}_B & 0_{2 \times 1} \\ 0_{1 \times 2} & 1 \end{pmatrix} (\bar{\mathbf{P}} - \bar{\mathbf{T}}) \quad (3.8)$$

where ${}^I\mathbf{R}_B$ denotes the rotation matrix from the inertial frame {I} to the body-fixed frame {B}.

The derivative of \mathbf{e} (expressed in {B}) is given by

$$\frac{d\mathbf{e}}{dt} = \bar{\mathbf{S}} \begin{pmatrix} {}^I\mathbf{R}_B & 0_{2 \times 1} \\ 0_{1 \times 2} & 1 \end{pmatrix} (\bar{\mathbf{P}} - \bar{\mathbf{T}}) + \begin{pmatrix} {}^I\mathbf{R}_B & 0_{2 \times 1} \\ 0_{1 \times 2} & 1 \end{pmatrix} \left(\frac{d\bar{\mathbf{P}}}{dt} - \frac{d\bar{\mathbf{T}}}{dt} \right) \quad (3.9)$$

where

$$\bar{\mathbf{S}} = \begin{pmatrix} -\mathbf{S}(\omega) & 0_{2 \times 0} \\ 0_{1 \times 2} & 0 \end{pmatrix} \quad (3.10)$$

By solving for $\frac{d\mathbf{e}}{dt}$ from Eq. (3.9), and by augmenting the error state model with the robot's dynamics (3.4), one can obtain the following error state model

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\psi}_e \\ \dot{v}_1 \\ \dot{v}_2 \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v_1 - v_r \cos(\psi_e) + \omega y_e \\ v_2 + v_r \sin(\psi_e) - \omega x_e \\ \omega - \omega_r \\ h_1(v_1, v_2, \omega) + b_1 u_1 + b_2 u_2 \\ h_2(v_1, v_2, \omega) + b_3 u_1 + b_4 u_2 \\ h_3(v_1, v_2, \omega) + b_5 u_1 + b_6 u_2 \end{bmatrix} \quad (3.11)$$

Assumption 4 *The reference velocities v_r and ω_r are bounded and differentiable with bounded derivatives \dot{v}_r and $\dot{\omega}_r$. In particular, we assume $|\omega_r| \leq \omega_r^{max}$ and $0 < v_r \leq v_r^{max}$.*

By formulating the tracking problem in terms of the error dynamics, the trajectory tracking control problem has become a stabilization problem. In particular, the trajectory-tracking problem is to find a control law such that, for an arbitrary initial error, the states (x_e, y_e, ψ_e) of system (3.11) converges to a neighborhood of the origin.

3.2.2 Backstepping Control Design

In this subsection we synthesize the backstepping-based controller such that the tracking error states, x_e , y_e , and ψ_e , are stabilized at a neighborhood of the origin.

Let $e = [x_e, y_e, \psi_e]^T$ such that (3.11) is expressed as

$$\dot{e} = F(t, e) + L(e)\eta \quad (3.12)$$

$$\dot{\eta} = H(\eta) + BU \quad (3.13)$$

where

$$F(t, e) = \begin{bmatrix} -v_r \cos(\psi_e) \\ v_r \sin(\psi_e) \\ -\omega_r \end{bmatrix}, \quad L(e) = \begin{bmatrix} 1 & 0 & y_e \\ 0 & 1 & -x_e \\ 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

Note that the time dependence in the system above stems from the time-varying reference trajectory. From (3.12)-(3.13), one can see that this system can be viewed as a cascade connection of two subsystems, where the first subsystem is given by Eq. (3.12) with η as input, and the second subsystem is given by Eq. (3.13) with U as input. By treating η as a virtual control input for the stabilization of e , the objective of backstepping control is to design a state feedback control law, such that $e \rightarrow 0$ as $t \rightarrow \infty$. Ideally one would seek a state feedback law $\eta = \phi(e)$ such that the system

$$\dot{e} = F(t, e) + L(e)\phi(e)$$

is asymptotically stable. However, although one could find a choice of ϕ that would stabilize e at the origin, this would entail utilizing three virtual inputs (v_1, v_2, ω). Given the under-actuated nature of the dynamics of η , this would lead to an over-constrained problem as one would be attempting to manipulate three virtual inputs with only two actual inputs u_1 and u_2 . The under-actuation of the dynamics thus implies that only two virtual inputs are available to stabilize e . Alternatively, one could attempt to stabilize only two of the states at the origin, for example, x_e and y_e ; however, the third state, in this case, ψ_e , is not guaranteed to converge to the origin or even be bounded.

The choice of virtual inputs should also consider practicality. For example, choosing the sway velocity v_2 as a virtual input is impractical in reality because either it can lead to unstable whirling of the robot or it is not directly actuated (when b_3 and b_4 are zero); therefore, the most viable virtual input choices are v_1 and ω . It is natural to use v_1 to regulate the error in the longitudinal

axis (x_e), which leaves ω as a virtual input to drive the lateral (y_e) and heading angle (ψ_e) errors to the origin.

The control scheme proposed in [93] for a ship, overcame a similar limitation by introducing a new error state that couples the lateral and heading angle errors. By stabilizing the longitudinal tracking error along with this new state, the authors were able to stabilize the whole error system at the origin. Inspired by this work, we propose a similar approach; however, whereas [93] only provided a specific form of this new error state, in this work, we provide a systematic approach to its design for a general class of under-actuated dynamics. In particular, we establish the sufficient conditions needed to achieve closed-loop stability and via time-scale analysis, we provide further insight into the structure that this new error state needs in order to achieve ultimate boundedness of the error state system.

We define a new state, referred to as a “modified error” z_ξ , which is a function of y_e and ψ_e and couples them in such a way that the convergence of one implies the convergence of the other. This new state can be viewed intuitively as a “correction angle”, where essentially the heading angle error ψ_e is “corrected” with an angle that is correlated to the magnitude of the error y_e , such that it determines the rotation needed for the robot to point towards the desired trajectory. Following this intuition, let the “modified error” state be defined as $z_\xi = q(t, y_e, \psi_e) = q_1(t, y_e) + q_2(t, \psi_e)$. Let $q(\cdot, \cdot, \cdot)$ satisfy the following assumption:

Assumption 5 *The function $q(t, y_e, \psi_e)$ is continuously differentiable with respect to all its arguments, $q_2(t, \psi_e)$ is invertible with respect to ψ_e , the range of q_1 is bounded, and $q(t, 0, 0) = 0$. Furthermore, write the time derivative of $q(\cdot, \cdot, \cdot)$ as*

$$\begin{aligned} \frac{dq}{dt} &= \frac{\partial q_1(t, y_e)}{\partial y_e} (v_2 + v_r \sin(\psi_e) - \omega x_e) + \frac{\partial q_2(t, \psi_e)}{\partial \psi_e} (\omega - \omega_r) + \frac{\partial q(t, y_e, \psi_e)}{\partial t} \\ \frac{dq}{dt} &= \omega k(t, \mathbf{e}) + p(t, \mathbf{e}, v_2) \end{aligned}$$

where $p(t, \mathbf{e}, v_2) = \frac{\partial q_1(t, y_e)}{\partial y_e} (v_2 + v_r \sin(\psi_e)) - \frac{\partial q_2(t, \psi_e)}{\partial \psi_e} (\omega_r) + \frac{\partial q(t, y_e, \psi_e)}{\partial t}$,

$k(t, \mathbf{e}) = \frac{\partial q_2(t, \psi_e)}{\partial \psi_e} - \frac{\partial q_1(t, y_e)}{\partial y_e} x_e$. It is assumed that $k(t, \mathbf{e}) \neq 0 \quad \forall \quad t \geq 0, \forall \quad \mathbf{e} \in D$, in which $D \subset \mathbb{R}^n$ is a domain that contains the origin and the initial error state $\mathbf{e}(0)$.

Using this new error variable, the error system (3.12)-(3.13) is augmented to the following

$$\dot{\mathbf{e}} = F(t, \mathbf{e}) + L(\mathbf{e})\boldsymbol{\eta} \quad (3.15a)$$

$$\dot{z}_\xi = p(t, \mathbf{e}, v_2) + \omega k(t, \mathbf{e}) \quad (3.15b)$$

$$\dot{\boldsymbol{\eta}} = H(\boldsymbol{\eta}) + BU \quad (3.15c)$$

The function $q(\cdot, \cdot, \cdot)$ should be defined such that driving z_ξ and x_e to a neighborhood of the origin guarantees that the error system \mathbf{e} converges to a neighborhood of the origin. The reason and mechanism to why this is true will be evident later with the time-scale analysis.

To guarantee that the dynamics of \dot{x}_e and \dot{z}_ξ are asymptotically stable, a state feedback law $\phi(\cdot)$ needs to be designed. Let $\boldsymbol{\eta} = \phi(\cdot) = [\phi_1 \ 0 \ \phi_3]^T$, where ϕ_1 and ϕ_3 will be determined shortly. Let the candidate Lyapunov function for the \dot{x}_e, \dot{z}_ξ system be chosen as

$$V(x_e, z_\xi) = \frac{1}{2}x_e^2 + \frac{1}{2}z_\xi^2 \quad (3.16)$$

The derivative of V with respect to time is then given by

$$\dot{V}(x_e, z_\xi) = x_e(-v_r \cos(\psi_e) + \phi_1 + y_e \phi_3) + z_\xi(\phi_3 k(t, \mathbf{e}) + p(t, \mathbf{e}, v_2)) \quad (3.17)$$

By choosing

$$\phi_3(t, \mathbf{e}, v_2, z_\xi) = \frac{-p(t, \mathbf{e}, v_2) - K_{z_\xi} z_\xi}{k(t, \mathbf{e})} \quad (3.18a)$$

$$\phi_1(t, \mathbf{e}, v_2, z_\xi) = v_r \cos(\psi_e) - y_e \phi_3 - K_{x_e} x_e \quad (3.18b)$$

one can get

$$\dot{V}(x_e, z_\xi) = -K_{x_e} x_e^2 - K_{z_\xi} z_\xi^2 \quad (3.19)$$

where $K_{x_e} > 0$ and $K_{z_\xi} > 0$ are tuning parameters.

From the above it can be concluded that z_ξ and x_e will asymptotically converge to the origin so long as $\boldsymbol{\eta}$ is identical to ϕ . However, given that there is no direct control over the virtual inputs, another ‘step’ in the backstepping design must be performed such that the actual input U

appears. By adding and subtracting $L(\mathbf{e})\phi(t, \mathbf{e}, v_2, z_\xi)$ on the right hand side of Eq. (3.15a) and $k(t, \mathbf{e})\phi_3(t, \mathbf{e}, v_2, z_\xi)$ on the right hand side of Eq. (3.15b), one can obtain the following:

$$\dot{\mathbf{e}} = F(t, \mathbf{e}) + L(\mathbf{e})\phi(t, \mathbf{e}, v_2, z_\xi) + L(\mathbf{e})\left(\boldsymbol{\eta} - \phi(t, \mathbf{e}, v_2, z_\xi)\right) \quad (3.20a)$$

$$\dot{z}_\xi = p(t, \mathbf{e}, v_2) + k(t, \mathbf{e})\phi_3 + k(t, \mathbf{e})\left(\omega - \phi_3\right) \quad (3.20b)$$

$$\dot{\boldsymbol{\eta}} = H(\boldsymbol{\eta}) + BU \quad (3.20c)$$

Let the virtual input error be defined as

$$\begin{aligned} \mathbf{e}_\eta = \begin{bmatrix} e_{\eta_1} \\ e_{\eta_2} \\ e_{\eta_3} \end{bmatrix} &= \begin{bmatrix} v_1 - \phi_1 + y_e(\omega - \phi_3) \\ 0 \\ k(t, \mathbf{e})(\omega - \phi_3) \end{bmatrix} \\ &= \bar{Q}(t, \mathbf{e})(\boldsymbol{\eta} - \phi(t, \mathbf{e}, v_2, z_\xi)) \end{aligned} \quad (3.21)$$

where

$$\bar{Q}(t, \mathbf{e}) = \begin{bmatrix} 1 & 0 & y_e \\ 0 & 0 & 0 \\ 0 & 0 & k(t, \mathbf{e}) \end{bmatrix} \quad (3.22)$$

The time derivative of \mathbf{e}_η is given by

$$\begin{aligned} \dot{\mathbf{e}}_\eta = \begin{bmatrix} \dot{e}_{\eta_1} \\ \dot{e}_{\eta_2} \\ \dot{e}_{\eta_3} \end{bmatrix} &= \bar{Q}(t, \mathbf{e})\left(\bar{H}(\boldsymbol{\eta}) + \bar{B}U\right) + \\ &\underbrace{\left(\dot{\bar{Q}}(t, \mathbf{e})\left(\boldsymbol{\eta} - \phi(t, \mathbf{e}, v_2, z_\xi)\right) - \bar{Q}(t, \mathbf{e})\dot{\phi}(t, \mathbf{e}, v_2, z_\xi)\right)}_{\psi(t, \mathbf{e}, v_2, z_\xi)} \\ &= \bar{Q}(t, \mathbf{e})\bar{H}(\boldsymbol{\eta}) + \bar{Q}(t, \mathbf{e})\bar{B}U + \psi(t, \mathbf{e}, v_2, z_\xi) \end{aligned} \quad (3.23)$$

where

$$\begin{aligned}\bar{H}(\boldsymbol{\eta}) &= \begin{bmatrix} h_1(\boldsymbol{\eta}) \\ 0 \\ h_3(\boldsymbol{\eta}) \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} b_1 & b_2 \\ 0 & 0 \\ b_5 & b_6 \end{bmatrix} \\ \dot{\phi}_1(t, \mathbf{e}, v_2, z_\xi) &= \begin{bmatrix} \frac{\partial \phi_1}{\partial \mathbf{e}} & \frac{\partial \phi_1}{\partial v_2} & \frac{\partial \phi_1}{\partial z_\xi} \end{bmatrix} \begin{bmatrix} F(t, \mathbf{e}) + L(\mathbf{e})\boldsymbol{\eta} \\ h_2(\boldsymbol{\eta}) + b_3u_1 + b_4u_2 \\ \omega k(t, \mathbf{e}) + p(t, \mathbf{e}, v_2) \end{bmatrix} + \frac{\partial \phi_1}{\partial t} \\ \dot{\phi}_3(t, \mathbf{e}, v_2, z_\xi) &= \begin{bmatrix} \frac{\partial \phi_3}{\partial \mathbf{e}} & \frac{\partial \phi_3}{\partial v_2} & \frac{\partial \phi_3}{\partial z_\xi} \end{bmatrix} \begin{bmatrix} F(t, \mathbf{e}) + L(\mathbf{e})\boldsymbol{\eta} \\ h_2(\boldsymbol{\eta}) + b_3u_1 + b_4u_2 \\ \omega k(t, \mathbf{e}) + p(t, \mathbf{e}, v_2) \end{bmatrix} + \frac{\partial \phi_3}{\partial t}\end{aligned}\tag{3.24}$$

Furthermore, since v_1 and ω are the only virtual inputs to be directly manipulated, the sway velocity v_2 is an uncontrolled state, so we let it signify a perturbation to the system and define

$$\bar{\boldsymbol{\eta}} = \begin{bmatrix} 0 \\ \bar{\eta} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ v_2 \\ 0 \end{bmatrix}\tag{3.25}$$

To ease the control synthesis and the analysis of the closed-loop system, using Eqs. (3.20)-(3.23) and (3.25), we rewrite the error system as

$$\dot{\mathbf{e}} = F(t, \mathbf{e}) + L(\mathbf{e})\phi(t, \mathbf{e}, v_2, z_\xi) + \bar{L}(t, \mathbf{e})\mathbf{e}_\eta + \bar{\boldsymbol{\eta}}\tag{3.26a}$$

$$\dot{z}_\xi = p(t, \mathbf{e}, \bar{\eta}) + k(t, \mathbf{e})\phi_3 + e_{\eta_3}\tag{3.26b}$$

$$\dot{\mathbf{e}}_\eta = \bar{Q}(t, \mathbf{e})\bar{H}(\boldsymbol{\eta}) + \bar{Q}(t, \mathbf{e})\bar{B}U + \psi(t, \mathbf{e}, v_2, z_\xi)\tag{3.26c}$$

$$\dot{\bar{\eta}} = h_2(\boldsymbol{\eta}) + b_3u_1 + b_4u_2\tag{3.26d}$$

where

$$\bar{L}(t, \mathbf{e}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -\frac{x_e}{k(t, \mathbf{e})} \\ 0 & 0 & \frac{1}{k(t, \mathbf{e})} \end{bmatrix}\tag{3.27}$$

When \mathbf{e}_η is zero (implying the virtual inputs v_1 and ω equal ϕ_1 and ϕ_3 , respectively), x_e and z_ξ approach zero asymptotically as previously shown with the Lyapunov function $V(x_e, z_\xi)$.

We thus would like to drive \mathbf{e}_η asymptotically to the origin. To ease the control design, let $\bar{Q}\bar{B}U = (\boldsymbol{\mu} - \psi(t, \mathbf{e}, v_2, z_\xi))$, where $\boldsymbol{\mu} \in \mathbb{R}^{3 \times 1}$ is the nominal control input defined as

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ 0 \\ \mu_2 \end{bmatrix} \quad (3.28)$$

Note that since the second element of $\boldsymbol{\mu}$ as well as the second row of of the right hand side of (3.26c) are zero, a unique expression for $\boldsymbol{\mu}$ can be obtained using the definition above. With (3.28), the error system (3.26) then becomes

$$\dot{\mathbf{e}} = F(t, \mathbf{e}) + L(\mathbf{e})\phi(t, \mathbf{e}, v_2, z_\xi) + \bar{L}(t, \mathbf{e})\mathbf{e}_\eta + \bar{\boldsymbol{\eta}} \quad (3.29a)$$

$$\dot{z}_\xi = p(t, \mathbf{e}, \bar{\boldsymbol{\eta}}) + k(t, \mathbf{e})\phi_3 + e_{\eta_3} \quad (3.29b)$$

$$\dot{\mathbf{e}}_\eta = \bar{Q}(t, \mathbf{e})\bar{H}(\boldsymbol{\eta}) + \boldsymbol{\mu} \quad (3.29c)$$

$$\dot{\boldsymbol{\eta}} = h_2(\boldsymbol{\eta}) + b_3u_1 + b_4u_2 \quad (3.29d)$$

Considering the system composed of $x_e, z_\xi, \mathbf{e}_\eta$, we define a new Lyapunov function V_2 as

$$V_2(x_e, z_\xi, \mathbf{e}_\eta) = V(x_e, z_\xi) + \frac{1}{2}\mathbf{e}_\eta^T \mathbf{e}_\eta \quad (3.30)$$

with its time derivative given by

$$\begin{aligned} \dot{V}_2 &= \left[\frac{\partial V(x_e, z_\xi, \mathbf{e}_\eta)}{\partial x_e} \quad \frac{\partial V(x_e, z_\xi, \mathbf{e}_\eta)}{\partial z_\xi} \right] \begin{bmatrix} -v_r \cos(\psi_e) + \phi_1 + \phi_3 y_e \\ p(t, \mathbf{e}, \bar{\boldsymbol{\eta}}) + k(t, \mathbf{e})\phi_3 \end{bmatrix} \\ &\quad + \left[\frac{\partial V(x_e, z_\xi, \mathbf{e}_\eta)}{\partial x_e} \quad \frac{\partial V(x_e, z_\xi, \mathbf{e}_\eta)}{\partial z_\xi} \right] \begin{bmatrix} e_{\eta_1} \\ e_{\eta_3} \end{bmatrix} + \mathbf{e}_\eta^T \dot{\mathbf{e}}_\eta \\ &= -K_{x_e}x_e^2 - K_{z_\xi}z_\xi^2 + x_e e_{\eta_1} + z_\xi e_{\eta_3} + \mathbf{e}_\eta^T (\bar{Q}(t, \mathbf{e})\bar{H}(\boldsymbol{\eta}) + \boldsymbol{\mu}) \end{aligned} \quad (3.31)$$

Let

$$\boldsymbol{\mu} = -\bar{Q}(t, \mathbf{e})\bar{H}(\boldsymbol{\eta}) - K_{\mathbf{e}_\eta} \mathbf{e}_\eta \quad (3.32)$$

where

$$K_{\mathbf{e}_\eta} = \begin{bmatrix} K_{e_{\eta_1}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & K_{e_{\eta_3}} \end{bmatrix}$$

which yields

$$\dot{V}_2 = -K_{x_e}x_e^2 - K_{z_\xi}z_\xi^2 + x_e e_{\eta_1} + z_\xi e_{\eta_3} - (K_{e_{\eta_1}}e_{\eta_1}^2 + K_{e_{\eta_3}}e_{\eta_3}^2) \quad (3.33)$$

After adding and subtracting $\frac{e_{\eta_1}^2}{4K_{x_e}}$ and $\frac{e_{\eta_3}^2}{4K_{z_\xi}}$ and completing the square, one can arrive at

$$\begin{aligned} \dot{V}_2 \leq & -K_{x_e}\left(x_e - \frac{e_{\eta_1}}{2K_{x_e}}\right)^2 - K_{z_\xi}\left(z_\xi - \frac{e_{\eta_3}}{2K_{z_\xi}}\right)^2 \\ & - e_{\eta_1}^2\left(K_{e_{\eta_1}} - \frac{1}{4K_{x_e}}\right) - e_{\eta_3}^2\left(K_{e_{\eta_3}} - \frac{1}{4K_{z_\xi}}\right) \end{aligned} \quad (3.34)$$

If $K_{x_e} > 0$, $K_{z_\xi} > 0$, $K_{e_{\eta_1}} > \frac{1}{4K_{x_e}}$ and $K_{e_{\eta_3}} > \frac{1}{4K_{z_\xi}}$, then $\dot{V}_2 < 0$ unless $x_e = z_\xi = e_{\eta_1} = e_{\eta_3} = 0$.

From Lyapunov theory ([129]), one can conclude the convergence of $(x_e, z_\xi, e_{\eta_1}, e_{\eta_3})$ to the origin.

Finally, let $\psi = [\psi_1 \ 0 \ \psi_2]^T$ such that the input U is given by

$$U = C(t, \mathbf{e}) \begin{bmatrix} \mu_1 - \psi_1 \\ \mu_2 - \psi_2 \end{bmatrix} \quad (3.35)$$

where $C(t, \mathbf{e})$ is given by

$$C(t, \mathbf{e}) = \begin{bmatrix} \frac{b_6}{b_1b_6 - b_2b_5} & -\frac{(b_2 + b_6y_e)}{k(t, \mathbf{e})(b_1b_6 - b_2b_5)} \\ -\frac{b_5}{b_1b_6 - b_2b_5} & \frac{(b_1 + b_5y_e)}{k(t, \mathbf{e})(b_1b_6 - b_2b_5)} \end{bmatrix} \quad (3.36)$$

Remark 1 Note that Assumption 3 guarantees that $b_1b_6 - b_2b_5$ is nonzero.

Theorem 3.2.1 Consider the error dynamics given by (3.11). If the system dynamics satisfy Assumptions 1 and 3 and the desired trajectory satisfies Assumption 4, then the backstepping control law (3.35) guarantees the convergence of the longitudinal tracking error x_e and the modified error z_ξ to the origin. Furthermore, if the system dynamics satisfies Assumptions 2, then (x_e, y_e, ψ_e) from the error system (3.11) are stabilized at a neighborhood of the origin and are ultimately bounded as

$$\|\mathbf{e}\| \leq \max\{\beta(\|\mathbf{e}(t_0)\|, t - t_0), \rho(\xi)\}, \quad \forall t \geq t_0 \quad (3.37)$$

where $\beta(\|\mathbf{e}(t_0)\|, t - t_0)$ is a class \mathcal{KL} function, $\rho(\xi)$ is a class \mathcal{K} function and ξ is an upper bound to be determined.

Remark 2 Note that typically for any set of inputs u_1 and u_2 , $|v_2| \leq |v_1|$ is satisfied in the absence of external disturbances. This assumption is often made in literature and it is reasonable in practice as many aquatic robotic systems satisfy this condition when no disturbances are present. Relevant examples include robotic fish [89] and under-actuated ships/surface vessels [130]. Therefore, the disturbance that arises from v_2 is expected to be relatively small.

Note that the backstepping controller guarantees the convergence of x_e and z_ξ to the origin. The following analysis in Section 3.3 shows how the rest of the error states y_e and ψ_e are stabilized.

3.3 Stability Analysis of the Closed-Loop System

We first show how the error system (3.11) is stable at the origin under the assumption that the perturbation arising from the uncontrolled state v_2 is identically zero, that is $\bar{\eta}(0) = 0$ and $\dot{\bar{\eta}} = 0$. The case of v_2 not being constantly zero will be considered in a later part of the stability analysis. With the control law μ substituted into Eq. (3.26), the closed-loop system is obtained as follows:

$$\dot{y}_e = v_r \sin(\psi_e) - x_e \phi_3 - \frac{x_e e_{\eta 3}}{k(t, y_e, z_\xi, x_e)} \quad (3.38a)$$

$$\dot{\psi}_e = -\omega_r + \phi_3 + \frac{e_{\eta 3}}{k(t, y_e, z_\xi, x_e)} \quad (3.38b)$$

$$\dot{x}_e = -K_{x_e} x_e + e_{\eta 1} \quad (3.38c)$$

$$\dot{z}_\xi = -K_{z_\xi} z_\xi + e_{\eta 3} \quad (3.38d)$$

$$\dot{e}_\eta = -K_{e_\eta} e_\eta \quad (3.38e)$$

$$\dot{\bar{\eta}} = 0 \quad (3.38f)$$

Eq. (3.38) allows one to see the time-scale structure of the closed-loop system as it is evident that e_η , z_ξ , and x_e are “fast” converging states if K_{e_η} , K_{z_ξ} and K_{x_e} are chosen adequately large. Inspired by this, we utilize singular perturbation methods for nonautonomous systems [131] to analyze the stability properties of the closed-loop system.

3.3.1 Asymptotic Stability of Multi-Time-Scale Systems: The Setup

In general, multi-time-scale analysis involves the decomposition of a system into stages dictated by the separation of different time scales. In particular, the system is decomposed into a “slow” reduced model that dominates the system behavior and a “fast” boundary layer which is composed of the states that evolve in faster time scales and represent deviations from the slow system. Following this intuition, let $\chi = [\chi_1 \ \chi_2 \ \chi_3 \ \chi_4]^T = [z_\xi \ x_e \ e_{\eta_1} \ e_{\eta_3}]^T$ and $\epsilon = \frac{1}{\max(K_{z_\xi}, K_{x_e}, K_{e_{\eta_1}}, K_{e_{\eta_3}})}$ such that the closed-loop system (3.38), excluding (3.38b), is transformed into a two-time scale singular perturbation model of the following form:

$$\dot{y}_e = f(t, y_e, \chi) \quad (3.39)$$

$$\epsilon \dot{\chi} = g(t, y_e, \chi, \epsilon) \quad (3.40)$$

$$\dot{\eta} = 0 \quad (3.41)$$

where

$$f(t, y_e, \chi) = V_r \sin(\psi_e) - \chi_2 \phi_3 - \frac{\chi_2 \chi_4}{k(t, y_e, \chi)} \quad (3.42a)$$

$$g(t, \chi, \epsilon) = \begin{bmatrix} -\bar{a}\chi_1 + \epsilon\chi_4 \\ -\bar{b}\chi_2 + \epsilon\chi_3 \\ -\bar{c}\chi_3 \\ -\bar{d}\chi_4 \end{bmatrix} \quad (3.42b)$$

$$= \underbrace{\begin{bmatrix} -\bar{a} & 0 & 0 & \epsilon \\ 0 & -\bar{b} & \epsilon & 0 \\ 0 & 0 & -\bar{c} & 0 \\ 0 & 0 & 0 & -\bar{d} \end{bmatrix}}_A \begin{bmatrix} \chi_1 \\ \chi_2 \\ \chi_3 \\ \chi_4 \end{bmatrix} \quad (3.42c)$$

where $\bar{a} = \frac{K_{z_\xi}}{\max(K_{z_\xi}, K_{x_e}, K_{e_{\eta_1}}, K_{e_{\eta_3}})}$, $\bar{b} = \frac{K_{x_e}}{\max(K_{z_\xi}, K_{x_e}, K_{e_{\eta_1}}, K_{e_{\eta_3}})}$, $\bar{c} = \frac{K_{\eta_1}}{\max(K_{z_\xi}, K_{x_e}, K_{e_{\eta_1}}, K_{e_{\eta_3}})}$, $\bar{d} = \frac{K_{\eta_3}}{\max(K_{z_\xi}, K_{x_e}, K_{e_{\eta_1}}, K_{e_{\eta_3}})}$, y_e is the slow variable, and χ is the fast variable. Furthermore, consider $\epsilon \ll 1$. Eqs. (3.39)-(3.40) represent the singularly perturbed full system, which for simplicity is denoted as the Υ_{SF} system. Similarly, the different sub-index combinations in Υ_*

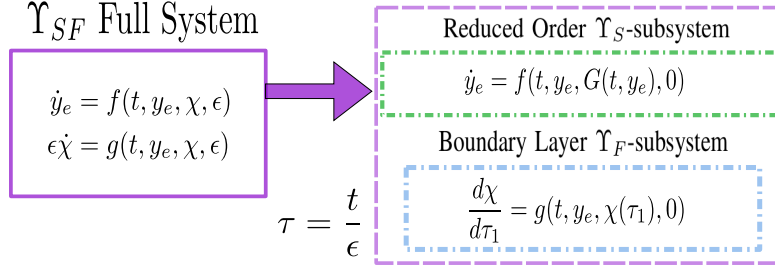


Figure 3.2: The *bottom-up* approach to the analysis of two-time-scale system.

denote the different subsystems. For example, Υ_S denotes the slow subsystem. Note that the state ψ_e has not been included in the perturbation model, since we exploit its relationship with the error state y_e and the new error coordinate z_ξ , and instead analyze the evolution of z_ξ . In other words, we utilize the relationship $z_\xi = q_1(t, y_e) + q_2(t, \psi_e)$, which is enforced via the backstepping feedback law.

By considering the singular perturbation parameter ϵ , the Υ_{SF} system can be broken into a pair of slow and fast subsystems, and its time-scale properties can be analyzed in an ascending manner. Figure 3.2 illustrates this methodology. The standard two-time scale stability analysis is based on the following result.

Lemma 1 ([131]) *Consider the singularly perturbed system*

$$\begin{aligned}\dot{\zeta}(t) &= f(t, \zeta, \gamma, \epsilon), \quad \zeta \in \mathbb{R}^n \\ \epsilon \dot{\gamma} &= g(t, \zeta, \gamma, \epsilon), \quad \gamma \in \mathbb{R}^m\end{aligned}\tag{3.43}$$

Assume that the following assumptions are satisfied for all

$$(t, \zeta, \gamma, \epsilon) \in [0, \infty) \times B_\zeta \times B_\gamma \times [0, \epsilon_0], \quad B_\zeta \subset \mathbb{R}^n, \quad B_\gamma \subset \mathbb{R}^m, \text{ and } 0 < \epsilon_0 \in \mathbb{R}$$

$$1. \quad f(t, 0, 0, \epsilon) \text{ and } g(t, 0, 0, \epsilon) = 0.$$

2. *The equation*

$$g(t, \zeta, \gamma, 0) = 0$$

has an isolated root $\gamma = G(t, \zeta)$, and there exists a class \mathcal{K} function p_1 such that

$$\|G(t, \zeta)\| \leq p_1(\|\zeta\|)$$

3. The functions f, g , and G are smooth and their partial derivatives up to the second order are bounded for $\gamma - G(t, \zeta) \in B_\gamma$.

4. The origin of the reduced order system

$$\dot{\zeta}(t) = f(t, \zeta, G(t, \zeta), 0)$$

is asymptotically stable; there exists a positive-definite Lyapunov function candidate V_R that satisfies

$$0 < s_1(\|\zeta\|) \leq V_R(t, \zeta) \leq s_2(\|\zeta\|)$$

for some class \mathcal{K} functions $s_1(\cdot)$ and $s_2(\cdot)$, and

$$\frac{\partial V_R(t, \zeta)}{\partial t} + \frac{\partial V_R(t, \zeta)}{\partial \zeta} f(t, \zeta, G(t, \zeta), 0) \leq -\alpha'_1 \Psi^2(\zeta)$$

where $\alpha'_1 > 0$ and $\Psi(\cdot)$ is a continuous scalar function of ζ that vanishes only at $\zeta = 0$.

5. The origin of the boundary-layer system

$$\frac{d\beta}{d\tau} = g(t, \zeta, \beta + G(t, \zeta), 0)$$

where $\beta = \gamma - G(t, \zeta)$ and $\tau = \frac{(t - t_0)}{\epsilon}$, is asymptotically stable at the origin, uniformly in (t, ζ) ; there exists a Lyapunov function candidate $W(t, \zeta, \gamma)$, that satisfies

$$0 < s_3(\|\gamma - G(t, \zeta)\|) \leq W(t, \zeta, \gamma) \leq s_4(\|\gamma - G(t, \zeta)\|)$$

for some class \mathcal{K} functions $s_3(\cdot)$ and $s_4(\cdot)$, and that satisfies the following:

$$\frac{\partial W(t, \zeta, \gamma)}{\partial \gamma} g(t, \zeta, \gamma, 0) \leq -\alpha'_2 \Phi^2(\gamma - G(t, \zeta)),$$

where $\alpha'_2 > 0$ and $\Phi(\cdot)$ is a continuous function of an \mathbb{R}^m vector z_f which vanishes only at $z_f = 0$.

6. $V_R(t, \zeta)$ satisfies the following interconnection condition:

$$\frac{\partial V_R}{\partial \zeta} [f(t, \zeta, \gamma, \epsilon) - f(t, \zeta, G(t, \zeta), 0)] \leq \beta'_1 \Psi(\zeta) \Phi(\gamma - G(t, \zeta)) + \epsilon \lambda'_1 \Psi^2(\zeta)$$

where the constants β'_1 and λ'_1 are nonnegative.

7. $W(t, \zeta, \gamma)$ satisfies the second and third interconnection conditions:

$$\frac{\partial W}{\partial \gamma} [g(t, \zeta, \gamma, \epsilon) - g(t, \zeta, G(t, \zeta), 0)] \leq \epsilon \lambda'_2 \Phi(\gamma - G(t, \zeta)) + \epsilon \beta'_2 \Psi(\zeta)' \Phi(\gamma - G(t, \zeta))$$

$$\frac{\partial W}{\partial t} + \frac{\partial W}{\partial \zeta} f(t, \zeta, \gamma, \epsilon) \leq \lambda'_3 \Phi^2(\gamma - G(t, \zeta)) + \beta'_3 \Psi(\zeta) \Phi(\gamma - G(t, \zeta))$$

where the constants $\lambda'_2, \lambda'_3, \beta'_2$, and β'_3 are nonnegative.

Then there exists $\epsilon^* > 0$ such that for all $\epsilon < \epsilon^*$, the origin of (3.43) is uniformly asymptotically stable. Moreover, there exists a composite Lyapunov function defined as

$$v(t, \zeta, \gamma) = (1 - d')V_R(t, \zeta) + d'W(t, \zeta, \gamma), \quad 0 < d' < 1$$

such that

$$\dot{v}(t, \zeta, \gamma) \leq - \left[\begin{array}{c} \Psi(\zeta) \\ \Phi(\gamma - G(t, \zeta)) \end{array} \right]^T \Lambda \left[\begin{array}{c} \Psi(\zeta) \\ \Phi(\gamma - G(t, \zeta)) \end{array} \right]$$

where

$$\Lambda = \left[\begin{array}{cc} (1 - d')(\alpha'_1 - \epsilon \lambda'_1) & -\frac{1}{2}(1 - d')\beta'_1 - \frac{1}{2}d'(\beta'_2 + \beta'_3) \\ -\frac{1}{2}(1 - d')\beta'_1 - \frac{1}{2}d'(\beta'_2 + \beta'_3) & \frac{d'}{\epsilon}(\alpha'_2 - \epsilon(\lambda'_2 + \lambda'_3)) \end{array} \right] \quad (3.44)$$

which is positive definite for

$$\epsilon < \frac{\alpha'_1 \alpha'_2}{\alpha'_1(\lambda'_2 + \lambda'_3) + \alpha'_2 \lambda'_1 + \frac{1}{4(1 - d')d'}[(1 - d')\beta'_1 + d'(\beta'_2 + \beta'_3)]^2} \stackrel{\text{def}}{=} \epsilon_d \quad (3.45)$$

The maximum of ϵ_d is given by

$$\epsilon_d|_{d'=\beta'_1/(\beta'_1+(\beta'_2+\beta'_3))} = \epsilon^* = \frac{\alpha'_1 \alpha'_2}{\alpha'_1(\lambda'_2 + \lambda'_3) + \alpha'_2 \lambda'_1 + \beta'_1(\beta'_2 + \beta'_3)}$$

The Υ_{SF} subsystem is formed by the reduced-order slow-subsystem Υ_S and the four-dimensional fast Υ_F boundary layer. Let $G(t, y_e)$ denote the unique equilibrium point of $g(t, y_e, G(t, y_e), 0)$ when $\epsilon = 0$, such that

$$0 = g(t, y_e, \chi, \epsilon) \rightarrow \chi = G(t, y_e) = 0 \quad (3.46)$$

By considering the stretched time scale defined by $\tau_1 = (t - t_0)/(\epsilon)$, the slow model is given by the Υ_S

$$\dot{y}_e = f(t, y_e, G(t, y_e)) = v_r \sin(\psi_e) \quad (3.47)$$

Subsystem	Condition
Assumption 4 of Lemma 1: Reduced System	
Υ_{SF}	$\frac{\partial V_S(t, y_e)}{\partial y_e} f(t, y_e, G(t, y_e), 0) + \frac{\partial V_S(t, y_e)}{\partial t} \leq -\alpha_1 \Psi_1^2$
Assumption 5 of Lemma 1: Boundary-Layer	
Υ_{SF}	$\frac{\partial V_F(t, x)}{\partial x} g(t, y_e, \chi, 0) \leq -\alpha_2 \Phi_1^2$
Assumption 6 of Lemma 1: Interconnection Condition 1	
Υ_{SF}	$\frac{\partial V_S(t, y_e)}{\partial y_e} \{ [f(t, y_e, \chi, \epsilon) - f(t, y_e, G(t, y_e), 0)] \} \leq \beta_1 \Psi_1 \Phi_1 + \epsilon \lambda_1 \Psi_1^2$
Assumption 7.a of Lemma 1: Interconnection Condition 2	
Υ_{SF}	$\frac{\partial V_F(t, \chi)}{\partial y_e} [g(t, y_e, \chi, \epsilon) - g(t, y_e, G(t, y_e), 0)] \leq \epsilon \lambda_2 \Phi_1 + \epsilon \beta_2 \Psi_1 \Phi_1$
Assumption 7.b of Lemma 1: Interconnection Condition 3	
Υ_{SF}	$\frac{\partial V_F(t, \chi)}{\partial y_e} f(t, y_e, \chi, \epsilon) + \frac{\partial V_F(t, x)}{\partial t} \leq \lambda_3 \Phi_1^2 + \beta_3 \Psi_1 \Phi_1$

Table 3.1: Comparison functions and inequalities that guarantee the asymptotic stability requirements for the two-time scale system.

while the fast model or boundary layer for the Σ_{SF} -subsystem is given by the Σ_F -subsystem as

$$\frac{d\chi}{d\tau_1} = g(t, y_e, \chi(\tau_1), 0) = \begin{bmatrix} -\bar{a}\chi_1 \\ -\bar{b}\chi_2 \\ -\bar{c}\chi_3 \\ -\bar{d}\chi_4 \end{bmatrix} \quad (3.48)$$

Recall that one of the fast variables, z_ξ , denotes the “modified error” defined as $z_\xi = q_1(t, y_e) + q_2(t, \psi_e)$. At this time-scale stage, z_ξ reaches equilibrium faster than y_e , i.e., $z_\xi \rightarrow 0$ which implies that $q_2(t, \psi_e) \rightarrow -q_1(t, y_e)$. Since $q_2(t, \psi_e)$ satisfies Assumption 5, this suggests that $\psi_e \rightarrow -q_2^{-1}(q_1(t, y_e))$, and implies that once z_ξ has reached equilibrium, ψ_e is forced to move in a manifold defined by $\psi_e = -q_2^{-1}(q_1(t, y_e))$ (enforced by the backstepping controller). By considering that ψ_e is constrained to evolve on this manifold, Eq. (3.47) can thus be rewritten as

$$\dot{y}_e = f(t, y_e, G(t, y_e)) \quad (3.49a)$$

$$= -v_r \sin(q_2^{-1}(q_1(t, y_e))) \quad (3.49b)$$

One can then carry out the stability analysis on the subsystem composed of the slow variable y_e and the fast variable z_ξ with ψ_e subject to evolving on the manifold defined by $-q_2^{-1}(q_1(t, y_e))$. Note this means that the convergence of ψ_e is implied once y_e converges.

3.3.2 Lyapunov Function Candidates and Two-Time-Scale Analysis

As seen from Lemma 1, the asymptotic stability analysis requires the existence of Lyapunov functions for each of the subsystems. These Lyapunov functions, in turn, will be utilized to form a composite Lyapunov function for the complete Υ_{SF} system.

To choose the Lyapunov functions at each stage, we consider each of the subsystems in a descending manner. In particular, we first consider the reduced-order slow Υ_S -subsystem given by Eq. (3.49), and let $V_S(t, y_e)$ represent its Lyapunov function, where the exact structure of $V_S(t, y_e)$ is dependent upon the choice of the q_1 and q_2 functions. Note that $V_S(t, y_e)$ is allowed to depend on time. An example of a candidate Lyapunov function will be given briefly. On the other hand, for the corresponding fast Υ_F -subsystem (3.48), a natural choice is

$$V_F(\chi) = \chi^T P \chi \quad (3.50)$$

where P is a real, symmetric positive-definite matrix and the solution to the Lyapunov equation $PA + A^T P = -S$ where $S = S^T > 0$. Note that since A is Hurwitz, P is the unique solution of the Lyapunov equation (Theorem 3.7 [129]).

Considering the above Lyapunov functions, one can then establish asymptotic stability of the origin by satisfying assumptions (1)-(7) of Lemma 1. Table 3.1 summarizes the equivalent assumptions by substituting the corresponding system dynamics.

It can be verified that $(0, 0)$ is an equilibrium of the system given by Eqs. (3.39)-(3.40). The time derivative of $V_S(t, y_e)$ is given

$$\dot{V}_S(t, y_e) = -\frac{\partial V_S}{\partial y_e} v_r \sin(q_2^{-1}(q_1(t, y_e))) + \frac{\partial V_S}{\partial t} \quad (3.51)$$

Depending on the choice of q_1 and q_2 , $V_S(t, y_e)$ can be determined such that $\dot{V}_S(t, y_e)$ is shown to be negative definite. In general, with the appropriate choice of α_1 and Ψ_1 , Assumption 4 of

Lemma 1 can be satisfied. Take for example $V_S(y_e) = (1 - \cos(q_2^{-1}(q_1(y_e))))$, where q_1 and q_2 are chosen to be independent of t , in which case

$$\dot{V}_S(y_e) = -v_r \sin^2(q_2^{-1}(q_1(y_e))) \frac{\partial(q_2^{-1}(q_1(y_e)))}{\partial y_e} \quad (3.52)$$

As a simple choice, let $q_2 = K_e \psi_e$, with $K_e > 0$, such that

$$\dot{V}_S(y_e) = -v_r \sin^2\left(\frac{q_1(y_e)}{K_e}\right) \frac{\partial(q_1(y_e))}{K_e \partial y_e} \quad (3.53)$$

By choosing q_1 such that $\frac{\partial(q_1(y_e))}{\partial y_e} > 0$, $\dot{V}_S(y_e)$ can be made negative definite. From Assumption 5, q_1 must be chosen such that it is continuously differentiable and well defined for all real numbers of y_e and has a bounded range. $\arcsin(\frac{y_e}{\sqrt{1+y_e^2}})$ is one viable option that would satisfy all the aforementioned assumptions. Note that $\arcsin(\cdot)$ has a limited domain; however, its argument is restricted since $-1 \leq \frac{y_e}{\sqrt{1+y_e^2}} \leq 1$. Let $q_1 = K_d \arcsin(\frac{y_e}{\sqrt{1+y_e^2}})$ such that

$$\dot{V}_S(y_e) = -v_r \frac{K_d}{K_e} \sin^2\left(\frac{K_d}{K_e} \arcsin\left(\frac{y_e}{\sqrt{1+y_e^2}}\right)\right) \left(\frac{1}{1+y_e^2}\right) \quad (3.54a)$$

$$\leq -v_r^{\min} \frac{K_d}{K_e} \sin^2\left(\frac{K_d}{K_e} \arcsin\left(\frac{y_e}{\sqrt{1+y_e^2}}\right)\right) \left(\frac{1}{1+y_e^2}\right) \quad (3.54b)$$

where $K_d > 0$ can be thought of as a scale to determine the effect of the “correction” term. $k(t, e) = \frac{\partial q_2(t, \psi_e)}{\partial \psi_e} - \frac{\partial q_1(t, y_e)}{\partial y_e} x_e = K_e - \frac{K_d x_e}{1+y_e^2}$ satisfies $k(t, e) \neq 0 \quad \forall \quad t \geq 0, \forall \quad |x_e| < \frac{K_e}{K_d}$. In this example, Assumption 4 of Lemma 1 can be satisfied with $\alpha_1 \geq v_r^{\min} \frac{K_d}{K_e}$ and $\Psi_1 = \sin^2\left(\frac{K_d}{K_e} \arcsin\left(\frac{y_e}{\sqrt{1+y_e^2}}\right)\right) \left(\frac{1}{1+y_e^2}\right)$. For another example we refer the reader to Section 3.4.3.

Using Eq. (3.50), it can be shown that by letting $\alpha_2 \geq \bar{\lambda}_{\min}(S)$, $\Phi_1 = \|\chi\|$, where $\bar{\lambda}_{\min}(S)$ is the smallest eigenvalue of S , Assumption 5 of Lemma 1 can be satisfied. Furthermore, with the appropriate choice β_1 and $\lambda_1 = 0$, Assumption 6 of Lemma 1 can also be satisfied. Finally, with $\lambda_2 = 0$, $\beta_2 = 0$, $\lambda_3 = 0$, and $\beta_3 = 0$, Assumption 7.a and Assumption 7.b of Lemma 1 can be satisfied.

A composite Lyapunov function for the Υ_{SF} -subsystem is given by

$$V_1(t, y_e, \chi) = (1 - d_1) V_S(t, y_e) + d_1 V_F(\chi) \quad (3.55)$$

with

$$\dot{V}_1(t, y_e, \chi) = - \begin{bmatrix} \Psi_1(y_e) \\ \Phi_1(\chi) \end{bmatrix}^T \Lambda_1 \begin{bmatrix} \Psi_1(y_e) \\ \Phi_1(\chi) \end{bmatrix} \leq -\lambda(\Lambda_1) \underbrace{\left\| \begin{bmatrix} \Psi_1(y_e) \\ \Phi_1(\chi) \end{bmatrix} \right\|^2}_{\sigma^2}$$

where Λ is given by

$$\Lambda = \begin{bmatrix} (1-d_1)\alpha_1 & -\frac{1}{2}(1-d_1)\beta_1 \\ -\frac{1}{2}(1-d_1)\beta_1 & d_1(\frac{\alpha_2}{\epsilon}) \end{bmatrix}$$

and $\lambda(\Lambda)$ is its smallest eigenvalue of Λ . The stability of the two time-scale Υ_{SF} -subsystem is then guaranteed for

$$\epsilon < \frac{4d_1\alpha_1\alpha_2}{(1-d_1)\beta_1^2} \stackrel{\text{def}}{=} \epsilon_d$$

3.3.2.1 Asymptotic Stability of the Closed-Loop Perturbed System

Throughout the stability analysis done so far, it has been assumed that the perturbation $\bar{\eta}$ that results from the uncontrolled sway velocity v_2 is identically zero. However, to complete the proof of Theorem 3.2.1 and show how the error system (3.11) is ultimately bounded we incorporate v_2 as a disturbance and treat the resultant closed-loop system as a disturbed system. Using the perturbed system theory as presented in [129] (Lemma 4.4), stability analysis can be carried out.

Let the perturbed Υ_{SF} system be expressed as

$$\dot{\Sigma} = Q(t, \Sigma) + r(t, \Sigma) = \begin{bmatrix} f(t, \Sigma) \\ g(t, \Sigma) \\ 0 \end{bmatrix} + r(t, \Sigma) \quad (3.56)$$

where $\Sigma = [y_e, \chi, \bar{\eta}]^T$

$$f(t, \Sigma) = \left[v_r \sin(\psi_e) - \frac{\chi_2(-\bar{p}(t, y_e, \chi) - k_{z\xi}\chi_1)}{k(t, y_e, \chi)} - \frac{\chi_2\chi_3}{k(t, y_e, \chi)} \right] \quad (3.57)$$

$$g(t, \Sigma) = \begin{bmatrix} -\bar{a}\chi_1 + \epsilon\chi_4 \\ -\bar{b}\chi_2 + \epsilon\chi_3 \\ -\bar{c}\chi_3 \\ -\bar{d}\chi_4 \end{bmatrix} \quad (3.58)$$

$$r(t, \Sigma) = \begin{bmatrix} -\frac{\partial q_1(t, y_e)}{\partial y_e} \bar{\eta} \chi_2 \\ \frac{1}{k(t, y_e, \chi)} + \bar{\eta} \\ 0 \\ 0 \\ 0 \\ 0 \\ h_2(v_1, \bar{\eta}, \omega) + b_3 u_1 + b_4 u_2 \end{bmatrix} \quad (3.59)$$

$$\bar{p}(t, y_e, \chi_1, \chi_2) = \frac{\partial q_1(t, y_e, \chi)}{\partial y_e} v_r \sin(\psi_e) - \frac{\partial q_2(t, y_e, \chi)}{\partial \psi_e} \omega_r + \frac{\partial q(t, y_e, \chi)}{\partial t} \quad (3.60)$$

Note that in this manner the nominal system is represented by $Q(t, \Sigma)$ while the perturbation is denoted by $r(t, \Sigma)$. In the previous section it was shown that the nominal system given by $\dot{\Sigma} = Q(\Sigma)$ is asymptotically stable. Furthermore, a composite Lyapunov function $V_1(t, \Sigma)$ for the full nominal system is found, for which it can be shown that

$$\begin{aligned} \Gamma_1(\|\Sigma\|) &\leq V_1(t, \Sigma) \leq \Gamma_2(\|\Sigma\|), \\ \frac{\partial V_1}{\partial \Sigma} Q(t, \Sigma) + \frac{\partial V_1}{\partial t} &\leq -\Gamma_3(\|\Sigma\|), \left\| \frac{\partial V_1}{\partial \Sigma} \right\| \leq \varsigma \end{aligned} \quad (3.61)$$

for all $\Sigma \in B_R$, where $B_R \subset \mathbb{R}^n$ and $\Gamma_i, i = 1, 2, 3$ are class \mathcal{K} functions.

Considering the same Lyapunov function V_1 for the perturbed system, we can express the derivative of V_1 along the trajectories of (3.56) as

$$\dot{V}_1(t, \Sigma) = \frac{\partial V_1}{\partial t} + \frac{\partial V_1}{\partial \Sigma} Q(\Sigma) + \frac{\partial V_1}{\partial \Sigma} r(t, \Sigma) \quad (3.62a)$$

$$\leq -\Gamma_3(\|\Sigma\|) + \left\| \frac{\partial V_1}{\partial \Sigma} \right\| \|r(t, \Sigma)\| \quad (3.62b)$$

Suppose that the perturbation term satisfies the bound

$$\|r(\Sigma, t)\| \leq \varphi < \frac{\theta \Gamma_3(\Gamma_2^{-1}(\Gamma_1(R)))}{\varsigma} \quad (3.63)$$

for all $t \geq 0$ and $\Sigma \in B_R$, with some positive constant $\theta < 1$. The derivative of V_1 along the

trajectories of (3.56) becomes

$$\dot{V}_1(t, \Sigma) = \frac{\partial V_1}{\partial \Sigma} Q(t, \Sigma) + \frac{\partial V_1}{\partial \Sigma} r(t, \Sigma) \quad (3.64a)$$

$$\leq -\Gamma_3(\|\Sigma\|) + \left\| \frac{\partial V_1}{\partial \Sigma} \right\| \|r(t, \Sigma)\| \quad (3.64b)$$

$$\leq -\Gamma_3(\|\Sigma\|) + \varsigma \varphi \quad (3.64c)$$

$$\leq -(1-\theta)\Gamma_3(\|\Sigma\|) - \theta\Gamma_3(\|\Sigma\|) + \varsigma \varphi \quad (3.64d)$$

$$\leq -(1-\theta)\Gamma_3(\|\Sigma\|), \quad \forall \|\Sigma\| \geq \Gamma_3^{-1}\left(\frac{\varsigma \varphi}{\theta}\right) \quad (3.64e)$$

Then for all $\Sigma(t_0) \in \{V_1(\Sigma) \leq \Gamma_1(R)\}$, the solution $\Sigma(t)$ of the perturbed system satisfies

$$\|\Sigma\| \leq \max\{\beta(\|\Sigma(t_0)\|, t - t_0), \rho(\varphi)\}, \quad \forall t \geq t_0 \quad (3.65)$$

for some class \mathcal{KL} functions β , and a class \mathcal{K} function ρ given by

$$\rho(\varphi) = \Gamma_1^{-1}(\Gamma_2(\Gamma_3^{-1}(\frac{\varphi \varsigma}{\theta})))$$

3.4 Backstepping Control of Robotic Fish

To demonstrate the effectiveness of the proposed approach, the backstepping control algorithm is implemented in a robotic fish, as this system has highly nonlinear dynamics with input coupling and is under-actuated.

3.4.1 Dynamic Model of Robotic Fish

The tail-actuated robotic fish is modeled as a rigid body with a rigid tail that is actuated at its base, and is considered to undergo planar motion [89]. The system has three degrees of freedom, namely surge, sway, and yaw. The tail deflection angle is assumed to have the following periodic pattern:

$$\alpha(t) = \alpha_0 + \alpha_a \sin(\omega_\alpha t) \quad (3.66)$$

where α_0 , α_a , and ω_α represent the bias, amplitude, and frequency of the tail beat, respectively.

The dynamics of the system are described with an average model [89] and is given by

$$\dot{v}_1 = h_1(v_1, v_2, \omega) + K_f c_7 \bar{f}_4(\alpha_0, \alpha_a, \omega_\alpha) \quad (3.67a)$$

$$\dot{v}_2 = h_2(v_1, v_2, \omega) + K_f c_8 \bar{f}_5(\alpha_0, \alpha_a, \omega_\alpha) \quad (3.67b)$$

$$\dot{\omega} = h_3(v_1, v_2, \omega) + K_m c_9 \bar{f}_6(\alpha_0, \alpha_a, \omega_\alpha) \quad (3.67c)$$

with

$$\left\{ \begin{array}{l} h_1(v_1, v_2, \omega) = \frac{m_2}{m_1} v_2 \omega - \frac{c_1}{m_1} v_1 \sqrt{v_1^2 + v_2^2} \\ \quad + \frac{c_2}{m_1} v_2 \sqrt{v_1^2 + v_2^2} \arctan\left(\frac{v_2}{v_1}\right) \\ h_2(v_1, v_2, \omega) = -\frac{m_1}{m_2} v_1 \omega - \frac{c_1}{m_2} v_2 \sqrt{v_1^2 + v_2^2} \\ \quad - \frac{c_2}{m_2} v_1 \sqrt{v_1^2 + v_2^2} \arctan\left(\frac{v_2}{v_1}\right) \\ h_3(v_1, v_2, \omega) = (m_1 - m_2) v_1 v_2 - c_4 \omega^2 \text{sgn}(\omega) \\ \bar{f}_4(\alpha_0, \alpha_a, \omega_\alpha) = \omega_\alpha^2 \alpha_a \left(3 - \frac{3}{2} \alpha_0^2 - \frac{3}{8} \alpha_a^2\right) \\ \bar{f}_5(\alpha_0, \alpha_a, \omega_\alpha) = \omega_\alpha^2 \alpha_a^2 \alpha_0 \\ \bar{f}_6(\alpha_0, \alpha_a, \omega_\alpha) = \omega_\alpha^2 \alpha_a^2 \alpha_0 \end{array} \right. \quad (3.68)$$

where $m_1 = m_b - m_{ax}$, $m_2 = m_b - m_{ay}$, $J_3 = J_{bz} - J_{az}$, $c_1 = \frac{1}{2} \rho S C_D$, $c_2 = \frac{1}{2} \rho S C_L$, $c_4 = \frac{K_D}{J_3}$, $c_7 = \frac{m L^2}{12 m_1}$, $c_8 = \frac{m L^2}{4 m_2}$, and $c_9 = -\frac{c m L^2}{4 J_3}$. Here m_b is the mass of the body, J_{bz} is the inertia of the body about the body-fixed z -axis, m_{ax} and m_{ay} are the hydrodynamic derivatives that represent the added masses of the robotic fish along the body-fixed x and y directions, respectively, and J_{az} represents the added inertia effect of the body about the body-fixed z direction. S denotes the reference surface area for the robot body, C_D , C_L and K_D represent the drag force coefficient, lift coefficient, and drag moment coefficient, respectively, ρ is the density of water, L is the tail length, c is the distance from the body center to the pivot point of the actuated tail and m represents the mass of water displaced by the tail per unit length and is approximated by $\frac{\pi}{4} \rho d^2$ with d denoting the tail depth. K_f is a scaling constant, and K_m is a scaling function affine in α_0 . As done in [92], to further facilitate control design, in this work K_m is considered as a constant by taking the average

of K_m for a given range of α_0 . Furthermore, let (x_p, y_p) denote the robot's center of mass and ψ the heading angle, formed by the body fixed axis relative to the inertial axis. The system's kinematics are thus given by

$$\dot{x}_p = v_1 \cos \psi - v_2 \sin \psi \quad (3.69a)$$

$$\dot{y}_p = v_1 \sin \psi + v_2 \cos \psi \quad (3.69b)$$

$$\dot{\psi} = \omega \quad (3.69c)$$

The robotic fish dynamics satisfy Assumption 1, and Assumption 2, the proof for which can be found in Appendix A.

3.4.2 Backstepping Control Design

To design the controller, the robot's simplified averaged model presented above is considered. Here the control inputs are functions of the actual physical variables, namely the tail-beat pattern parameters α_0 , α_a , and ω_α . To simplify discussion, a fixed tail-beat frequency ω_α is assumed. The control inputs are then chosen as

$$u_1 = \alpha_a \left(3 - \frac{3}{2} \alpha_0^2 - \frac{3}{8} \alpha_a^2 \right) \quad (3.70)$$

$$u_2 = \alpha_a^2 \alpha_0 \quad (3.71)$$

which are present in functions $\bar{f}_4(\alpha_0, \alpha_a, \omega_\alpha)$ to $\bar{f}_6(\alpha_0, \alpha_a, \omega_\alpha)$ in Eqs. (3.67a)-(3.67c). Note that the control inputs defined this way appear linearly in the dynamic equations. In this manner, (3.67a)-(3.67c) thus resembles (3.4) with $b_1 = K_f c_7 \omega_\alpha^2$, $b_2 = 0$, $b_3 = 0$, $b_4 = K_f c_8 \omega_\alpha^2$, $b_5 = 0$ and $b_6 = K_m c_9 \omega_\alpha^2$.

The modified error function q for the robotic fish should be chosen such that the heading angle of the robot can be used to minimize y_e , but still allow tracking of ψ_r when y_e is small enough. For simplicity, we choose $q_2 = \psi_e$, and consider the Lyapunov function V_S proposed in Section 3.3.2 to determine possible choices for q_1 . In particular the derivative of V_S with respect to time is

given by

$$\dot{V}_S(y_e) = -v_r \sin^2(q_2^{-1}(q_1(y_e))) \frac{\partial(q_1(t, y_e))}{\partial y_e} + \frac{\partial V_S}{\partial t} \quad (3.72)$$

From the above, q_1 need to be chosen such that \dot{V}_S is negative definite. For simplicity, we let q_1 be only a function of y_e . Furthermore, taking into consideration Assumption 5, q_1 needs to be a continuously differentiable function that is well defined for all real numbers of y_e and has a bounded range. The function $\arctan(y_e)$ is an attractive choice since it satisfies all of the aforementioned assumptions. Furthermore, its derivative with respect to y_e , $\frac{\partial(q_1(t, y_e))}{\partial y_e} = \frac{1}{1 + y_e^2}$, is positive for all y_e . The modified error for the robotic fish system is thus chosen as

$$z_\xi = q(y_e, \psi_e) = \psi_e + q_2(y_e) \quad (3.73)$$

where q_2 is given by

$$q_2(y_e) = k_{\delta_1} \arctan(k_{\delta_2} y_e) \quad (3.74)$$

where k_{δ_1} and k_{δ_2} are positive constants. In particular, k_{δ_1} can be thought of as a scale for the “correction” angle within the total error z_ξ , while k_{δ_2} tunes the magnitude of the angle. Note that with this choice we also satisfy $k(t, \mathbf{e}) = \frac{\partial q_2(t, \psi_e)}{\partial \psi_e} - \frac{\partial q_1(t, y_e)}{\partial y_e} x_e = 1 - \frac{k_{\delta_1} k_{\delta_2} x_e}{1 + (k_{\delta_2} y_e)^2}$ such that $k(t, \mathbf{e}) \neq 0 \quad \forall \quad t \geq 0, \forall \quad |x_e| < \frac{1}{k_{z_{\delta_1}} k_{z_{\delta_2}}}$.

3.4.2.1 Control Synthesis

The backstepping controller for the robotic fish can be designed by following the procedure from Section 3.2.2. First, to stabilize the (x_e, z_ξ) subsystem, the following candidate Lyapunov function is chosen

$$V = \frac{1}{2} x_e^2 + \frac{1}{2} z_\xi^2 \quad (3.75)$$

Let $v_1 = \phi_1$ and $\omega = \phi_3$ represent the virtual inputs, which will be chosen shortly. The time derivative of Eq. (3.75) is given by

$$\begin{aligned} \dot{V} &= x_e \dot{x}_e + z_\xi \dot{z}_\xi \\ &= x_e (\phi_1 - V_r \cos(\psi_e) + \phi_3 y_e) + z_\xi (k(t, \mathbf{e}) \phi_3 + p(t, \mathbf{e}, v_2)) \end{aligned} \quad (3.76)$$

where

$$p(t, \mathbf{e}, v_2) = k_{\delta_1} k_{\delta_2} \frac{(V_r \sin(\psi_e) + v_2)}{(k_{\delta_2} y_e)^2 + 1} - \omega_r \quad (3.77)$$

$$k(t, \mathbf{e}) = 1 - \frac{k_{\delta_1} k_{\delta_2} x_e}{(k_{\delta_2} y_e)^2 + 1} \quad (3.78)$$

Let

$$\phi_1 = V_r \cos(\psi_e) - \phi_3 y_e - K_{x_e} x_e \quad (3.79)$$

$$\phi_3 = \frac{-p(t, \mathbf{e}, v_2) - K_{z_\xi} z_\xi}{k(t, \mathbf{e})} \quad (3.80)$$

so that

$$\dot{V} = -K_{x_e} x_e^2 - K_{z_\xi} z_\xi^2 \quad (3.81)$$

Let the virtual input error be given by

$$\begin{bmatrix} e_{\eta_1} \\ e_{\eta_3} \end{bmatrix} = \begin{bmatrix} v_1 - \phi_1 + y_e(\omega - \phi_3) \\ k(t, \mathbf{e})(\omega - \phi_3) \end{bmatrix} \quad (3.82)$$

and define a new Lyapunov function

$$V_2 = V + \frac{1}{2} e_{\eta_1}^2 + \frac{1}{2} e_{\eta_3}^2 \quad (3.83)$$

The time derivative of Eq. (3.83) is given by

$$\dot{V}_2 = \dot{V}_1 + e_{\eta_1} \dot{e}_{\eta_1} + e_{\eta_3} \dot{e}_{\eta_3} \quad (3.84)$$

With Eqs. (3.67a)-(3.67c) along with the input definition Eq. (3.70)- (3.71), one can expand the \dot{v}_1 , \dot{v}_2 and $\dot{\omega}$ terms that appear in Eq. (3.84). To make \dot{V}_2 negative definite, let u_1 and u_2 be chosen as

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} K_{fc7} & K_{mc9} Y_e \\ 0 & \eta \end{bmatrix}^{-1} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} \quad (3.85)$$

with

$$T_1 = -h_1(\cdot) + \dot{V}_r \cos(\psi_e) - V_r \sin(\psi_e) \dot{\psi}_e - h_3(\cdot) y_e - \omega \dot{y}_e - K_{x_e} \dot{x}_e - K_{e\eta_1} e_{\eta_1} \quad (3.86)$$

$$T_2 = -h_3(\cdot) k(\cdot) - \dot{k}(\cdot) (\omega - \phi_3) - k(\cdot) \Delta - K_{e\eta_3} e_{\eta_3} \quad (3.87)$$

$$\dot{k}(\cdot) = -k_{z\delta 1} k_{z\delta 2} \frac{\dot{x}_e}{(k_{z\delta 2} y_e)^2 + 1} + k_{z\delta 1} k_{z\delta 2}^2 \frac{2x_e y_e \dot{y}_e}{((k_{z\delta 2} y_e)^2 + 1)^2} \quad (3.88)$$

$$\Delta = k_{z\delta 1} k_{z\delta 2} \frac{-\dot{p} - K_{z\xi} \dot{z}_\xi}{k(\cdot)} - k_{z\delta 1} k_{z\delta 2} \dot{k}(\cdot) \frac{-p - K_{z\xi} z_\xi}{k(\cdot)^2} \quad (3.89)$$

$$\dot{p} = k_{z\delta 1} k_{z\delta 2} \frac{h_2(\cdot) + \dot{V}_r \sin(\psi_e) + V_r \cos(\psi_e) \dot{\psi}_e}{(k_{z\delta 2} y_e)^2 + 1} - k_{z\delta 1} k_{z\delta 2}^2 \frac{2y_e \dot{y}_e (V_r \sin(\psi_e) + v_2)}{((k_{z\delta 2} y_e)^2 + 1)^2} \quad (3.90)$$

$$\eta = k_{z\delta 1} k_{z\delta 2} \frac{K_f c_8}{(k_{z\delta 2} y_e)^2 + 1} + K_m c_9 k(t, \mathbf{e}) \quad (3.91)$$

3.4.2.2 Control Synthesis Incorporating Input Constraints

Given that robot fish actuators have physical limitations, the backstepping-based controller design should accommodate these constraints so that the control scheme can be successfully implemented. In this section, we present the control synthesis incorporating the input constraints. However, note that the time-scale stability analysis assumes that the nominal inputs are achievable.// The latter can be true given that the desired trajectory is feasible. In other words, when the desired trajectory is such that the nominal values will lie within the actuator constraints, then the achievable input will converge to the nominal input.

In order to address magnitude constraints on the control inputs, we employ the following scheme inspired by [132] and [133]. Let v_1 and v_2 represent the nominal backstepping control inputs, and let u_1 and u_2 be the inputs that can be practically implemented. To obtain the value for u_1 and u_2 , first the values given by v_1 and v_2 are used to solve for the tail-beat parameters α_0, α_a using Eqs. (3.70)-(3.71), and then the tail-beat parameter values are saturated such that they lie within the range $[\alpha_{0_{min}}, \alpha_{0_{max}}]$, and $[\alpha_{a_{max}}, \alpha_{a_{min}}]$. Finally u_1 and u_2 are obtained using Eqs. (3.70)-(3.71)

with the saturated values.

To analyze the influence of the input constraints, the following auxiliary system is chosen,

$$\begin{cases} \dot{\lambda}_1 = -\zeta_1 \lambda_1 + \lambda_2 \\ \dot{\lambda}_2 = -\zeta_2 \lambda_2 + K_f c_7(u_1 - v_1) + Y_e K_m c_9(u_2 - v_2) \\ \dot{\lambda}_3 = -\zeta_3 \lambda_3 + \lambda_4 \\ \dot{\lambda}_4 = -\zeta_4 \lambda_4 - \eta(u_2 - v_2) \end{cases} \quad (3.92)$$

The variables λ_1 - λ_4 defined above represent the filtered effect of the non-achievable portion of the virtual and control inputs. In particular, the additional tracking error that arises because of mismatch between the nominal and implementable inputs is represent by λ_1 and λ_3 , while λ_2 and λ_4 represent the error propagated to the virtual inputs. As a result, the modified tracking errors are defined as follows:

$$\bar{x}_e = x_e - \lambda_1 \quad (3.93)$$

$$\bar{z}_\xi = z_\xi - \lambda_3 \quad (3.94)$$

Furthermore, let the modified virtual errors be given as

$$\bar{e}_{\eta_1} = v_1 - \phi_1 + y_e(\omega - \phi_3) - \lambda_2 \quad (3.95)$$

$$\bar{e}_{\eta_3} = k(t, \mathbf{e})(\omega - \phi_3) - \lambda_4 \quad (3.96)$$

To stabilize the (\bar{x}_e, \bar{z}_ξ) subsystem, the following Lyapunov function is chosen

$$\bar{V} = \frac{1}{2} \bar{x}_e^2 + \frac{1}{2} \bar{z}_\xi^2 \quad (3.97)$$

With these new definitions, similar stability analysis as done in Eq. (3.76)-(3.81) is carried out.

Let the virtual inputs be defined as

$$\phi_1 = V_r \cos(\psi_e) - \omega y_e - K_{x_e} \bar{x}_e - \zeta_1 \lambda_1 \quad (3.98)$$

$$\phi_3 = \frac{-p(t, \mathbf{e}, v_2) - K_{z_\xi} \bar{z}_\xi - \zeta_3 \lambda_3}{k(t, \mathbf{e})} \quad (3.99)$$

so that

$$\dot{\bar{V}} = \bar{x}_e(\bar{e}_{\eta_1} - K_{x_e}\bar{x}_e) + \bar{z}_e(\bar{e}_{\eta_3} - K_{z_\xi}\bar{z}_e) \quad (3.100)$$

Then a new Lyapunov function is defined as

$$\bar{V}_2 = \bar{V} + \frac{1}{2}\bar{e}_{\eta_1}^2 + \frac{1}{2}\bar{e}_{\eta_3}^2 \quad (3.101)$$

The time derivative of Eq. (3.101) is given by

$$\dot{\bar{V}}_2 = \dot{\bar{V}} + \bar{e}_{\eta_1}\dot{\bar{e}}_{\eta_1} + \bar{e}_{\eta_3}\dot{\bar{e}}_{\eta_3} \quad (3.102)$$

As previously done, the terms \dot{v}_1 , \dot{v}_2 and $\dot{\omega}$ that appear in Eq. (3.102) can be further expanded.

After simplifying the above, v_1 and v_2 can be chosen as

$$\begin{aligned} K_f c_7 v_1 + K_m c_9 y_e v_2 = & -h_1(\cdot) + \dot{V}_r \cos(\psi_e) - V_r \sin(\psi_e) \dot{\psi}_e - h_3(\cdot) y_e - \omega_z \dot{y}_e - K_{x_e} \dot{\bar{x}}_e \\ & + \zeta_1^2 \lambda_1 - (\zeta_1 + \zeta_2) \lambda_2 - K_{e\eta_1} \bar{e}_{\eta_1} \end{aligned} \quad (3.103)$$

$$\eta v_2 = -h_3(\cdot) k(\cdot) - \dot{k}(\cdot)(\omega - \phi_3) - k(\cdot) \bar{\Delta} + \zeta_3^2 \lambda_3 - (\zeta_3 + \zeta_4) \lambda_4 - K_{e\eta_3} \bar{e}_{\eta_3} \quad (3.104)$$

where

$$\bar{\Delta} = k_{z_{\delta 1}} k_{z_{\delta 2}} \frac{-\dot{p} - K_{z_\xi} \dot{\bar{z}}_\xi}{k(\cdot)} - k_{z_{\delta 1}} k_{z_{\delta 2}} \dot{k}(\cdot) \frac{-p - K_{z_\xi} \bar{z}_\xi}{k(\cdot)^2} \quad (3.105)$$

$$\dot{p} = k_{z_{\delta 1}} k_{z_{\delta 2}} \frac{h_2(\cdot) + \dot{V}_r \sin(\psi_e) + V_r \cos(\psi_e) \dot{\psi}_e}{(k_{z_{\delta 2}} y_e)^2 + 1} - k_{z_{\delta 1}} k_{z_{\delta 2}} \frac{2y_e \dot{y}_e (V_r \sin(\psi_e) + v_2)}{((k_{z_{\delta 2}} y_e)^2 + 1)^2} \quad (3.106)$$

By following similar stability analysis as previously done for the constraints-free case, one can arrive at the following

$$\dot{\bar{V}}_2 = -K_{x_e} (\bar{x}_e - \frac{1}{2K_{x_e}} \bar{e}_{\eta_1})^2 - K_{z_\xi} (\bar{z}_\xi - \frac{1}{2K_{z_\xi}} \bar{e}_{\eta_3})^2 - \bar{e}_{\eta_1}^2 (K_{e\eta_1} - \frac{1}{4K_{x_e}}) - \bar{e}_{\eta_3}^2 (K_{e\eta_3} - \frac{1}{4K_{z_\xi}}) \quad (3.107)$$

If $K_{x_e} > 0$, $K_{z_\xi} > 0$, $K_{e\eta_1} > \frac{1}{4K_{x_e}}$ and $K_{e\eta_3} > \frac{1}{4K_{z_\xi}}$, then $\dot{\bar{V}}_2 < 0$ unless when $\bar{x}_e = \bar{z}_\xi = \bar{e}_{\eta_1} = \bar{e}_{\eta_3} = 0$, implying the convergence of $(\bar{x}_e, \bar{z}_\xi, \bar{e}_{\eta_1}, \bar{e}_{\eta_3})$ to zero as time approaches infinity.

Furthermore, since $0 \leq \bar{V}_2(t) \leq \bar{V}_2(0)$, one can conclude that $(\bar{x}_e, \bar{z}_\xi, \bar{e}_{\eta_1}, \bar{e}_{\eta_3})$ are each in \mathcal{L}_2 . This shows that even when physical limitations do not allow the the desired control signals to be implemented, the quantities \bar{x}_e and \bar{z}_ξ do not diverge. In other words, this guarantees convergence for the compensated tracking errors \bar{x}_e and \bar{z}_ξ but not the actual tracking errors x_e and z_ξ . The latter may actually increase during periods when input limitations are in effect given that the desired control signal is not being implemented (i.e. $u_1 \neq v_1$ and/or $u_2 \neq v_2$). However, when the control signal limitations are not in effect, (i.e. $u_1 = v_1$ and $u_2 = v_2$), λ_1 - λ_4 approach zero, and (\bar{x}_e, \bar{z}_ξ) converges towards (x_e, z_ξ) . Once the control input limitations are not in the effect, the time-scale analysis previously presented will apply. Finally the nominal inputs v_1 and v_2 can be obtained from (3.105).

In the following section we establish the stability analysis for the robotic fish under the assumption that the nominal input values are achieved.

3.4.3 Stability Analysis of Closed-Loop Robotic Fish System

Following the time-scale analysis presented in Section 3.3, it can be shown that by choosing

$$V_S(y_e) = (1 - \cos(k_{\delta_1} \arctan(k_{\delta_2} y_e))) \quad (3.108a)$$

$$V_F(\chi) = \chi^T P \chi \quad (3.108b)$$

and a composite Lyapunov function

$$V_1(t, y_e, \chi) = (1 - d_1)V_S(t, y_e) + d_1 V_F(\chi) \quad (3.109)$$

then for all $\Sigma(t_0) \in \{V_1(t, \Sigma) \leq \Gamma_1(R)\}$, the solution $\Sigma(t)$ of the robotic fish perturbed system satisfies

$$\|\Sigma\| \leq \max\{\beta(\|\Sigma(t_0)\|, t - t_0), \rho(\varphi)\}, \quad \forall t \geq t_0 \quad (3.110)$$

where ρ is given by

$$\rho(\varphi) = \left(\sqrt{\frac{\varphi S}{\theta \lambda_{\min}(\Lambda)}} \right)$$

and

$$\begin{aligned} \varsigma &= \sqrt{(c_1(1-d_1)v_r^{max}k_{\delta_1}k_{z\delta_2})^2 + (2d_1\bar{\lambda}_{max}(p))^2\|R\|} \\ \varphi &= \frac{k_{z\delta_1}k_{z\delta_2}\bar{\vartheta}}{D_e}\|R\| \end{aligned} \quad (3.111)$$

with D_e defined such that

$$\frac{|x_e|(1+k_{z\delta_1}k_{z\delta_2})}{(k_{z\delta_2}y_e)^2+1-k_{ze1}k_{z\delta_2}x_e} \leq \frac{|x_e|}{D_e} \quad (3.112)$$

for $|x_e| < \frac{1}{k_{z\delta_1}k_{z\delta_2}}$. The reader is referred to Appendix B for a detailed proof of the time-scale analysis without perturbation and Appendix C for the stability analysis of the perturbed robotic fish system.

3.5 Simulation Results

To evaluate the effectiveness of the designed controller, simulations were carried out using MATLAB. Furthermore, a PI controller was implemented to provide performance comparison. The robotic fish parameters used for simulation are listed Table 3.2.

The tunable backstepping and PI controller parameters for the line trajectory (defined briefly) in the simulation were chosen as follows:

$K_{x_e}=1.1$	$K_{z_\xi}=1.2$	$K_{e\eta_1}=0.69$	$K_{e\eta_3}=0.41$
$\zeta_1=0.4$	$\zeta_2=0.8$	$\zeta_3=0.9$	$\zeta_4=0.9$
$k_{z\delta_1}=5$	$k_{z\delta_2}=63.3$	$K_{P1}=3.5$	$K_{P2}=0.85$
$K_{I1}=0.021$	$K_{I2}=0.023$	$\alpha_{0_{\min}}=-50^\circ$	$\alpha_{0_{\max}}=50^\circ$
$\alpha_{a_{\min}}=0^\circ$	$\alpha_{a_{\max}}=30^\circ$	$t_s=0.66$ s	$\omega_\alpha=3\pi$

where $\alpha_{0_{\min}}$, $\alpha_{0_{\max}}$, $\alpha_{a_{\min}}$ and $\alpha_{a_{\max}}$ are the physical limits on the tail-beat bias and amplitude respectively. Furthermore, K_{P1} , K_{P2} , K_{I1} , and K_{I2} are the PI controller tunable parameters. The variable t_s is the sampling interval which pertains to the amount of time between an update to the control inputs. In this design, we chose $t_s = 0.66$ seconds given that the tail-beat frequency is

1.5 Hz. The controller parameters were chosen such that under the right values the backstepping controller was able to regulate the error system to the origin. We found that varying $\zeta_1 - \zeta_4$ controlled the convergence rate of the errors $\lambda_1 - \lambda_4$, which are a consequence of the effect of the input constraints. For the PI controller, its gain parameters were tuned carefully in simulation using Ziegler-Nichols' closed-loop method [134].

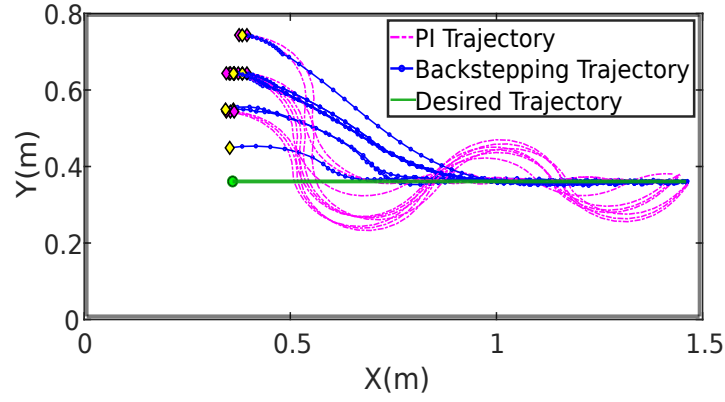
Note that although the backstepping controller was designed using the simplified averaged model, the simulations were performed on the original dynamic model. The following line and circular trajectories were considered

$$\begin{cases} \dot{x}_r = v_r \cos(\psi_r), & \dot{y}_r = 0, & \dot{\psi}_r = \omega_r, \text{ with } v_r = 0.02, \omega_r = 0 \text{ and} \\ \dot{x}_r = v_r \cos(\psi_r), & \dot{y}_r = v_r \sin(\psi_r), & \dot{\psi}_r = \omega_r, \text{ with} \\ v_r = R_1 |\omega_r|, & R_1 = 0.2, & \omega_r = -0.09, \end{cases} \quad (3.113)$$

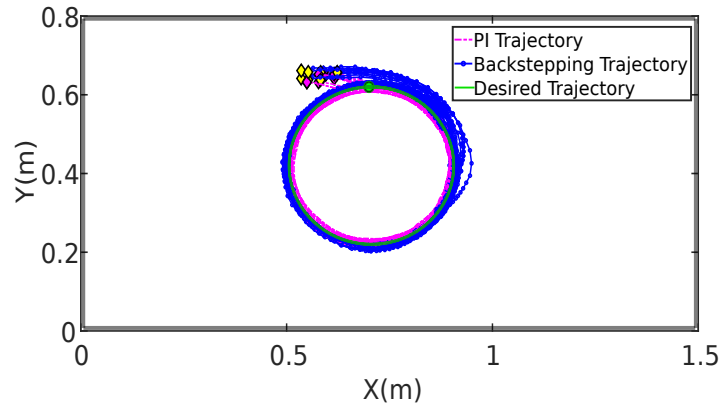
where \dot{x}_r and \dot{y}_r represent the velocity of the trajectory in the $\{I\}$ frame.

Figures 3.3a-3.3b depict the desired and the closed-loop trajectories of the robotic fish for the backstepping-based and PI controller in both the line and circular cases. Note that the diamonds represents the starting position of the robotic fish, while the green circle represents the starting point of the path. In particular, 10 simulations trials were run for each type of trajectory with different initial conditions.

Figures 3.4-3.5 illustrate the averaged magnitude of the position errors over time along with the corresponding standard deviations for line and circular tracking, respectively, for both the proposed backstepping scheme and the PI controller. Similarly, Figures 3.6-3.7 illustrate the averaged magnitude of the angle errors over time along with the corresponding standard deviations for line and circular tracking, respectively, for both the proposed backstepping scheme and the PI controller. From simulation results, in particular from Figures 3.4-3.7, one can see that, with the proposed backstepping scheme, smaller position tracking error in the allotted time is obtained, as well as a faster convergence to the desired trajectory.



(a) Line-tracking trajectories.



(b) Circle-tracking trajectories.

Figure 3.3: Simulation: line and circle-tracking trajectory results, respectively, for backstepping-based and PI control.

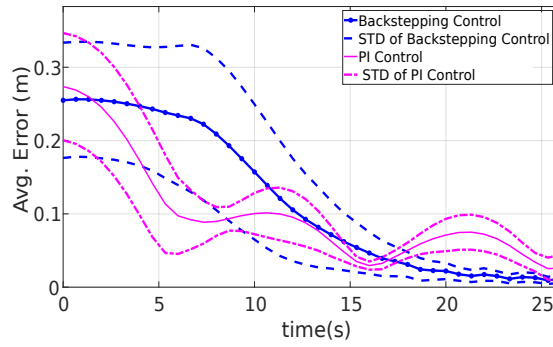


Figure 3.4: Simulation: line-tracking position error for backstepping-based and PI control. “STD” curves represent the standard deviation envelopes for the average error under each controller.

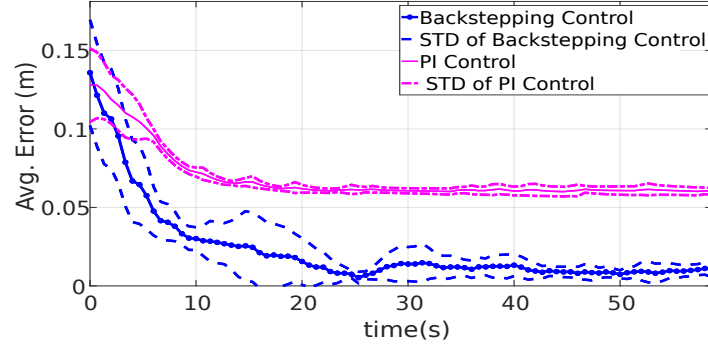


Figure 3.5: Simulation: circle-tracking position error for backstepping-based and PI control. “STD” curves represent the standard deviation envelopes for the average error under each controller.

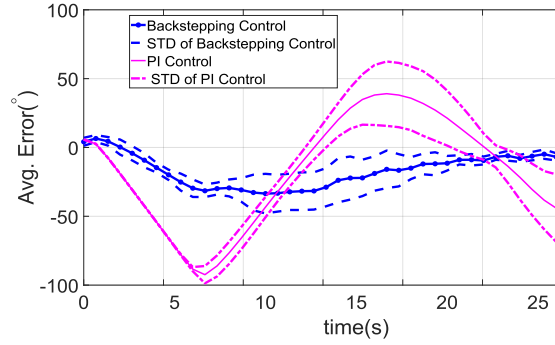


Figure 3.6: Simulation: line-tracking angle error for backstepping-based and PI control. “STD” curves represent the standard deviation envelopes for the average error under each controller.

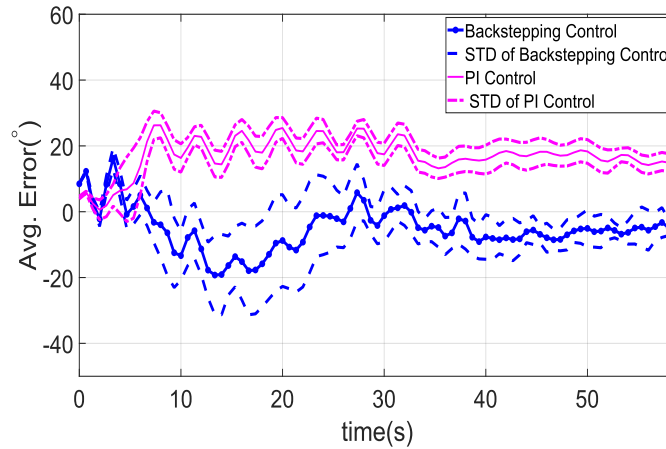


Figure 3.7: Simulation: circle-tracking angle error for backstepping-based and PI control. “STD” curves represent the standard deviation envelopes for the average error under each controller.

Table 3.2: Parameters of the robotic fish.

PARAMETER	VALUE	PARAMETER	VALUE
m_b	0.725 kg	m_{ax}	-0.217 kg
m_{ay}	-0.7888 kg	c	0.105 m
J_{az}	$-7.93 \times 10^{-4} \text{ kg} \cdot \text{m}^2$	L	0.071 m
J_{bz}	$2.66 \times 10^{-3} \text{ kg} \cdot \text{m}^2$	d	0.04 m
ρ	1000 kg/m ³	S	0.03 m ²
C_D	0.97	C_L	3.9047
K_D	$4.5 \times 10^{-3} \text{ kg} \cdot \text{m}^2$	K_f	0.7
$K_m(\text{averaged})$	0.45		.

3.6 Experimental Results

In order to evaluate the validity of the proposed scheme, experiments were carried out using the robotic fish presented in the previous Chapter (Figure 2.13). Furthermore, a similar experimental set up was utilized (Figure. 2.12). Finally, the parameters for the robotic fish are as shown in Table 3.2.

To demonstrate the effectiveness of the proposed approach, experiments were carried out to compare the performance of the backstepping-based controller with that of a proportional-integral (PI) controller in terms of tracking performance. It is important to emphasize that the robotic fish has actuator constraints that must be considered when implementing the controller. To address this, the backstepping controller was modified as shown in Section 3.4.2.2. Furthermore, the inputs were saturated in a similar fashion as described in Section 3.5. In the case of the PI controller, the original calculated values for the tail-beat parameters were saturated to obtain the realizable values and thus the viable inputs.

The tunable controller parameters were chosen as follows:

$$\begin{array}{llll}
K_{x_e} = 0.6 & K_{Z_\xi} = 0.6 & K_{e\eta_1} = 1.02 & K_{e\eta_2} = 1.02 \\
\zeta_1 = 0.6 & \zeta_2 = 0.6 & \zeta_3 = 0.5 & \zeta_4 = 0.5 \\
k_{\delta_1} = 1 & k_{\delta_2} = 1.6 & K_{P1} = 0.25 & K_{P2} = 0.31 \\
K_{I1} = 0.01 & K_{I2} = 0.0071 & \alpha_{0_{\min}} = -50^\circ & \alpha_{0_{\max}} = 50^\circ \\
\alpha_{a_{\min}} = 0^\circ & \alpha_{a_{\max}} = 30^\circ & t_s = 0.66 \text{ s} &
\end{array}$$

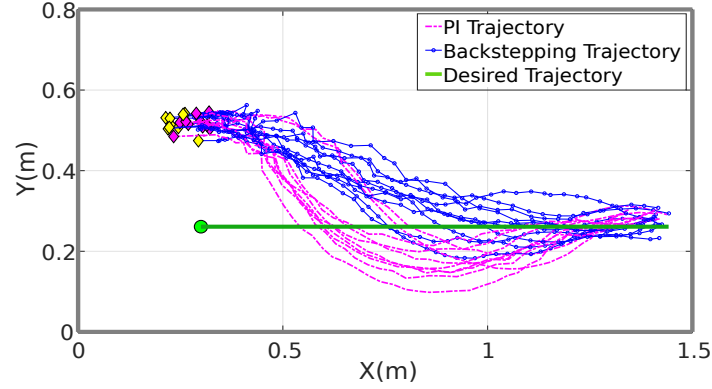
where $\alpha_{0_{\min}}, \alpha_{0_{\max}}, \alpha_{a_{\min}}$ and $\alpha_{a_{\max}}$ are the physical limits on the tail-beat bias and amplitude respectively. Furthermore, K_{P1}, K_{P2}, K_{I1} , and K_{I2} are the PI controller tunable parameters. t_s is the sampling interval that pertains to the amount of time between an update to the control inputs. In this design, we chose $t_s = 0.66$ seconds given that the tail-beat frequency is 1.5 Hz. Finally, $\zeta_1 - \zeta_4$ are the controller parameters used to handle the input constraints. The PI gain parameters were tuned in experiments using Ziegler-Nichols' closed-loop method [134].

The following line and circular trajectories were considered for the tracking tasks

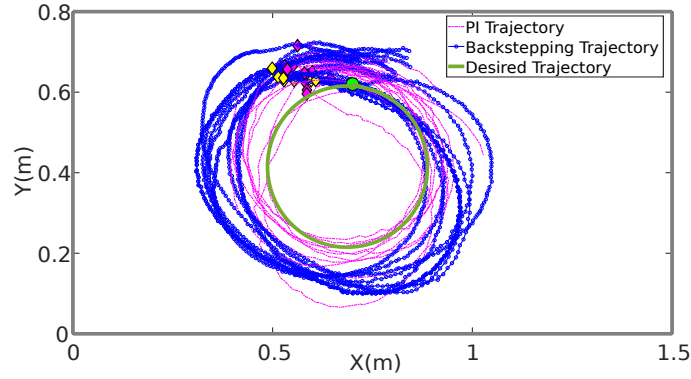
$$\begin{cases} \dot{x}_r = v_{1r} \cos(\psi_r), & \dot{y}_r = 0, & \dot{\psi}_r = \omega_r, \text{ with } v_{1r} = 0.02, \omega_r = 0 \text{ and} \\ \dot{x}_r = v_{1r} \cos(\psi_r), & \dot{y}_r = v_{1r} \sin(\psi_r), & \dot{\psi}_r = \omega_r, \text{ with} \\ v_{1r} = R_1 |\omega_r|, & R_1 = 0.2, & \omega_r = -0.09, \end{cases} \quad (3.114)$$

In particular, ten different trials with similar initial conditions were performed for both the backstepping and PI controllers. In Figure 3.8, the desired and the closed-loop robotic fish trajectories for all ten trials are depicted for both the line and circular trajectories, while in Figure 3.9 the tracking results for only one of the trials is depicted. In particular, Figure 3.9a illustrates the tracking results for backstepping-based control, and Figure 3.9b illustrates those obtained for PI control. In both figures the yellow diamond depicts the starting position of the robot while the green circle depicts the start of the trajectory. Furthermore, the blue solid line depicts the desired trajectory while the red solid line depicts that of the robotic fish. Finally, the arrowheads on the lines denote the direction of progression.

In addition, Figure 3.10 depicts the averaged position tracking error along with its corresponding standard deviation for both control schemes in each trajectory case. In particular, the solid blue line with circular markers represents the averaged tracking error obtained for backstepping-based control, while the dotted blue lines represent its standard deviation. Similarly, the solid magenta line represents the averaged error obtained for PI control, while the dotted-dashed magenta lines represent its standard deviation. Similarly, Figure 3.11 depicts the averaged angle tracking error along with its corresponding standard deviation for both control schemes in each trajectory case.



(a) Line-tracking trajectories.



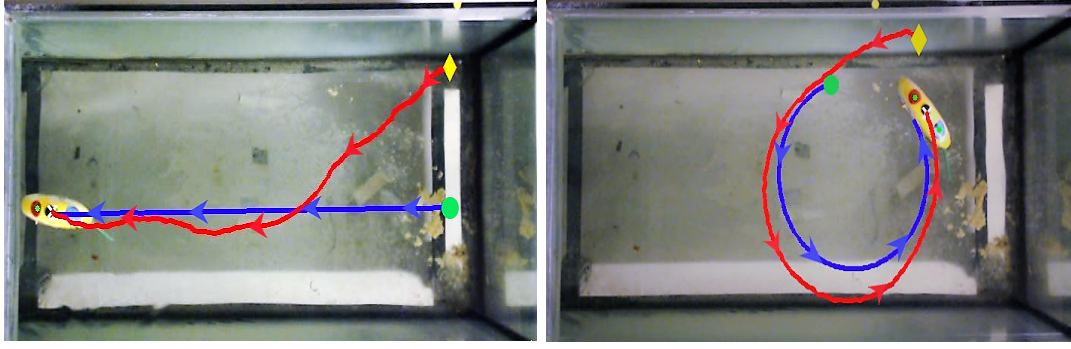
(b) Circle-tracking trajectories.

Figure 3.8: Experiments: line and circle-tracking trajectory results, respectively, for backstepping-based and PI control.

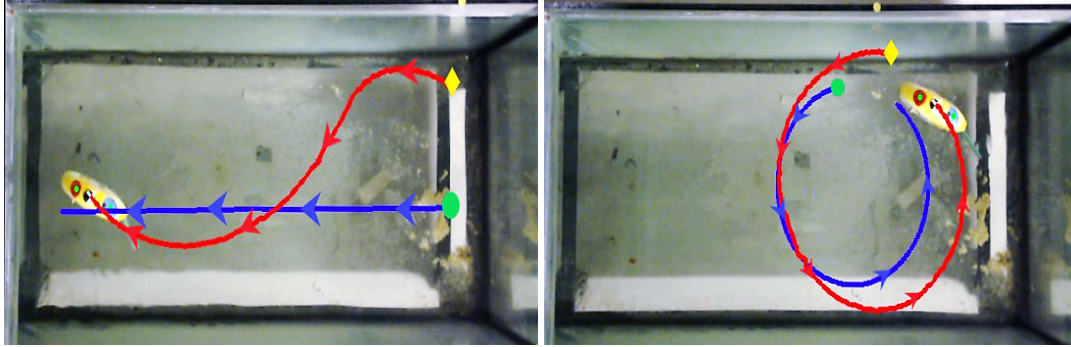
From the average errors depicted in Figures 3.10-3.11 one can see that overall the proposed backstepping scheme resulted in faster convergence to the desired trajectory and achieves a smaller steady-state error for both types of trajectory.

3.7 Chapter Summary

In this chapter, we proposed a backstepping-based scheme to achieve trajectory tracking of under-actuated planar robotic systems. In particular, we proposed a scheme that would guarantee the ultimate boundedness of the position and heading tracking errors to a neighborhood of the origin. This was achieved via defining a new error coordinate that coupled the lateral and heading errors. The controller was then synthesized such that it guaranteed the convergence of the longitudinal and new coordinate error. Via time-scale analysis of perturbed system, it was shown that the



(a) Backstepping-based control.

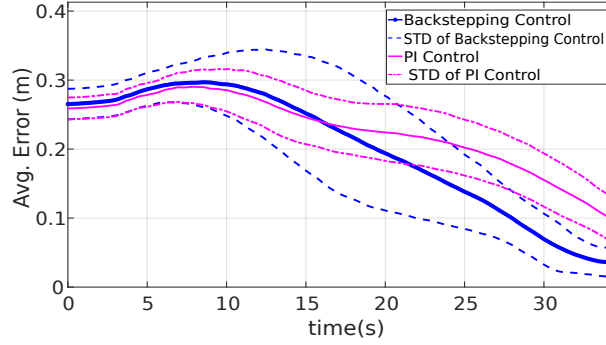


(b) PI control.

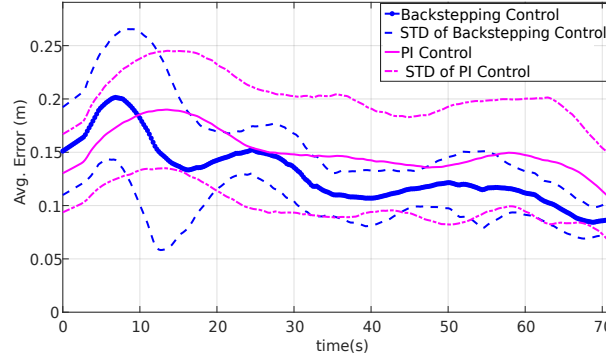
Figure 3.9: A snapshot of an experimental run for line and circle-tracking using backstepping-based (a) and PI control (b), respectively.

convergence of these then guarantees the stabilization of the whole error system to a neighborhood around the origin.

The proposed scheme was implemented on a tail-actuated robotic fish, where a high-fidelity averaged nonlinear dynamic model was used for controller design. The actuator constraints of the system were handled via an auxiliary system. Finally, real-time experimental results demonstrated the effectiveness of the proposed scheme and showed its value over standard PI control. While most of the discussions in this work were framed in the context of aquatic robots and vehicles, the approach applies to ground robots and vehicles with similar under-actuated constraints.

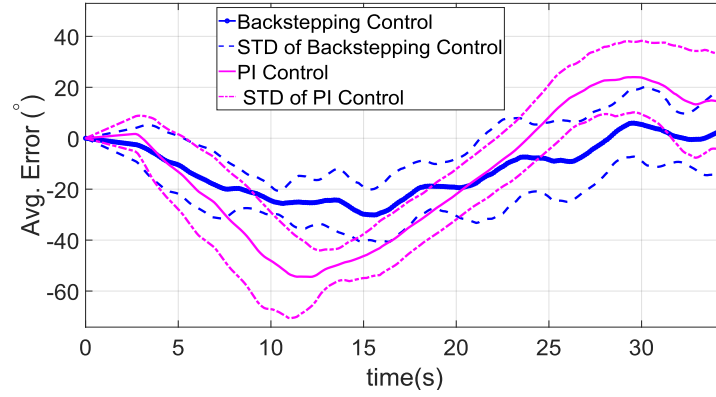


(a) Line-tracking averaged position error and its standard deviation.

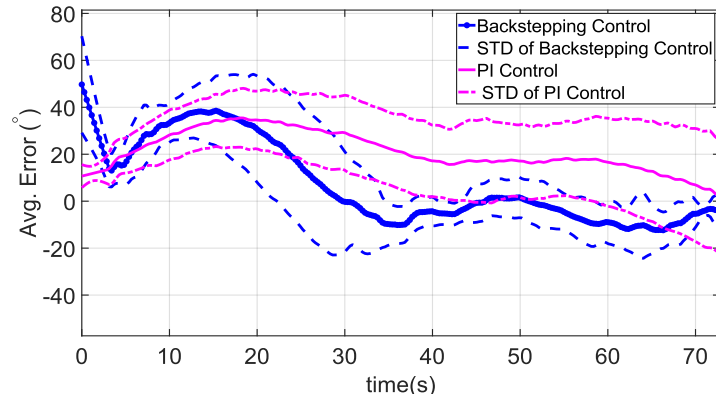


(b) Circle-tracking averaged position error and its standard deviation.

Figure 3.10: Experiments: line-tracking (3.10a) and circle-tracking (3.10b) averaged position error for backstepping-based control and PI control. The solid blue line with circular markers represents the averaged position error obtained for backstepping-based control, while the dashed blue lines represent its standard deviation. Similarly, the solid magenta line represents the averaged position error obtained for PI control, while the dotted-dashed magenta lines represent its standard deviation.



(a) Line-tracking averaged angle error and its standard deviation.



(b) Circle-tracking averaged angle error and its standard deviation.

Figure 3.11: Experiments: line-tracking (3.11a) and circle-tracking (3.11b) averaged angle error for backstepping-based control and PI control. The solid blue line with circular markers represents the averaged angle error obtained for backstepping-based control, while the dashed blue lines represent its standard deviation. Similarly, the solid magenta line represents the averaged angle error obtained for PI control, while the dotted-dashed magenta lines represent its standard deviation.

CHAPTER 4

AVERAGED MODELING AND TRACKING CONTROL OF PECTORAL FIN-ACTUATED ROBOTIC FISH

The approaches proposed in Chapter 2 and Chapter 3 focused on the control of tail fin-actuated robotic fish. While caudal fins have proven to be an efficient propulsion mode at higher speeds, pectoral fins play a vital role in maneuvering and stability while providing or assisting propulsion at lower speeds [25].

Pectoral fin motions can generally be classified into three modes based on the axis of rotation: rowing, feathering, and flapping. Feathering and flapping involve fin rotation about the transverse and longitudinal axis, respectively, while rowing motion involves fin rotation about the vertical axis. Rowing motion is classified as a “drag-based” swimming mechanism, where the drag element of fluid dynamics generates the thrust and can be utilized for a number of in-plane locomotion and maneuvering tasks, such as forward swimming, sideways swimming, and turning [66, 67].

The fin beat cycle in the rowing motion of pectoral fins comprises two sub-movements: a power stroke and a recovery stroke. During the power stroke, the pectoral fin rotates towards the back of the robot to produce thrust through induced drag on the pectoral fin surface, while during the recovery stroke, the fin moves toward the front of the body, ideally with minimal loading, to get ready for the next fin-beat cycle.

Considering the cyclic nature of typical actuation modes, it is of interest to develop a dynamic average model that is amenable to controller design, where the control inputs are actuation pattern parameters. Furthermore, in practical applications it is more natural to control the parameters of periodic fin beats than to directly control the fin position at every moment, which makes an averaged model best suited for trajectory planning and tracking control. In this chapter, we present a nonlinear dynamic average model for robotic fish propelled by a pair of rigid pectoral fins undergoing rowing motion. In particular, we consider the robot undergoing planar motion, with its original dynamics incorporating pectoral fin-generated hydrodynamic forces evaluated via

the blade element theory. Inspired by the work in [89], which deals with averaged dynamics for tail-actuated robotic fish, we seek scaling factors, as functions of fin-beat parameters, for the original hydrodynamic forces and moment, such that when classical averaging is applied to the resulting modified dynamics the obtained average model produces locomotion behaviors close to those of the original dynamic model. One fundamental step in identifying the scaling functions is estimating the scaling values for a given fin-beat pattern. [89] used a trial-and-error approach for the tail-actuated robotic fish, which is time-consuming. We propose a novel systematic approach to finding optimal scaling values by formulating a nonlinear model-predictive control (NMPC) problem, which can be readily solved with NMPC packages. Once the scaling values are found for a set of fin-beat patterns, nonlinear regression is used to determine the scaling functions with minimal complexity. Simulation comparison between the averaged model and the original dynamic model, using fin-beat patterns not used in identifying the scaling functions, supports the efficacy of the developed averaged model. Furthermore, we conduct experiments on a pectoral fin-actuated robotic fish and compare the experimental results with simulation predictions when considering the forward swimming motion, where both fins are actuated symmetrically.

Finally, to demonstrate the utility of the dynamic average model, we design a controller for trajectory tracking of a pectoral fin-actuated robotic fish. In particular, the proposed scheme uses a backstepping-based controller that finds the needed inputs for the robot to track the desired trajectory based upon the averaged model. In this design, the physical control inputs involve two of the fin-beat parameters, the bias, and the amplitude, while the other parameters (angular frequency and power/recover stroke ratio) are kept constant. We further use a multi-variable minimization solver to determine the optimal fin-beat parameters such that the achieved inputs are closed to the needed values. Simulation results are presented to demonstrate the effectiveness of the proposed model-based tracking control scheme.

The rest of the chapter is organized as follows. We first review the dynamic model of the pectoral fin-actuated robotic fish in Section 4.1. In Section 4.2 we present the development of the proposed averaged model. In Section 4.3 we present the scheme to determine the model parameters

and the averaged scaling functions, as well as the experimental and simulation validation of the resulting average model. In Section 4.4 we detail the proposed backstepping-based tracking control approach, with simulation results in Section 4.5. Finally, we provide some concluding remarks in Section 4.6.

4.1 Dynamic Model For Pectoral Fin-actuated Robotic Fish

4.1.1 Rigid Body Dynamics

We consider the robot to be a rigid body with rigid pectoral fins that are actuated at the base, and we assume that the robot operates in an inviscid, irrotational, and incompressible fluid within an infinite domain.

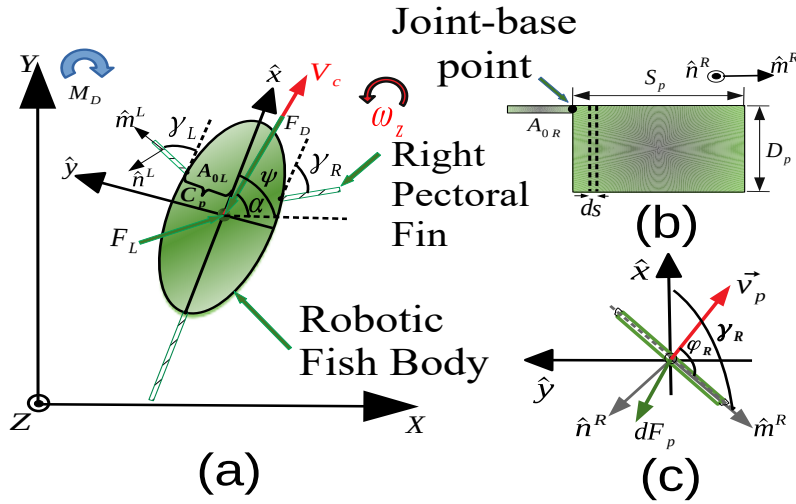


Figure 4.1: (a) Top view of the pectoral fin-actuated robotic fish undergoing planar motion; (b) side view and blade element of the right pectoral fin with its parameters and variables; (c) top view of the pectoral fin with its associated forces and angles.

As illustrated in Figure 4.1(a), we define $[X, Y, Z]^T$ and $[\hat{x}, \hat{y}, \hat{z}]^T$ as the inertial coordinate system and the body-fixed coordinate system, respectively. The velocity of the center of mass in the body-fixed coordinates is expressed as $V_c = [V_{c_x}, V_{c_y}, V_{c_z}]$, where V_{c_x} , V_{c_y} , and V_{c_z} indicate surge, sway, and heave velocities, respectively. The angular velocity expressed in the body-fixed coordinate system is given by $\omega = [\omega_x, \omega_y, \omega_z]$, which is composed of roll (ω_x), pitch (ω_y), and

yaw (ω_z). We let α denote the angle of attack, formed by the direction of V_c with respect to the \hat{x} -axis and is given by $\alpha = \arctan(\frac{V_{cy}}{V_{cx}})$. Let ψ denote the heading angle, formed by the \hat{x} -axis relative to the X-axis, C_p be the distance between the pectoral fin base and the body's center of mass, and A_{0L} and A_{0R} denote the pivot points for the left and right fins, respectively. Finally, γ_L and γ_R represent the angles between the left pectoral fin and the body-fixed \hat{x} -axis and the right pectoral fin and the body-fixed \hat{x} -axis, respectively.

We only consider the robot's planar motion, and further assume that the body is symmetric with respect to the $\hat{x}\hat{z}$ -plane and that the pectoral fins move in the $\hat{x}\hat{y}$ -plane, such that the system only has three degrees of freedom, surge (V_{cx}), sway (V_{cy}), and yaw (ω_z). Furthermore, we assume that we can neglect the inertial coupling between yaw, sway and surge motions [102], and arrive at the following equations of planar motion

$$(m_b - m_{ax})\dot{V}_{cx} = (m_b - m_{ay})V_{cy}\omega_z + f_x \quad (4.1)$$

$$(m_b - m_{ay})\dot{V}_{cy} = -(m_b - m_{ax})V_{cx}\omega_z + f_y \quad (4.2)$$

$$(J_{bz} - J_{az})\dot{\omega}_z = (m_{ay} - m_{ax})V_{cx}V_{cy} + \tau_z \quad (4.3)$$

where m_b is the mass of the body, J_{bz} is the inertia of the body about the \hat{z} -axis, m_{ax} and m_{ay} are the hydrodynamic derivatives that represent the added masses of the robotic fish along the \hat{x} and \hat{y} directions, respectively, and J_{az} represents the added inertia effect of the body about the \hat{z} direction. Finally, the hydrodynamic forces and moment due to the pectoral fin actuation and the interaction of the body itself with the fluid are captured by f_x , f_y , and τ_z and are given by

$$f_x = f_{hx} - F_D \cos(\alpha) + F_L \sin(\alpha) \quad (4.4)$$

$$f_y = f_{hy} - F_D \sin(\alpha) - F_L \cos(\alpha) \quad (4.5)$$

$$\tau_z = \tau_{hz} + M_D \quad (4.6)$$

where f_{hx} , f_{hy} , and τ_{hz} are the hydrodynamic forces and moment transmitted to the fish body by the right and left pectoral fins, while F_D , F_L , and M_D are the body drag, lift, and moment, respectively.

4.1.2 Drag and Lift on the Robot Body

The lift force F_L , drag force F_D , and drag moment M_D acting on the robotic fish can be captured by ([31, 135])

$$F_D = \frac{1}{2} \rho |V_c|^2 S_A C_D \quad (4.7)$$

$$F_L = \frac{1}{2} \rho |V_c|^2 S_A C_L \alpha \quad (4.8)$$

$$M_D = -C_M \omega_z^2 \text{sgn}(\omega_z) \quad (4.9)$$

where ρ is the density of water, $|V_c|$ is the linear velocity magnitude of the body in the body-fixed frame and is defined as $|V_c| = \sqrt{V_{cx}^2 + V_{cy}^2}$, S_A is the wetted surface area for the robot, C_D is the drag force coefficient, C_L is the lift force coefficient, C_M is the drag moment coefficient, and $\text{sgn}(\cdot)$ is the signum function.

4.1.3 Hydrodynamic Forces from Rowing Pectoral Fins

As shown in Figure 4.1(b), we consider the pectoral fins to be rectangular with span length S_p and chord length D_p , and assume that they perform pure rowing motion. To evaluate their hydrodynamic forces, we adopt the procedure proposed in [136]. Furthermore, we illustrate the force calculations using only the right pectoral fin, since they can be trivially extended to the left pectoral fin.

We consider a coordinate system with unit vectors \hat{m}^R and \hat{n}^R that are attached to the pectoral fin. The relationship between these unit vectors and the robotic fish body-fixed coordinates is given by

$$\hat{m}^R = \cos \gamma_R \hat{x} - \sin \gamma_R \hat{y} \quad (4.10)$$

$$\hat{n}^R = -\sin \gamma_R \hat{x} - \cos \gamma_R \hat{y} \quad (4.11)$$

In the following calculations we assume an anchored robotic fish body, often adopted in literature as it simplifies calculation without incurring significant error [25, 137]. The velocity and

acceleration at the point s along the fin are then given by

$$v_{pR}(s, t) = s\dot{\gamma}_R \hat{n}^R \quad (4.12)$$

$$a_R(s, t) = s\ddot{\gamma}_R \hat{n}^R - s\dot{\gamma}_R^2 \hat{m}^R \quad (4.13)$$

where $\dot{\gamma}_R$ and $\ddot{\gamma}_R$ indicate the first and second time derivatives of γ_R , respectively.

The hydrodynamic forces on the pectoral fin have both span-wise and normal components. However, the fins are considered to have pure rowing motion which implies that the span-wise force that arises from friction is very small and can thus be neglected [138]. Using blade theory, we can then calculate the differential normal force $dF_{nR}(s, t)$ on each blade element ds on the pectoral fin at time t as

$$dF_{nR}(s, t) = -\frac{1}{2}C_n(\varphi_R(s, t))\rho D_p |v_{pR}(s, t)|^2 ds \hat{n}^R \quad (4.14)$$

where $C_n(\varphi_R(s, t)) = \lambda \sin \varphi_R(s, t)$ is the normal force coefficient, which depends on the angle of attack of each arbitrary blade, $\varphi_R(s, t)$, and λ is a parameter that can be evaluated empirically through experiments. The angle of attack of the right pectoral fin at each point, $\varphi_R(s, t)$, is defined as

$$\tan \varphi_R(s, t) = \frac{\langle v_{pR}(s, t), \hat{n}^R \rangle}{\langle v_{pR}(s, t), \hat{m}^R \rangle} \quad (4.15)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. Note that in this work we assume $\langle v_{pR}(s, t), \hat{m}^R \rangle = 0$.

The total hydrodynamic force acting on each pectoral fin is calculated by integrating the force density along the span length of the fin such that

$$\begin{aligned} F_{nR}(t) &= - \int_0^{S_p} dF_{nR}(s, t) \\ &= - \int_0^{S_p} \frac{1}{2} \lambda \rho D_p |v_{pR}(s, t)|^2 ds \hat{n}^R \\ &= - \frac{1}{6} \lambda \rho D_p S_p^3 \dot{\gamma}_R^2 \sin \varphi_R \hat{n}^R \\ &= - \frac{1}{6} \lambda \rho D_p S_p^3 \dot{\gamma}_R^2 \text{sgn}(\dot{\gamma}_R) \hat{n}^R \end{aligned} \quad (4.16)$$

where we used $\sin \varphi_R = \sin(\arctan(\frac{s\dot{\gamma}_R}{0})) = \frac{s\dot{\gamma}_R}{\sqrt{(s\dot{\gamma}_R)^2}} = \text{sgn}(\dot{\gamma}_R)$. The total force acting on the right fin is determined by

$$\vec{F}_R = F_{nR} \hat{n}^R - \vec{F}_{A_0R} = m_p a_R(s, t) \Big|_{s=\frac{S_p}{2}} \quad (4.17)$$

where \vec{F}_{A_0} represents the force applied by the rigid pectoral fin on the servo joint, and m_p is the effective mass of the rigid fin (the fin mass m_{pf} and the added mass, where the added mass is calculated based on a rigid plate moving in water [139]).

The moment of the fin relative to its pivot point (A_{0R}) is given by

$$\vec{M}_{nR} = \int_0^{S_p} s \hat{m}_R \times dF_{nR} \quad (4.18)$$

Finally, the force and moment exerted on the robotic fish body by the right pectoral fin is given by

$$\begin{aligned} f_{h_x R} &= \langle \vec{F}_{A_0R}, \hat{x} \rangle \\ &= \frac{1}{6} \lambda \rho D_p S_p^3 \dot{\gamma}_R^2 \sin \gamma_R \text{sgn}(\dot{\gamma}_R) - m_p \left(-\frac{S_p}{2} \ddot{\gamma}_R \sin \gamma_R - \frac{S_p}{2} \dot{\gamma}_R^2 \cos \gamma_R \right) \hat{x} \end{aligned} \quad (4.19)$$

$$\begin{aligned} f_{h_y R} &= \langle \vec{F}_{A_0R}, \hat{y} \rangle \\ &= \frac{1}{6} \lambda \rho D_p S_p^3 \dot{\gamma}_R^2 \cos \gamma_R \text{sgn}(\dot{\gamma}_R) - m_p \left(-\frac{S_p}{2} \ddot{\gamma}_R \cos \gamma_R + \frac{S_p}{2} \dot{\gamma}_R^2 \sin \gamma_R \right) \hat{y} \end{aligned} \quad (4.20)$$

$$\begin{aligned} \tau_{h_z R} &= C_p \hat{y} \times \vec{F}_{A_0} \\ &= -C_p \left(\frac{1}{6} \lambda \rho D_p S_p^3 \dot{\gamma}_R^2 \sin \gamma_R \text{sgn}(\dot{\gamma}_R) - m_p \left(-\frac{S_p}{2} \ddot{\gamma}_R \sin \gamma_R - \frac{S_p}{2} \dot{\gamma}_R^2 \cos \gamma_R \right) \right) \hat{k} \end{aligned} \quad (4.21)$$

For a more comprehensive derivation of the hydrodynamic forces, we refer the reader to [136]. By considering the kinematic equations of the robotic fish, the final dynamic model can be summarized as follows:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\psi} \\ \dot{v}_{cx} \\ \dot{v}_{cy} \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} v_{cx} \cos \psi - v_{cy} \sin \psi \\ v_{cx} \sin \psi + v_{cy} \cos \psi \\ \omega_z \\ f_1(v_{cx}, v_{cy}, \omega_z) + \frac{f_{hxR} + f_{hxL}}{m_1} \\ f_2(v_{cx}, v_{cy}, \omega_z) + \frac{f_{hyR} + f_{hyL}}{m_2} \\ f_3(v_{cx}, v_{cy}, \omega_z) + \frac{\tau_{hR} + \tau_{hL}}{J_3} \end{bmatrix} \quad (4.22)$$

where

$$\begin{cases} f_1(v_{cx}, v_{cy}, \omega_z) = \frac{m_2}{m_1} v_{cy} \omega_z - \frac{c_1}{m_1} v_{cx} \sqrt{v_{cx}^2 + v_{cy}^2} + \frac{c_2}{m_1} v_{cy} \sqrt{v_{cx}^2 + v_{cy}^2} \arctan\left(\frac{v_{cy}}{v_{cx}}\right) \\ f_2(v_{cx}, v_{cy}, \omega_z) = -\frac{m_1}{m_2} v_{cx} \omega_z - \frac{c_1}{m_2} v_{cy} \sqrt{v_{cx}^2 + v_{cy}^2} - \frac{c_2}{m_2} v_{cx} \sqrt{v_{cx}^2 + v_{cy}^2} \arctan\left(\frac{v_{cy}}{v_{cx}}\right) \\ f_3(v_{cx}, v_{cy}, \omega_z) = \frac{(m_1 - m_2)}{J_3} v_{cx} v_{cy} - c_4 \omega_z^2 \text{sgn}(\omega_z) \end{cases} \quad (4.23)$$

with $m_1 = m_b - m_{ax}$, $m_2 = m_b - m_{ay}$, $J_3 = J_{bz} - J_{az}$, $c_1 = \frac{1}{2} \rho S C_D$, $c_2 = \frac{1}{2} \rho S C_L$, $c_4 = \frac{1}{(J_3)} C_M$.

4.2 Averaging with Scaled Forcing

In this section we present the scaled-averaging approach and the proposed scheme to estimate the scaling functions. Finally, we present experimental and simulation results to validate the proposed averaged model.

4.2.1 Averaged Model

In order to generate a net thrust over each cycle, the pectoral fins need to be actuated differently in the power and recovery strokes. For example, to generate forward thrust the fin has to be actuated faster in the power stroke than in the recovery stroke. We specify the fin beat pattern as

$$\gamma(t) = \begin{cases} \gamma_0 - \gamma_A \cos\left[\pi \frac{(\zeta + 1)}{T_p} t\right], & 0 \leq t \leq \frac{T_p}{\zeta + 1} \\ \gamma_0 + \gamma_A \cos\left[\pi \left(\frac{\zeta + 1}{\zeta T_p}\right) \left(t - \frac{T_p}{\zeta + 1}\right)\right], & \frac{T_p}{\zeta + 1} < t \leq T_p \end{cases} \quad (4.24)$$

where γ_0 is the fin-beat bias, γ_A is the fin-beat amplitude, T_P is the fin-beat period, and ζ is a parameter defining the ratio of angular velocities of the fin during the power and recovery strokes, respectively.

Under this periodic fin movement, averaging can be a useful tool for gaining insight into the effect of the input parameters (such as the beat bias γ_0 , amplitude γ_A , period T_P , and ratio ζ) on the dynamics and for designing controllers. First-order averaging [96] tends to generate prohibitively complex model for control [89]. On the other hand, classical averaging (directly averaging the vector field over one period of the fin-beat) cannot be directly applied since the dynamics is not slow in typical scenarios as shown in [89]. Therefore, we first scale the original forcing terms with functions that are potentially dependent on the fin-beat parameters, and then apply classical averaging over the modified dynamics.

Specifically, let the original system (4.22) be modified as

$$\begin{aligned}\dot{v}_{cx} = & f_1(v_{cx}, v_{cy}, \omega_z) + K_{f_x R}(\gamma_{0R}, \gamma_{AR}, T_{pR}, \zeta_R) \cdot f_{h_x R}(t) \\ & + K_{f_x L}(\gamma_{0L}, \gamma_{AL}, T_{pL}, \zeta_L) \cdot f_{h_x L}(t)\end{aligned}\quad (4.25)$$

$$\begin{aligned}\dot{v}_{cy} = & f_2(v_{cx}, v_{cy}, \omega_z) + K_{f_y R}(\gamma_{0R}, \gamma_{AR}, T_{pR}, \zeta_R) \cdot f_{h_y R}(t) \\ & + K_{f_y L}(\gamma_{0L}, \gamma_{AL}, T_{pL}, \zeta_L) \cdot f_{h_y L}(t)\end{aligned}\quad (4.26)$$

$$\begin{aligned}\dot{\omega}_z = & f_3(v_{cx}, v_{cy}, \omega_z) + K_{m_R}(\gamma_{0R}, \gamma_{AR}, T_{pR}, \zeta_R) \cdot \tau_{h_R}(t) \\ & + K_{m_L}(\gamma_{0L}, \gamma_{AL}, T_{pL}, \zeta_L) \cdot \tau_{h_L}(t)\end{aligned}\quad (4.27)$$

where $K_{f_x R}(\cdot)$, $K_{f_y R}(\cdot)$, $K_{m_R}(\cdot)$, $K_{f_x L}(\cdot)$, $K_{f_y L}(\cdot)$, $K_{m_L}(\cdot)$ are scaling functions to be determined later (Section 4.3.3), and are $\{\gamma_{0R}, \gamma_{AR}, T_{pR}, \zeta_R\}$ and $\{\gamma_{0L}, \gamma_{AL}, T_{pL}, \zeta_L\}$ are the fin-beat parameters of the right fin and left fin, respectively. For brevity, the arguments of the functions $f_1(\cdot)$, $f_2(\cdot)$, $f_3(\cdot)$ and the scaling functions are omitted in the remainder of the chapter, and the calculations are illustrated using only the right fin since they can be extended for the left fin in a straightforward manner.

To avoid the integration of nested sin functions and facilitate the computation of the averaging, we first use the second-order Taylor series expansion to approximate the $\cos(\gamma)$ and $\sin(\gamma)$ terms

that appear in the forcing terms in f_{h_x} and f_{h_y} . After conducting classical averaging, the following averaged system is obtained

$$\begin{cases} \dot{\bar{v}}_{c_x} = f_1 + K_{f_x R} \cdot \bar{f}_{h_x R}(\gamma_{0R}, \gamma_{AR}, T_{pR}, \zeta_R) + K_{f_x L} \cdot \bar{f}_{h_x L}(\gamma_{0L}, \gamma_{AL}, T_{pL}, \zeta_L) \\ \dot{\bar{v}}_{c_y} = f_2 + K_{f_y R} \cdot \bar{f}_{h_y R}(\gamma_{0R}, \gamma_{AR}, T_{pR}, \zeta_R) + K_{f_y L} \cdot \bar{f}_{h_y L}(\gamma_{0L}, \gamma_{AL}, T_{pL}, \zeta_L) \\ \dot{\bar{\omega}}_z = f_3 + K_{m_R} \cdot \bar{\tau}_{h_R}(\gamma_{0R}, \gamma_{AR}, T_{pR}, \zeta_R) + K_{m_L} \cdot \bar{\tau}_{h_L}(\gamma_{0L}, \gamma_{AL}, T_{pL}, \zeta_L) \end{cases} \quad (4.28)$$

where

$$\bar{f}_{h_x R} = \frac{D_p \lambda l_p^3 S_p^3 \pi^2 \rho \gamma_0 \gamma_A^2 (-4\gamma_0^2 - 3(-8 + \gamma_A^2))(-1 + \zeta^2)}{288 m_1 \zeta T_p^2} \quad (4.29)$$

$$\bar{f}_{h_y R} = \frac{-\gamma_A^2 l_p \pi^2}{96 m_2 T_p^2 \zeta} \left(4 D_p \gamma_0^2 \lambda l_p^2 \rho (-1 + \zeta^2) + \right. \quad (4.30)$$

$$\left. D_p (-8 + \gamma_A^2) \lambda l_p^2 \rho (-1 + \zeta^2) - 4 \gamma_0^3 m_p (1 + \zeta)^2 - 3 \gamma_0 \gamma_A^2 m_p (1 + \zeta)^2 \right) \\ \bar{\tau}_{h_R} = \frac{C_p D_p \lambda l_p^3 \pi^2 \rho \gamma_0 \gamma_A^2 (-4\gamma_0^2 - 3(-8 + \gamma_A^2))(-1 + \zeta^2)}{288 J_3 \zeta T_p^2} \quad (4.31)$$

Note that the model (4.28) can be expressed in a control-affine form if one defines the control inputs as $\alpha_1 = K_{f_x R} \bar{f}_{h_x R}$, $\alpha_2 = K_{f_x L} \bar{f}_{h_x L}$, $\alpha_3 = K_{f_y R} \bar{f}_{h_y R}$ and $\alpha_4 = K_{f_y L} \bar{f}_{h_y L}$. As an example, we refer the reader to [92], where the authors show how one can express an averaged model for a trail-actuated robotic fish in a control-affine form and use it to design a model-predictive controller.

4.3 Simulation and Experimental Model Validation

In order to validate the presented averaged model, we must first identify the hydrodynamic parameters (C_D , C_L , and C_M), the fin parameter (λ), and the scaling functions ($K_{f_x R}$, $K_{f_y R}$, K_{m_R} , $K_{f_x L}$, $K_{f_y L}$, K_{m_L}) for the model. In this section we present the experimental setup and discuss the experimental identification and validation of the hydrodynamic and fin model parameters and present an estimation scheme to obtain the scaling coefficients. Finally, we present simulation and experimental results to evaluate the effectiveness of the proposed averaged model.

4.3.1 Experimental Setup

To validate the presented averaged model, we conduct experiments using the free-swimming robotic fish depicted in Figure 4.2. The robot consists of a rigid-shell body, a tail and two rigid pectoral fins, which were all 3D-printed. Although the robot also has a servo-actuated caudal fin, the tail-actuation is not included in this work. The body and fins dimensions are shown in Table 4.1. Two Tenenergy Li-Ion rechargeable batteries (7.4V, 3350mAh) are used to power the robot, and two Hitec digital micro waterproof servos (HS-5086WP) are used to actuate the pectoral fins. Finally, a Microchip Digital Signal Processor and Controller (DSPIC30F6014) is used to realize the control of the servos and a Xbee module is used for wireless communication with a computer.

The robotic fish is run in a 2.30 m by 1.2 m space enclosed within a tank equipped with an overhead Logitech C930E camera as seen in Figure 4.2. To obtain the robotic fish's position and orientation in the tank, two markers were attached to the anterior and posterior of the robotic fish body. An overhead video of the robotic fish swimming in the tank is captured using the camera, and Visual C++ with the OpenCV library is used to implement an image processing algorithm. The algorithm detects the positions of the two markers and uses their average to obtain the center position of the robotic fish. In addition, the heading angle of the robot is estimated using the positions of the two markers, and a high gain observer is used to estimate the linear and angular velocities of the robot based on the measured position and heading.

4.3.2 Original Dynamics Parameter Identification

Before identifying the scaling functions and validating the averaged model, the hydrodynamic parameters present in the dynamic and averaged model must first be identified. These parameters are either measured directly or calculated based on measurements and are summarize in Table 4.1. As typically done in literature [102, 140], the body inertia about \hat{z} -axis is evaluated as $J_{bz} = \frac{1}{5}m_b(a^2 + c^2)$, where $a = \frac{\text{Body length}}{2}$ and $c = \frac{\text{Body width}}{2}$ are the semiaxis lengths. Furthermore, the added masses, added inertia, and wetted surface are calculated based on a prolate spheroid approximation of the robotic fish body [102, 130].

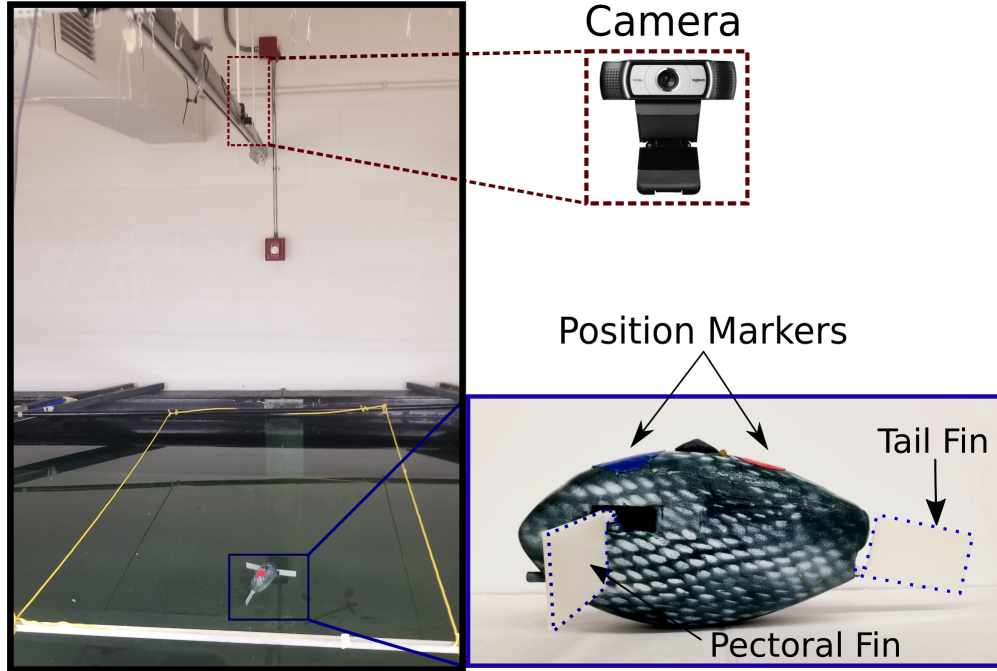


Figure 4.2: The experimental setup. During experiments the pectoral-fin actuated robotic fish swims within the enclosed area (denoted by the yellow lines) in the tank, and the overhead Logitech camera captures a video of the robot swimming. An image processing algorithm detects the red and blue markers placed on top of the robot to localize it and determine its heading.

The fin normal force coefficient λ , as well as the body drag and lift coefficients (C_D , C_L , and C_M) are identified empirically from data collected using the robotic fish described above. In particular, we consider only turning motions which are achieved by activating only one fin at a time according to (4.24).

To determine the body drag, lift and moment coefficients (C_D , C_L , and C_M), we let the robotic fish swim for some time (approximately 35 s) to reach the steady-state motion, and then stop actuating the pectoral fin such that the robot slowly halts to a stop. In particular, considering (5.2), once the actuation has stopped, the dynamics equations become

$$\begin{bmatrix} \dot{v}_{cx} \\ \dot{v}_{cy} \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} \frac{m_2}{m_1} v_{cy} \omega_z - \frac{c_1}{m_1} v_{cx} \sqrt{v_{cx}^2 + v_{cy}^2} + \frac{c_2}{m_1} v_{cy} \sqrt{v_{cx}^2 + v_{cy}^2} \\ -\frac{m_1}{m_2} v_{cx} \omega_z - \frac{c_1}{m_2} v_{cy} \sqrt{v_{cx}^2 + v_{cy}^2} - \frac{c_2}{m_2} v_{cx} \sqrt{v_{cx}^2 + v_{cy}^2} \\ \frac{(m_1 - m_2)}{J_3} v_{cx} v_{cy} - c_4 \omega_z^2 \text{sgn}(\omega_z) \end{bmatrix} \quad (4.32)$$

where $m_1 = m_b - m_{ax}$, $m_2 = m_b - m_{ay}$, $J_3 = J_{bz} - J_{az}$, $c_1 = \frac{1}{2} \rho S C_D$, $c_2 = \frac{1}{2} \rho S C_L$, $c_4 = \frac{1}{3} C_M$.

Let $\phi = [C_D \ C_L \ C_M]^T$ and rearrange (4.32) such that

$$Y = X\phi \quad (4.33)$$

where

$$Y = \begin{bmatrix} \dot{v}_{cx} - \frac{m_2}{m_1} v_{cy} \\ \dot{v}_{cy} - \frac{m_1}{m_2} v_{cx} \omega_z \\ \dot{\omega}_z - \frac{(m_1 - m_2)}{J_3} v_{cx} v_{cy} \end{bmatrix} \quad (4.34)$$

$$X = \begin{bmatrix} -\frac{\bar{c}_1}{m_1} v_{cx} \sqrt{v_{cx}^2 + v_{cy}^2} & \frac{\bar{c}_2}{m_1} v_{cy} \sqrt{v_{cx}^2 + v_{cy}^2} & 0 \\ -\frac{\bar{c}_1}{m_2} v_{cy} \sqrt{v_{cx}^2 + v_{cy}^2} & -\frac{\bar{c}_2}{m_2} v_{cx} \sqrt{v_{cx}^2 + v_{cy}^2} & 0 \\ 0 & 0 & -\bar{c}_4 \omega_z^2 \text{sgn}(\omega_z) \end{bmatrix} \quad (4.35)$$

where $\bar{c}_1 = \frac{1}{2}\rho S$, $\bar{c}_2 = \frac{1}{2}\rho S$, $\bar{c}_4 = \frac{1}{J_3}$. We use the captured video along with the image processing algorithm to determine the body-fixed velocities, and a high gain observer to estimate the body-fixed acceleration for different sets of fin amplitudes (γ_{AR}): 15° , 18° , 20° , 25° , 30° ; biases (γ_{0R}): 80° , 90° , 100° , 110° ; periods (T_{pR}): 0.5, 0.66, 1 s; and power/ recovery stroke ratios (ζ): 5, 6.

Furthermore, to estimate the fin parameter λ , we formulate another parameter estimation paradigm, where we consider constant actuation. In particular, using the hydrodynamic parameters estimated above along with the dynamic equations and Eqs. (4.19)-(4.21), we can obtain the following

$$\begin{bmatrix} \dot{V}_{cx} - f_1(V_{cx}, V_{cy}, \omega_z) - \frac{\hat{f}_{hx}}{m_1} \\ \dot{V}_{cy} - f_2(V_{cx}, V_{cy}, \omega_z) - \frac{\hat{f}_{hy}}{m_2} \\ \dot{\omega}_z - f_3(V_{cx}, V_{cy}, \omega_z) - \frac{\hat{\tau}_{hz}}{J_3} \end{bmatrix} = \lambda \begin{bmatrix} \frac{\frac{1}{6}\rho D_p S_p^3 \dot{\gamma}_R^2 \sin \gamma_R \text{sgn}(s\dot{\gamma}_R)}{m_1} \\ \frac{\frac{1}{6}\rho D_p S_p^3 \dot{\gamma}_R^2 \cos \gamma_R \text{sgn}(s\dot{\gamma}_R)}{m_2} \\ \frac{-C_p \frac{1}{6}\rho D_p S_p^3 \dot{\gamma}_R^2 \sin \gamma_R \text{sgn}(s\dot{\gamma}_R)}{J_3} \end{bmatrix} \quad (4.36)$$

where

$$\begin{aligned} \hat{f}_{hxR} &= -m_p \left(-\frac{S_p}{2} \ddot{\gamma}_R \sin \gamma_R - \frac{S_p}{2} \dot{\gamma}_R^2 \cos \gamma_R \right) \\ \hat{f}_{hyR} &= -m_p \left(-\frac{S_p}{2} \ddot{\gamma}_R \cos \gamma_R + \frac{S_p}{2} \dot{\gamma}_R^2 \sin \gamma_R \right) \\ \hat{\tau}_{hzR} &= -C_p \left(-m_p \left(-\frac{S_p}{2} \ddot{\gamma}_R \sin \gamma_R - \frac{S_p}{2} \dot{\gamma}_R^2 \cos \gamma_R \right) \right) \end{aligned} \quad (4.37)$$

We collect the robotic fish steady-state body-fixed velocities for another set of fin amplitudes (γ_{AR}): $8^\circ, 10^\circ, 13^\circ, 15^\circ, 18^\circ$; biases (γ_{0R}): $85^\circ, 93^\circ, 98^\circ, 100^\circ, 103^\circ$; periods (T_{pR}): 0.5, 0.66 s ; and power/ recovery stroke ratios (ζ): 4, 5, and estimate the parameter λ using (4.36). The resulting coefficients are listed in Table 4.1. These parameters are then used in independent model validation for the dynamic model.

Table 4.1: Identified model parameters for the robotic fish.

Robot Body		
Parameter	Value	unit
Body Length	0.198	m
Body Height	0.1	m
Body Width (C_P)	0.03	m
Mass (m_b)	0.795	kg
Inertia (J_{bz})	4.26×10^{-4}	$\text{kg} \cdot \text{m}^2$
$-m_{ax}$	0.095	kg
$-m_{ay}$	0.1794	kg
$-J_{az}$	2.7×10^{-5}	kg/m^2
Wet surface area (S_A)	0.325	m^2
Drag coef. (C_D)	0.3870	-
Lift coef. (C_L)	0.0808	-
Moment coef. (C_M)	8.5×10^{-3}	kg/m^2
Pectoral Fin		
Parameter	Value	unit
Fin Length (S_p)	0.061	m
Fin Height (D_p)	0.041	m
Fin Mass (m_{pf})	0.008	kg
Effective mass (m_p)	0.008	kg
Water density (ρ)	1000	kg/m^3
λ	4.1464	-

To validate the dynamic model, we conduct experiments based on different fin actuation parameters in both forward swimming and turning. In the forward swimming case both left and right fins are actuated in sync with the same fin-beat patterns. We compare the steady state swimming speeds predicted by the model and those obtained from experiments in the forward swimming case, whilst in the turning cases we compare the turning radius and period. Each experiment is repeated four times to obtain the average and standard deviation. Table 4.2 and Table 4.3 list the percent errors between the values obtained from experiments and those obtained from simulation using the parameters estimated earlier. The comparison indicates that the dynamic model has acceptable accuracy.

Table 4.2: Model validation results: relative model prediction error for turning radius and turning period.

$(\gamma_{0L}, \gamma_{AL}, T_{pL}, \zeta_L)$	Turning Radius Error(%)	Turning Period Error(%)
$(85^\circ, 18^\circ, 1 \text{ s}, 4)$	10.96	5.96
$(85^\circ, 18^\circ, 1 \text{ s}, 3)$	4.66	17.92
$(80^\circ, 22^\circ, 1 \text{ s}, 4)$	12.67	4.91
$(80^\circ, 22^\circ, 1 \text{ s}, 3)$	16.15	2.12
$(85^\circ, 22^\circ, 1 \text{ s}, 4)$	2.60	1.39
$(85^\circ, 22^\circ, 1 \text{ s}, 3)$	13.61	4.82

Table 4.3: Model validation results: surge velocities predicted by the original model and measured from experiments, and their relative error.

$(\gamma_{0L}, \gamma_{AL}, T_{pL}, \zeta_L)$	Surge Velocity(Experiments)	Surge Velocity(Model)	Error (%)
$(95^\circ, 12^\circ, 1 \text{ s}, 4)$	0.0383	0.0397	3.60
$(95^\circ, 12^\circ, 0.66 \text{ s}, 5)$	0.0423	0.0449	5.77
$(100^\circ, 12^\circ, 0.8 \text{ s}, 4)$	0.0536	0.0494	8.52
$(100^\circ, 12^\circ, 0.8 \text{ s}, 5)$	0.0614	0.0558	9.96
$(100^\circ, 12^\circ, 0.66 \text{ s}, 5)$	0.0490	0.0447	9.74

4.3.3 Identification of Scaling Functions

To identify the corresponding scaling functions for the averaged model, we conduct simulations using the original dynamic model (with the experimentally identified parameters presented in the previous subsection) considering only one fin (the right fin) actuated with a given fin-beat pattern, and seek the corresponding values of the scaling functions (i.e., $K_{f_x R}, K_{f_y R}, K_{m_R}$) such that the resulting average model (4.28) produces the best match in the turning radius and turning period with those of the original dynamics (4.22).

Under a given fin actuation pattern, instead of conducting blanket-search of the scaling parameters as done in [89], we propose a novel formulation that treats the scaling values for the given actuation pattern as constant control inputs to the averaged model and solve for these values through nonlinear model predictive control [100], such that the averaged system tracks the surge, sway, and angular velocities extracted from the simulated trajectory of the original dynamics (4.22) under the same fin-actuation pattern. We elaborate on this below.

Simulations are first conducted using the original dynamic model with different sets of fin-beat parameters $(\gamma_0, \gamma_A, T_P, \zeta)$, where we use body parameters identified in the previous subsection (Table 4.1). Only the right fin is actuated, since the scaling functions are considered left-right symmetric. In particular, 560 simulations are conducted with the combination of the following fin-beat patterns: 7 different amplitudes (γ_{AR}) : $10^\circ, 15^\circ, 20^\circ, 22^\circ, 25^\circ, 28^\circ, 30^\circ$; 5 different biases (γ_{0R}) : $50^\circ, 60^\circ, 70^\circ, 80^\circ, 90^\circ$; 4 different periods (T_{pR}) : 0.5, 0.66, 1, 2 s; and 4 different power/recovery stroke ratios (ζ) : 2, 3, 4, 5.

For each simulation, we extract the turning radius R , turning period T_b (time taken to complete one turn) and angle of attack α of the robot at the steady state, and then use the following relationships to determine the corresponding steady-state body-fixed linear (v_{cx}, v_{cy}) and angular (ω_z) velocities for the original dynamics:

$$T_b = \frac{2\pi}{\omega_z}, \quad R = \frac{\sqrt{v_{cx}^2 + v_{cy}^2}}{\omega_z}, \quad \alpha = \arctan \frac{v_{cy}}{v_{cx}} \quad (4.38)$$

These velocities $(v_{cx}, v_{cy}, \omega_z)$ are then considered as desired values to be tracked by the averaged model: $\bar{v}_{xr} = v_{cx}, \bar{v}_{yr} = v_{cy}, \bar{\omega}_r = \omega_z$. We further let $u_1 = K_{fxR}, u_2 = K_{fyR}, u_3 = K_{mR}$ such that (4.28) can be rewritten as

$$\dot{\bar{v}}_{cx} = f_1 + u_1 \bar{f}_{hxR} \quad (4.39)$$

$$\dot{\bar{v}}_{cy} = f_2 + u_2 \bar{f}_{hyR} \quad (4.40)$$

$$\dot{\bar{\omega}}_z = f_3 + u_3 \bar{\tau}_{hR} \quad (4.41)$$

We construct the velocity tracking error as

$$\mathbf{e}_\eta = \begin{bmatrix} \bar{v}_{cx} - \bar{v}_{xr} \\ \bar{v}_{cy} - \bar{v}_{yr} \\ \bar{\omega}_z - \bar{\omega}_r \end{bmatrix} \quad (4.42)$$

The objective is to determine u_1, u_2, u_3 such that the tracking error states of system (4.42) are driven to zero. To do so, we define an objective function with the following stage cost $F(\cdot)$ and

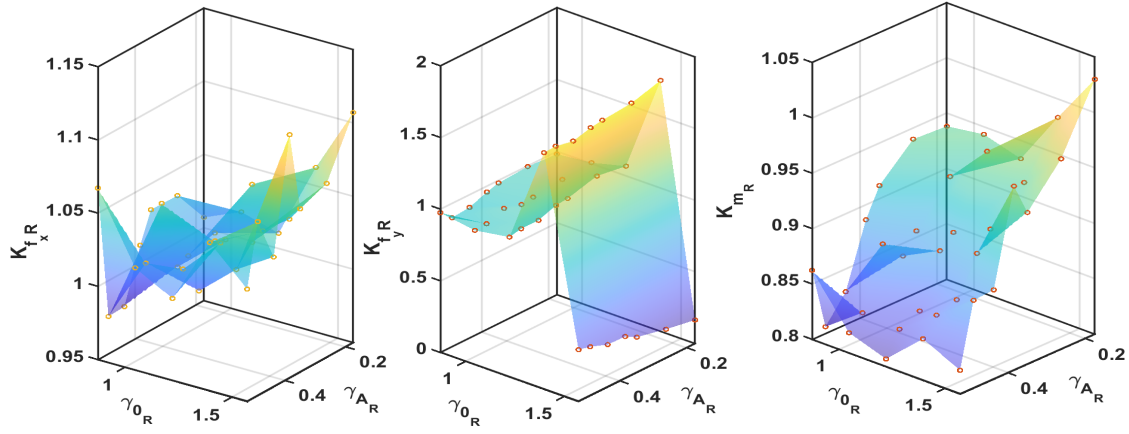


Figure 4.3: Obtained scaling values K_{fxR} , K_{fyR} , K_{mR} versus the amplitude γ_{AR} and bias γ_{0R} , for fixed actuation period $T_{pR} = 1$ s and ratio $\zeta = 2$.

terminal penalty $E(\cdot)$

$$F(\mathbf{e}_\eta, \mathbf{u}) = \mathbf{e}_\eta(\tau)^T \mathbf{Q} \mathbf{e}_\eta(\tau) \quad (4.43)$$

$$E(\mathbf{e}_\eta(t+T)) = (\mathbf{e}_\eta(t+T))^T \mathbf{Q}_T (\mathbf{e}_\eta(t+T)) \quad (4.44)$$

where T is the prediction horizon, $\mathbf{u}(\tau) = [u_1 \ u_2 \ u_3]^T$, and \mathbf{Q} and \mathbf{Q}_T are positive-definite weighting matrices that penalize deviations from the desired values. By solving the nonlinear model predictive control (NMPC) problem (for example, using ACADO Model Predictive Control Toolkit [90]), we obtain the optimal inputs u_1, u_2, u_3 and thus the values for K_{fxR}, K_{fyR}, K_{mR} for a given fin-beat parameter combination. Specifically, the following NMPC parameters are used in solving for the scaling values:

Length of optimization horizon : $T = 10$ s

Sampling interval : $t_s = 1$ s

Weighting matrix : $\mathbf{Q} = 2000\mathbf{I}_3$

Terminal Penalty Weighting matrix : $\mathbf{Q}_T = 80\mathbf{I}_3$

where \mathbf{I}_3 is a 3 by 3 identity matrix. Figure 4.3 shows the resultant 3D surfaces of optimal coefficients obtained for different fin amplitudes and fin biases when the fin actuation period and

Table 4.4: Averaged model validation results: relative model prediction error between the original and averaged models for turning radius and turning period.

$(\gamma_{0L}, \gamma_{AL}, T_{pL}, \zeta_L)$	Turning Radius Error(%)	Turning Period Error(%)
$(90^\circ, 22^\circ, 1 \text{ s}, 4)$	3.51	0.21
$(90^\circ, 22^\circ, 1 \text{ s}, 3)$	3.58	3.13
$(90^\circ, 20^\circ, 1 \text{ s}, 4)$	3.45	1.34
$(90^\circ, 20^\circ, 1 \text{ s}, 3)$	3.77	5.58
$(80^\circ, 22^\circ, 1 \text{ s}, 4)$	2.93	1.84
$(80^\circ, 22^\circ, 1 \text{ s}, 3)$	3.31	2.56
$(80^\circ, 20^\circ, 1 \text{ s}, 4)$	2.86	0.73
$(80^\circ, 20^\circ, 1 \text{ s}, 3)$	3.78	2.17
$(60^\circ, 22^\circ, 1 \text{ s}, 4)$	2.14	4.06
$(60^\circ, 22^\circ, 1 \text{ s}, 3)$	2.29	2.00
$(60^\circ, 20^\circ, 1 \text{ s}, 4)$	1.97	0.19
$(60^\circ, 20^\circ, 1 \text{ s}, 3)$	2.28	4.60

power/recover stroke ratio are fixed at some particular values. Results for other period and ratio combinations are similar and thus omitted in the interest of brevity.

To obtain closed-form expressions for the relationships between each scaling coefficient and the fin-beat parameters, we implement a multivariate nonlinear regression scheme using the MATLAB command *fitnlm*, where we seek the lowest-degree polynomials that provide adequate match with the computed scaling coefficients, and obtain the following:

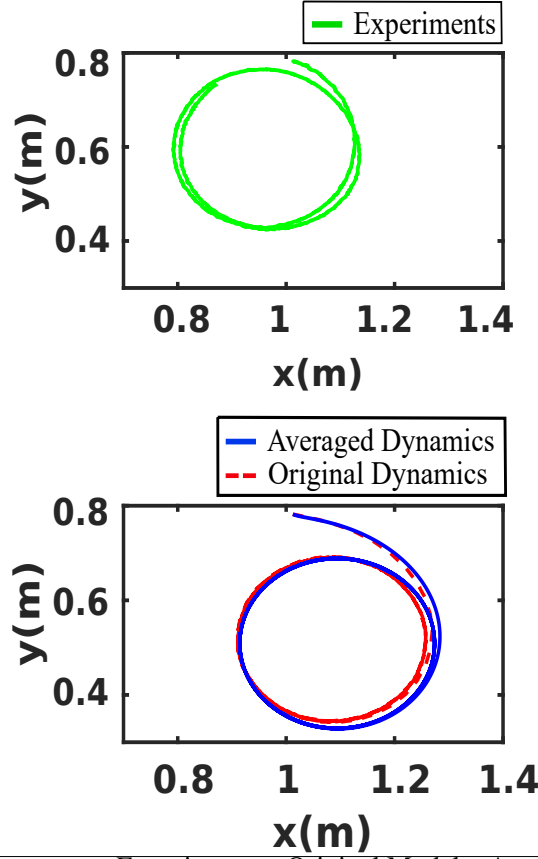
$$K_{fxR} = 0.9801 - 0.0653\gamma_{AR} + 0.0828\gamma_{0R} - 0.0007\zeta_R - 0.0069T_{pR} \quad (4.45)$$

$$K_{fyR} = -529.2 + 0.6\gamma_{AR} + 1893.4\gamma_{0R} - 3.8\gamma_{AR}^2 - 2500.6\gamma_{0R}^2 - 0.0000276\zeta_R^2 \\ - 0.0001324T_{pR}^2 + 9.1\gamma_{AR}^3 + 1446.6\gamma_{0R}^3 - 8.2\gamma_{AR}^4 - 309.1\gamma_{0R}^4 \quad (4.46)$$

$$K_{mR} = 0.9869 - 0.4004\gamma_{AR} + 0.0661\gamma_{0R} - 0.0007\zeta_R - 0.0059T_{pR} \quad (4.47)$$

4.3.4 Validation of the Averaged Model

We first compare the turning radius and period predictions between the averaged and the dynamic model for the turning case. Table 4.4 lists the errors between the predictions obtained from the averaged model and those obtained from the dynamic model for different sets of fin parameters. Furthermore, Figure 4.4 depicts circular trajectories obtained from experiments and simulation using the dynamic and averaged models. From the figure, we can see how the trajectory behavior



Measurement	Experiments	Original Model	Averaged Model
Turning Radius	0.169 m	0.17 m	0.18 m
Turning Period	26.59 s	31.55 s	32.44 s

Figure 4.4: Circular trajectories captured in experiments and predicted by the original and average dynamic models. In this case the right fin remains still and the left fin undergoes actuation with the following parameters fixed: $\gamma_{AL} = 22^\circ$, $\gamma_{0L} = 85^\circ$, $T_{PL} = 1$ and $\zeta = 4$.

and steady-state radius of the dynamic and averaged model predictions match that of experiments, which suggest that the average model is able to capture well the behavior of the original dynamics and the dynamics of the robot under the new actuation patterns.

To further validate the average model, we conduct simulations and experiments considering the forward swimming motion. The forward swimming case was not used in obtaining the scaling functions, and thus provides independent validation for the proposed average model. Figure 4.5 compares the simulated steady-state forward swimming speeds predicted with the original dynamic and the average models, and those obtained from the experiments. Note that the steady state is considered to be reached after the first 15 s of the robot swimming. In the experiments, for the given

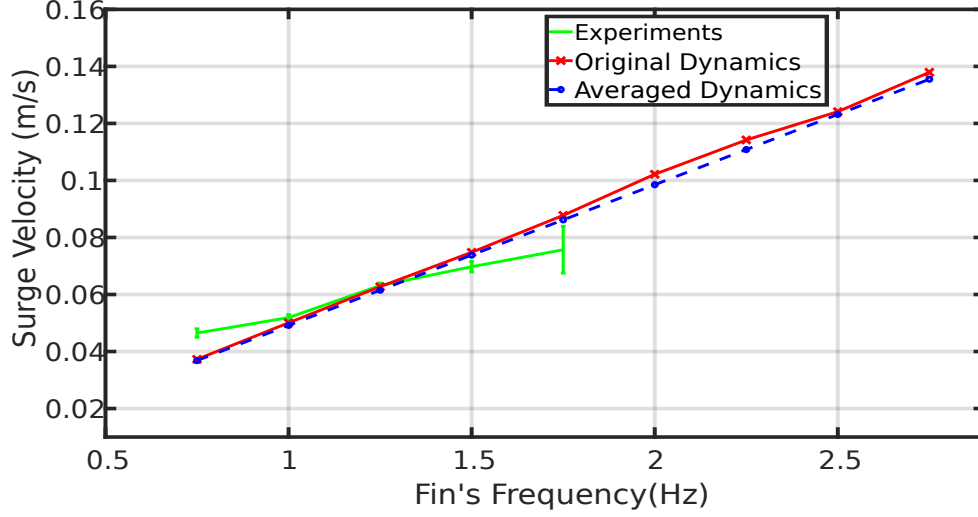


Figure 4.5: Comparison between experimental results and the predictions of the steady-state forward swimming speed based on the average model and the original dynamic model. In this study the left and right fins undergo symmetric actuation with the following parameters fixed: $\gamma_A = 15^\circ$, $\gamma_0 = 95^\circ$ and $\zeta = 4$.

$\zeta = 4$, to prevent exceeding the speed limit of the servo motors, a maximum actuation frequency 1.75 Hz is used. We have extended the simulation results to fin-beat frequency up to 2.75 Hz in order to capture the performance trend of the robotic fish.

4.4 Average Dynamic Model-based Trajectory Tracking Control

To illustrate the utility and further evaluate the validity of the proposed averaged model, we consider the trajectory tracking control problem as an example.

4.4.1 Trajectory Tracking Problem

Let the vectors $\bar{\mathbf{F}}(t) = [X \ Y \ \psi]^T$ and $\bar{\mathbf{T}}(t) = [X_r \ Y_r \ \psi_r]^T$ denote the position (of the center) and orientation of the robotic fish and the desired position/orientation, respectively, with respect to the inertial frame $\{I\}$ at a given time t . Let $\mathbf{e}_\chi = [X_e \ Y_e \ \psi_e]^T$ denote the tracking error vector expressed in the robot's body-fixed frame:

$$\mathbf{e}_\chi = {}^I\mathbf{R}_B(\bar{\mathbf{F}} - \bar{\mathbf{T}}) \quad (4.48)$$

where ${}^I\mathbf{R}_B$ is the rotation matrix from the inertial frame {I} to the body-fixed frame {B} and it is defined as where

$${}^I\mathbf{R}_B = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.49)$$

Following the procedure outlined in Chapter 3, the following error state model augmented with the averaged dynamics (Eq. (4.28)), is obtained

$$\begin{bmatrix} \dot{X}_e \\ \dot{Y}_e \\ \dot{\psi}_e \\ \dot{\bar{v}}_{cx} \\ \dot{\bar{v}}_{cy} \\ \dot{\bar{\omega}}_z \end{bmatrix} = \begin{bmatrix} \bar{v}_{cx} - \mathbf{V}_r \cos(\psi_e - \bar{\psi}_r) + \bar{\omega}_z Y_e \\ \bar{v}_{cy} + \mathbf{V}_r \sin(\psi_e - \bar{\psi}_r) - \bar{\omega}_z X_e \\ \bar{\omega}_z - \omega_r \\ f_1 + K_{fxR} \cdot \bar{f}_{hxR} + K_{fxL} \cdot \bar{f}_{hxL} \\ f_2 + K_{fyR} \cdot \bar{f}_{hyR} + K_{fyL} \cdot \bar{f}_{hyL} \\ f_3 + K_{mR} \cdot \bar{\tau}_{hR} + K_{mL} \cdot \bar{\tau}_{hL} \end{bmatrix} \quad (4.50)$$

where $\mathbf{V}_r = \sqrt{\dot{u}_r^2 + \dot{v}_r^2}$, $\bar{\psi}_r = \arctan(\frac{v_r}{u_r})$, $\omega_r = \dot{\psi}$, and u_r , v_r and ω_r are some surge, sway and angular velocities, respectively, used to generate the desired trajectory such that $\dot{X}_r = u_r \cos \psi_r - v_r \sin \psi_r$ and $\dot{Y}_r = u_r \sin \psi_r + v_r \cos \psi_r$.

By formulating the tracking problem in terms of the error dynamics, the control objective has become a stabilization problem, where the task is to find a control law such that, for an arbitrary initial error, the position and orientation error states (X_e, Y_e, ψ_e) of system (4.50) converge to the neighborhood of the origin.

4.4.2 Trajectory Tracking Control Algorithm

To design the controller, let the control inputs be chosen as

$$u_1 = K_{fxR} \cdot \bar{f}_{hxR} + K_{fxL} \cdot \bar{f}_{hxL} \quad (4.51)$$

$$u_2 = K_{fyR} \cdot \bar{f}_{hyR} + K_{fyL} \cdot \bar{f}_{hyL} \quad (4.52)$$

$$u_3 = K_{mR} \cdot \bar{\tau}_{hR} + K_{mL} \cdot \bar{\tau}_{hL} \quad (4.53)$$

such that the averaged dynamic model (4.28) is expressed in the control-affine form as

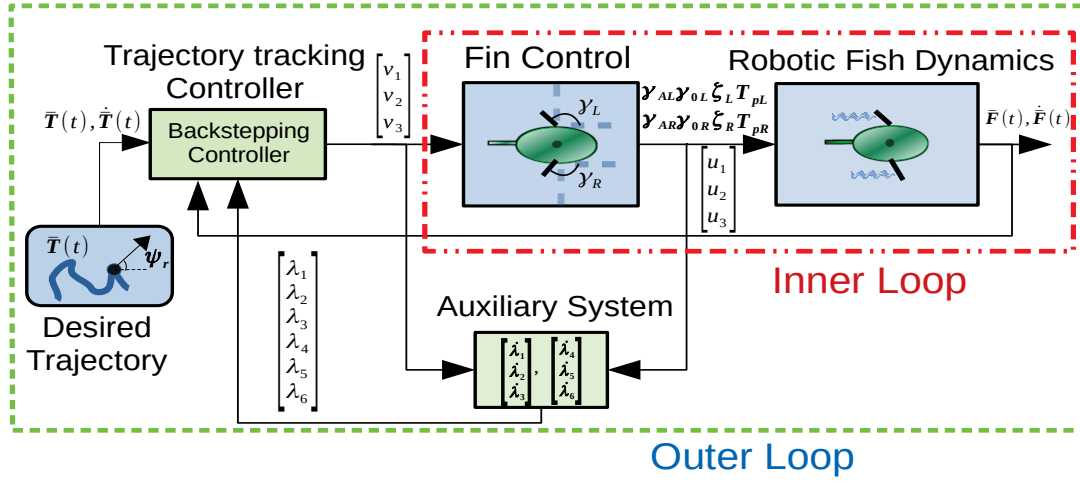


Figure 4.6: Illustration of the proposed dual-loop pectoral fin-actuated robotic fish control scheme. The green dashed line encompasses the outer loop trajectory tracking controller, while the inner red dotted and dashed line encompasses the fin parameter optimization algorithm.

$$\begin{bmatrix} \dot{v}_{cx} \\ \dot{v}_{cy} \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} f_1 + u_1 \\ f_2 + u_2 \\ f_3 + u_3 \end{bmatrix} \quad (4.54)$$

where the inputs appearing linearly.

To achieve trajectory tracking, we use a backstepping controller that determines the inputs u_1, u_2 and u_3 needed such that $(X_e, Y_e, \psi_e) \rightarrow 0$. We then determine the fin-beat parameters, considering their practical constraints, such that the generated inputs are close to the desired values. The difference between the generated and desired input values is captured by the states of an auxiliary system, which is used along with the backstepping controller to guarantee closed-loop stability, as is similarly done in [94, 132]. Figure 4.6 illustrates the control scheme. We elaborate the controller design below.

Let v_1, v_2 and v_3 represent the nominal inputs from the backstepping design, and let u_1, u_2 and u_3 be the inputs that can be practically implemented. To analyze the influence of the mismatch

between the nominal and actual inputs, the following auxiliary system is chosen

$$\begin{cases} \dot{\lambda}_1 = -\zeta_1 \lambda_1 + \lambda_2 \\ \dot{\lambda}_2 = -\zeta_2 \lambda_2 + (u_1 - v_1) + Y_e(u_3 - v_3) \\ \dot{\lambda}_3 = -\zeta_3 \lambda_3 + \lambda_4 \\ \dot{\lambda}_4 = -\zeta_4 \lambda_4 + (u_2 - v_2) - X_e(u_3 - v_3) \\ \dot{\lambda}_5 = -\zeta_5 \lambda_5 + \lambda_6 \\ \dot{\lambda}_6 = -\zeta_6 \lambda_6 + (u_3 - v_3) \end{cases} \quad (4.55)$$

where $\zeta_1, \zeta_2, \zeta_3, \zeta_4, \zeta_5, \zeta_6$ are positive tuning constants. The variables λ_1, λ_3 and λ_5 represent the filtered effect of the non-achievable portion of the inputs as well as the additional tracking error that arises because of the mismatch between the nominal and implementable inputs.

To stabilize the e_χ error while also considering the difference between nominal and achievable inputs, we define the following Lyapunov function

$$\begin{aligned} V_1(\bar{X}_e, \bar{Y}_e, \bar{\psi}_e) &= \frac{1}{2} \bar{X}_e^2 + \frac{1}{2} \bar{Y}_e^2 + \frac{1}{2} \bar{\psi}_e^2 \\ &= \frac{1}{2} (X_e - \lambda_1)^2 + \frac{1}{2} (Y_e - \lambda_3)^2 + \frac{1}{2} (\psi_e - \lambda_5)^2 \end{aligned} \quad (4.56)$$

where \bar{X}_e, \bar{Y}_e and $\bar{\psi}_e$ are the modified tracking errors. The time derivative of Eq. (4.56) is given by

$$\begin{aligned} \dot{V}_1 &= \bar{X}_e(\bar{v}_{cx} - \mathbf{V}_r \cos(\psi_e - \bar{\psi}_r) + \bar{\omega}_z Y_e + \zeta_1 \lambda_1 - \lambda_2) \\ &\quad + \bar{Y}_e(\bar{v}_{cy} + \mathbf{V}_r \sin(\psi_e - \bar{\psi}_r) - \bar{\omega}_z X_e + \zeta_3 \lambda_3 - \lambda_4) \\ &\quad + \bar{\psi}_e(\bar{\omega}_z - \omega_r + \zeta_5 \lambda_5 - \lambda_6) \end{aligned} \quad (4.57)$$

Let the $\alpha_1 = \bar{v}_{cx}$, $\alpha_2 = \bar{v}_{cy}$ and $\alpha_3 = \bar{\omega}_z$ denote the virtual inputs with α_{d1} , α_{d2} and α_{d3} as the corresponding desired virtual inputs. The modified virtual inputs errors are then defined as

$$\bar{Z}_1 = \alpha_1 - \alpha_{d1} - \lambda_2 \quad (4.58)$$

$$\bar{Z}_2 = \alpha_2 - \alpha_{d2} - \lambda_4 \quad (4.59)$$

$$\bar{Z}_3 = \alpha_3 - \alpha_{d3} - \lambda_6 \quad (4.60)$$

Let the desired virtual inputs be given by

$$\alpha_{d1} = V_r \cos(\psi_e - \bar{\psi}_r) - \bar{\omega}_z Y_e - \zeta_1 \lambda_1 - K_{\bar{X}_e} \bar{X}_e \quad (4.61)$$

$$\alpha_{d2} = -V_r \sin(\psi_e - \bar{\psi}_r) + \bar{\omega}_z X_e - \zeta_3 \lambda_3 - K_{\bar{Y}_e} \bar{Y}_e \quad (4.62)$$

$$\alpha_{d3} = \omega_r - \zeta_5 \lambda_5 - K_{\bar{\psi}_e} \bar{\psi}_e \quad (4.63)$$

such that (4.57) becomes

$$\dot{V}_1 = \bar{X}_e(\bar{Z}_1 - K_{\bar{X}_e} \bar{X}_e) + \bar{Y}_e(\bar{Z}_2 - K_{\bar{Y}_e} \bar{Y}_e) + \bar{\psi}_e(\bar{Z}_3 - K_{\bar{\psi}_e} \bar{\psi}_e) \quad (4.64)$$

where $K_{\bar{X}_e}$, $K_{\bar{Y}_e}$ and $K_{\bar{\psi}_e}$ are positive tuning constants.

To account for the virtual input errors, we then define a new Lyapunov function

$$\bar{V}_2 = \bar{V}_1 + \frac{1}{2} \bar{Z}_1^2 + \frac{1}{2} \bar{Z}_2^2 + \frac{1}{2} \bar{Z}_3^2 \quad (4.65)$$

with its time derivative is given by

$$\dot{\bar{V}}_2 = \dot{\bar{V}}_1 + \dot{\bar{Z}}_1 \bar{Z}_1 + \dot{\bar{Z}}_2 \bar{Z}_2 + \dot{\bar{Z}}_3 \bar{Z}_3 \quad (4.66)$$

Eq. (4.66) can be further expanded using Eq. (4.28) along with the input definition Eq. (4.51). After simplification, v_1 , v_2 and v_3 can be obtain from

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & Y_e \\ 0 & 1 & -X_e \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \end{bmatrix} \quad (4.67)$$

where

$$\begin{aligned} \Gamma_1 = & -f_1 - f_3 Y_e + \dot{V}_r \cos(\psi_e - \bar{\psi}_r) - V_r \sin(\psi_e - \bar{\psi}_r)(\dot{\psi}_e - \dot{\bar{\psi}}_r) - \bar{\omega}_z \dot{Y}_e \\ & - K_{\bar{X}_e} \dot{\bar{X}}_e - \zeta_1(-\zeta_1 \lambda_1 + \lambda_2) - \zeta_2 \lambda_2 - K_1 \bar{Z}_1 \end{aligned} \quad (4.68a)$$

$$\begin{aligned} \Gamma_2 = & -f_2 + f_3 X_e - \dot{V}_r \sin(\psi_e - \bar{\psi}_r) + \bar{\omega}_z \dot{X}_e - V_r \cos(\psi_e - \bar{\psi}_r)(\dot{\psi}_e - \dot{\bar{\psi}}_r) \\ & - K_{\bar{Y}_e} \dot{\bar{Y}}_e - \zeta_3(-\zeta_3 \lambda_3 + \lambda_4) - \zeta_4 \lambda_4 - K_2 \bar{Z}_2 \end{aligned} \quad (4.68b)$$

$$\Gamma_3 = -f_3 + \dot{\omega}_r - \zeta_5(-\zeta_5 \lambda_5 + \lambda_6) - \zeta_6 \lambda_6 - K_{\bar{\psi}_e} \dot{\bar{\psi}}_e - K_3 \bar{Z}_3 \quad (4.68c)$$

By adding and subtracting $\frac{1}{4K_{\bar{X}_e}}\bar{Z}_1^2, \frac{1}{4K_{\bar{Y}_e}}\bar{Z}_2^2$ and $\frac{1}{4K_{\bar{\psi}_e}}\bar{Z}_3^2$ to Eq. (4.66), and after completing the square one can arrive at

$$\begin{aligned}\dot{\bar{V}}_2 = & -K_{\bar{X}_e}(\bar{X}_e - \frac{1}{2K_{\bar{X}_e}}\bar{Z}_1)^2 - K_{\bar{Y}_e}(\bar{Y}_e - \frac{1}{2K_{\bar{Y}_e}}\bar{Z}_2)^2 - K_{\bar{\psi}_e}(\bar{\psi}_e - \frac{1}{2K_{\bar{\psi}_e}}\bar{Z}_3)^2 - \bar{Z}_1^2(K_1 - \frac{1}{4K_{\bar{X}_e}}) \\ & - \bar{Z}_2^2(K_2 - \frac{1}{4K_{\bar{Y}_e}}) - \bar{Z}_3^2(K_3 - \frac{1}{4K_{\bar{\psi}_e}})\end{aligned}\quad (4.69)$$

If $K_{\bar{X}_e} > 0$, $K_{\bar{Y}_e} > 0$, $K_{\bar{\psi}_e} > 0$, $K_1 > \frac{1}{4K_{\bar{X}_e}}$, $K_2 > \frac{1}{4K_{\bar{Y}_e}}$ and $K_3 > \frac{1}{4K_{\bar{\psi}_e}}$, then $\dot{\bar{V}}_2 < 0$ unless when $\bar{X}_e = \bar{Y}_e = \bar{\psi}_e = \bar{Z}_1 = \bar{Z}_2 = \bar{Z}_3 = 0$, implying the convergence of $(\bar{X}_e, \bar{Y}_e, \bar{\psi}_e)$ to zero as time approaches infinity. Given that $0 \leq \bar{V}_1(t) \leq \bar{V}_1(0)$, one can conclude that $(\bar{X}_e, \bar{Y}_e, \bar{\psi}_e)$ belongs to \mathcal{L}_2 , which implies that even when the desired force and moment are not implemented, the quantities \bar{X}_e, \bar{Y}_e and $\bar{\psi}_e$ do not diverge. Note that while the convergence for the modified tracking errors \bar{X}_e, \bar{Y}_e and $\bar{\psi}_e$ is guaranteed, that of the actual tracking error \mathbf{e}_χ is not. The latter may actually increase during periods when input limitations are in effect and the desired values cannot be implemented (i.e. $u_1 \neq v_1$, $u_2 \neq v_2$ and/or $u_3 \neq v_3$). However, when the control signal limitations are not in effect, and λ_1, λ_2 and λ_3 approach zero, $(\bar{X}_e, \bar{Y}_e, \bar{\psi}_e)$ converges towards (X_e, Y_e, ψ_e) and the tracking errors can be stabilized.

Let v_1, v_2 and v_3 represent the nominal inputs obtained from the backstepping design. One needs to determine the (feasible) fin-beat parameters such that the resultant inputs u_1, u_2 , and u_3 are close to v_1, v_2 , and v_3 , respectively. Note that to ease the discussion, in this work we fix T_{pL}, ζ_L, T_{pR} and ζ_R , leaving $(\gamma_{0R}, \gamma_{AR})$ and $(\gamma_{0L}, \gamma_{AL})$ as the fin parameters to be found. We use a constrained multi-variable minimization solver, in particular, a controlled, elitist genetic algorithm available in MATLAB (a variant of NSGA-II [141]), to find the best fin parameters. Given v_1, v_2

and v_3 , the constrained multi-variable optimization problem at time t is posed as

$$\begin{aligned} & \underset{\gamma_{0R}, \gamma_{AR}, \gamma_{0L}, \gamma_{AL}}{\operatorname{argmin}} \begin{cases} (K_{f_x R} \cdot \bar{f}_{h_x R} + K_{f_x L} \cdot \bar{f}_{h_x L} - v_1)^2 \\ (K_{f_y R} \cdot \bar{f}_{h_y R} + K_{f_y L} \cdot \bar{f}_{h_y L} - v_2)^2 \\ (K_{m_R} \cdot \bar{\tau}_{h_R} + K_{m_L} \cdot \bar{\tau}_{h_L} - v_3)^2 \end{cases} \\ & \text{subject to: } \gamma_{AR}, \gamma_{AL} \in [\gamma_{A \min}, \gamma_{A \max}] \\ & \gamma_{0R}, \gamma_{0L} \in [\gamma_{0 \min}, \gamma_{0 \max}] \end{aligned} \quad (4.70)$$

where $[\gamma_{A \min}, \gamma_{A \max}]$ and $[\gamma_{0 \min}, \gamma_{0 \max}]$ represent the ranges of the fin-beat amplitude and bias, respectively.

4.5 Trajectory Tracking Simulation Results

Simulation is conducted to evaluate the effectiveness of the designed controller, where the following parameters for the simulation are used:

$$\begin{array}{cccc} K_{\bar{X}e}=0.3 & K_{\bar{Y}e}=0.2 & K_{\bar{\psi}e}=0.3 & K_1=1.03 \\ K_2=1.45 & K_3=1.03 & \zeta_1=0.1 & \zeta_2=0.1 \\ \zeta_3=0.1 & \zeta_4=0.2 & \zeta_5=0.2 & \zeta_6=0.2 \\ \gamma_{A \min} = 0^\circ & \gamma_{A \max} = 50^\circ & \gamma_{0 \min} = 40^\circ & \gamma_{0 \max} = 100^\circ \\ T_p=1 \text{ s} & \zeta = 3 & t_s=1 \text{ s} & \end{array}$$

where $K_{\bar{X}e}, K_{\bar{Y}e}, K_{\bar{\psi}e}, K_1, K_2, K_3, \zeta_1, \zeta_2, \zeta_3, \zeta_4, \zeta_5, \zeta_6$ are tuning parameters for the backstepping controller an auxiliary system. Note that although the controller is designed using the averaged model, inputs from the controller are applied to the original dynamics described by Eq. (4.22).

The following line and circular trajectories are considered

$$\begin{cases} \dot{X}_r = u_r, & \dot{Y}_r = v_r, & \dot{\psi}_r = \omega_r, \text{ with} \\ u_r = 0.03, & v_r = 0, & \omega_r = 0 \text{ and} \\ \dot{X}_r = R_1 \omega_r \cos(\omega_r t), & \dot{Y}_r = -R_1 \omega_r \sin(\omega_r t), \\ \dot{\psi}_r = \omega_r, & \text{with } R_1 = 0.25, & \omega_r = 0.09, \end{cases} \quad (4.71)$$

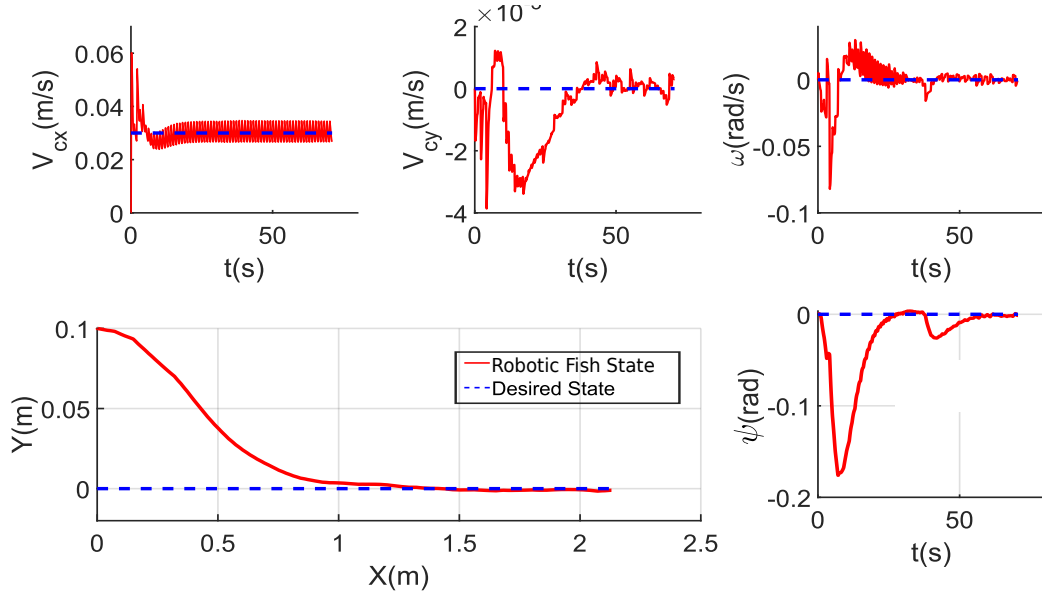


Figure 4.7: Simulation: position, heading angle and velocity tracking for the line-tracking case.

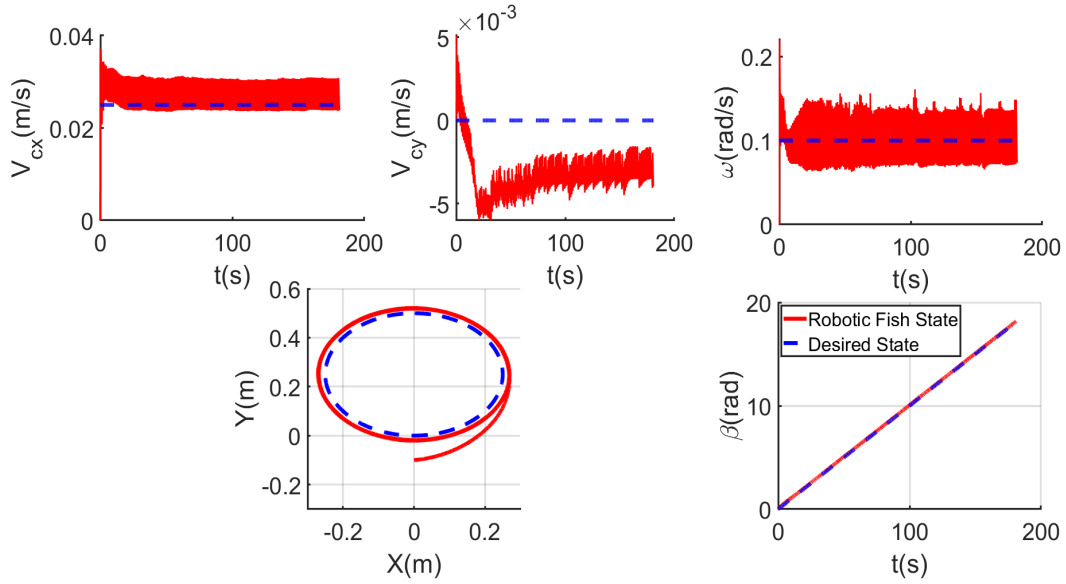


Figure 4.8: Simulation: position, heading angle and velocity tracking for the circle-tracking case.

where \dot{X}_r and \dot{Y}_r represent the velocity of the trajectory in the $\{I\}$ frame. In Figures 4.7-4.8 the desired and the closed-loop trajectories of the robotic fish are compared for both the line and circle-tracking cases. Furthermore, the surge, sway and angular velocities for the desired trajectory and robotic fish are presented. Finally, Figure 4.9 depicts the desired versus the achievable inputs

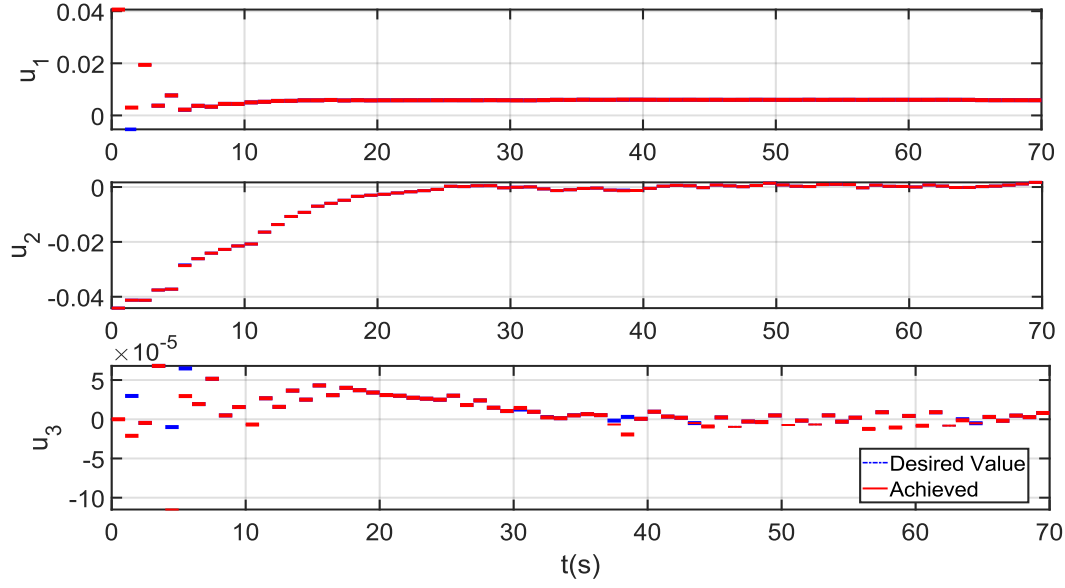


Figure 4.9: Simulation: desired input versus achievable inputs for the line tracking case.

for the line tracking case. From the results, we can see that the robot is able to track the desired position and heading angle trajectories.

4.6 Conclusion

In this work, we presented a nonlinear dynamic average model for a pectoral fin-actuated robotic fish. In particular, we proposed a scaling averaging scheme, where the pectoral fin-generated hydrodynamic forces and moment are first scaled using functions of the fin-beat parameters, and classical averaging is then conducted over the resulting dynamics. Furthermore, we proposed a novel estimation scheme employing a nonlinear model predictive controller and a multivariate nonlinear regression scheme to determine the scaling functions. To evaluate the averaged model, simulation and experimental results comparing the predictions from the original and average models were presented. Furthermore, the utility of the proposed modeling scheme was demonstrated via the design of a backstepping-based trajectory tracking controller.

CHAPTER 5

RAPID MANEUVERING CONTROL OF PECTORAL FIN-ACTUATED ROBOTIC FISH

Although useful in practical applications, cyclic fin movement limits precise manipulation of the fin movements and the thrust profile that can be generated, which impedes full exploitation of the maneuverability of pectoral fin-actuated robotic fish. On the other hand, acyclic thrust or moments could be more instrumental for generating a quick maneuvering response, which can be valuable in scenarios like counteracting disturbances or avoiding fast obstacles.

While rowing motion is beneficial in maneuvering, the actuation constraints (i.e., angular position, velocity, and acceleration limitations) and the mechanism in which the “drag-based” swimming method is used to generate thrust give rise to challenges for maneuvering control of robotic fish. In particular, the range constraint of the fin movement can often inhibit the robot from generating thrust in a direction required for maneuvering. The latter could necessitate the fin moving first in a direction opposite to the desired one (which in turn generates unwanted drag) in order to “back up” and create enough room for accelerating.

While seeming natural for fish or humans, such fin maneuvers are difficult to engineer with existing control design methods. To overcome these challenges and achieve quick maneuvering control, in this work, we propose a dual-loop control approach composed of a backstepping-based controller in the outer loop and a fin movement-planning algorithm in the inner loop. In particular, for the inner loop, we propose a model-predictive planning scheme based on a randomized sampling algorithm that accommodates the fins’ constraints and “intelligently” determines the necessary fins’ movements to produce a desired thrust despite the fins’ current configuration. Simulation results are presented to demonstrate the performance of the proposed scheme via comparison with a nonlinear model predictive controller (NMPC) in rapid velocity maneuvering.

The rest of this chapter is organized as follows. We first review the dynamic model of the pectoral fin-actuated robotic fish in Section 5.1. In Section 5.2, we present the proposed control approach in detail. In Section 5.3, simulation results are discussed. Finally, we provide some

concluding remarks in Section 5.4.

5.1 Dynamic Model of Pectoral Fin-actuated Robotic Fish

The dynamic model of the robotic fish is presented in Section 4.1 of Chapter 4. By letting $\ddot{\gamma}_R = \dot{\omega}_R$, $\ddot{\gamma}_L = \dot{\omega}_L$, $u_1 = \dot{\omega}_R$ and $u_2 = \dot{\omega}_L$, and by considering the kinematic equations of the robotic fish, the dynamic model can be summarized as follows:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\psi} \\ \dot{V}_{cx} \\ \dot{V}_{cy} \\ \dot{\omega}_z \\ \dot{\gamma}_L \\ \dot{\gamma}_R \\ \dot{\omega}_L \\ \dot{\omega}_R \end{bmatrix} = \begin{bmatrix} V_{cx} \cos \psi - V_{cy} \sin \psi \\ V_{cx} \sin \psi + V_{cy} \cos \psi \\ \omega_z \\ f_1(V_{cx}, V_{cy}, \omega_z) + \frac{f_{hx}}{m_1} \\ f_2(V_{cx}, V_{cy}, \omega_z) + \frac{f_{hy}}{m_2} \\ f_3(V_{cx}, V_{cy}, \omega_z) + \frac{\tau_{hz}}{J_3} \\ \omega_L \\ \omega_R \\ u_1 \\ u_2 \end{bmatrix} \quad (5.1)$$

with

$$\begin{cases} f_1(V_{cx}, V_{cy}, \omega_z) = \frac{m_2}{m_1} V_{cy} \omega_z - \frac{c_1}{m_1} V_{cx} \sqrt{V_{cx}^2 + V_{cy}^2} + \\ \quad \frac{c_2}{m_1} V_{cy} \sqrt{V_{cx}^2 + V_{cy}^2} \arctan\left(\frac{V_{cy}}{V_{cx}}\right) \\ f_2(V_{cx}, V_{cy}, \omega_z) = -\frac{m_1}{m_2} V_{cx} \omega_z - \frac{c_1}{m_2} V_{cy} \sqrt{V_{cx}^2 + V_{cy}^2} \\ \quad - \frac{c_2}{m_2} V_{cx} \sqrt{V_{cx}^2 + V_{cy}^2} \arctan\left(\frac{V_{cy}}{V_{cx}}\right) \\ f_3(V_{cx}, V_{cy}, \omega_z) = \frac{(m_1 - m_2)}{J_3} V_{cx} V_{cy} - c_4 \omega_z^2 \text{sgn}(\omega_z) \end{cases} \quad (5.2)$$

where $m_1 = m_b - m_{ax}$, $m_2 = m_b - m_{ay}$, $J_3 = J_{bz} - J_{az}$, $c_1 = \frac{1}{2} \rho S C_D$, $c_2 = \frac{1}{2} \rho S C_L$, $c_4 = \frac{1}{(J_3)} C_M$.

Note that $f_{hx} = f_{hxR} + f_{hxL}$, $f_{hy} = f_{hyR} + f_{hyL}$ and $\tau_{hz} = \tau_{hzR} + \tau_{hzL}$.

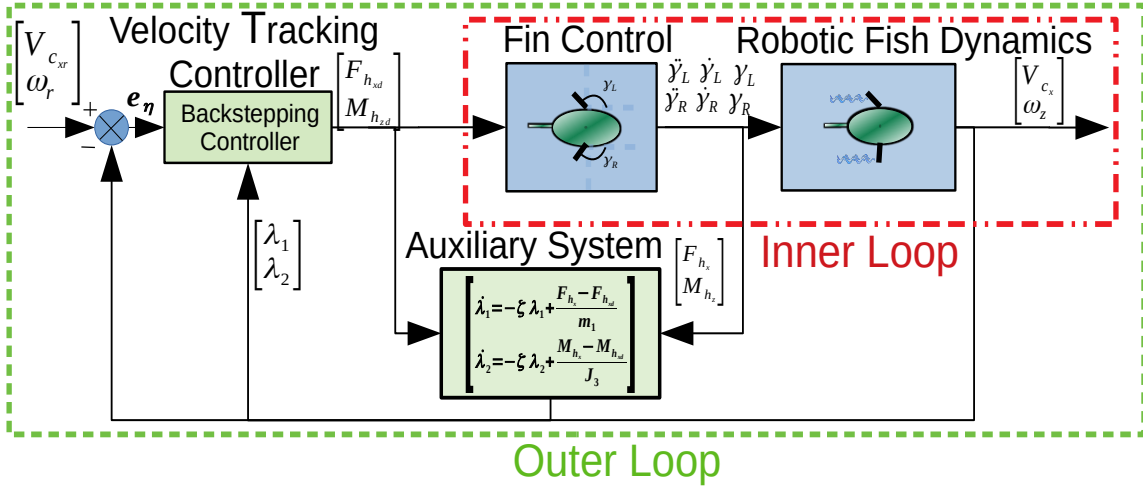


Figure 5.1: Illustration of the proposed dual-loop pectoral fin-actuated robotic fish control scheme. The green dashed line encompasses the outer loop velocity tracking controller, while the inner red dotted and dashed line encompasses the fin movement planning and control algorithm.

5.2 Dual-Loop Fin Control Scheme

Using the validated model proposed in Section 4.1, in this section we propose a dual-loop fin control scheme to achieve velocity tracking for a pectoral fin-actuated robotic fish.

5.2.1 Velocity Tracking Problem

The velocity tracking problem involves controlling the robot to track desired body-fixed velocity trajectories that are parameterized in time t . Given the underactuated nature and input coupling of the robotic fish, tracking three velocities is challenging [94]; since the main focus of this work is in fin control we only consider tracking of the surge and angular velocity and leave the former for future work. Let the velocity tracking error at time t be given by

$$e_\eta = \begin{bmatrix} e_v \\ e_\omega \end{bmatrix} = \begin{bmatrix} V_{cx} - V_{cxr}(t) \\ \omega_z - \omega_r(t) \end{bmatrix} \quad (5.3)$$

For brevity we drop the time-dependence for the remainder of the chapter. By formulating the tracking problem in terms of the error state, the control objective becomes a stabilization problem, where the task is to find suitable control laws for u_1, u_2 such that for an arbitrary initial error, the states (e_v, e_ω) of system (5.3) can be held near the origin $(0,0)$.

5.2.2 Velocity Tracking Control Algorithm

To achieve velocity tracking, a dual loop control structure is proposed. The outer loop is composed of a backstepping controller that determines the thrust and moment needed such that $\mathbf{e}_\eta \rightarrow 0$. In particular, with sampling time t_s , the robotic fish and auxiliary system states are sampled. The robot's velocities are compared to the reference velocities $V_{c_{xr}}$ and ω_r and the tracking error \mathbf{e}_η is then calculated. The error \mathbf{e}_η is used by the outer loop controller (i.e., the backstepping controller) to determine the thrust $F_{h_{xd}}$ and moment $M_{h_{zd}}$ needed to drive \mathbf{e}_η to the origin. On the other hand, the inner loop is composed of a pectoral fin controller that determines the inputs u_1 and u_2 such that the pectoral fins generate a thrust F_{h_x} and moment M_{h_z} that are close to the desired values $F_{h_{xd}}$ and $M_{h_{zd}}$. The robotic fish dynamics are then propagated using u_1 and u_2 , and the difference between the generated and desired thrust is captured by the states of an auxiliary system which is used along with the backstepping controller to guarantee closed-loop stability. At time $t + t_s$ the process repeats. Figure 5.1 illustrates the proposed method. A detailed overview of proposed control approach will be provided in the following sub-sections.

5.2.3 Outer-loop Tracking Control Synthesis

Backstepping control is a practical and systematic approach that provides stability guarantees, which makes it an attractive choice for velocity tracking. In this work, a backstepping-based controller is proposed to determine the thrust and moments necessary to drive the states (e_v, e_ω) of the error system (5.3) to a neighborhood of the origin.

In order to successfully stabilize the error states to the origin and guarantee closed-loop stability, the backstepping controller must accommodate magnitude constraints on the thrust f_{h_x} and moments τ_{h_z} that can be generated by the pectoral fins at a given time. In order to address this limitations, we adopt a similar scheme to that proposed in [94, 132].

Let $F_{h_{xd}}$ and $M_{h_{zd}}$ represent the nominal backstepping control inputs, and let F_{h_x} and M_{h_z} be the forces and moments that can be practically implemented by the fins. Note that the F_{h_x} and M_{h_z} will be determined by the inner loop fin controller. To analyze the influence of the input

difference, the following auxiliary system is chosen

$$\dot{\lambda}_1 = -\zeta_1 \lambda_1 + \frac{F_{hx} - F_{hxd}}{m_1} \quad (5.4a)$$

$$\dot{\lambda}_2 = -\zeta_2 \lambda_2 + \frac{M_{hz} - M_{hzd}}{J_3} \quad (5.4b)$$

The auxiliary system composed of the variables λ_1 and λ_2 defined above represents the filtered effect of the non-achievable portion of the control inputs. In other words, they represent the additional tracking error that arises because of the mismatch between the nominal and implementable forces and moments.

To stabilize the (e_v, e_ω) subsystem while also considering the difference in inputs, the following candidate Lyapunov function is proposed

$$V_1 = \frac{1}{2} \bar{e}_v^2 + \frac{1}{2} \bar{e}_\omega^2 = \frac{1}{2} (e_v - \lambda_1)^2 + \frac{1}{2} (e_\omega - \lambda_2)^2 \quad (5.5)$$

where \bar{e}_v and \bar{e}_ω are the modified tracking errors. The time derivative of Eq. (5.5) is given by

$$\begin{aligned} \dot{V}_1 &= \bar{e}_v \dot{\bar{e}}_v + \bar{e}_\omega \dot{\bar{e}}_\omega \\ &= \bar{e}_v (f_1(V_{cx}, V_{cy}, \omega_z) - \dot{V}_{cxr} + \frac{F_{hxd}}{m_1} + \zeta_1 \lambda_1) + \bar{e}_\omega (f_3(V_{cx}, V_{cy}, \omega_z) - \dot{\omega}_r + \frac{M_{hzd}}{J_3} + \zeta_2 \lambda_2) \end{aligned} \quad (5.6)$$

Let F_{hxd} and M_{hzd} be given by

$$F_{hxd} = m_1 (-f_1(V_{cx}, V_{cy}, \omega_z) + \dot{V}_{cxr} - \zeta_1 \lambda_1 - K_{\bar{e}_v} \bar{e}_v) \quad (5.7a)$$

$$M_{hzd} = J_3 (-f_3(V_{cx}, V_{cy}, \omega_z) + \dot{\omega}_r - \zeta_2 \lambda_2 - K_{\bar{e}_\omega} \bar{e}_\omega) \quad (5.7b)$$

such that (5.6) becomes

$$\dot{V}_1 = -K_{\bar{e}_v} \bar{e}_v^2 - K_{\bar{e}_\omega} \bar{e}_\omega^2 \quad (5.8)$$

If $K_{\bar{e}_v} > 0$ and $K_{\bar{e}_\omega} > 0$, then $\dot{V}_1 < 0$ except when $\bar{e}_v = \bar{e}_\omega = 0$ implying the convergence of $(\bar{e}_v, \bar{e}_\omega)$ to zero as time approaches infinity. Given that $0 \leq \bar{V}_1(t) \leq \bar{V}_1(0)$, one can conclude that $(\bar{e}_v, \bar{e}_\omega)$ belongs to \mathcal{L}_2 , which implies that even when the desired force and moment are not implemented, the quantities \bar{e}_v and \bar{e}_ω do not diverge. While the convergence for the modified tracking errors \bar{e}_v and \bar{e}_ω is guaranteed, that of the actual velocity tracking errors e_v and e_ω is not,

as the latter may actually increase during periods when the force and moment limitations are in effect and the desired values cannot be implemented (i.e. $F_{h_x} \neq F_{h_xd}$ or $M_{h_z} \neq M_{h_zd}$). On the other hand, when the control signal limitations are not in effect (i.e., $F_{h_x} = F_{h_xd}$ and $M_{h_z} = M_{h_zd}$), λ_1 and λ_2 approach zero, and $(\bar{e}_v, \bar{e}_\omega)$ converges towards (e_v, e_ω) and thus the velocity error can be stabilized.

From Eq. (4.21) and Eqs. (5.7a)-(5.7b), the desired left and right pectoral fin forces can be determined as

$$f_{h_{xd}L} = \frac{J_3}{2C_p}(-f_3(V_{cx}, V_{cy}, \omega_z) + \dot{\omega}_r + \zeta_2\lambda_2 - K\bar{e}_\omega\bar{e}_\omega) - \frac{m_1}{2}(-f_1(V_{cx}, V_{cy}, \omega_z) + \dot{V}_{cxr} - \zeta_1\lambda_1 - K\bar{e}_v\bar{e}_v) \quad (5.9)$$

$$f_{h_{xd}R} = \frac{J_3}{2C_p}(-f_3(V_{cx}, V_{cy}, \omega_z) + \dot{\omega}_r + \zeta_2\lambda_2 - K\bar{e}_\omega\bar{e}_\omega) + \frac{m_1}{2}(-f_1(V_{cx}, V_{cy}, \omega_z) + \dot{V}_{cxr} - \zeta_1\lambda_1 - K\bar{e}_v\bar{e}_v) \quad (5.10)$$

5.2.4 Pectoral Fin Control Algorithm

The goal of the pectoral fin control algorithm is to determine the inputs u_1 and u_2 such that pectoral fin movement generates forces that track the desired forces $f_{h_{xd}L}$ and $f_{h_{xd}R}$.

To achieve this goal, we propose a model-predictive planning algorithm to determine an angular acceleration trajectory for each fin that accommodates the feasibility of the fin movement while producing a thrust that is close to the desired value for a given interval of time T_p , where $T_p = t_s$.

The general procedure to determine the angular trajectory is as follows: T_p is discretized into multiple evenly spaced sub-intervals. For the first sub-interval, a fixed number of B different possible (constant) fin acceleration choices are generated by randomly sampling a distribution (the design of which will be discussed later), and their corresponding angular velocities, positions and force trajectories within the said sub-interval are calculated based on a constant-acceleration model. For the next sub-interval, for each choice previously generated, a new set of B choices for the acceleration are generated and once again the fins' angular velocities, positions and force trajectories within the sub-interval are calculated for all choices. The process repeats until the total

number of sub-intervals is reached. In this manner the number of choices per sub-interval increases exponentially with the sub-interval stage. Figure 5.2 depicts an example of the general idea, where different possible acceleration, velocity and position trajectories generated for a planning interval T_P are shown. The assemblage of individual angular accelerations values (and corresponding position and velocities) from each interval is considered a plausible angular trajectory for the period T_P . Each possible angular trajectory is assigned a cost that is dependent on the difference between its corresponding generated force trajectory and the desired value. The trajectory that yields the lowest cost is selected as the solution. To elaborate on the control algorithm in detail, we utilize the right fin as an example; however, the same approach can be trivially extended to the left pectoral fin.

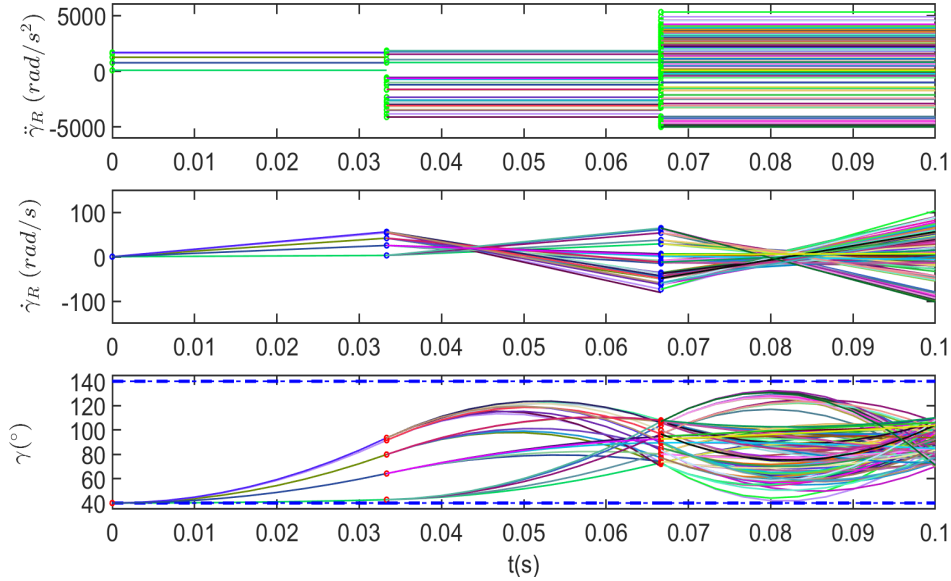


Figure 5.2: Example of different possible fin acceleration, velocity and position trajectories generated within a planning interval T_P that is divided into three sub-intervals.

The outer loop control inputs are updated every t_s seconds and new $f_{h_{xd}L}$ and $f_{h_{xd}R}$ forces are calculated. For the duration of t_s , the desired forces remain constant and become the tracking reference for the fin control algorithm. Note that $T_P = t_s$. Let T_P be discretized into n_0 evenly spaced sub-intervals of length Δt and let $\ddot{\gamma}_{Ri}$ be constant throughout the i th sub-interval, where $i = 1, \dots, n_0$. At a given i th sub-interval there will be B^i different choices for $\ddot{\gamma}_R$, such that for a given period T_P there are a total of B^{n_0} possible choices of (piecewise constant) acceleration

trajectories.

For a given choice of angular accelerations up to the beginning of the i th sub-interval, the following elaborates on the procedure for the related computation:

1. Determine allowed range for $\ddot{\gamma}_R$: Given that the pectoral fins position γ_R is physically limited, the allowed angular velocities $\dot{\gamma}_R$ and angular acceleration $\ddot{\gamma}_R$ are constrained to lie within a range dependent on the current fin's position and velocity. Since $\ddot{\gamma}_R$ is constant throughout each sub-interval, using the standard constant acceleration model the allowed $\ddot{\gamma}_R$ within Δt can be determined as follows

$$\ddot{\gamma}_{R(max)i} = \frac{2(\gamma_{Rmax} - \gamma_{R0}^{(i)} - \dot{\gamma}_{R0}^{(i)}\Delta t)}{(\Delta t)^2} \quad (5.11a)$$

$$\ddot{\gamma}_{R(min)i} = \frac{2(\gamma_{Rmin} - \gamma_{R0}^{(i)} - \dot{\gamma}_{R0}^{(i)}\Delta t)}{(\Delta t)^2} \quad (5.11b)$$

where γ_{Rmax} and γ_{Rmin} denote the maximum and minimum allowed fin position, respectively, while $\gamma_{R0}^{(i)}$ and $\dot{\gamma}_{R0}^{(i)}$ denote the value of γ_R and $\dot{\gamma}_R$ at the beginning of the i th sub-interval, respectively.

2. Calculate desired $\ddot{\gamma}_{Rd}$: Given $f_{hx dR}$, a desired $\ddot{\gamma}_{Rd}$ is calculated from Eq. (4.19) as follows

$$\begin{aligned} \ddot{\gamma}_{Rdi} = & - \frac{\lambda \rho D_p S_p^2 (\dot{\gamma}_{R0}^{(i)})^2 \text{sgn}(\dot{\gamma}_{R0}^{(i)})}{3m_p} \\ & - \frac{(\dot{\gamma}_{R0}^{(i)})^2 \cos \gamma_{R0}^{(i)}}{\sin \gamma_{R0}^{(i)}} + \frac{2f_{hx} R_d}{S_p m_p \sin \gamma_{R0}^{(i)}} \end{aligned} \quad (5.12)$$

3. Generate the sampling distribution and sample B values : The B different $\ddot{\gamma}_R$ values are sampled from a distribution which is generated by dividing the allowable $\ddot{\gamma}_{Ri}$ range into N evenly spaced discrete values. Each possible n th value is assigned a normalized weight W_n determined by the following:

$$W_n = \frac{e^{-d_n}}{\sum_{n=1}^N e^{-d_n}} \quad (5.13)$$

where $n = 1, \dots, N$, and $d_n = \sqrt{(\ddot{\gamma}_{Ri}^{(n)} - \ddot{\gamma}_{Rd})^2}$ is the Euclidean distance between the n th $\ddot{\gamma}_{Ri}$ value and the desired $\ddot{\gamma}_{Rd}$. Note $\sum_{n=1}^N W_n = 1$.

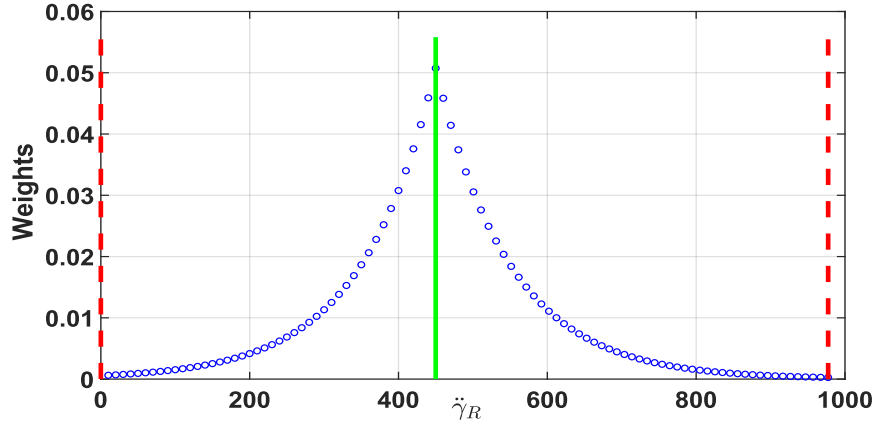


Figure 5.3: Example of weights for each value of $\ddot{\gamma}_{Ri}$ within a given allowed range. The vertical dotted red lines indicate the maximum and minimum values allowed for $\ddot{\gamma}_{Ri}$ in the i th interval, while the green solid line represents the desired value.

To sample from this distribution, generate P “particles”, where each “particle” is representative of each discrete value of $\ddot{\gamma}_{Ri}$. The number of particles generated for a particular discrete value of $\ddot{\gamma}_{Ri}$ is given by

$$P_n = W_n P \quad (5.14)$$

Note that P_n must be rounded to the nearest integer and that $\sum_{n=1}^N P_n = P$. Figure 5.3 illustrates an example of the weights calculated for a given range of $\ddot{\gamma}_{Ri}$. Finally, B “particles” are uniformly sampled from the whole set of P particles, which results in random choices of $\ddot{\gamma}_{Ri}$ with preference towards values closer to $\ddot{\gamma}_{Rd}$.

4. Calculate γ_{Ri} and $\dot{\gamma}_{Ri}$: For a given $\ddot{\gamma}_{Ri}$, the resultant $\dot{\gamma}_R^{(i)}$ and $\gamma_R^{(i)}$ trajectories can be obtained from the constant acceleration model as follows

$$\gamma_R^{(i)}(t_i, t_{i+1}) = \gamma_{R0} + \dot{\gamma}_{R0}^{(i)} t + \frac{1}{2} \ddot{\gamma}_{Ri} t^2 \quad (5.15a)$$

$$\dot{\gamma}_R^{(i)}(t_i, t_{i+1}) = \dot{\gamma}_{R0}^{(i)} + \ddot{\gamma}_{Ri} t \quad (5.15b)$$

where $t \in [t_i, t_{i+1}]$.

5. Determine the force $f_{hxR}^{(i)}$ for a given $\ddot{\gamma}_{Ri}$: From Eq. (4.19), the force generated by the

fin in the i th interval is calculated as

$$f_{h_x R}^{(i)}(t_i, t_{i+1}) = \frac{1}{6} \lambda \rho D_p S_p^3 (\dot{\gamma}_R^{(i)})^2 \sin \gamma_R^{(i)} \operatorname{sgn}(\dot{\gamma}_R^{(i)}) - m_p \left(-\frac{S_p}{2} \dot{\gamma}_R^{(i)} \sin \gamma_R^{(i)} - \frac{S_p}{2} (\dot{\gamma}_R^{(i)})^2 \cos \gamma_R^{(i)} \right) \quad (5.16)$$

6. Calculate cost for each $\ddot{\gamma}_R$: A cost is assigned to each generated acceleration choice within the i th sub-interval. The cost function is given by

$$K^i = \left(\int_{t_i}^{t_{i+1}} f_{h_x R}^{(i)}(\tau) - f_{h_x R d} \right)^2 d\tau \quad (5.17)$$

7. Determine the best $\ddot{\gamma}_R$ trajectory among the B^{n_0} candidates :

$$\min_{\ddot{\gamma}_R(\cdot)} \sum_{p=1}^{n_0} K^i \quad (5.18)$$

Note that in steps 1-5 sets of possible accelerations for each interval are generated, while in the steps 6-7 the best candidate is determined. Furthermore, by sampling from the skewed distribution presented in step 3, we take into consideration the ideal $\ddot{\gamma}_{Rd}$, which allows us to make an educated guess as to what possible $\ddot{\gamma}_R$ should be generated in order to find a good solution. Finally, since the algorithm considers the angular position constraints, a plausible $\ddot{\gamma}_R$ trajectory is always generated.

Once the best $\ddot{\gamma}_R$ and $\ddot{\gamma}_L$ are selected, the total hydrodynamic force F_{h_x} and moment M_{h_z} that will be exerted by the robotic fish can be calculated, and the auxiliary system states can be updated.

5.3 Simulation Results

To evaluate the effectiveness of the designed controller, simulations were carried out using MATLAB. The robotic fish parameters used for simulation are listed in Table 4.1. Furthermore, the backstepping controller and planning algorithm parameters were chosen as follows:

$$\begin{array}{ll} K_{\tilde{e}_v} = 1.5 & K_{\tilde{e}_\omega} = 2 \\ \zeta_1 = 0.1 & \zeta_2 = 0.2 \\ \gamma_{Rmin} = \gamma_{Lmin} = 45^\circ & \gamma_{Rmax} = \gamma_{Lmax} = 120^\circ \\ t_s = 0.2 \text{ s} & T_P = 0.2 \text{ s} \\ n_0 = 3 & B = 5 \\ N = 20 & P = 100 \end{array}$$

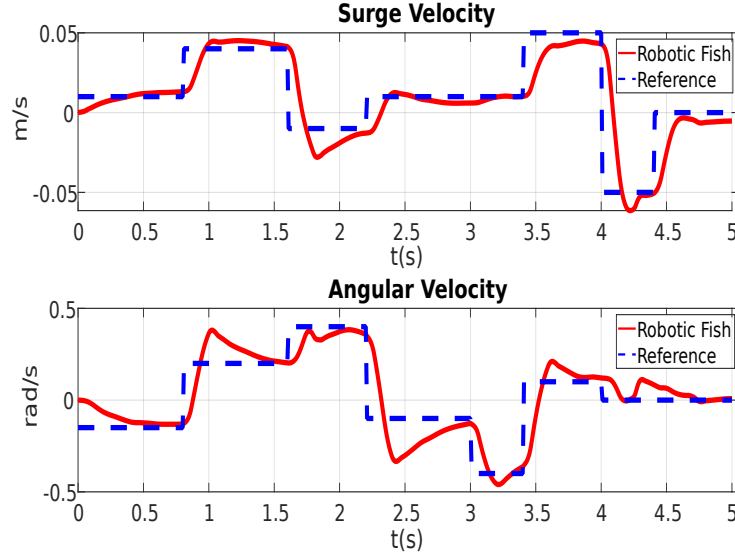
where the variable t_s is the sampling interval which pertains to the amount of time between an update to the desired force and moment values, $F_{h_{xd}}$ and $M_{h_{zd}}$. The backstepping controller parameters were chosen such that under the right values the velocity error system was stabilized to a neighborhood of the origin. We found that $K_{\bar{e}_v}$ and $K_{\bar{e}_\omega}$ regulate the balance between the convergence rate of the e_v and e_ω error, respectively, while varying ζ_1 and ζ_2 regulates the convergence rate of the control-deviation errors λ_1 and λ_2 , respectively. The randomized model-predictive planning scheme parameters, n_0 , B , N and P , were chosen as to balance the trade-off between computational effort and performance.

To demonstrate the effectiveness of the proposed approach, simulations were carried out to compare its performance with an alternative approach, where NMPC is used as the inner loop fin controller. The cost function Eq. (5.17) was adopted to the NMPC algorithm, and the NMPC parameters were chosen as follows:

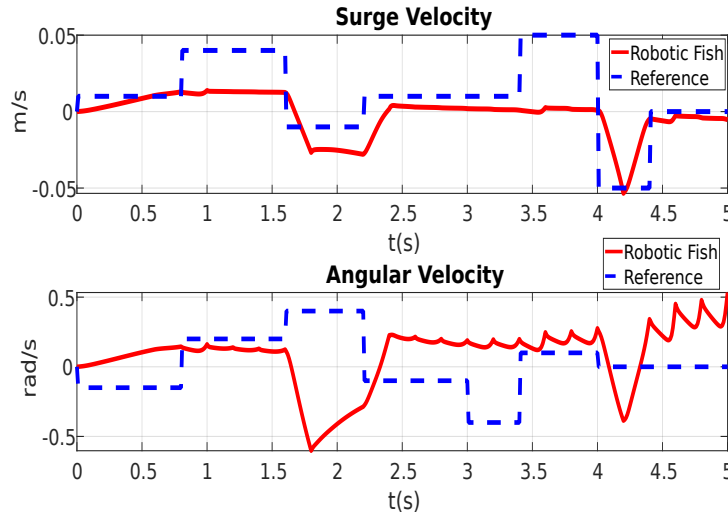
$$\begin{aligned} \text{Prediction Horizon} &= 0.2 \text{ s} & \text{Control Intervals} &= 30 \\ Q &= 5000 & t_s &= 0.2 \text{ s} \end{aligned}$$

where the weighting matrix Q was chosen to heavily penalize the deviation from the desired force and the force being generated, and the prediction horizon was chosen as the length of the sampling time t_s . The following reference velocity trajectory, with abrupt changes, was considered in simulations

$$\left\{ \begin{array}{ll} V_{c_{x_r}} = 0.01 \text{ m/s}, \omega_r = -0.15 \text{ rad/s} & t < 0.8 \\ V_{c_{x_r}} = 0.04 \text{ m/s}, \omega_r = 0.2 \text{ rad/s} & 0.8 \leq t < 1.6 \\ V_{c_{x_r}} = -0.01 \text{ m/s}, \omega_r = 0.4 \text{ rad/s} & 1.6 \leq t < 2.2 \\ V_{c_{x_r}} = 0.01 \text{ m/s}, \omega_r = -0.1 \text{ rad/s} & 2.2 \leq t < 2.4 \\ V_{c_{x_r}} = 0.01 \text{ m/s}, \omega_r = 0 \text{ rad/s} & 2.4 \leq t < 3 \\ V_{c_{x_r}} = 0.01 \text{ m/s}, \omega_r = -0.04 \text{ rad/s} & 3 \leq t < 3.4 \\ V_{c_{x_r}} = 0.04 \text{ m/s}, \omega_r = 0.1 \text{ rad/s} & 3.4 \leq t < 4 \\ V_{c_{x_r}} = -0.05 \text{ m/s}, \omega_r = 0.1 \text{ rad/s} & 4 \leq t < 4.4 \\ V_{c_{x_r}} = 0 \text{ m/s}, \omega_r = 0 \text{ rad/s} & 4.4 \leq t \leq 5 \end{array} \right.$$



(a) Proposed approach.

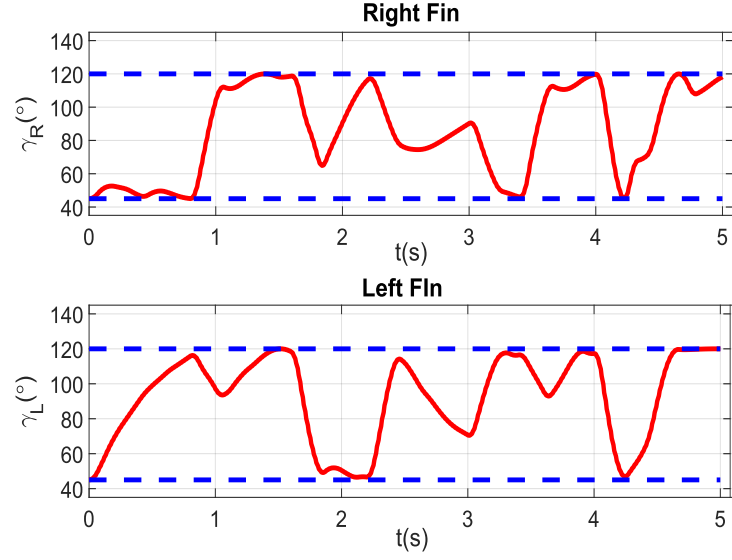


(b) NMPC approach.

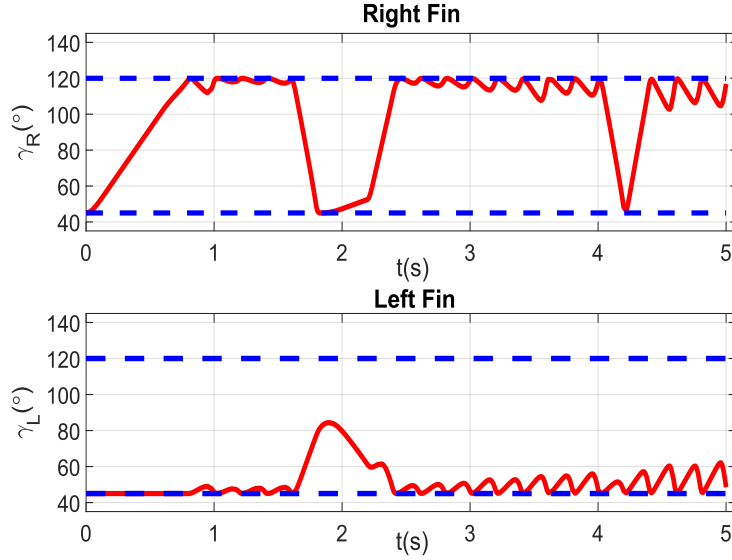
Figure 5.4: Simulation: velocity-tracking trajectories results for the proposed scheme (a) and the alternative scheme with NMPC for the inner loop controller (b).

Note that the above desired velocities include scenarios like backwards swimming ($1.6 \leq t < 2.2$), quick left and right turning, and stopping ($4.4 \leq t \leq 5$).

In Figure 5.4 the desired and the closed-loop velocity trajectories of the robotic fish are depicted for the proposed method (Figure 5.4a) and the alternative NMPC scheme (Figure 5.4b) , while Figure 5.5 illustrates the fin positions over time for each scheme. From Figure 5.4 one can see that



(a) Proposed approach.



(b) NMPC approach.

Figure 5.5: Simulation: fin position trajectories resulting from the proposed scheme (a) and the alternative scheme with NMPC for the inner loop controller (b). The dashed blue lines depict the fin's position constraints.

the proposed approach allows the robot to respond to the sudden changes of the desired velocities. In particular, although there are slight overshoot the robot was able to change its surge and angular velocity velocities to keep up with the quickly varying reference. In contrast, although the NMPC approach is able to handle the fin range constraints, its tracking performance is inferior when

compared to the proposed approach. Finally, from Figure 5.5, one can note that for both approaches the fin's positions constraints are not violated.

5.4 Chapter Summary

In this work, a systematic approach for maneuvering control of a pectoral-fin actuated robotic fish was proposed. Specifically, we proposed a dual loop control scheme consisting of an outer-loop backstepping controller and an inner loop fin movement-planning algorithm. In particular, the outer loop backstepping-controller finds the thrust and moment required to stabilize the velocity tracking error, while the inner loop plans the motion of the fin for a given time-interval to produce a thrust and a moment close to their desired values by utilizing a randomized sampling algorithm. A parameter estimation scheme was employed to empirically identify the hydrodynamic parameters of the model. To illustrate the challenges in control, the velocity tracking problem with abrupt velocity changes for a robotic fish was considered. Simulation results showed the effectiveness of the proposed scheme and its superiority over an alternative employing an NMPC in the inner loop.

CHAPTER 6

SUMMARY AND FUTURE WORK

6.1 Summary

This dissertation presented several systematic model-based control approaches for tail-actuated and pectoral fin-actuated robotic fish that guarantee closed-loop system stability, accommodate input constraints, and are computationally viable for these robots.

We first proposed and implemented in real-time a path-following NMPC scheme for a tail-actuated robotic fish, where a high-fidelity averaged nonlinear dynamic model was used for controller design. A novel parameter estimation scheme was employed to empirically identify the averaged model's hydrodynamic parameters and scaling coefficients. Furthermore, given that the control inputs were functions of two of the tail-beat parameters, specifically the tail bias and tail amplitude, a control projection strategy was implemented to handle these nonlinear input constraints and maximize the use of the admissible control region in a computationally efficient manner. Finally, simulation and experimental results demonstrated the effectiveness of the proposed scheme.

Despite being a promising tracking control approach, NMPC's computational complexity posed challenges in implementation in resource-constrained robots. We thus proposed a backstepping-based control that is practical, systematic, and computationally inexpensive, especially when compared to NMPC. We first considered the trajectory tracking of under-actuated planar robotic systems and demonstrated how the controller guarantees the ultimate boundedness of the position and heading tracking errors to a neighborhood of the origin. In particular, the controller was synthesized to guarantee the convergence of the longitudinal error and a new coordinate error that coupled the lateral error and the heading error. Via time-scale analysis of perturbed systems, it was shown that the convergence of these errors guarantees the stabilization of the whole error system to a neighborhood around the origin. Furthermore, the proposed scheme was implemented on a tail-actuated robotic fish, where a high-fidelity averaged nonlinear dynamic model was used for controller design.

Finally, real-time experimental results demonstrated the effectiveness of the proposed scheme and showed its value over standard PI control.

Moving forward, we then turned our attention to pectoral fin-actuated robotic fish. In practical applications like trajectory tracking, it is more natural to control the parameters of periodic fin beats. To this end, we presented a nonlinear dynamic average model for a pectoral fin-actuated robotic fish that is amenable to controller design, where the control inputs are actuation pattern parameters. In particular, we proposed a scaling averaging scheme, where the pectoral fin-generated hydrodynamic forces and moment are first scaled using functions of the fin-beat parameters, and classical averaging is then conducted over the resulting dynamics. Furthermore, we proposed a novel estimation scheme employing a nonlinear model predictive controller and a multivariate nonlinear regression scheme to determine the scaling functions. To validate the averaged model simulation comparing the predictions from the original and average models were presented. Furthermore, both models were validated with experimental results. Finally, the utility of the proposed modeling scheme was demonstrated via the design of a dual-loop backstepping-based trajectory tracking controller. Simulation results demonstrate the effectiveness of the proposed control approach.

Acyclic actuation could be more instrumental than cyclic actuation in tasks such as rapid maneuvering. To overcome the challenges pectoral fins pose in control, a systematic approach for maneuvering control of a pectoral-fin actuated robotic fish was proposed. Specifically, we proposed a dual-loop control scheme consisting of an outer-loop backstepping controller and an inner-loop fin movement-planning algorithm. The outer-loop backstepping-controller finds the thrust and moment required to stabilize the velocity tracking error, while the inner-loop plans the motion of the fin for a given time interval to produce a thrust and a moment close to their desired values by utilizing a randomized sampling algorithm. To illustrate the challenges in control, the velocity tracking problem with abrupt velocity changes for a robotic fish was considered. Simulation results showed the effectiveness of the proposed scheme and its superiority over an alternative employing an NMPC in the inner loop.

6.2 Future Work

For future work, experimental trials that validate the proposed trajectory tracking control scheme for a pectoral fin-actuated robotic fish will be implemented. Furthermore, the proposed pectoral fin-actuated robotic fish maneuvering scheme will be optimized to accommodate the trade-off between performance and computational efficiency. In addition, experiments will be implemented on a robotic fish prototype to verify the effectiveness of the proposed approach.

The work presented in this dissertation can also be expanded in different directions. First, it would be interesting to compare the trajectory tracking performance of controllers using the original dynamics versus the averaged dynamics when considering trajectories with sharp turns or sudden changes. Furthermore, since the original dynamic model captures transient behavior while the averaged dynamic model captures the steady state behavior, it would be of interest to develop frameworks that allow the coordination of two separate model-based controllers that are designed to implement pectoral fin-based locomotion. Finally, since the tail fin tends to be most beneficial at higher speeds, while pectoral fins tend to be more effective for maneuvering, it would be interesting to develop control algorithms that allow the coordination of the two forms of locomotion.

In addition, it also would be interesting to investigate the use of Linear Parameter Varying (LPV) models for control of robotic fish [142, 143]. For example, one could explore the use of model predictive control or linear quadratic regulator control approaches along with the LPV model to implement trajectory tracking control.

In another direction, it would be interesting to utilize the trajectory tracking control algorithms in an environmental sensing application, where an upper-level path planning scheme is integrated with one of the proposed trajectory tracking schemes for active sensing. We have done some preliminary work along this direction, where we used an ergodic exploration algorithm for trajectory planning along with an NMPC trajectory tracking controller [144].

Finally, while model-based control tends to be effective, it is challenging to obtain high-fidelity and accurate models that are amenable to control synthesis especially for dynamic, uncertain environmental conditions. Furthermore, it can be difficult to estimate parameters, either offline

and online, accurately. Data-driven methods such as Koopman operators are promising approaches that can facilitate learning control-affine models. In our recent work, we have explored the use of Koopman operators for developing linear models and implementing real-time velocity control of robotic fish [145, 146]. Exploring these data-driven techniques can be useful for robotic fish due to the highly nonlinear dynamics and the need for controllers that use limited computation.

APPENDICES

APPENDIX A

PROOF OF THE ROBOTIC FISH'S INPUT-TO-STATE STABILITY

Consider the robotic fish dynamics given by Eq. (3.67a)-(3.67c), where f is locally Lipschitz in x and u , and $u(t)$ is a piece-wise continuous, bounded function of t for all $t \geq 0$. To determine if the system is input-to-state stable we apply the result from [129](Lemma 4.6).

Unforced System

Let $\bar{x} = [\bar{x}_1 \ \bar{x}_2 \ \bar{x}_3]^T = [v_1 \ v_2 \ \omega]$. The unforced system for the robotic fish is then given by

$$\dot{\bar{x}} = f(\bar{x}, 0) \quad (\text{A.1})$$

and in particular

$$\dot{\bar{x}}_1 = \frac{m_2}{m_1} \bar{x}_2 \bar{x}_3 - \frac{c_1}{m_1} x_1 \sqrt{\bar{x}_1^2 + \bar{x}_2^2} + \frac{c_2}{m_1} \bar{x}_2 \sqrt{\bar{x}_1^2 + \bar{x}_2^2} \arctan \frac{\bar{x}_2}{\bar{x}_1} \quad (\text{A.2a})$$

$$\dot{\bar{x}}_2 = -\frac{m_1}{m_2} \bar{x}_1 \bar{x}_3 - \frac{c_1}{m_2} \bar{x}_2 \sqrt{\bar{x}_1^2 + \bar{x}_2^2} - \frac{c_2}{m_2} \bar{x}_1 \sqrt{\bar{x}_1^2 + \bar{x}_2^2} \arctan \frac{\bar{x}_2}{\bar{x}_1} \quad (\text{A.2b})$$

$$\dot{\bar{x}}_3 = \frac{(m_1 - m_2)}{J_3} \bar{x}_1 \bar{x}_2 - c_4 \bar{x}_3^2 \text{sgn}(\bar{x}_3) \quad (\text{A.2c})$$

To prove that this system has a global asymptotically stable equilibrium point at the origin, consider the following Lyapunov function

$$V(x) = \frac{m_1}{2J_3} \bar{x}_1^2 + \frac{m_2}{2J_3} \bar{x}_2^2 + \frac{1}{2} \bar{x}_3^2 \quad (\text{A.3})$$

such that

$$\begin{aligned} \dot{V}(x) &= \frac{m_1}{J_3} x_1 \dot{\bar{x}}_1 + \frac{m_2}{J_3} \bar{x}_2 \dot{\bar{x}}_2 + \bar{x}_3 \dot{\bar{x}}_3 \\ &= \left(\frac{m_2}{J_3} - \frac{m_1}{J_3} \right. \\ &\quad \left. + \frac{(m_1 - m_2)}{J_3} \right) \bar{x}_1 \bar{x}_2 \bar{x}_3 + \left(\frac{c_2}{J_3} - \frac{c_2}{J_3} \right) \bar{x}_1 \bar{x}_2 \sqrt{\bar{x}_1^2 + \bar{x}_2^2} \arctan \frac{\bar{x}_2}{\bar{x}_1} \\ &\quad - \frac{c_1}{J_3} \bar{x}_1^2 \sqrt{\bar{x}_1^2 + \bar{x}_2^2} - \frac{c_1}{J_3} \bar{x}_2^2 \sqrt{\bar{x}_1^2 + \bar{x}_2^2} - c_4 \bar{x}_3^3 \text{sgn}(\bar{x}_3) \\ &= -\frac{c_1}{J_3} \bar{x}_1^2 \sqrt{\bar{x}_1^2 + \bar{x}_2^2} - \frac{c_1}{J_3} \bar{x}_2^2 \sqrt{\bar{x}_1^2 + \bar{x}_2^2} - c_4 \bar{x}_3^3 \text{sgn}(\bar{x}_3) \end{aligned} \quad (\text{A.4})$$

$V(x)$ is thus negative definite globally and it can be concluded that the origin is globally asymptotically stable (G.A.S).

State Boundedness

Let $\dot{\bar{x}} = f(\bar{x}, u)$ be a perturbation of the unforced system $\dot{x} = f(\bar{x}, 0)$. The function $f(\bar{x}, u)$ satisfies the following

$$\begin{aligned} \|f(\bar{x}, u) - f(\bar{x}, 0)\| &\leq L\|u\| \\ \left\| \begin{array}{c} K_f \frac{mL^2}{12m_1} \omega_\alpha^2 u_1 \\ K_f \frac{mL^2}{4m_2} \omega_\alpha^2 u_2 \\ -K_m \frac{cmL^2}{4J_3} \omega_\alpha^2 u_2 \end{array} \right\| &\leq L\|u\| \end{aligned} \quad (\text{A.5})$$

where L is a Lipschitz constant. From Definition 4.7 and Theorem 4.19 in [129], the ultimate bound of the states can be determined. Considering the derivative of Eq.(A.3) along $f(\bar{x}, u)$

$$\begin{aligned}
\frac{\partial V}{\partial \bar{x}} f(\bar{x}, u) &= \frac{m_1}{J_3} \bar{x}_1 \dot{\bar{x}}_1 + \frac{m_2}{J_3} \bar{x}_2 \dot{\bar{x}}_2 + \bar{x}_3 \dot{\bar{x}}_3 \\
&= -\frac{c_1}{J_3} \bar{x}_1^2 \sqrt{\bar{x}_1^2 + \bar{x}_2^2} - \frac{c_1}{J_3} \bar{x}_2^2 \sqrt{\bar{x}_1^2 + \bar{x}_2^2} \\
&\quad - c_4 \bar{x}_3^3 \text{sgn}(\bar{x}_3) + \bar{x}_1 \frac{m_1}{J_3} K_f \frac{mL^2}{12m_1} \omega_\alpha^2 u_1 \\
&\quad + \bar{x}_2 \frac{m_2}{J_3} K_f \frac{mL^2}{4m_2} \omega_\alpha^2 u_2 - x_3 K_m \frac{cmL^2}{4J_3} \omega_\alpha^2 u_2 \\
&\leq -\frac{c_1}{J_3} \bar{x}_1^2 \sqrt{\bar{x}_1^2 + \bar{x}_2^2} - \frac{c_1}{J_3} \bar{x}_2^2 \sqrt{\bar{x}_1^2 + \bar{x}_2^2} - c_4 \bar{x}_3^3 \text{sgn}(\bar{x}_3) \\
&\quad + \frac{K_f mL^2}{12J_3} \omega_\alpha^2 |\bar{x}_1| |u_1| + \frac{K_f mL^2}{4J_3} \omega_\alpha^2 |\bar{x}_2| |u_2| \\
&\quad + K_m \frac{cmL^2}{4J_3} \omega_\alpha^2 |\bar{x}_3| |u_2| \\
&\leq - (1 - \theta_1) \frac{c_1}{J_3} \bar{x}_1^2 - (1 - \theta_2) \frac{c_1}{J_3} \bar{x}_2^2 \\
&\quad - (1 - \theta_3) c_4 \bar{x}_3^3 \text{sgn}(\bar{x}_3) - \theta_1 \frac{c_1}{J_3} \bar{x}_1^2 \\
&\quad - \theta_2 \frac{c_1}{J_3} \bar{x}_2^2 - \theta_3 c_4 \bar{x}_3^3 \text{sgn}(\bar{x}_3) \\
&\quad + \frac{K_f mL^2}{12J_3} \omega_\alpha^2 |\bar{x}_1| |u_1| + \frac{K_f mL^2}{4J_3} \omega_\alpha^2 |\bar{x}_2| |u_2| \\
&\quad + K_m \frac{cmL^2}{4J_3} \omega_\alpha^2 |\bar{x}_3| |u_2| \\
&\leq - (1 - \theta_1) \frac{c_1}{J_3} \bar{x}_1^2 - (1 - \theta_2) \frac{c_1}{J_3} \bar{x}_2^2 \\
&\quad - (1 - \theta_3) c_4 \bar{x}_3^3 \text{sgn}(\bar{x}_3) - \theta_1 \frac{c_1}{J_3} \bar{x}_1^2 - \theta_2 \frac{c_1}{J_3} \bar{x}_2^2 \\
&\quad - \theta_3 c_4 \bar{x}_3^3 \text{sgn}(\bar{x}_3) \quad \forall \quad |\bar{x}_1| \geq \mu_1, \quad |\bar{x}_2| \geq \mu_2, \quad |\bar{x}_3| \geq \mu_3
\end{aligned} \tag{A.6}$$

where when $\mu_1 = \frac{K_f mL^2}{12\theta_1 c_1} \omega_\alpha^2 |u_1|$, $\mu_2 = \frac{K_f mL^2}{4\theta_2 c_1} \omega_\alpha^2 |u_2|$ and $\mu_3 = \frac{K_m cmL^2}{4K_D \theta_3} \omega_\alpha^2 |u_2|$.

From Definition 4.4 of [129], for any initial state $x(t_0)$ and a bounded input $u(t)$, the solution $x(t)$ exists for all $t \geq t_0$ and satisfies

$$\|x\| \leq \max \left\{ \beta(\|x(t_0)\|, t - t_0), \alpha_1^{-1}(\alpha_2(\mu)) \right\} \quad \forall \quad t \geq t_0 \tag{A.7}$$

where $\alpha_1(r) = \lambda_{\min}(P)r^2$, $\alpha_2(r) = \lambda_{\max}(P)r^2$. In particular, for \bar{x}_1 , \bar{x}_2 , and \bar{x}_3 , respectively, the ultimate bounds $b_1 - b_3$ are given by

$$b_1 = \frac{K_f m L^2}{12\theta_1 c_1} \omega_\alpha^2 |u_1| \quad (\text{A.8a})$$

$$b_2 = \frac{K_f m L^2}{4\theta_2 c_1} \omega_\alpha^2 |u_2| \quad (\text{A.8b})$$

$$b_3 = \frac{K_m c m L^2}{4K_D \theta_3} \omega_\alpha^2 |u_2| \quad (\text{A.8c})$$

APPENDIX B

PROOF OF STABILITY OF THE PERTURBATION-FREE ROBOTIC FISH CLOSED-LOOP SYSTEM

The asymptotic stability of the origin for the robotic fish system can be demonstrated by following the procedure from Section 3.3. Using the following definitions

$$v_1 = \chi_3 + \phi_1 - y_e(\omega - \phi_3) \quad (\text{B.1})$$

$$\omega = \chi_4 + \varrho\phi_3 \quad (\text{B.2})$$

we can write the closed-loop system as

$$\begin{bmatrix} \dot{y}_e \\ \dot{\chi}_1 \\ \dot{\chi}_2 \\ \dot{\chi}_3 \\ \dot{\chi}_4 \\ \dot{\bar{\eta}} \end{bmatrix} = \begin{bmatrix} v_2 + v_r \sin(\psi_e) - \frac{\chi_2 \chi_4}{k(t, e)} - \chi_2 \phi_3 \\ -K_{z_\xi} \chi_1 + \chi_4 \\ -K_{x_e} \chi_2 + \chi_3 \\ -K_{z_1} \chi_3 \\ -K_{z_2} \chi_4 \\ h_2(v_1, \bar{\eta}, \omega) + b_4 \bar{f}_5(\alpha_0, \alpha_a, \omega_\alpha) \end{bmatrix} \quad (\text{B.3})$$

where

$$\psi_e = \chi_1 - k_{\delta_1} \arctan(k_{\delta_2} y_e) \quad (\text{B.4})$$

$$k(t, e) = 1 - \frac{k_{z\delta_1} k_{z\delta_2} \chi_2}{(k_{\delta_2} y_e)^2 + 1} \quad (\text{B.5})$$

The reduced Υ_S -subsystem for the robotic fish is defined as

$$\begin{aligned} \dot{y}_e &= f(t, y_e, G(t, y_e), 0) \\ &= -v_r \sin(k_{\delta_1} \arctan(k_{\delta_2} y_e)) \end{aligned}$$

Its corresponding boundary layer is given by

$$\frac{d\chi}{d\tau_1} = g(t, y_e, G(t, \chi(\tau_1)), 0) = \begin{bmatrix} -\bar{a}\chi_1 \\ -\bar{b}\chi_2 \\ -\bar{c}\chi_3 \\ -\bar{d}\chi_4 \end{bmatrix}$$

Assumption 4 of Lemma 1:

$$\begin{aligned}\dot{V}_S &= \frac{\partial V_S}{\partial y_e} f(t, y_e, G(t, y_e)) + \frac{\partial V_S}{\partial t} \\ &= -\frac{v_r k_{\delta_1} k_{\delta_2} \sin(k_{\delta_1} \arctan(k_{\delta_2} y_e))}{(k_{\delta_2} y_e)^2 + 1}\end{aligned}\quad (\text{B.6})$$

Let y_e lie within compact set such that $\bar{K}_{y_e} = \max_{y_e \in \Omega} |y_e|$. One can then find c_0 such that in an interval of y_e

$$\frac{\sin(k_{\delta_1} \arctan(k_{\delta_2} y_e))^2}{(k_{\delta_2} y_e)^2 + 1} \geq c_0 y_e^2 \quad (\text{B.7})$$

such that

$$\dot{V}_S \leq -v_r^{\min} k_{\delta_1} k_{\delta_2} c_0 y_e^2 \quad (\text{B.8})$$

Since $v_r^{\min}, k_{\delta_1}, k_{\delta_2} > 0$, Assumption 4 of Lemma 1 can be satisfied with $\alpha_1 \geq v_r^{\min} k_{\delta_1} k_{\delta_2} c_0$ and $\Psi_1(y_e) = |y_e|$

Assumption 5 of Lemma 1:

$$\frac{\partial V_F}{\partial x} g(t, y_e, \chi, 0) = -\chi^T S \chi \leq -\bar{\lambda}_{\min}(S) \|\chi\|^2 \quad (\text{B.9})$$

where $\bar{\lambda}_{\min}(S)$ is the smallest eigenvalue of S . Let $\alpha_2 \geq \bar{\lambda}_{\min}(S)$ and $\Phi_1 = \|\chi\|$

Assumption 6 of Lemma 1:

Let $q = k_{\delta_1} \arctan(k_{\delta_2} y_e)$, and

$$\tau = k_{\delta_1} k_{\delta_2} \frac{(v_r \sin(\psi_e) + v_2)}{(k_{\delta_2} y_e)^2 + 1} \quad (\text{B.10})$$

$$\begin{aligned}
& \frac{\partial V_S}{\partial y_e} [f(t, y_e, \chi) - f(t, y_e, G(t, y_e))] \\
&= \frac{k_{\delta_1} k_{\delta_2} \sin(q)}{(k_{\delta_2} y_e)^2 + 1} \left[v_r \sin(\psi_e) - \chi_2 \left(\frac{\chi_4 + \omega_r - \tau - K_{z\xi} \chi_1}{k(t, \mathbf{e})} \right) + v_r \sin(q) \right] \\
&= \frac{k_{\delta_1} k_{\delta_2} \sin(q)}{(k_{\delta_2} y_e)^2 + 1} \left[v_r \left(\sin(\psi_e) + \sin(q) \right) - \chi_2 \left(\frac{\chi_4 ((k_{\delta_2} y_e)^2 + 1)}{(k_{\delta_2} y_e)^2 + 1 - k_{z\delta_1} k_{z\delta_2} \chi_2} \right. \right. \\
&\quad - \frac{\omega_r ((k_{\delta_2} y_e)^2 + 1)}{(k_{\delta_2} y_e)^2 + 1 - k_{z\delta_1} k_{z\delta_2} \chi_2} - \frac{k_{z\delta_1} k_{z\delta_2} v_r \sin(\psi_e)}{(k_{\delta_2} y_e)^2 + 1 - k_{z\delta_1} k_{z\delta_2} \chi_2} - \\
&\quad \left. \left. \frac{K_{z\xi} \chi_1 ((k_{\delta_2} y_e)^2 + 1)}{(k_{\delta_2} y_e)^2 + 1 - k_{z\delta_1} k_{z\delta_2} \chi_2} \right) \right] \\
&= k_{\delta_1} k_{\delta_2} \sin(q) \left[\frac{v_r \left(\sin(\chi_1) \cos(q) - \cos(\chi_1) \sin(q) + \sin(q) \right)}{(k_{\delta_2} y_e)^2 + 1} \right. \\
&\quad - \frac{\chi_2 \omega_r}{(k_{\delta_2} y_e)^2 + 1 - k_{z\delta_1} k_{z\delta_2} \chi_2} \\
&\quad \left. - \frac{\chi_2}{(k_{\delta_2} y_e)^2 + 1 - k_{z\delta_1} k_{z\delta_2} \chi_2} \left(\chi_4 - K_{z\xi} \chi_1 - \frac{k_{z\delta_1} k_{z\delta_2} v_r \sin(\psi_e)}{(k_{\delta_2} y_e)^2 + 1} \right) \right] \\
&= \frac{v_r k_{\delta_1} k_{\delta_2} \sin(q) \left(\sin(\chi_1) \cos(q) + (1 - \cos(\chi_1)) \sin(q) \right)}{(k_{\delta_2} y_e)^2 + 1} \\
&\quad - \frac{\chi_2 k_{\delta_1} k_{\delta_2} \sin(q) \omega_r}{(k_{\delta_2} y_e)^2 + 1 - k_{z\delta_1} k_{z\delta_2} \chi_2} - \frac{k_{\delta_1} k_{\delta_2} \sin(q) \chi_2}{(k_{\delta_2} y_e)^2 + 1 - k_{z\delta_1} k_{z\delta_2} \chi_2} \left(\chi_4 - K_{z\xi} \chi_1 \right. \\
&\quad \left. - \frac{(k_{z\delta_1} k_{z\delta_2}) v_r \sin(\psi_e)}{(k_{\delta_2} y_e)^2 + 1} \right)
\end{aligned} \tag{B.11}$$

Since $\bar{K}_{y_e} = \max_{y_e \in \Omega} |y_e|$, one can always find an interval for all y_e within the compact set where the term $\frac{|\sin(k_{\delta_1} \arctan(k_{\delta_2} y_e))|}{(k_{\delta_2} y_e)^2 + 1}$ is increasing and is bounded such that

$$c_0 |y_e| \leq \frac{|\sin(k_{\delta_1} \arctan(k_{\delta_2} y_e))|}{(k_{\delta_2} y_e)^2 + 1} \tag{B.12}$$

Furthermore, one can find c_1 such that

$$\frac{|\sin(k_{\delta_1} \arctan(k_{\delta_2} y_e))|}{(k_{\delta_2} y_e)^2 + 1} \leq c_1 |y_e| \tag{B.13}$$

so that $\frac{|\sin(q)|}{(k_{\delta_2}y_e)^2 + 1} \leq c_1|y_e|$, and $|\sin(q)| \leq k_1|y_e|$. Similarly, let $\bar{K}_{\chi_1} = \max_{\chi \in \Omega} |\chi_1|$ such that

$$d_0|\chi_1| \leq (1 - \cos(\chi_1)) \leq d_1|\chi_1| \quad (\text{B.14})$$

$$b_0|\chi_1| \leq \sin(\chi_1) \leq b_1|\chi_1| \quad (\text{B.15})$$

Furthermore, given $|\sin(\psi_e)| \leq 1$, $|\cos(\chi)| \leq 1$, and $|\cos(q)| \leq 1$, (B.11) can be rewritten as

$$\begin{aligned} \frac{\partial V_S}{\partial y_e} [f(t, y_e, \chi) - f(t, y_e, G(t, y_e))] &\leq \frac{v_r k_{\delta_1} k_{\delta_2} \sin(q) (b_1|\chi_1| \cos(q) + d_1|\chi_1| \sin(q))}{(k_{\delta_2}y_e)^2 + 1} + \\ &\quad \frac{|\chi_2| k_{\delta_1} k_{\delta_2} |\sin(q)| |\omega_r|}{(k_{\delta_2}y_e)^2 + 1 - k_{z\delta_1} k_{z\delta_2} \chi_2} \\ &\quad + \frac{k_{\delta_1} k_{\delta_2} |\sin(q)| |\chi_2|}{(k_{\delta_2}y_e)^2 + 1 - k_{z\delta_1} k_{z\delta_2} \chi_2} (|\chi_4| + K_{z\xi} |\chi_1| + \\ &\quad \frac{k_{z\delta_1} k_{z\delta_2} v_r |\sin(\psi_e)|}{(k_{\delta_2}y_e)^2 + 1}) \\ &\leq v_r^{\max} (b_1 + d_1) c_1 k_{\delta_1} k_{\delta_2} |y_e| |\chi_1| \\ &\quad + \frac{k_1 k_{\delta_1} k_{\delta_2} |\omega_r| |\chi_2| |y_e|}{(k_{\delta_2}y_e)^2 + 1 - k_{z\delta_1} k_{z\delta_2} \chi_2} \\ &\quad + \frac{k_1 k_{\delta_1} k_{\delta_2} |y_e| |\chi_2|}{(k_{\delta_2}y_e)^2 + 1 - k_{z\delta_1} k_{z\delta_2} \chi_2} (|\chi_4| + K_{z\xi} |\chi_1|) \\ &\quad + \frac{v_r^{\max} k_1 (k_{\delta_1} k_{\delta_2})^2 |\chi_2| |y_e|}{(k_{\delta_2}y_e)^2 + 1 - k_{z\delta_1} k_{z\delta_2} \chi_2} \end{aligned} \quad (\text{B.16})$$

Assume that χ_2 is restricted to a compact set such that $\bar{K}_{\chi_2} = \max_{\chi \in \Omega} |\chi_2|$, and let

$$\frac{k_{z\delta_1} k_{z\delta_2} |\chi_2|}{(k_{z\delta_2}y_e)^2 + 1 - k_{z\delta_1} k_{z\delta_2} \chi_2} < \frac{|\chi_2|}{D_e} \quad (\text{B.17})$$

for $|\chi_2| \leq \frac{1}{k_{z\delta_1} k_{z\delta_2}}$ such that (B.16) becomes

$$\begin{aligned}
&\leq 2c_1 v_r^{\max}(b_1 + d_1)k_{\delta_1}k_{\delta_2}|y_e||\chi_1| + \frac{c_1 k_{\delta_1} k_{\delta_2} |\omega_r| |\chi_2| |y_e|}{D_e} \\
&+ \frac{c_1 k_{\delta_1} k_{\delta_2} |y_e| K_{\chi_2}}{D_e} (|\chi_4| + K_{z_\xi} |\chi_1|) + \frac{v_r c_1 (k_{\delta_1} k_{\delta_2})^2 |\chi_2| |y_e|}{D_e} \\
&\leq K_\delta |y_e| \|\chi\|
\end{aligned} \tag{B.18}$$

where

$$\begin{aligned}
K_\delta = & v_r^{\max}(b_1 + d_1)c_1 k_{\delta_1} k_{\delta_2} \\
& + \frac{k_1 k_{\delta_1} k_{\delta_2}}{D_e} \left(|\omega_r| + K_{\chi_2} + K_{\chi_2} K_{z_\xi} + v_r^{\max}(k_{\delta_1} k_{\delta_2}) \right)
\end{aligned} \tag{B.19}$$

Let $\beta_1 \geq K_\delta$ and $\lambda_1 = 0$ to satisfy Assumption 6 of Lemma 1.

Assumption 7.a of Lemma 1

$$\frac{\partial V_F}{\partial y_e} [g(t, y_e, G(t, y_e, \cdot), \epsilon) - g(t, y_e, G(t, y_e), 0)] = 0 \tag{B.20}$$

Let $\lambda_2 = \beta_2 = 0$ to satisfy the second interconnection condition from Assumption 7 of Lemma 1.

Assumption 7.b of Lemma 1

$$\frac{\partial V_F}{\partial y_e} f(t, y_e, G(t, y_e)) = 0 \tag{B.21}$$

Let $\lambda_3 = \beta_3 = 0$ to satisfy third interconnection condition from Assumption 7 of Lemma 1.

The composite Lyapunov function for the Υ_{SF} -subsystem is given by

$$\begin{aligned}
V_1(y_e, x, t) &= (1 - d_1)V_S(y_e) + d_1V_F(x), \quad d_1 \in (0, 1) \\
&= (1 - d_1)(1 - \cos(k_{\delta_1} \arctan(k_{\delta_2} y_e))) + d_1(\chi^T P \chi)
\end{aligned} \tag{B.22}$$

and its derivative is given by,

$$\begin{aligned}
\dot{V}_1 &\leq - \begin{bmatrix} \sqrt{|y_e|^2} \\ \|\chi\| \end{bmatrix}^T \Lambda \begin{bmatrix} \sqrt{|y_e|^2} \\ \|\chi\| \end{bmatrix} \\
&\leq -\lambda_{\min}(\Lambda) \left(\sqrt{y_e^2 + \|\chi\|^2} \right)^2
\end{aligned} \tag{B.23}$$

APPENDIX C

PROOF OF STABILITY OF THE ROBOTIC FISH'S PERTURBED CLOSED-LOOP SYSTEM

Let the perturbed Σ_{SF} system be expressed as

$$\dot{\Sigma} = Q(t, \Sigma) + r(t, \Sigma) = \begin{bmatrix} f(t, \Sigma) \\ g(t, \Sigma) \\ 0 \end{bmatrix} + r(t, \Sigma) \quad (C.1)$$

where

$$\Sigma = [y_e, \chi, \bar{\eta}]^T \quad (C.2)$$

$$f(t, \Sigma) = \begin{bmatrix} v_r \sin(\psi_e) - (\frac{\chi_4 + \bar{\phi}_3}{\varrho}) \chi_2 \end{bmatrix} \quad (C.3)$$

$$g(t, \Sigma) = \begin{bmatrix} -\bar{a}\chi_1 + \epsilon_1\chi_4 \\ -\bar{b}\chi_2 + \epsilon_1\chi_3 \\ -\bar{c}\chi_3 \\ -\bar{d}\chi_4 \end{bmatrix} \quad (C.4)$$

$$r(t, \Sigma) = \begin{bmatrix} \frac{-k_{z\delta_1} k_{z\delta_2} \bar{\eta} \chi_2}{(k_{z\delta_2} C y_e)^2 + 1 - k_{z\delta_1} k_{z\delta_2} \chi_2} + \bar{\eta} \\ 0 \\ 0 \\ 0 \\ 0 \\ h_2(v_1, \bar{\eta}, \omega) + b_4 u_2 \end{bmatrix} \quad (C.5)$$

$$\bar{\phi}_3 = \omega_r - k_{z\delta_1} k_{z\delta_2} \frac{(v_r \sin(\psi_e))}{(k_{z\delta_2} y_e)^2 + 1} - K_{Z\xi} \chi_1 \quad (C.6)$$

From Eq. (B.22)

$$\frac{\partial V_1}{\partial \Sigma} = \left[(1 - d_1) \frac{v_r k_{\delta_1} k_{\delta_2} \sin(k_{\delta_1} \arctan(k_{\delta_2} y_e))}{(k_{\delta_2} y_e)^2 + 1} \quad d_1 2\chi^T P \right] \quad (C.7)$$

Since it is assumed that y_e lies within a compact set such that $\bar{K}_{y_e} = \max_{y_e \in \Omega} |y_e|$, one can bound the $\sin(k_{\delta_1} \arctan(k_{\delta_2} y_e))$ term in (C.7) by (B.13), such that

$$\left\| \frac{\partial V_1}{\partial \Sigma} \right\| \leq \left[c_1(1-d_1)v_r^{\max} k_{\delta_1} k_{\delta_2} |y_e| \quad d_1 2 \|P\| \|\chi\| \right] \quad (\text{C.8a})$$

$$\leq \left[c_1(1-d_1)v_r^{\max} k_{\delta_1} k_{\delta_2} |y_e| \quad d_1 2 \bar{\lambda}_{\max}(P) \|\chi\| \right] \quad (\text{C.8b})$$

$$\leq N \|\Sigma\| \quad (\text{C.8c})$$

where we write (C.8b) as $y^T w$ and use the inequality $|y^T w| \leq \|w\| \|y\|$, $\bar{\lambda}_{\max}(P)$ is the maximum eigenvalue of P and $N = \sqrt{(c_1(1-d_1)v_r^{\max} k_{\delta_1} k_{\delta_2})^2 + (2d_1 \bar{\lambda}_{\max}(P))^2}$. Furthermore, recall

$$\frac{\partial V_1}{\partial \Sigma} Q(t, \Sigma) \leq -\lambda_{\min}(\Lambda) \left(\sqrt{(y_e^2 + \|\chi\|^2)} \right)^2 \quad (\text{C.9})$$

Since y_e lies within a compact set such that $\bar{K}_{y_e} = \max_{y_e \in \Omega} |y_e|$, one could find q_0 and q_1 such that

$$\begin{aligned} V_1(\Sigma) &= (1-d_1)(1 - \cos(k_{\delta_1} \arctan(k_{\delta_2} y_e))) + d_1 \chi^T P \chi \\ &\leq (1-d_1) q_1 y_e^2 + d_1 \bar{\lambda}_{\max}(P) \chi^2 \\ &\leq \sqrt{((1-d_1) q_1)^2 + (d_1 \bar{\lambda}_{\max}(P))^2} \|\Sigma\|^2 \end{aligned} \quad (\text{C.10})$$

and

$$\begin{aligned} V_1(\Sigma) &\geq (1-d_1) q_0 y_e^2 + d_1 \bar{\lambda}_{\min}(P) \chi^2 \\ &\geq \sqrt{((1-d_1) q_0)^2 + (d_1 \bar{\lambda}_{\min}(P))^2} \|\Sigma\|^2 \end{aligned} \quad (\text{C.11})$$

Therefore, for $\Sigma \in B_R$, the Lyapunov function V_1 satisfies inequalities (3.61) with

$$\begin{aligned}
\Gamma_1 &= \underbrace{\sqrt{((1-d_1)q_0)^2 + (d_1\bar{\lambda}_{\min}(P))^2}}_{\bar{a}} \|\Sigma\|^2 \\
\Gamma_2 &= \underbrace{\sqrt{((1-d_1)q_1)^2 + (d_1\bar{\lambda}_{\max}(P))^2}}_{\bar{b}} \|\Sigma\|^2 \\
\Gamma_3 &= \lambda_{\min}(\Lambda) \|\Sigma\|^2 \\
\varsigma &= N \|\Sigma\|
\end{aligned} \tag{C.12}$$

Recall

$$\frac{k_{z\delta_1} k_{z\delta_2} |\chi_2|}{(k_{z\delta_2} y_e)^2 + 1 - k_{z\delta_1} k_{z\delta_2} \chi_2} < \frac{|\chi_2|}{D_e} \tag{C.13}$$

for $|\chi_2| \leq \frac{1}{k_{z\delta_1} k_{z\delta_2}}$. Eq. (C.5) can thus be rewritten as

$$\begin{aligned}
\|r(t, \Sigma)\| &\leq \begin{bmatrix} \frac{k_{z\delta_1} k_{z\delta_2} |\bar{\eta}| |\chi_2|}{D_e} + |\bar{\eta}| \\ 0 \\ 0 \\ 0 \\ 0 \\ h_2(v_1, \bar{\eta}, \omega) + b_4 u_2 \end{bmatrix} \\
&\leq \left(\frac{k_{z\delta_1} k_{z\delta_2} \bar{\theta}}{D_e} + 1 \right) \|\Sigma\|
\end{aligned}$$

The right-hand side of (3.63) is given by

$$\frac{\theta \Gamma_3(\Gamma_2^{-1}(\Gamma_1(R)))}{\varsigma} = \frac{\lambda_{\min}(\Lambda)}{N} \sqrt{\frac{\bar{a}}{\bar{b}}} \theta R \tag{C.14}$$

The derivative of V_1 along the trajectories of (C.1) satisfies

$$\begin{aligned}
\dot{V}_1(t, \Sigma) &= \frac{\partial V_1}{\partial \Sigma} Q(\Sigma) + \frac{\partial V_1}{\partial \Sigma} r(\Sigma, t) \\
&\leq -\lambda_{\min}(\Lambda) \|\Sigma\|^2 + \varphi N \|\Sigma\| \\
&\leq -(1-\theta) \lambda_{\min}(\Lambda) \|\Sigma\|^2 - \theta \lambda_{\min}(\Lambda) \|\Sigma\|^2 + \varphi N \|\Sigma\| \\
&\leq -(1-\theta) \lambda_{\min}(\Lambda) \|\Sigma\|^2 \quad \forall \quad \|\Sigma\| \geq \frac{\varphi N}{\theta \lambda_{\min}(\Lambda)}
\end{aligned} \tag{C.15}$$

The solution Σ of the perturbed system will be ultimately bounded with the bound given by (3.110).

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] Naomi Kato, Bugi Wiku Wicaksono, and Yoshihiro Suzuki. Development of biology-inspired autonomous underwater vehicle “BASS III” with high maneuverability. In *Proceedings of the 2000 International Symposium on Underwater Technology (Cat. No. 00EX418)*, pages 84–89. IEEE, 2000.
- [2] Parasar Kodati, Jonathan Hinkle, Aaron Winn, and Xinyan Deng. Microautonomous robotic ostraciiform (MARCO): Hydrodynamics, design, and fabrication. *IEEE Transactions on Robotics*, 24(1):105–117, 2008.
- [3] George Lauder, Peter GA Madden, Rajat Mittal, Haibo Dong, and Meliha Bozkurttas. Locomotion with flexible propulsors: I. Experimental analysis of pectoral fin swimming in sunfish. *Bioinspiration & Biomimetics*, 1(4):S25, 2006.
- [4] Kristi A Morgansen, Timothy M La Fond, and Jennifer X Zhang. Agile maneuvering for fin-actuated underwater vehicles. In *Proceedings of the 2006 Second International Symposium on Communications, Control and Signal Processing*. Citeseer, 2006.
- [5] Kin H Low and A. Renandya Willy. Biomimetic motion planning of an undulating robotic fish fin. *Journal of Vibration and Control*, 12(12):1337–1359, 2006.
- [6] Xiaobo Tan. Autonomous robotic fish as mobile sensor platforms: Challenges and potential solutions. *Marine Technology Society Journal*, 45(4):31–40, 2011.
- [7] Feitian Zhang, Osama Ennasr, Elena Litchman, and Xiaobo Tan. Autonomous sampling of water columns using gliding robotic fish: Algorithms and harmful-algae-sampling experiments. *IEEE Systems Journal*, 10(3):1271–1281, 2016.
- [8] Stefano Marras and Maurizio Porfiri. Fish and robots swimming together: attraction towards the robot demands biomimetic locomotion. *Journal of The Royal Society Interface*, 9(73):1856–1868, 2012.
- [9] Giovanni Polverino, Mert Karakaya, Chiara Spinello, Vrishin R Soman, and Maurizio Porfiri. Behavioural and life-history responses of mosquitofish to biologically inspired and interactive robotic predators. *Journal of the Royal Society Interface*, 16(158):20190359, 2019.
- [10] Michael S. Triantafyllou and George S. Triantafyllou. An efficient swimming machine. *Scientific American*, 273(3):64–70, 1995.
- [11] Michael Sfakiotakis, David M Lane, and J Bruce C Davies. Review of fish swimming modes for aquatic locomotion. *IEEE Journal of Oceanic Engineering*, 24(2):237–252, 1999.
- [12] Michael S Triantafyllou, George S Triantafyllou, and Dick K. P. Yue. Hydrodynamics of fishlike swimming. *Annual review of fluid mechanics*, 32(1):33–53, 2000.

- [13] Michael Rufo. Ghostswimmer: Tactically relevant, biomimetically inspired, silent, highly efficient and maneuverable autonomous underwater vehicle. *Boston Engineering Corporation*, 411, 2012.
- [14] Sander C. van den Berg, Rob B. N. Scharff, Zoltán Rusák, and Jun Wu. Biomimetic design of a soft robotic fish for high speed locomotion. In Vasiliki Vouloutsis, Anna Mura, Falk Tauber, Thomas Speck, Tony J. Prescott, and Paul F. M. J. Verschure, editors, *Biomimetic and Biohybrid Systems*, pages 366–377, Cham, 2020. Springer International Publishing.
- [15] Shuxiang Guo, Toshio Fukuda, and Kinji Asaka. A new type of fish-like underwater microbot. *IEEE/ASME Transactions on Mechatronics*, 8(1):136–141, 2003.
- [16] Huosheng Hu, Jindong Liu, Ian Dukes, and George Francis. Design of 3D swim patterns for autonomous robotic fish. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2406–2411, 2006.
- [17] Daniele Costa, Matteo Franciolini, Giacomo Palmieri, Andrea Crivellini, and David Scaradozzi. Computational fluid dynamics analysis and design of an ostraciiform swimming robot. In *Robotics and Biomimetics (ROBIO), 2017 IEEE International Conference on*, pages 135–140. IEEE, 2017.
- [18] Xingyu Chen, Junzhi Yu, Zhengxing Wu, Yan Meng, and Shihan Kong. Toward a maneuverable miniature robotic fish equipped with a novel magnetic actuator system. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 13(3):1–11, 2018.
- [19] Pan Liao, Shiwu Zhang, and Dong Sun. A dual caudal-fin miniature robotic fish with an integrated oscillation and jet propulsive mechanism. *Bioinspiration & Biomimetics*, 13(3):036007, 2018.
- [20] Alessio Alessi, Angelo Sudano, Dino Accoto, and Eugenio Guglielmelli. Development of an autonomous robotic fish. In *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on*, pages 1032–1037. IEEE, 2012.
- [21] Tetsuo Ichikizaki and Ikuo Yamamoto. Development of robotic fish with various swimming functions. In *Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies, 2007. Symposium on*, pages 378–383. IEEE, 2007.
- [22] Kin Huat Low and Willy A. Renandya. Development and initial investigation of NTU robotic fish with modular flexible fins. In *Mechatronics and Automation, 2005 IEEE International Conference*, volume 2, pages 958–963. IEEE, 2005.
- [23] Xiaobo Tan, Drew Kim, Nathan Usher, Dan Laboy, Joel Jackson, Azra Kapetanovic, Jason Rapai, Benjamin Sabadus, and Xin Zhou. An autonomous robotic fish for mobile sensing. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5424–5429. IEEE, 2006.
- [24] Palmani Duraisamy, Rakesh Kumar Sidharthan, and Manigandan Nagarajan Santhanakrishnan. Design, modeling, and control of biomimetic fish robot: A review. *Journal of Bionic Engineering*, 16(6):967–993, 2019.

- [25] Sanaz Bazaz Behbahani. *Role of Flexibility in Robotic Fish*. PhD thesis, Michigan State University, 2016.
- [26] Patar Ebenezer Sitorus, Yul Yunazwin Nazaruddin, Edi Leksono, and Agus Budiyo. Design and implementation of paired pectoral fins locomotion of labriform fish applied to a fish robot. *Journal of Bionic Engineering*, 6(1):37–45, 2009.
- [27] Zhihang Ye, Piqui Hou, and Zheng Chen. 2D maneuverable robotic fish propelled by multiple ionic polymer–metal composite artificial fins. *International Journal of Intelligent Robotics and Applications*, 1(2):195–208, 2017.
- [28] Naomi Kato. Locomotion by mechanical pectoral fins. *Journal of Marine Science and Technology*, 3(3):113–121, 1998.
- [29] Daisy Lachat, Alessandro Crespi, and Auke Jan Ijspeert. Boxybot: a swimming and crawling fish robot controlled by a central pattern generator. In *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006.*, pages 643–648. IEEE, 2006.
- [30] Xinyan Deng and Srinath Avadhanula. Biomimetic micro underwater vehicle with oscillating fin propulsion: System design and force measurement. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3312–3317. IEEE, 2005.
- [31] Kristi A Morgansen, Benjamin I Triplett, and Daniel J Klein. Geometric methods for modeling and control of free-swimming fin-actuated underwater vehicles. *IEEE Transactions on Robotics*, 23(6):1184–1199, 2007.
- [32] Kristi A Morgansen, Benjamin I Triplett, and Daniel J Klein. Geometric methods for modeling and control of free-swimming underwater vehicles. *IEEE Transactions on Robotics*, 23(6):1184–1199, 2007.
- [33] Zheng Chen, Stephan Shatara, and Xiaobo Tan. Modeling of biomimetic robotic fish propelled by an ionic polymer-metal composite caudal fin. *IEEE/ASME Transactions on Mechatronics*, 15(3):448–459, 2010.
- [34] Scott D Kelly and Richard M Murray. Modeling efficient pisciform swimming for control. *International Journal of Robust and Nonlinear Control*, 10:217–241, 2000.
- [35] Jianxun Wang, Freddie Alequin-Ramos, and Xiaobo Tan. Dynamic modeling of robotic fish and its experimental validation. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 588–594, San Francisco, California, 2011.
- [36] Jianxun Wang, Philip K McKinley, and Xiaobo Tan. Dynamic modeling of robotic fish with a flexible caudal fin. In *ASME 2012 Dynamic Systems and Control Conference*, pages 203–212, Fort Lauderdale, FL, 2012.
- [37] Sanaz Bazaz Behbahani and Xiaobo Tan. Bio-inspired flexible joints with passive feathering for robotic fish pectoral fins. *Bioinspiration & Biomimetics*, 11(3):036009, 2016.

- [38] Vladislav Kopman, Jeffrey Laut, Francesco Acquaviva, Alessandro Rizzo, and Maurizio Porfiri. Dynamic modeling of a robotic fish propelled by a compliant tail. *IEEE Journal of Oceanic Engineering*, 40(1):209–221, 2015.
- [39] David Barrett, Mark Grosenbaugh, and Michael Triantafyllou. The optimal control of a flexible hull robotic undersea vehicle propelled by an oscillating foil. In *Autonomous Underwater Vehicle Technology, 1996. AUV'96., Proceedings of the 1996 Symposium on*, pages 1–9. IEEE, 1996.
- [40] Jamie M Anderson and Narender K Chhabra. Maneuvering and stability performance of a robotic tuna. *Integrative and Comparative Biology*, 42(1):118–126, 2002.
- [41] Shuo Wang, Zhigang Zhang, and Haiquan Sang. Analysis of velocity control algorithms for biomimetic robot fish. In *Robotics and Biomimetics, 2004. ROBIO 2004. IEEE International Conference on*, pages 972–976. IEEE, 2004.
- [42] Chao Zhou, Zhiqiang Cao, Shuo Wang, and Min Tan. The posture control and 3D locomotion implementation of biomimetic robot fish. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5406–5411. IEEE, 2006.
- [43] Koichi Hirata, Tadanori Takimoto, and Kenkichi Tamura. Study on turning performance of a fish robot. In *First International Symposium on Aqua Bio-Mechanisms*, pages 287–292, 2000.
- [44] Kristi A Morgansen, V Duidam, Richard J Mason, Joel W Burdick, and Richard M Murray. Nonlinear control methods for planar carangiform robot fish locomotion. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 1, pages 427–434. IEEE, 2001.
- [45] Junzhi Yu, Long Wang, and Min Tan. A framework for biomimetic robot fish's design and its realization. In *American Control Conference, 2005. Proceedings of the 2005*, pages 1593–1598. IEEE, 2005.
- [46] Feitian Zhang, Fumin Zhang, and Xiaobo Tan. Tail-enabled spiraling maneuver for gliding robotic fish. *Journal of Dynamic Systems, Measurement, and Control*, 136(4):041028, 2014.
- [47] Michail Makrodimitris, Kostas Nanos, and Evangelos Papadopoulos. A novel trajectory planning method for a robotic fish. In *Control and Automation (MED), 2017 25th Mediterranean Conference on*, pages 1119–1124. IEEE, 2017.
- [48] Long Wang, Shuo Wang, Zhiqiang Cao, Min Tan, Chao Zhou, Haiquan Sang, and Zhizhong Shen. Motion control of a robot fish based on CPG. In *Industrial Technology, 2005. ICIT 2005. IEEE International Conference on*, pages 1263–1268. IEEE, 2005.
- [49] Wei Zhao, Yonghui Hu, L Zhang, and Long Wang. Design and CPG-based control of biomimetic robotic fish. *IET Control Theory & Applications*, 3(3):281–293, 2009.
- [50] Wei Wang, Jiajie Guo, Zijian Wang, and Guangming Xie. Neural controller for swimming modes and gait transition on an ostraciiform fish robot. In *Advanced Intelligent Mechatronics (AIM), 2013 IEEE/ASME International Conference on*, pages 1564–1569. IEEE, 2013.

- [51] Yonghui Hu, Weicheng Tian, Jianhong Liang, and Tianmiao Wang. Learning fish-like swimming with a CPG-based locomotion controller. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1863–1868. IEEE, 2011.
- [52] Qinyuan Ren, Jianxin Xu, and Xuefang Li. A data-driven motion control approach for a robotic fish. *Journal of Bionic Engineering*, 12(3):382–394, 2015.
- [53] Wei Wang, Dongbing Gu, and Guangming Xie. Autonomous optimization of swimming gait in a fish robot with multiple onboard sensors. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(99):1–13, 2017.
- [54] Kristi A Morgansen, Patricio A Vela, and Joel W Burdick. Trajectory stabilization for a planar carangiform robot fish. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 1, pages 756–762. IEEE, 2002.
- [55] Jian-Xin Xu, Xue-Lei Niu, and Zhao-Qin Guo. Sliding mode control design for a carangiform robotic fish. In *2012 12th International Workshop on Variable Structure Systems*, pages 308–313. IEEE, 2012.
- [56] Junzhi Yu, Min Tan, Shuo Wang, and Erkui Chen. Development of a biomimetic robotic fish and its control algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(4):1798–1810, 2006.
- [57] Saroj Saimek and Perry Y Li. Motion planning and control of a swimming machine. *The International Journal of Robotics Research*, 23(1):27–53, 2004.
- [58] Yonghui Hu, Wei Zhao, and Long Wang. Vision-based target tracking and collision avoidance for two autonomous robotic fish. *IEEE Transactions on Industrial Electronics*, 56(5):1401–1410, 2009.
- [59] Pichet Suebsaiprom and Chun-Liang Lin. Maneuverability modeling and trajectory tracking for fish robot. *Control Engineering Practice*, 45:22–36, 2015.
- [60] Kexu Zou, Chen Wang, Guangming Xie, Tianguang Chu, Long Wang, and Yingmin Jia. Cooperative control for trajectory tracking of robotic fish. In *American Control Conference, 2009. ACC'09.*, pages 5504–5509. IEEE, 2009.
- [61] Pichet Suebsaiprom and Chun-Liang Lin. Sliding mode path tracking control for fish-robot under ocean current perturbation. In *Control and Automation (ICCA), 2016 12th IEEE International Conference on*, pages 972–977. IEEE, 2016.
- [62] Songlin Chen, Jianxun Wang, and Xiaobo Tan. Backstepping-based hybrid target tracking control for a carangiform robotic fish. In *ASME 2013 Dynamic Systems and Control Conference*, pages V002T32A005–V002T32A005. American Society of Mechanical Engineers, 2013.
- [63] Erfu Yang, Takayuki Ikeda, and Tsutomu Mita. Nonlinear tracking control of a nonholonomic fish robot in chained form. In *SICE 2003 Annual Conference (IEEE Cat. No. 03TH8734)*, volume 2, pages 1260–1265. IEEE, 2003.

- [64] Janet E Fontanella, Frank E Fish, Elizabeth I Barchi, Regina Campbell-Malone, Rachel H Nichols, Nicole K DiNenno, and John T Beneski. Two-and three-dimensional geometries of batoids in relation to locomotor mode. *Journal of experimental marine biology and ecology*, 446:273–281, 2013.
- [65] Lisa J Rosenberger. Pectoral fin locomotion in batoid fishes: undulation versus oscillation. *Journal of Experimental Biology*, 204(2):379–394, 2001.
- [66] Hongan Wang. *Design and Implementation of Biomimetic Robotic Fish*. Citeseer, 2009.
- [67] Steven Vogel. *Life in moving fluids: the physical biology of flow*. Princeton, N.J. : Princeton University Press, 1996.
- [68] John Palmisano, Ravi Ramamurti, Kerr-Jia Lu, Jonah Cohen, William Sandberg, and Banahalli Ratna. Design of a biomimetic controlled-curvature robotic pectoral fin. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 966–973. IEEE, 2007.
- [69] Rajat Mittal, Haibo Dong, Meliha Bozkurtas, GeorgeV Lauder, and Peter Madden. Locomotion with flexible propulsors: II. computational modeling of pectoral fin swimming in sunfish. *Bioinspiration & Biomimetics*, 1(4):S35, 2006.
- [70] Yong Zhong, Zheng Li, and Ruxu Du. Robot fish with two-DOF pectoral fins and a wire-driven caudal fin. *Advanced Robotics*, 32(1):25–36, 2018.
- [71] Navinder Singh, Ankur Gupta, and Sujoy Mukherjee. A dynamic model for underwater robotic fish with a servo actuated pectoral fin. *SN Applied Sciences*, 1(7):659, 2019.
- [72] Shusheng Bi, Hongwei Ma, Yueri Cai, Chuanmeng Niu, and Yuliang Wang. Dynamic modeling of a flexible oscillating pectoral fin for robotic fish. *Industrial Robot: An International Journal*, 2014.
- [73] Xinmin Liu, Tetsuya Iwasaki, and Frank Fish. Dynamic modeling and gait analysis of batoid swimming. In *2013 American Control Conference*, pages 566–571. IEEE, 2013.
- [74] Naomi Kato and Tadahiko Inaba. Guidance and control of fish robot with apparatus of pectoral fin motion. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, volume 1, pages 446–451. IEEE, 1998.
- [75] Naomi Kato and Motonori Furushima. Pectoral fin model for maneuver of underwater vehicles. In *Proceedings of Symposium on Autonomous Underwater Vehicle Technology*, pages 49–56. IEEE, 1996.
- [76] Ming Wang, Junzhi Yu, and Min Tan. Modeling neural control of robotic fish with pectoral fins using a CPG-based network. In *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 6502–6507. IEEE, 2009.

- [77] Ming Wang, Junzhi Yu, Min Tan, and Jianwei Zhang. Design and implementation of a novel cpg-based locomotion controller for robotic dolphin. In *2010 8th World Congress on Intelligent Control and Automation*, pages 1611–1616. IEEE, 2010.
- [78] Ming Wang, JunZhi Yu, Min Tan, and JianWei Zhang. Multimodal swimming control of a robotic fish with pectoral fins using a CPG network. *Chinese Science Bulletin*, 57(10):1209–1216, 2012.
- [79] Junzhi Yu, Kai Wang, Min Tan, and Jianwei Zhang. Design and control of an embedded vision guided robotic fish with multiple control surfaces. *The Scientific World Journal*, 2014, 2014.
- [80] Shiwu Zhang, Yun Qian, Pan Liao, Fenghua Qin, and Jiming Yang. Design and control of an agile robotic fish with integrative biomimetic mechanisms. *IEEE/ASME Transactions on Mechatronics*, 21(4):1846–1857, 2016.
- [81] Ming Wang, Junzhi Yu, and Min Tan. CPG-based sensory feedback control for bio-inspired multimodal swimming. *International Journal of Advanced Robotic Systems*, 11(10):170, 2014.
- [82] Naomi Kato. Control performance in the horizontal plane of a fish robot with mechanical pectoral fins. *IEEE journal of oceanic engineering*, 25(1):121–129, 2000.
- [83] Fu-dong Gao, Cun-yun Pan, Yan-yan Han, and Xiang Zhang. Nonlinear trajectory tracking control of a new autonomous underwater vehicle in complex sea conditions. *Journal of Central South University*, 19(7):1859–1868, 2012.
- [84] Xianbo Xiang, Lionel Lapierre, Chao Liu, and Bruno Jouvencel. Path tracking: Combined path following and trajectory tracking for autonomous underwater vehicles. In *In Proceedings of 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3558–3563, 2011.
- [85] Gu Dongbing and Hu Huosheng. A stabilizing receding horizon regulator for nonholonomic mobile robots. *IEEE Transactions on Robotics*, 21(5):1022–1028, 2005.
- [86] Kiattisin Kanjanawanishkul and Andreas Zell. Path following for an omnidirectional mobile robot based on model predictive control. In *In Proceedings of 2009 IEEE International Conference on Robotics and Automation*, pages 3341–3346, 2009.
- [87] Xiang Li, Kiattisin Kanjanawanishkul, and Andreas Zell. Nonlinear model predictive control of an omnidirectional mobile robot. In *In Proceedings of the 10th International Conference on Intelligent Autonomous Systems (IAS 2008)*, pages 92–99, 2008.
- [88] Kiattisin Kanjanawanishkul. MPC-based path following control of an omnidirectional mobile robot with consideration of robot constraints. *Advances in Electrical and Electronic Engineering*, 13(1):54–63, 2015.
- [89] Jianxun Wang and Xiaobo Tan. Averaging tail-actuated robotic fish dynamics through force and moment scaling. *IEEE Transactions on Robotics*, 31(4):906–917, 2015.

- [90] Boris Houska, Hans Joachim Ferreau, and Moritz Diehl. ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.
- [91] Maria L Castaño and Xiaobo Tan. Model predictive control of a tail-actuated robotic fish. In *Dynamic Systems and Control Conference*, volume 50695, page V001T03A006. American Society of Mechanical Engineers, 2016.
- [92] Maria L Castaño and Xiaobo Tan. Model predictive control-based path-following for tail-actuated robotic fish. *Journal of Dynamic Systems, Measurement, and Control*, 141(7), 2019.
- [93] Khac Duc Do and Jie Pan. *Control of Ships and Underwater Vehicles: Design for Underactuated and Nonlinear Marine Systems*. Springer Science & Business Media, 2009.
- [94] Maria L Castaño and Xiaobo Tan. Backstepping control-based trajectory tracking for tail-actuated robotic fish. In *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 839–844. IEEE, 2019.
- [95] Jan A Sanders, Ferdinand Verhulst, and James Murdock. *Averaging Methods in Nonlinear Dynamical Systems*, volume 59. Springer, 2007.
- [96] Francesco Bullo and Andrew D Lewis. *Geometric Control of Mechanical Systems: Modeling, Analysis, and Design for Simple Mechanical Control systems*, volume 49. Springer, 2019.
- [97] Maria L Castaño and Xiaobo Tan. Averaged modeling of pectoral fin-actuated robotic fish. In *The 2021 Inaugural Modeling, Estimation and Control Conference (MECC)*, page In press. AACC, 2021.
- [98] Maria L Castaño and Xiaobo Tan. Rapid maneuvering control of pectoral fin-actuated robotic fish. In *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 705–712. IEEE, 2021.
- [99] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: Theory and practice - a survey. *Automatica*, 25(3):335–348, 1989.
- [100] Frank. Allgöwer, Rolf Findeisen, and Zoltan K Nagy. Nonlinear model predictive control: From theory to application. *J. Chin. Inst. Chem. Engrs*, 35(3):299–315, 2004.
- [101] Shuyou Yu, Xiang Li, Hong Chen, and Frank Allgower. Nonlinear model predictive control for path following problems. *IFAC Proceedings Volumes*, 45(17):145 – 150, 2012. 4th IFAC Conference on Nonlinear Model Predictive Control.
- [102] Matteo Aureli, Vladislav Kopman, and Maurizio Porfiri. Free-locomotion of underwater vehicles actuated by ionic polymer metal composites. *IEEE/ASME Transactions on Mechatronics*, 15(4):603–614, 2009.
- [103] Filoktimon Repoulas and Evangelos Papadopoulos. Planar trajectory planning and tracking control design for underactuated AUVs. *Ocean Engineering*, 34(11-12):1650–1667, 2007.

- [104] Filoktimon Repoulas and Evangelos Papadopoulos. Trajectory planning and tracking control of underactuated AUVs. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1610–1615. IEEE, 2005.
- [105] Jawhar Ghommam, Faical Mnif, and N Derbel. Global stabilisation and tracking control of underactuated surface vessels. *IET Control Theory & Applications*, 4(1):71–88, 2010.
- [106] Adriaan A J Lefeber. *Tracking Control of Nonlinear Mechanical Systems*. Universiteit Twente Eindhoven, The Netherlands, 2000.
- [107] Kristin Y Pettersen and Henk Nijmeijer. Tracking control of an underactuated surface vessel. In *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No. 98CH36171)*, volume 4, pages 4561–4566. IEEE, 1998.
- [108] Zhong-Ping Jiang. Global tracking control of underactuated ships by Lyapunov’s direct method. *Automatica*, 38(2):301–309, 2002.
- [109] Kristin Y Pettersen and Henk Nijmeijer. Underactuated ship tracking control: Theory and experiments. *International Journal of Control*, 74(14):1435–1446, 2001.
- [110] Ruiting Yu, Qidan Zhu, Guihua Xia, and Zhe Liu. Sliding mode tracking control of an underactuated surface vessel. *IET Control Theory & Applications*, 6(3):461, 2012.
- [111] Erjen Lefeber, Kristin Ytterstad Pettersen, and Henk Nijmeijer. Tracking control of an underactuated ship. *IEEE Transactions on Control Systems Technology*, 11(1):52–61, Jan 2003.
- [112] Elena Panteley and Antonio Loria. Growth rate conditions for uniform asymptotic stability of cascaded time-varying systems. *Automatica*, 37(3):453–460, 2001.
- [113] Jian Xu, Man Wang, and Lei Qiao. Dynamical sliding mode control for the trajectory tracking of underactuated unmanned underwater vehicles. *Ocean Engineering*, 105:54–63, Sep 2015.
- [114] John-Morten Godhavn. Nonlinear tracking of underactuated surface vessels. In *Proceedings of 35th IEEE Conference on Decision and Control*, volume 1, pages 975–980. IEEE, 1996.
- [115] John-Morten Godhavn, Thor I Fossen, and Svein P Berge. Non-linear and adaptive backstepping designs for tracking control of ships. *International Journal of Adaptive Control and Signal Processing*, 12(8):649–670, 1998.
- [116] Jia-Zhong Zhang, Yingjie Wei, F.Y. Bi, and Weijin Cao. Position-tracking control of underactuated autonomous underwater vehicles in the presence of unknown ocean currents. *IET Control Theory & Applications*, 4(11):2369–2380, Nov 2010.
- [117] Hashem Ashrafiuon, Kenneth R Muske, Lucas C McNinch, and Reza A Soltan. Sliding-mode tracking control of surface vessels. *IEEE Transactions on Industrial Electronics*, 55(11):4004–4012, 2008.

- [118] Taha Elmokadem, Mohamed Zribi, and Kamal Youcef-Toumi. Trajectory tracking sliding mode control of underactuated AUVs. *Nonlinear Dynamics*, 84(2):1079–1091, Apr 2016.
- [119] H Sira-Ramfrez. On the control of the underactuated ship: A trajectory planning approach. In *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No. 99CH36304)*, volume 3, pages 2192–2197. IEEE, 1999.
- [120] Pedro A Aguiar and Joao P Hespanha. Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. *IEEE Transactions on Automatic Control*, 52(8):1362–1379, 2007.
- [121] Aman Behal, Darren M. Dawson, Bin Xian, and Pradeep Setlur. Adaptive tracking control of underactuated surface vessels. In *Proceedings of the 2001 IEEE International Conference on Control Applications (CCA'01) (Cat. No.01CH37204)*, page 645–650. IEEE, 2001.
- [122] Khac Duc Do, Zhong-Ping Jiang, and Jie Pan. Underactuated ship global tracking under relaxed conditions. *IEEE Transactions on Automatic Control*, 47(9):1529–1536, Sep 2002.
- [123] Khac D. Do, Zhong-Ping. Jiang, and Jie Pan. Universal controllers for stabilization and tracking of underactuated ships. *Systems & Control Letters*, 47(4):299–317, Nov 2002.
- [124] Khac D Do, Zhong-Ping Jiang, and Jie Pan. Global partial-state feedback and output-feedback tracking controllers for underactuated ships. *Systems & Control Letters*, 54(10):1015–1036, 2005.
- [125] Zhang Wei, Xu Da, Tan Meiling, Wang Conglin, and Yan Zheping. Trajectory tracking control of underactuated uuv for underwater recovery. In *2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control*, pages 386–391. IEEE, 2012.
- [126] Dongkyoung Chwa. Global tracking control of underactuated ships with input and velocity constraints using dynamic surface control method. *IEEE Transactions on Control Systems Technology*, 19(6):1357–1370, Nov 2011.
- [127] Reza Olfati-Saber. *Nonlinear Control of Underactuated Mechanical Systems with Application to Robotics and Aerospace Vehicles*. PhD thesis, Massachusetts Institute of Technology, 2001.
- [128] Jianxun Wang and Xiaobo Tan. A dynamic model for tail-actuated robotic fish with drag coefficient adaptation. *Mechatronics*, 23(6):659–668, 2013.
- [129] Hassan Khalil. *Nonlinear Control*. Pearson New York, 2014.
- [130] Thor I Fossen. *Guidance and Control of Ocean Vehicles*. PhD thesis, University of Trondheim, Norway, 1994.
- [131] Petar Kokotovic, Hassan K Khali, and John O'reilly. *Singular Perturbation Methods in Control: Analysis and Design*, volume 25. Siam, 1999.

- [132] Jay Farrell, Marios Polycarpou, and Manu Sharma. Adaptive backstepping with magnitude, rate, and bandwidth constraints: Aircraft longitude control. In *Proceedings of the 2003 American Control Conference*, volume 5, pages 3898–3904. IEEE, 2003.
- [133] Rong Mei and Qingliang Cui. Backstepping control for a 3DOF model helicopter with input and output constraints. *International Journal of Advanced Robotic Systems*, 14(1):1729881416687133, 2016.
- [134] Finn Haugen. *PID Control*, chapter 4, pages 94–98. Tapir Academic Press, 2004.
- [135] Jianxun Wang, Philip K McKinley, and Xiaobo Tan. Dynamic modeling of robotic fish with a base-actuated flexible tail. *Journal of Dynamic Systems, Measurement, and Control*, 137(1):11004(1)–11004(11), 2015.
- [136] Sanaz Bazaz Behbahani and Xiaobo Tan. Role of pectoral fin flexibility in robotic fish performance. *Journal of Nonlinear Science*, 27(4):1155–1181, 2017.
- [137] Pablo Valdivia y Alvarado and Kamal Youcef-Toumi. Design of machines with compliant bodies for biomimetic locomotion in liquid environments. *Journal of Dynamic Systems, Measurement, and Control*, 128(1):3–13, 09 2005.
- [138] Eliot G Drucker, Jeffrey A Walker, and Mark W Westneat. Mechanics of pectoral fin swimming in fishes. *Fish Physiology*, 23:369–423, 2005.
- [139] R.G Dong. Effective mass and damping of submerged structures. Technical report, California Univ., Livermore (USA). Lawrence Livermore Lab., 1978.
- [140] Sanaz Bazaz Behbahani and Xiaobo Tan. Design and modeling of flexible passive rowing joint for robotic fish pectoral fins. *IEEE Transactions on Robotics*, 32(5):1119–1132, 2016.
- [141] Kalyanmoy Deb. *Multi-objective Optimization Using Evolutionary Algorithms*, volume 16. John Wiley & Sons, 2001.
- [142] Francesco Casella and Marco Lovera. LPV/LFT modelling and identification: overview, synergies and a case study. In *2008 IEEE International Conference on Computer-Aided Control Systems*, pages 852–857. IEEE, 2008.
- [143] Yucai Zhu and Zuhua Xu. A method of LPV model identification for control. *IFAC Proceedings Volumes*, 41(2):5018–5023, 2008.
- [144] Maria L Castaño, Anastasia Mavrommati, Todd D Murphey, and Xiaobo Tan. Trajectory planning and tracking of robotic fish using ergodic exploration. In *2017 American Control Conference (ACC)*, pages 5476–5481. IEEE, 2017.
- [145] Giorgos Mamakoukas, Maria Castano, Xiaobo Tan, and Todd Murphey. Local koopman operators for data-driven control of robotic systems. In *Robotics: science and systems*, 2019.
- [146] Giorgos Mamakoukas, Maria L Castano, Xiaobo Tan, and Todd D Murphey. Derivative-based koopman operators for real-time control of robotic systems. *IEEE Transactions on Robotics*, 2021.