From 2D to 3D Maneuverable Robotic Fish: A Systems Perspective

by
Wenyu Zuo

A dissertation submitted to the Department of Mechanical Engineering,

Cullen College of Engineering

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in Mechanical Engineering

Chair of Committee: Dr. Zheng Chen

Committee Member: Dr. Matthew A. Franchek

Committee Member: Dr. Miao Pan

Committee Member: Dr. Gangbing Song

Committee Member: Dr. Karolos Grigoriadis

University of Houston
May 2022

ACKNOWLEDGMENTS

I would like to acknowledge and express my sincere gratitude to those who have helped me with this dissertation and supported me through my PhD study.

My first round thanks go to my advisor, Dr. Zheng Chen, who gave me the opportunity to be a team member in his lab. His enthusiastic encouragement and thoughtful guidance carried me through all the stages of my projects. Dr. Chen inspired me with his envision and expertise and guided me with his patience during the academic years. Besides providing technical guidance, he also offered me invaluable advises on how to pursue my academic career.

I would also like to thank my committee members, Dr. Miao Pan, Dr. Matthew Franchek, Dr. Karolos Grigoriadis, and Dr. Gangbing Song for providing insightful suggestions and comments, making my defense an enjoyable moment. As for the colleagues who helped and supported me with my projects and researches, my thanks go to my colleagues Dr. Alicia Keow, Dr. Xiongfeng Yi, Theophilus Kaaya, Shengbin Wang. Thanks also go to the collaborators at Rice University, West Chester University, and The University of Texas at Austin, in particular Dr. Frank Fish, Dr. Animesh Chakravarthy, and Dr. Fathi Ghorbel , who gave me great help during my research.

I owe my greatest thanks to the financial support of my studies and research from the Subsea Systems Institute in UH Energyand National Science Foundation

Special thanks must go to my dear wife Xiang for her understanding and sacrifice when I was undertaking my research and doing my projects. Her continuous supporting and patience were what sustained me this far. The last but not the least, I am grateful to my parents for their support and sacrifice that enabled me to be here to pursue my PhD degree.

ABSTRACT

Robotic fish, as an emerging member of marine robots, have received lots of attention in recent years. Because of its unique propulsion mechanism, a large amount of research work today focuses on robotic fish design. Due to the complex hydrodynamics, the modeling of the robotic fish has become a challenging topic, and the research on control and application is still in its beginning.

This study systematically introduces the development and application of a robotic fish from the perspective of design, modeling, and control. A three-joint robotic fish propelled by a Double-Slider-Crank (DSC) mechanism, which uses one DC motor to achieve oscillating foil propulsion, is designed. From the design aspect, DSC helps the robotic fish in mimicking a real fish's two-dimensional free-swimming. The robotic fish's top speed is 0.35 m/s at 3 Hz, equivalent to 0.98 body length (BL) per second. DSC also benefits the control of the robotic fish by independently adjusting its steering and swimming speed. This characteristic is studied in a hydrodynamic model that derives the thrust within a DSC frame. A semi-physics-based and data-driven linear model is established to connect the bias angle to the robotic fish's steering. A linear model is used to design a controller, called event-trigger-control, to overcome the adverse effects of communication drop-off.

Furthermore, the work is extended to a robotic fish application study that uses robotic fish to estimate the flow field. Besides, the three-dimensional maneuverability is also addressed by developing a buoyancy control device to change depth. Overall, the proposed robotic fish has an excellent performance in free-swimming and shows great application value in environmental surveys.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Background and Motivation

Ocean covers approximately 71 % surface of the earth and carries nearly 90 % of world shipping. The ocean-based industries, including ocean transportation, fishing, aquaculture, and offshore energy, are closely related to human life. As one of the most important tools for humans to explore the ocean, autonomous underwater vehicles (AUVs) have attracted increasing attention in recent decades [5]. Most AUVs obtain thrust through propeller, a heavy device with high energy consumption and large acoustic noise [6]. These shortcomings hinder the development of compact-size AUVs and limit their applications in large-scale and long-duration missions which require less impact on marine habitats.

In contrast, millions of years of evolution have endowed fish with highly energy-efficient and maneuverable swimming skills [7, 8]. Their propulsion mechanism is promising for improving the propulsive efficiency and maneuvering capability of AUV. Inspired by fish swimming, many engineers and researchers have designed and built a flapping-based propulsion mechanism that can inherit the advantages of fish swimming and is favorable for AUV applications.

The primary efforts on robotic fish research can be divided into three categories: design, modeling, and control [9]. Researchers adopt various layouts and attempt different actuators to imitate the swimming of fish, which is characterized by a unique wave-like undulatory locomotion. The choices of propulsion mechanism include pectoral fin propulsion and caudal fin propulsion, and the choices of actuator include motors, hydraulics or pneumatic, magnetic actuators, and smart materials. Although robotic fish still has a gap to match the swimming performance of the real fish, its advantages, such as propulsive efficiency and simple mechanism, exhibit great potential in applications.

An accurate dynamic model that reflects the status of the system and predict the direction of evaluation would be of great benefit for a robot in its controller design and applications. However, for a robotic fish, establishing such a model that calculates the thrust produced by the fishtail is very difficult due to the complex interaction between the muscles' internal forces and the fluid's external forces [10, 11]. An alternative approach is to establish a control-oriented model using well-developed system model reduction and identification algorithms to bypass the complicated physical model. The obtained model would be formatted in terms of a state-of-art representation of differential equations that balances mathematical structure complicity and controller design compatibility. Thus, the modeling work can be led in a control-oriented direction so that the model can be used in controlling the robotic fish with the help of modern control theory.

With a solid design and accurate model, robotic fish have demonstrated great application value. A robotic fish can dive or rise with changeable speed and flexible steering. Current studies have shown that robotic fish can effectively complete trajectory tracking, target following, and team swarming. With continuous improvements of hardware performance, the direction of robotic fish study has gradually shifted to establishing a comprehensive intelligent platform that integrates robotics, information sensing, and wireless communication. Moreover, due to the high propulsive efficiency and low cost of robotic fish, robotic fish is more suitable for large-scale and long time deployment. This feature gives robotic fish a significant advantage in environmental monitoring and data collecting. In this trend, overcoming the obstacles of underwater communication and finding a practical application for robotic fish are required to be sufficiently addressed.

1.2 Literature Review

1.2.1 Review of robotic fish design

It is well known that fish gain thrust by flapping their body and tail. Fish can be classified into two categories according to the parts of their bodies that participate in flapping: the median or paired fin (MPF) swimmers or body and caudal fin (BCF) swimmers [12]. Compared to MPF swimmers, BCF swimmers have more body parts involved in flapping. Their flapping is more aggressive so that they can obtain more thrusts to accelerate and have better stability. Considering the main purpose of researching robotic fish is to take advantage of the high propulsive efficiency of the flapping, BCF swimmers are ideal models for robotic fish design.

There are four types BCF swimmers: anguilliform, subcarangiform, subcarangiform, and thunniform [13]. Anguilliform and subcarangiform fish have the most significant swing amplitude. Their entire body participates in the swing, which gives them the best flexibility and maneuverability. Carangiform and thunniform swimmers only swing their tail parts. They usually have a large and strong caudal fin to gain thrust. Most large-scale fish adopt the thunniform method to enhance their long-distance swimming ability. In contrast, small-sized fish prefer anguilliform or subcarangiform methods to survive in a compact environment and escape predators. The design of robotic fish focuses on mimicking the carangiform or thunniform swimmer. Anguilliform and subcarangiform swimmers require high orders of degrees-of-freedom (DOF), which is too complicated and less energy efficient. Carangiform and thunniform swimmers have a better balance between the design complexity and the propulsive efficiency.

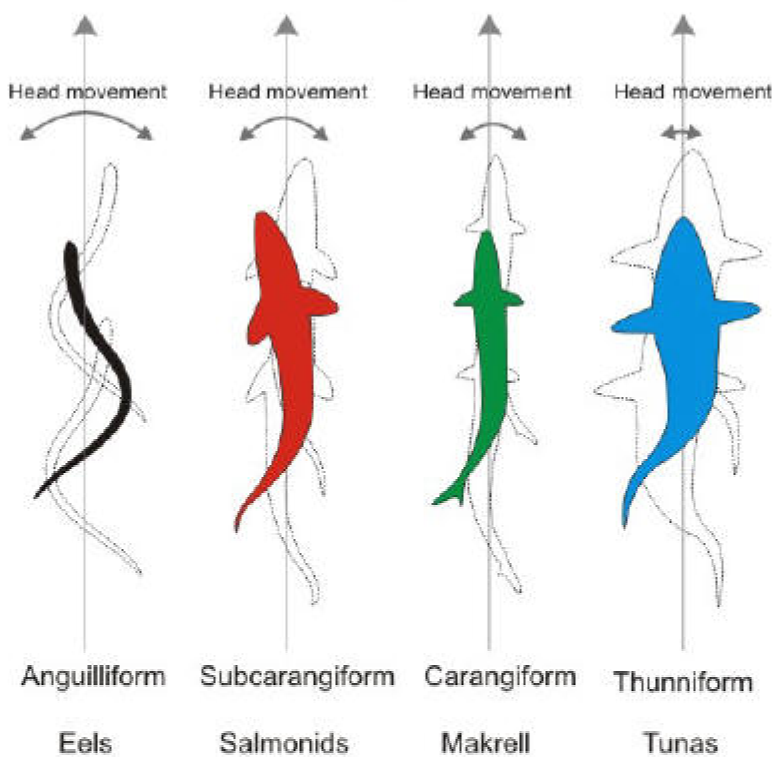


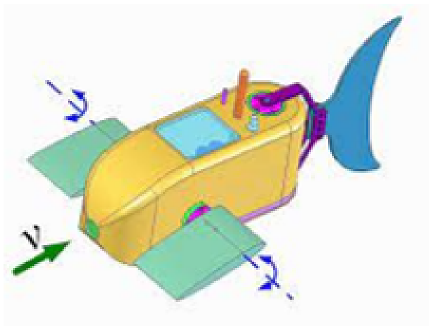
Figure 1.1: BCF swim style [1].

The most straightforward configuration of designing a robotic fish is to add a device that can generate flapping to make the robotic fish swim close to the category of thunniform. This type of design is often focused on a single-joint robotic fish because its flapping only has one DOF [14]. The fishtail could be a mechanism that converts a rotary motion to a reciprocating motion, or a servo system that precisely outputs a sinuous motion. A stiff flapping motion and low propulsive efficiency are the weaknesses of such design because most of the force is wasted in the lateral direction. The use of passive flexible fin helps improve efficiency.

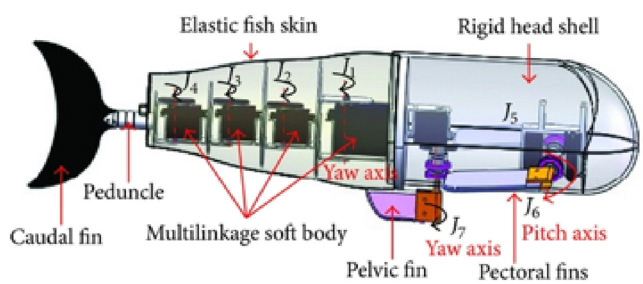
Carangiform and subcarangiform swimmers are characterized by a unique type of undulatory locomotion. During swimming, the peduncle actuates the caudal fin to form a rhythmic wave that passes from the head to the tail. Their spine can be described as a traveling wave governed by a sinusoidal function. Fish benefit from this undulatory locomotion to gain thrust [13] and reduce drag [15, 16]. The most effective way to mimic a carangiform fish is designing a multi-joint robotic fish that has multiple DOFs to imitate the undulatory locomotion [17]. In this layout, each joint of the robotic fish has a specific angle, and the overall robotic fish body curve matches the swimming curve. A typical multi-joint layout is the robotic fish adopting several servos assembled serially to approximate a sine-wave-like poly-line. The phase and amplitude of the flapping can be adjusted with an accurate synchronization between servos.

A special design other than single-joint and multi-joint is the robotic fish driven by a soft propulsion mechanism; their propulsion mechanism uses pneumatic/pump [18], or tendon [19] to generate tail flapping. In this design, the flapping kinematics is infinite dimensional. However, its flapping pattern, such as phase and amplitude, is unchangeable compared to the multi-joint robotic fish.

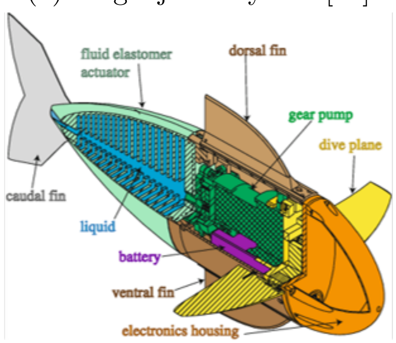
The third type of design is smart material actuated robotic fish. Their flapping is generated by the material's deformation. Materials such as shape-memory alloy[20], ionic polymer-metal composite (IPMC) [21], and piezoelectric materials [22] have been tested to form a propulsion mechanism. The design prospect of the smart material actuated robotic fish is to pursue the ultimate energy efficiency and compact size. The swimming speed is the major weakness of smart material actuated robotic fish.



(a) Single-joint layout. [14]



(b) Multi-joint layout [17]



(c) Soft robotic fish [18]

(d) Smart material fish [21]

Figure 1.2: Review of robotic fish design.

The servo-driven multi-joint layout is the mainstream design idea for robotic fish. It has the best performance in mimicking the real fish with variable flapping frequency and amplitude. It also has the best maneuverability proven by achieving special maneuvers, such as quick start and sharp turn. Numerous research topics have been studied using this type of robotic fish, such as propulsion efficiency, locomotion control, and special maneuvering. However, this type of design has a high energy consumption due to multiple actuators being used. An accurate synchronization between actuators is required, without which the propulsive efficiency could be jeopardized.

Overall, the idea that using multiple actuators to design a multi-joint robotic fish greatly increases the complexity of fabrication and control and reduces the superiority of robotic fish compared to AUVs. An AUV could be fully actuated using multiple actuators, but robotic fish is still under-actuated. Therefore, soft multi-joint robotic fish, propelled by a pump or tendon to realize infinite DOFs flapping, has received increasing attention in recent years.

Another trend in robotic fish design is a multi-joint design using one actuator. This design aims to make the most of the motor's speed to optimize the flapping frequency, so that the robotic fish's swimming speed is highly boosted. Some studies have demonstrated that a multi-joint robotic fish driven by a DC motor can swim at a very high-speed [23].

1.2.2 Review of robotic fish modelling

The kinematics of robotic fish could follow the model of the underwater robot, whose rigid body dynamics are established in the Cartesian coordinate system. More specifically, the positions are calculated in the inertial coordinate system, and the velocity is obtained in the body-fixed coordinate. The major challenge in modeling of robotic fish is calculating the force and acceleration governed by hydrodynamics. The thrust obtained by tail flapping during swimming is highly complicated by the hydrodynamic interaction between the fish body and the surrounding fluid.

When a fish displays significant undulatory movement, its caudal fin acts as an oscillating foil that sheds momentum into the wake as alternating thrust-type vortices for propulsion [13] (Fig. 1.3). It can be described as a two-DOF heave-pitch motion with a phase shift between the heave and the pitch. Substantial research examines the oscillating foil's hydrodynamic insight in generating force and momentum. They use a special yoke mechanism that outputs two-DOF flapping to measure the force of the oscillating foil at various frequencies and amplitudes. These studies bring the connections between the magnitude of force and flapping configuration. Magnitude of force may include thrust and drag, and flapping configuration may include frequency and phase. The particle image velocimetry (PIV) studies use high-speed cameras to analyze the flow motion around the fish body. They conclude that the fish's oscillating foil amplifies the thrust by enlarging the wake effort and strengthening vorticity [24]. Some hydrodynamics studies concludes that forward thrust is contributed by lift-based and added mass forces [24], and the angle-of-attack (AOA) plays a significant role in determining propulsive efficiency [25]. The scaling law is emphasized to identify the connection between the propulsion and oscillating foil with thrust coefficient such as Strouhal number [26, 27]. However, these studies are unable to propose an analytical method to calculate caudal fin propulsion due to the complex hydrodynamics such as vortex ring behavior.

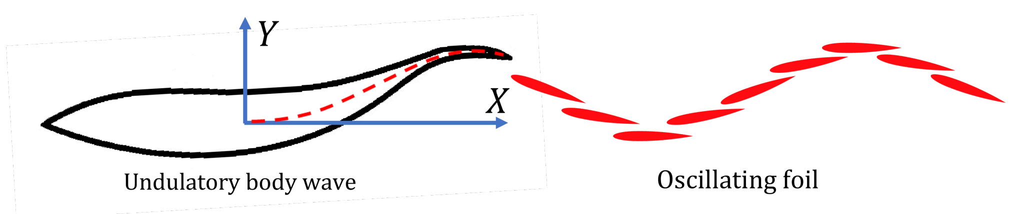


Figure 1.3: Undulatory locomotion.

In need of testing and controlling the robotic fish, researchers are investigating a comprehensive fish model that includes kinematics and hydrodynamics to estimate fish's acceleration and speed mathematically. Most of these models are inspired by the elongated body theory (EBT) proposed by Lighthill [28]. EBT captures the reactive force between the fish body and the surrounding fluids with an added-mass effect from the energy conversion aspect. Some assumptions from EBT simplify the model complexity so that the thrust can be calculated, as long as the shape of the fish body is known. Some studies also include the quasi-steady wing theory to derive a hydrodynamic lift force for flapping fin [29, 30]. This method is also very simple and intuitive, but it ignores the tail's deformation caused by body-flow interaction. In total, most current models include more or less model reductions in both hydrodynamics and kinetic aspect. The fish model's key points have been concentrated on several elements, such as forward swimming speed, tail flapping frequency, and AOA.

The data-driven empirical models have also been investigated. They borrow the ideas from either neutral network or linear regression to skip the complicated hydrodynamics. Even though these models may have limitations that only validate a specific design or circumstance, they still perform promising outcomes in optimization and controller design [31].

1.2.3 Review of robotic fish two-dimensional control

The control of robotic fish includes three levels of control task: locomotion control, maneuvering control, and mission-level control. The locomotion control's objective is to achieve high propulsion efficiency. For example, in multi-joint robotic fish, locomotion control is necessary to coordinate each actuator's work following a specific order [32, 33]. The locomotion control is the foundation of a robotic fish's operation; it ensures the tail flapping generate effective thrust to maneuver.

The second level is the maneuvering control that navigates the robotic fish through a specific path or reaches a destination. Existing steering controls for robotic fish have been mainly focused on error-based proportionalintegralderivative (PID) control [34, 35, 36] and active disturbance rejection control[37]. The PID control can skip the complex hydrodynamic physics to control the steering and speed using feedback and error. The model-based nonlinear control, using simplified dynamic models or identified empirical models, can optimize the control performance in making the best use of energy. The empirical model, obtained from input-output identification, has been implemented to develop adaptive control [38] and sliding-mode control [39] . Castano *et al.*[40, 41] developed a model predictive control and a back-stepping control based on a nonlinear model, which is a simplified version of EBT hydrodynamic model [42]. A linear quadratic regulator yaw control was developed for a three-dimensional robotic fish [43]. However, experimental validation for this study was not provided.

With a well-developed low-level control, the ultimate goal is to control the robotic fish to interact with environment and other robots in a certain application, which is the third level control of the robotic fish. Various constraints such as energy and communication, require the robotic to be operated with smart path planning and condition monitoring. For example, path planning aims to find an optimal path that navigates the robotic fish to the destination with the minimum usage of energy or time. Swarm control [37, 44] and collision avoidance [45, 14] focus on control of multiple robotic fish to safely cooperate with each other in a bounded area.

In recent years, the attention of robotic fish has gradually shifted to application. The robotic fish could be deployed in large numbers and could work for a long time because of their low-cost and high-efficiency characteristics. Inspired by the application of drones in agriculture and environment monitoring, robotic fish can be widely used in information collection and environmental monitoring. Researchers from Massachusetts Institute of Technology use robots to observe underwater ecology [46]. Researchers from New York University let the robotic fish swim with real zebrafish, so that they could study their behaviors [47]. Researchers from Michigan State University build robotic fish as a part of underwater communication network [48]. The above examples illustrate that the robotic fish can be used as a remote intelligent platform to provide information and data support for the offshore industry in many categories.

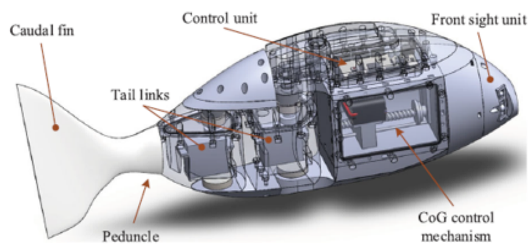
1.2.4 Review of robotic fish depth control

Most current robotic fish research focuses on optimizing robots' two-dimensional (2D) maneuvering capabilities. However, three-dimensional (3D) maneuverability is necessary not only for robotic fish but also for all other underwater robots. In nature, fish use two sets of organs to adjust depth: gills and bladders are used to change their density, and fins and tails to generate thrust in the vertical direction. Inspired by these mechanisms, researchers developed two types of depth control methods to inherit the 3D maneuvering capability for the underwater robot.

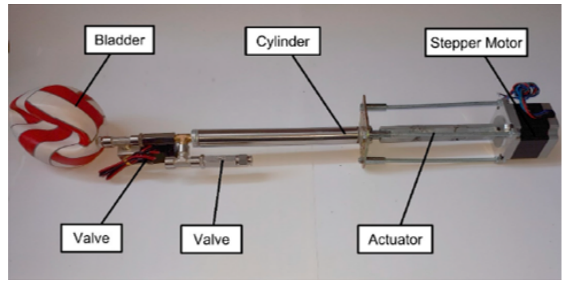
The first method is dynamic depth control. In this method, the robot is initially assumed to be neutrally buoyant or nearly neutrally buoyant, then the depth is changed by vertical thrust obtained from actuators. The actuator can be a vertically installed propeller so that the robot can directly maneuvers up and down [49, 50]. This design is very straightforward and effective, but it is not ideal for robotic fish whose main purpose is to avoid using propellers. The alternative approach is adding a rudder [51] to borrow some horizontal thrust to the vertical direction, or a specific center mass device to control the robot's altitude orientation [52, 53, 46]. The main idea for this approach is changing the pitch angle to borrow part of the thrust from the horizontal direction to the vertical direction. Studies show that the dynamic depth control is very effective and reliable to adjust depth, but it is unable to adapt to the density change of the surrounding fluid. This means if the robot's internal density changes, such as leaking or unloading, the actuators have to keep consuming power to maintain depth.

The second method is buoyancy control. In this method, depth is changed by controlling the volume of the robot. In theory, the buoyancy acting on a submerged object is proportional to fluid displacement volume. The robot needs to reduce its volume to sink and gain volume to float. One way to do this is using a motorized piston or pump to push out and suck in water into an internal chamber to change the overall robot's volume. This method is very quick and highly sensitive. However, pushing water out of a fluid-filled chamber will result in pressure dropping and temporary vacuuming, which bring a strict standard requirement for robot's structure design and a high volume power source to power the piston or pump [54, 55, 56]. Another way is using compressed air or compressible fluid to replace the volume occupied by water. This method is widely adopted in submarines and many underwater robots [57, 58], in which the water will be pushed out by the compressed air stored on-board. The compressible fluid method needs an artificial bladder that could expand when the compressible fluid is heated up or shrink when the fluid is cooling down [59, 60]. However, due to space constraints, adding a compressed air tank or fluid tank into a small robotic fish is difficult.

The most effective way to control depth is combining both the dynamic method and buoyancy method. Every individual method has its own pros and cons. When both methods are combined, they can be used to overcome each other's limitations. The buoyancy depth control method can be used to overcome the density change between the robot and surrounding water when the robot changes its load. It saves the power consumed in maintaining depth using propellers in the long run. Consequently,



(a) dynamic depth control [61]



(b) buoyancy control. [62]

Figure 1.4: Review of robotic fish depth control.

the dynamic depth control method has a much faster response, allowing the robot to go to its desired depth sooner, while the buoyancy control method slowly compensates for the density differences.

For the robotic fish, the current existing buoyancy methods are not ideal due to the compact size and limited energy storage. A viable solution to deploy the buoyancy method is to produce the required gas on-board by using electrolysis to convert the surrounding water into hydrogen and oxygen gases. In 2003, a micro robotic fish developed by Nagoya University brought the idea of adjusting the depth by electrolysis of water [63]. The fish were utilizing an ionic conducting polymer film actuator which can electrolyze the water when the frequency of the applied voltage is at a specific range. Cameron et al. [64] patented the concept of using water electrolysis for the buoyancy control device (BCD). Um et al. [65] built a prototype of the buoyancy device equipped with an IPMC electrolyzer. They showed that depth control using electrolysis of water is achievable through an open-loop test. Chen et al. [66] modeled the buoyancy device's dynamics and developed the controller for depth control. Keow et al. [3] improved the design of the buoyancy device and realized the depth control in real-time, successfully positioning and maintaining the device at the desired depth.

1.3 Contributions of this Dissertation

1.3.1 Robotic fish design using a novel Double-Slider-Crank mechanism

This work summarizes the advantages and disadvantages of single-joint robotic fish and multi-joint robotic fish and explores the benefit of adopting only one DC motor as the main actuator for a multi-joint robotic fish. A novel propulsion mechanism, Double-Slider-Crank (DSC) that uses one DC motor to drive two joints, is designed for a three-joint robotic fish. The mechanism converts a rotary motion into two flapping motions with a constant phase shift to realize oscillating foil flapping. It is inspired by large ocean fish, such as tuna and sharks, who make small turns by producing a bent at the peduncle's anterior while the peduncle and tail keep oscillating to produce thrust. A servo motor is added in the first joint to direct the thrust generated by DSC.

The DSC is designed in a compact size to act as peduncle and tail in this robotic fish. The servo motor outputs a constant angle and does not participate in the flapping. The servo motor and DSC form a hybrid propulsion system. This design reduces the recoil of the fish head by moving the flapping motion of the caudal fin far away from the robot's mass center. It is worth noticing that using one DC motor could restrict undulatory locomotion into a fixed pattern: the amplitude of each joint and the phase shift between two joints are unchangeable. Swimming with a fixed pattern is acceptable because the real fish normally use a fixed pattern in straight swimming. The restricted undulatory locomotion is a trade-off to pursue fast and stable forward swimming.

In experiments, the robotic fish exhibit extraordinary stability and maneuverability. The DSC flaps like a real fish, and the robotic fish can achieve a maximum swimming speed of 0.35 m/s (0.98 BL/s) at a 3 Hz flapping frequency, which is impressive for a compact size robotic fish.

1.3.2 Linear model of the robotic fish using Nomoto model

This work explores the feasibility of modeling the robotic fish from a marine vessel perspective. The robotic fish developed in previous work has a hybrid propulsion system in which the robotic fish can be classified into a marine vessel; the speed and steering are independently controlled by the DC motor and the servo motor, respectively. When deriving the dynamic model, the hydrodynamic model is calculated in a third sub-coordinate, which is the coordinate of DSC. Since the actuation pattern of DSC is fixed, its hydrodynamic model is relatively unchanged. An assumption is proposed that the thrust acquired by the DSC is slightly affected by the servo's angle, therefore the thrust derivation is decoupled from the robotic fish's kinematics.

Besides, a steering model expressed by a transfer function is investigated. Previously established linear models for steering of the robotic fish lack physical insight and only consider simple maneuvering [67, 68]. The proposed model employs the Nomoto model [69], a well-developed perturbation model for vessel control, to predict fish's steering. It is obtained from a system identification approach for various simulated maneuverings. The identified model is tested in an observed-state feedback controller to investigate its effectiveness. The results of the model identification and validation, as well as a control experiment, are presented.

1.3.3 Depth control of the robotic fish using soft actuator

This work addresses the challenge of depth control for a robotic fish. We develop a compact 3D maneuverable robotic fish. The fish adjusts the buoyancy by using an IPMC water electrolyzer to generate hydrogen and oxygen gases on board. When positive buoyancy is needed, the hydrogen and oxygen gases are stored inside the fish body, which increases the robot's volume. To reduce buoyancy, the gases are released by a solenoid valve to surrender volume. The robotic fish also consists of a servo-driven tail to achieve 2D maneuvering. A 3D dynamic model is developed to capture the robotic fish's dynamics with buoyancy changes. Both experiments and simulations have been conducted to validate the proposed robotic fish design and model. This is the first time to demonstrate a 3D motion performed by a compact robotic fish equipped with the water electrolyzer.

The model of the BCD is carefully examined to ensure its controllability. Considering the gas generation rate is limited by the physical characteristics of the electrolyzer, an optimal trajectory is developed for the BCD with consideration of control constrain and state constrain. This trajectory is obtained by using a virtual linear model to solve a bang-bang-off-bang-bang minimum-time optimal control problem. Simulation results have shown that the BCD is stable during the tracking, and the control input and state are staying bounded within the allowable ranges.

1.3.4 Event-triggered-control of robotic fish in inconsistent communication

We have conducted two studies on robotics applications. The first work addresses the challenges of communication in deploying robotic fish. One of the most common implementations of robotic fish is to form a sensing network for environmental monitoring, such as oil leak detection and carbon dioxide monitoring. Among the existing marine robots for environmental monitoring, the bio-inspired robotic fish has the advantage of better maneuvering capabilities and less impact on marine ecosystems [70, 71]. However, a challenge is raised when deploying robotic fish in a long-range and multi-agent task. The operation of a multi-agent control system requires frequent communications for path planning, collision avoidance, and data uploading [72, 45]. Hence, establishing a robot-involved monitoring system has a high requirement in communication resources. The most current control algorithms require the robot to be operated under a constant sampling period, and all robots need to be well synchronized with a remote controller to exchange information. Thus, communication failures such as delays and packet losses are unavoidable when intensive information flooding into the network [73, 74]. In this work, we address this real-world challenge that a communication network with limited bandwidth could reduce the performance of robotic fish; and a failure to maintain the constant-sampling rate (CSR) could significantly undermine the robotic fish's maneuvering, even leading to damage. We develop a model-based event-triggered-control (ETC) to control the robotic fish's steering with fewer communication times between the remote controller and the robotic fish. The robotic fish is tested in this configuration to reduce communication usage and maintain performance when CSR fails.

1.3.5 Robotic fish application in motion tomography

Ocean-current plays an important role in the operation of marine vehicles [75]. When a marine vehicle, especially an underwater vehicle, is operated in a global positioning system (GPS) denied environment, it relies on inertial sensors to track its position. The ocean-current induced flow, which will not be captured by the inertial unit, would lead to a tremendous error in position estimation [76]. On the other hand, ocean current impacts marine vehicles' energy efficiency. When the vehicle's heading is along the direction of the flow, the vehicle's energy consumption is at the least, and the control of the vehicle is light. When the vehicle is traveling against the flow, the vehicle needs to consume extra energy to cancel the movement caused by the current. Considering the vehicle can only carry a limited amount of energy, ocean-current brings an extra burden to vehicles. Overall, the perception of the flow field could greatly help the vehicles in path planning[77] and improve energy efficiency.

Traditional ocean-current observations rely on information obtained through buoys array and satellites. It is expensive and time-inefficient. An interest in using the position and velocity information of the vehicles to predict the ocean current is growing by developing ocean-current observers for marine vehicles [78, 79]. However, their sensing area is limited to where the vehicle passes, and a constant current is assumed in these cases. Motion tomography is a technique that uses the vehicle's navigation information to estimate the flow field [80, 81]. By collecting the velocity and position information from multiple vehicles, it can estimate the flow field of a larger area. This technology provides a time-efficient and convenient way to monitor ocean currents.

One limitation of motion tomography is that it ignores the vehicle's rigid-body dynamics by considering the vehicle's model as a first-order particle. In the real application, the flow-induced forces and moments are non-neglectable, especially at a small flow-vehicle scale. We introduce an active heading control to enable the robotic fish to offset the error caused by flow-induced force. The simulation analysis is provided, proving that using robotic fish can obtain an appropriate integration error that is reasonable for motion tomography. Proposed controllers are tested in an experiment, and the obtained robotic fish trajectories lead to a good estimation of the flow field.

1.4 Dissertation Organization

The rest of this dissertation is organized as follows: Chapter 2 introduces the design and fabrication of the robotic fish; Chapter 3 introduces dynamic modeling and model reduction; Chapter 4 discusses the model-based control design, including event-trigger control; The depth control device and 3D robotic fish are discussed in Chapter 5; In Chapter 6, the application of robotic fish in motion tomography is discussed; Conclusion and future work are discussed in Chapter 7.

Chapter 2

Robotic Fish Design

This chapter presents a robotic fish propelled by a double-slider-crank (DSC) driven fishtail. The DSC is a mechanism that can convert DC motor's rotation to two-joint flapping. This character can be used to design a multi-joint robotic fish that uses one DC motor to achieve undulatory locomotion. The design of DSC is guided by a traveling wave equation that describes the flapping of real fish. After multiple tests, the DSC has been proven to be an efficient propulsion mechanism for robotic fish. The proposed robotic fish has a good performance in mimicking real fish swimming and boosting swimming speed. By adding a servo motor to form a hybrid propulsion system, the robotic fish can achieve two-dimensional free swimming. The experimental result demonstrates that the robot can achieve a maximum 0.98 body-length (BL) per second forward swimming speed.

2.1 Oscillating Foil

Fish obtain thrust through body flapping. Thus, the key in designing a robotic fish is to make an actuator to generate a reasonable flapping motion. When fish is swimming, its wave-like body shape can be described as a traveling wave equation,

$$y(x,t) = (c_1 x + c_2 x^2)[\sin(kx + \omega t)]. \tag{2.1}$$

Where y is lateral displacement, x is forward displacement, c_1 and c_2 are the quadratic parameter, k is the body wave number, and ω is the angular frequency. When designing a three-joint robotic fish (Fig. 2.1), one can discretize this wave into $i = 1, 2, 3$ segments at length l_i in the body-fixed coordinate, with origin $[x_0, y_0]$ at the mass center. The location of each joints is described as follows:

$$\begin{aligned} y_i(x_i, t) &= (c_1 x_i + c_2 x_i^2)[\sin(kx_i + \omega t)], \\ l_i &= \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}. \end{aligned} \tag{2.2}$$

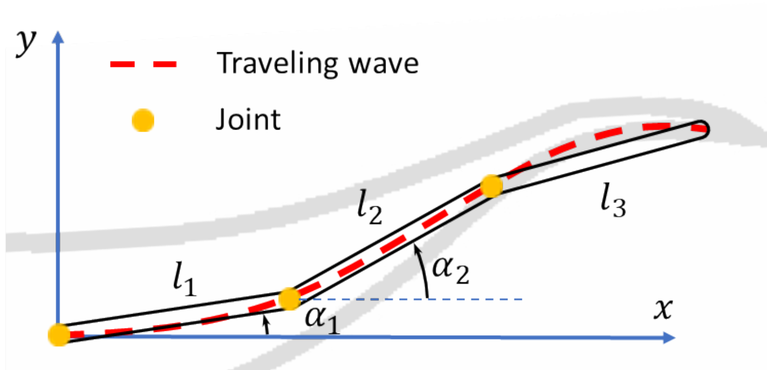


Figure 2.1: An illustration of a discretized traveling wave with three joints.

Let's denote α_i to be the absolute joint angle of i-th segment with respect to the x-axis. For undulatory locomotion, α_i can be written as a sinusoidal function with an amplitude A_i and frequency ω,

$$\alpha_i(t) = A_i \sin(\omega t + \phi_i), i \geq 1, \tag{2.3}$$

where ϕ_i is a constant phase shaft [82].

The undulatory locomotion makes the fishtail to flap like an oscillating foil. A robotic fish needs at least two joints to realize an oscillating foil. The first joint's flapping can inherit a heave motion to the second joint's pitch motion, so that the tail flapping is a motion of 2 degrees of freedom (DOFs). An illustration of oscillating foil achieved by a two-joint kinematics is shown in Fig. 2.2. The kinematics has a first joint angle $\alpha_1 = A_1 \sin(\omega t), \phi_1 = 0$. The first joint does not need large amplitude, so A_1 is small and α_1 varies at a small range. Thus, the second joint's horizontal motion can be neglected and the lateral displacement of the second joint can be approximated by $y \approx l_1 \sin \alpha_1$. The second joint has the angle $\alpha_2 = A_2 \sin(\omega t + \phi_2)$. The pitch and heave motions can be represented as the time derivative of the second joint's lateral displacement and the time derivative of α_2

$$\begin{aligned}
&\text{Pitch: } \dot{\alpha}_2 = \omega A_2 \cos(\omega t + \phi_2), \\
&\text{Heave: } \dot{y} = \frac{\mathrm{d}}{\mathrm{d}t} l_1 \sin[A_1 \sin(\omega t)] = \omega l_1 A_1' \cos(\omega t).
\end{aligned}$$

The A_1' represents the amplitude of $\sin[A_1 \sin(\omega t)]$, at $0 < A_1 \leq 1$. It is noted that ϕ_2 produces the phase shift between two joints.

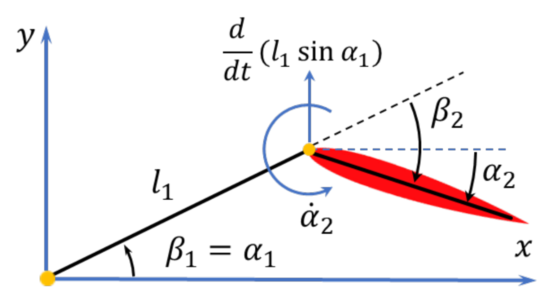


Figure 2.2: Kinematics of a three joint mechanism.

Define $\beta_2 = \alpha_2 - \alpha_1$ as the angle increment between two joints, named relative angle. The β_2 can also be represented by a sinusoidal function

$$\beta_2(t) = B_2 \sin(\omega t + \Psi_2), \tag{2.4}$$

where B_2 is the amplitude decided by A_1 and A_2. The phase Ψ_2 is

$$\Psi_2 = \arctan \frac{A_2 \sin \phi_2}{A_2 \cos \phi_2 - A_1}. \tag{2.5}$$

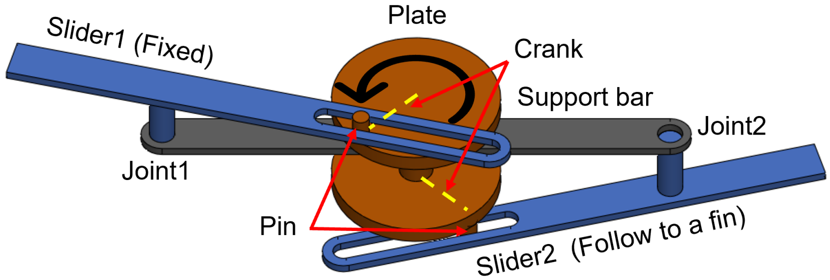
Eq. 2.5 concludes that when a two-joint robotic fish is swimming with its tail flaps like an oscillating foil, its flapping amplitude A_1 and A_2 are unchanged, and the tail's relative angle has a constant phase shift.

2.2 Double Slider Crank Mechanism

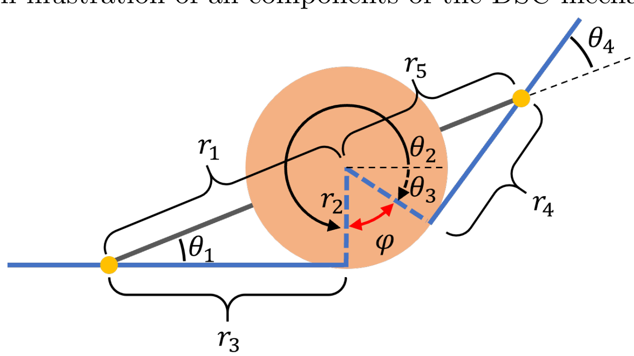
The design of DSC uses the conclusion of Eq. 2.5 to construct a constant phase shift between the first and second joints relative angle. As shown in Fig. 2.3a, it includes two slider-crank mechanisms driven by a pair of rotatory uni-axial plates. Each slider-crank mechanism has a slider that is pin-slot mated with a double-layer plate, which is driven by a DC motor. The mechanism diagram of DSC is shown in Fig. 2.3b. The sliders are marked in a solid line and their sliding displacements are marked as r_3 and r_4. The rotatory plate has a radius r_2, and the crank is marked in dashed lines. The support bar's length is a sum of r_1 and r_5. The joints are marked by yellow circles. θ_1 is the relative angle of the first joint and θ_4 is the relative angle of the second joint. φ is the angle difference between two cranks. The kinematics of

DSC are described in the following equations,

$$
\begin{aligned}
r_3 &= \sqrt{r_1 - (r_2 \sin\theta_2)^2} + r_2 \cos\theta_2, \\
\theta_1 &= \arctan\left(\frac{-r_2 \sin\theta_2}{r_3 - r_2 \cos\theta_2}\right), \\
\theta_3 &= \theta_2 + \varphi, \\
\theta_4 &= \arctan\left(\frac{r_2 \sin\theta_3 - r_5 \sin\theta_1}{r_2 \cos\theta_3 - r_5 \cos\theta_1}\right) - \theta_1.
\end{aligned}
\tag{2.6}
$$



(a) An illustration of all components of the DSC mechanism.



(b) 2D kinematic diagram of DSC mechanism

Figure 2.3: An illustration of DSC mechanism.

When using the DSC to realize the oscillating foil, the ‘Slider1’ is connected to the fish body. The plates and support bar are the first segment that oscillates about ‘Joint1’. While ‘Slider2’ connects to the caudal fin, and they form the second segments. At this circumstance, the $\theta_1 = \beta_1$ and $\theta_4 = \beta_2$ if one considers that the DSC is a two-joint mechanism similar to Fig. 2.2. When the rotatory plate is spinning at a consent frequency, β_1 and β_2 can be approximated as

$$\begin{aligned}\beta_1(t) &= \arcsin \frac{r_2}{r_1} \sin(\omega t),\\ \beta_2(t) &= \arcsin \frac{r_2}{r_5} \sin(\omega t + \varphi),\end{aligned} \tag{2.7}$$

where φ denotes a constant phase shift. The β_1 and β_2 are numerically plotted in Fig. 2.4, where the solid line indicates the β_i simulated by Eq. 2.6 by considering $\theta_1 = \beta_1$ and $\theta_4 = \beta_2$. The dashed line indicates the β_i simulated by Eq. 2.7. One can conclude from the plot that there is a phase shift between β_2 and β_1. Noticing that r_5 is shorter than r_1 so that the amplitude of 'Joint2' is larger than 'Joint1'. However, a shortened distance between the joint and plate axis could distort the sinusoidal wave's shape due to the forward stroke being unbalanced to the return stroke. To moderate the distortion, the amplitude of oscillating is bounded within 0.5 rad.

The phase shift φ between the DSC's two relative angles is determined by the angle difference between two cranks, where φ is shown in Fig. 2.3b. φ can be selected numerically to ensure that the phase shift in Eq. 2.3 is close to $\phi_2 = 0.5\pi$, which has been concluded as an optimal phase shift for propulsive efficiency of oscillating foil [83, 84].

2.3 Hybrid Propulsion System

DSC can use one motor to achieve oscillation foil. However, its oscillating amplitude and phase are fixed so that the fish's turning ability is limited. To overcome this limitation, a servo motor is added in front of the DSC to enhance the turning capability so that the servo motor and the DSC form a hybrid propulsion system;

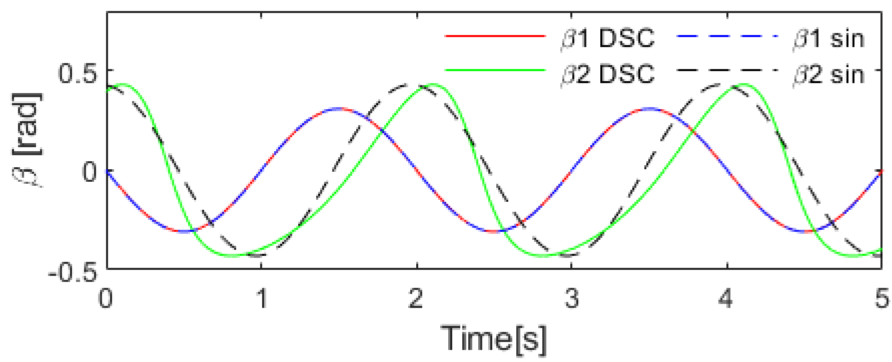


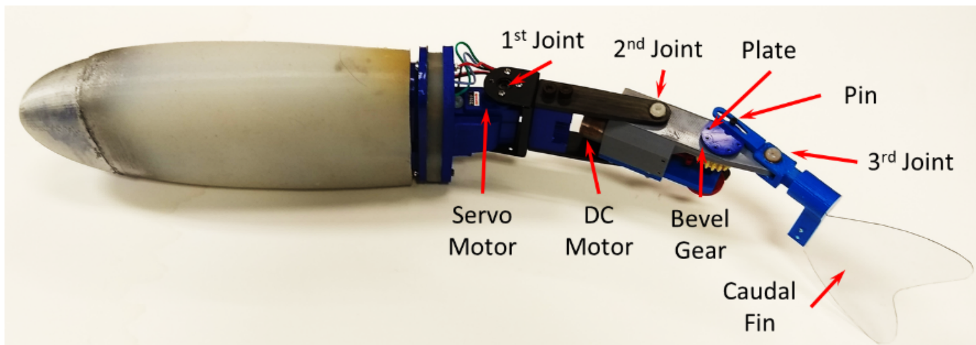
Figure 2.4: The plot of β_1 and β_2 with approximated sin wave.

the servo motor actuates the first joint, and the DSC actuates the second and third joints, as shown in Fig. 2.5a. The operation of this hybrid propulsion system has two modes; the 'two-joint flapping' mode indicates the servo motor does not participate in flapping. The robotic fish only use the DSC to gain thrust, and the servo motor is held at a constant angle. The 'three-joint flapping' mode indicates both servo motor and the DSC participate in the flapping with the following manner,

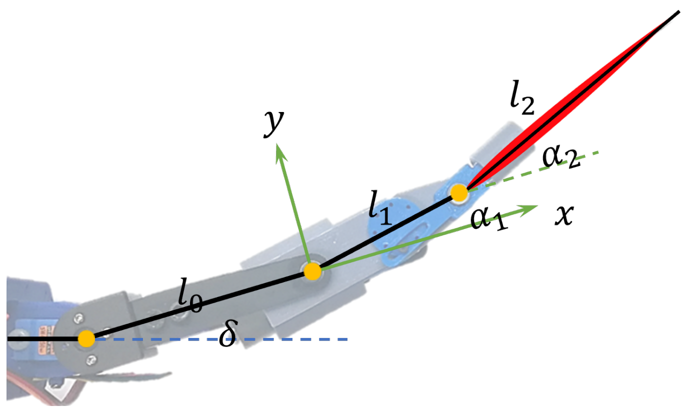
$$\delta(t) = B_1 \sin(\omega t - \varphi'), \tag{2.8}$$

where δ represents the output angle of servo motor, as shown in Fig. 2.5b. φ' represents the reverse phase shift and $\varphi' \leq \varphi$. B_1 is a constant coefficient with the condition that $B_1 < \arcsin \frac{r_2}{r_1}$. Under this mode, the servo motor flaps at the same frequency as the DSC with a constant phase shift. An in-air demonstration of undulatory locomotion achieved in 'three-joint flapping' mode is shown in Fig. 2.6. From the snapshot, one can clearly observe that the servo motor lead the flapping of DSC.

The 'three-joint flapping' generates a flapping with a larger wake area. It can boost up the speed compared to the 'two-joint flapping' mode. However, the 'two-joint flapping' is a more favorable mode. Holding the servo motor at a constant angle



(a) Hybrid propulsion system in a robotic fish.



(b) Top view of the hybrid propulsion system.

Figure 2.5: The illustration of hybrid propulsion system.

benefits the robotic fish in two aspects: on one hand, since the servo motor is closer to the fish's center of mass, its oscillation could aggravate the fish's head movements, on the other hand, resting the servo motor can save power.

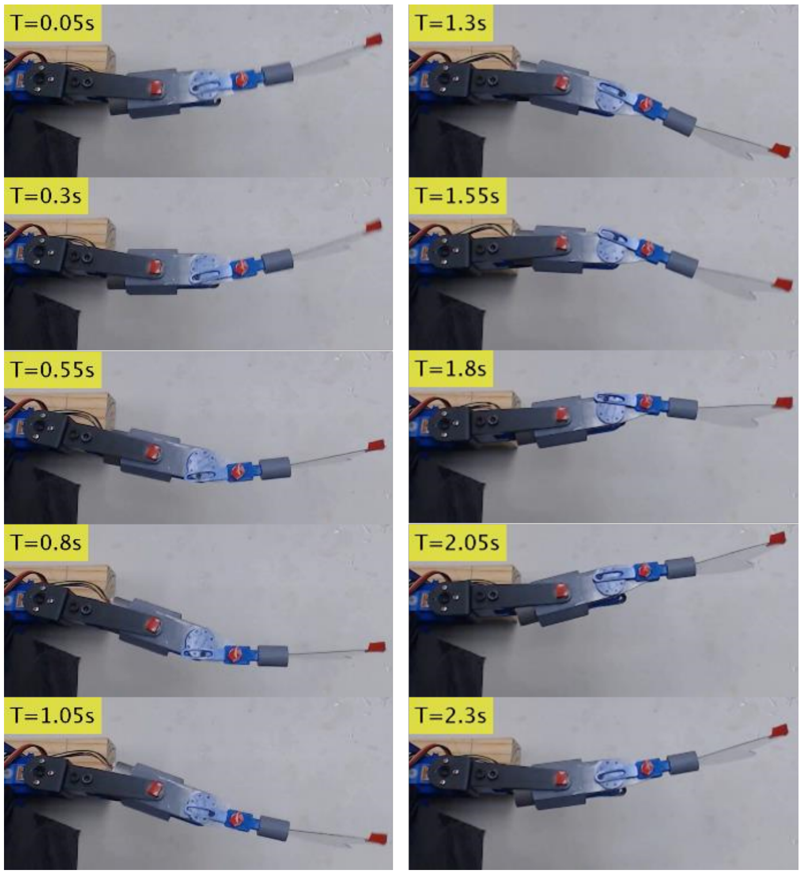


Figure 2.6: Snapshots of the 'three-joint flapping' in an air test.

2.4 Robotic Fish

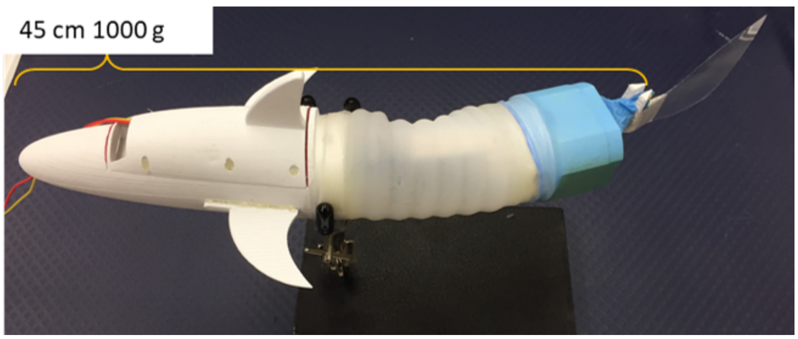
There are three versions of robotic fish that have been developed. The first version, in Fig. 2.7, was developed to verify the effectiveness of the DSC. This robotic fish uses a 12 V DC motor and a 6V servo motor. It could achieve maximum 0.48 BL/s at 3

Hz flapping frequency. The second version, in the bottom of Fig. 2.7a, was developed to analyze the swimming speed between 'three-joint flapping' mode and 'two-joint flapping' at low frequency. This robotic fish has a 12 V DC motor (FIT0441) with an encoder. The encoder reads the DC motor's angle position for the servo motor to catch up the DSC's flapping frequency and phase. The third version, in the top of Fig. 2.7b, was developed for application study. This robotic fish does not equip with a motor encoder and it was used to measure the maximum speed at 'two-joint flapping' mode. The robotic fish's head and skeleton were printed by a 3D printer (Ultimaker 3). The fish body was made of silicon rubber for waterproof purpose. The DSC mechanism was fabricated by a CNC machine (Roland MDX540). The caudal fin was made of high-density polyethylene terephthalate, which has relatively high tensile (around 60 MPa). A pair of side fins were added to prevent rolling. The control circuits consisted of a micro-controller, which translates the wireless command into a corresponding pulse width modulation (PWM), and a Li-ion battery. The control command was sent from a computer through Wi-Fi.

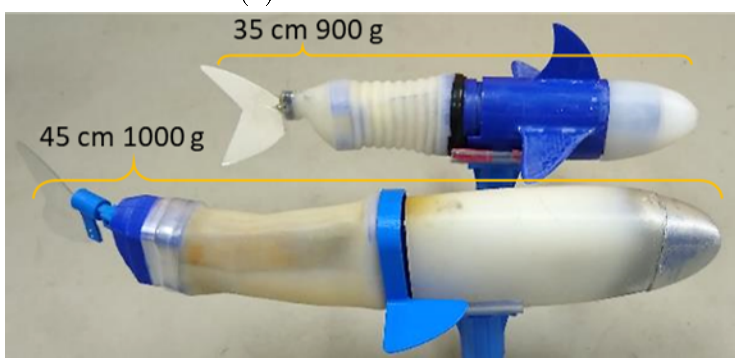
2.5 Swimming Test

In experiments, the robotic fish swam in a 2.4 m $\times$ 1.6 m pool. The robot's position was obtained through computer vision using a camera installed 2.4 m above the water surface. Fish's heading angle was derived from a series of positions to reduce the heading angle error caused by the lateral recoil movements. The camera acquisition frequency was 10 Hz, and the control signal was transmitted at a frequency of 2 Hz.

The 'three-joint flapping' and 'two-joint flapping' are compared in forwarding



(a) The first version.



(b) The second version (bottom) and third version (top).

Figure 2.7: Robotic fish

speed. Since the robotic fish need time to achieve a stable speed, the measurement counted the speed at the last two seconds in each test at the specific flapping frequency. The result is shown in Fig. 2.8. In 'three-joint flapping' mode, the microcontroller reads the DC motor's rotary speed through the encoder to calculate the flapping frequency. That frequency was translated to a sinusoidal equation to guide the servo motor's flapping. The DC motor's speed of the larger robotic fish was limited so that the servo motor can catch up the DC motor's flapping frequency. At a frequency range from 0.5 Hz to 1 Hz, the surge velocity of the 'three-joint flapping' mode in each frequency was about 50% higher than its speed in 'two-joint flapping' mode. Then maximum velocity was 0.178 m/s at 1 Hz, which was 0.4 BL/s.

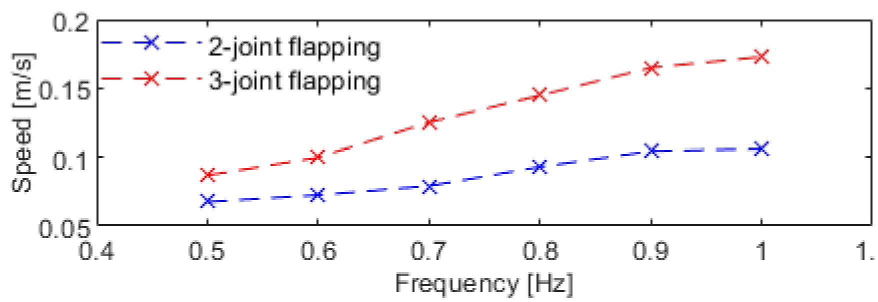


Figure 2.8: Experimental result of forward speed in 'three-joint flapping' and 'two-joint flapping' mode.

The fish of the third version was tested at a higher frequency range (from 1.8 Hz to 3 Hz) for top speed and steering. The robotic fish accelerated from stationary at a constant frequency and zero bias angle in the forward swimming test. The maximum surge speed was 0.35 m/s, which was equivalents to 0.98 BL/s and achieved at 3 Hz. As shown in Fig. 2.9, the fish swam about 2 m in 6 s. Due to the pool size, the robotic fish was only allowed to free swim in 6 s. The achievable maximum speed was believed higher than the current result.

In the steering experiment, the fish of the third version is tested from 1.8 Hz to

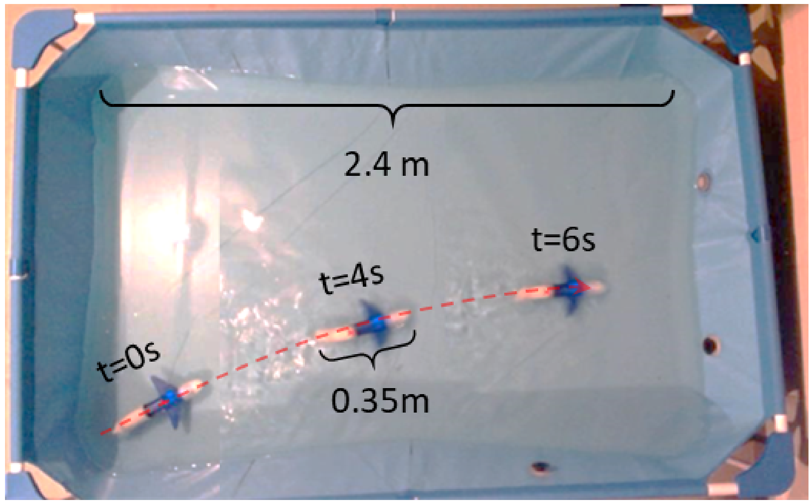
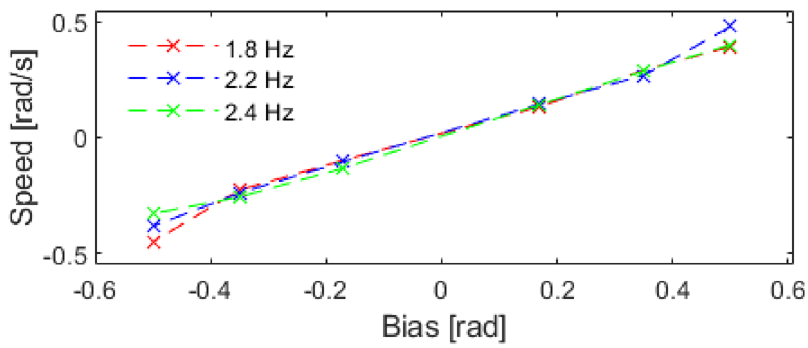
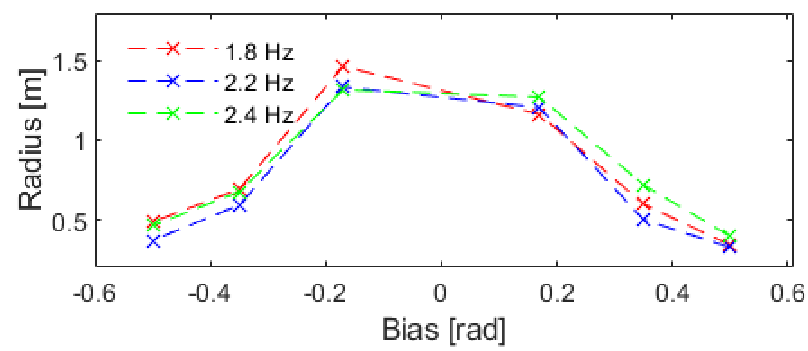


Figure 2.9: Experimental snapshot of robotic fish swims at 3 Hz.

2.4 Hz, to maximize the swimming time. When the robotic fish is steering, the servo motor kept a constant bias angle and the DSC works at a constant frequency. The bias angle was assigned from -0.5 rad to 0.5 rad to test both left turning and right turning. Robotic fish's steering performance was evaluated by turning radius and turning rate. The turning radius was denoted by fitting a circle to the trajectory of the latest 3/4 of the testing period. The turning rate was derived by calculating the heading angle changes over this period. The robot's left and right turns cannot be perfectly symmetrical because of the thrust from the first several flapping in the experiment was not symmetrical. In Fig. 2.10a, the turning radius gradually decreases when the bias angle is increasing. The turning rate, in Fig.2.10b, is almost linearly increased by bias angle. It is worth noticing that the flapping frequency has a very small impact on the radius, which is consistent with the study in [85]. The maximum turning rate was 0.48 rad/s at a 0.5 rad bias angle, and the corresponding turning radius was 0.37 m.



(a) Steering speed versus bias.



(b) Steering radius versus bias.

Figure 2.10: Experimental result of steering.

2.6 Chapter Summary

In this chapter, we introduce how the DSC mechanism is designed by following the traveling wave equation. The DSC mechanism has a phase shift φ between its two cranks so that it can realize a flapping in two DOFs, which makes the fishtail flap like an oscillating foil. In robotic fish design, a servo motor is added in front of the DSC to control thrust direction. Servo motor and DSC form a hybrid propulsion system that enables robotic fish to swim on the water surface freely. The proposed robotic fish is tested in experiments to examine its swimming speed and steering capability. Overall, the proposed robotic fish propelled by the DSC mechanism demonstrates strong maneuverability. In an experimental examination, the robotic fish can achieve a maximum swimming speed of 0.35 m/s (0.98 BL/s).

Chapter 3

Robotic Fish Modeling

The proposed robotic fish has a hybrid propulsion system to realize oscillating foil flapping. The use of the hybrid propulsion system benefits the modeling work by simplifying the flapping pattern and moderating the head shaking. In this chapter, the dynamic model of robotic fish's two-dimensional maneuvering is discussed. A vectorized elongated body theory (BET) is used to calculate the thrust, while considering a normalized forward speed to decouple the thrust derivation from the robot's kinematics. The hybrid propulsion system also allows the robotic fish to be modelled from the vessel's perspective. The vectorized EBT model is reducted by inheriting the ideal of the Namoto model to reflect robotic fish's steering in terms of actuation. The obtained linear model is validated in both simulation and experiment, and it is further tested in designing a heading angle control using in air controllers.

3.1 Kinematics

The model of robotic fish is built in a rigid-body system with action of external forces and momentum. The kinematics of robotic fish is described in the Cartesian coordinate system in 6 DOFs. As shown in Fig. 3.1, the position $[X, Y, Z]^T$ and orientation $[\phi, \theta, \psi]^T$ of robotic fish is analyzed in an inertial coordinate, whose origin is defined by O. Respectively, the linear velocity $[u, v, w]^T$ and angular velocity $[p, q, r]^T$ is analyzed in the body-fixed coordinate, whose origin is defined by G.

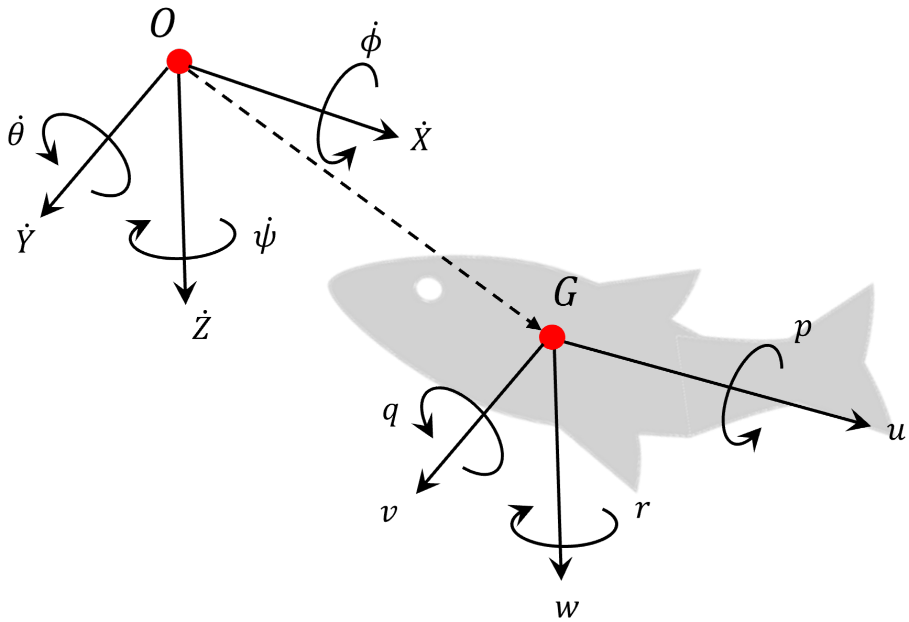


Figure 3.1: Coordinates system.

The velocity are transferred from the body coordinate to the inertial coordinate using the following equations:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = J(\phi, \theta, \psi) \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \tag{3.1}$$

where the $J(\phi, \theta, \psi)$ is the coordinate transformation matrix.

Several assumptions are introduced to simplify the modeling process of two-dimensional maneuvering. Firstly, the pitch motion q and roll motion p are set constantly to be zero, because the pitch angle θ and roll angle ϕ are uncontrollable. Secondly, the origin of the body-fixed coordinate coincides with the center mass of robotic fish, hence the steering of the robotic fish is respect to the z-axis in body-fixed coordinate. Finally, the fluid around the fish is considered inviscid so that the viscous resistance on the fish body is ignored. Construct a velocity vector $\nu = [u, v, r]^T$ for two-dimensional maneuvering, and a vector of actuation $\tau = [T_x, T_y, N_T]^T$ that includes the thrust from x and y direction in body frame and rotational moment caused by thrust. The motion of the rigid body system can be expressed as

$$\mathbf{M}\dot{\nu} + \mathbf{C}(\nu)\nu + \mathbf{D}(\nu)\nu = \tau. \tag{3.2}$$

$\mathbf{M}$ is the inertia matrix including the mass of rigid body and fluid added mass

$$\begin{gathered}\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A, \\ \mathbf{M}_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{bmatrix}, \mathbf{M}_A = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & 0 \\ 0 & 0 & -N_{\dot{r}} \end{bmatrix}.\end{gathered} \tag{3.3}$$

m is the robot's mass and I_z is moment of inertia about z-axis. $X_{\dot{u}}$ is defined as a constant hydrodynamic added mass for fish body in x-axis, caused by acceleration $\dot{u}$, and $Y_{\dot{v}}$ and $N_{\dot{r}}$ are defined analogously. The added masses are calculated by considering the entire body to an ellipsoid shape with a length of 0.03 meter and a radius of 0.015 meter. With these added mass coefficients k_{11}, k_{22} and k_{55} obtained from Fig. 3.2, the added masses are calculated as

$$X_{\dot{u}} = -k_{11}m, \quad Y_{\dot{v}} = -k_{22}m, \quad N_{\dot{r}} = -k_{55}I_z. \tag{3.4}$$

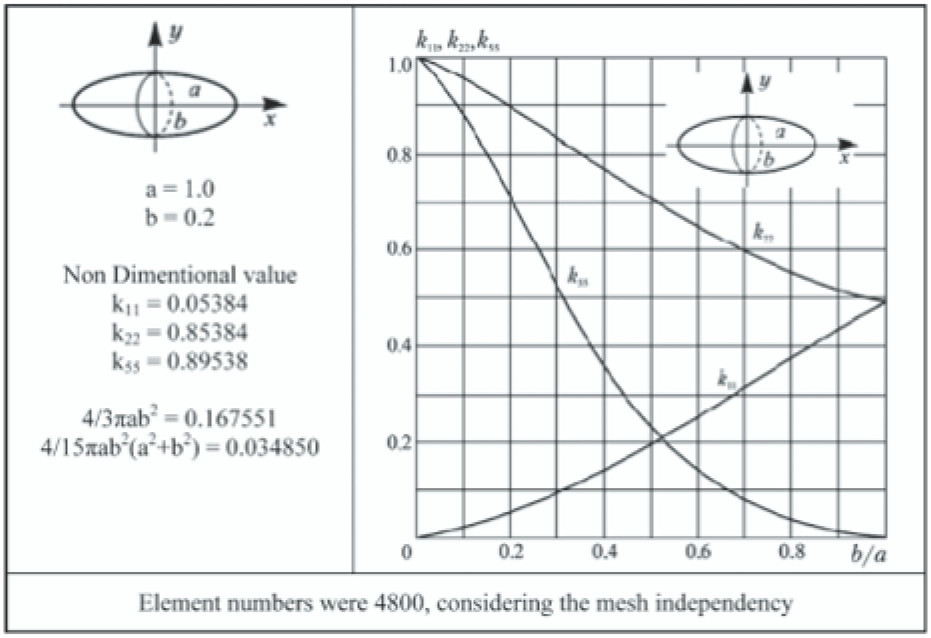


Figure 3.2: Chart of added mass coefficients [2].

C is a matrix term representing the Coriolis force. It describes the motion of the robotic fish due to rotation about inertial coordinate.

$$
\begin{aligned}
&\mathbf{C} = \mathbf{C}_{RB} + \mathbf{C}_A, \\
&\mathbf{C}_{RB} = \begin{bmatrix} 0 & 0 & -mv \\ 0 & 0 & mu \\ mv & -mu & 0 \end{bmatrix}, \mathbf{C}_A = \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v \\ 0 & 0 & -X_{\dot{u}}u \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 \end{bmatrix}.
\end{aligned} \tag{3.5}
$$

D is the matrix of hydrodynamic drag,

$$
\mathbf{D} = \begin{bmatrix} X_{|u|} & 0 & 0 \\ 0 & Y_{|v|} & 0 \\ 0 & 0 & N_{|r|} \end{bmatrix}. \tag{3.6}
$$

$X_{|u|}, Y_{|v|}, N_{|r|}$ are the linearized damping coefficients obtained through

$$\begin{aligned} X_{|u|} &= \frac{\mathrm{d}\frac{1}{2}\rho_w s_x c_x |u|u}{\mathrm{d}u}\Big|_{\bar{u}}, \\ Y_{|v|} &= \frac{\mathrm{d}\frac{1}{2}\rho_w s_y c_y |v|v}{\mathrm{d}v}\Big|_{\bar{v}}, \\ N_{|r|} &= \frac{\mathrm{d}c_r |r|r}{\mathrm{d}r}\Big|_{\bar{r}}. \end{aligned} \tag{3.7}$$

c_x, c_y, c_r are the drag coefficients, ρ_w is the density of water, s_x and s_y are robot's project area in x and y axes respectively. $\bar{u}$ is the median of the range of the forward speed, so as $\bar{v}$ and $\bar{r}$.

Combine Eq. 3.3, Eq. 3.5 and Eq. 3.6 together, the dynamic model of robotic fish in 3 DOF is

$$\begin{aligned} (m - X_{\dot{u}})\dot{u} - (m - Y_{\dot{v}})vr + X_{u|u|} &= T_x, \\ (m - Y_{\dot{v}})\dot{v} + (m - X_{\dot{u}})ur + Y_{v|v|} &= T_y, \\ (I_z - N_{\dot{r}})\dot{r} + (X_{\dot{u}} - Y_{\dot{v}})vr + N_{r|r|} &= N_T. \end{aligned} \tag{3.8}$$

The derivation of T_x, T_y and N_T are discussed in the next section.

3.2 Elongated Body Theory

EBT obtains the thrust by estimating the momentum convection from the body of the fish to the surrounding flow. In EBT [85], the force is derived in an inertial coordinate, as shown in Fig. 3.3. An index $\zeta \in [0, L]$, where L is the body length, is defined to indicate the r column of fish from tail to head. $[X(\zeta), Y(\zeta)]$ indicates the movement of a selected body segment whose velocity vector is defined as $V_\zeta = [\frac{\partial X}{\partial t}, \frac{\partial Y}{\partial t}]^T$. The reactive force F_{rf} that summaries all the forces obtained from flapping can be written as

$$F_{rf} = \left(m_i V_n V_m \hat{n} - m_i \frac{1}{2} V_n^2 \hat{m} \right)_{\zeta=L} - \frac{\mathrm{d}}{\mathrm{d}t} \int_0^L m_i V_n \hat{n} d\zeta. \tag{3.9}$$

The $m_i \simeq \frac{1}{4}\pi\rho d^2$ denotes the add-mass caused by the body segment's motion, ρ is the fluid density, d is the depth of segment, $\hat{n}$ is the unit vector that indicates the direction perpendicular to the curve of the segment, and $\hat{m}$ is the unit vector that indicates the direction tangential to the curve of the segment

$$\hat{m} = \begin{bmatrix} \frac{\partial X}{\partial \zeta} \\ \frac{\partial Y}{\partial \zeta} \end{bmatrix}, \quad \hat{n} = \begin{bmatrix} -\frac{\partial Y}{\partial \zeta} \\ \frac{\partial X}{\partial \zeta} \end{bmatrix}. \tag{3.10}$$

$V_n = V_\zeta^T \hat{n}$ and $V_m = V_\zeta^T \hat{m}$ are the magnitudes of the component velocity along $\hat{n}$ and $\hat{m}$ respectively.

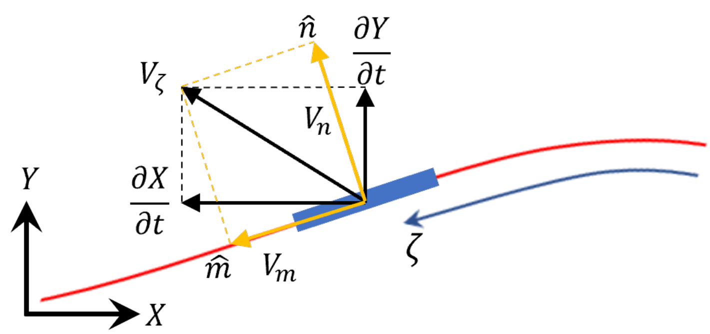


Figure 3.3: Derivation of EBT.

Although EBT is widely accepted, it still has limitations in control-oriented applications because it is defined in infinite-dimension. For robotic fish, obtaining the instantaneous body shape is impossible due to the complicated fluid-structure interaction. A feasible way to simplify the model is to consider the caudal fin at a quasi-steady state, in which the fin has high stiffness and the fluid-structure interaction can be ignored at low frequency. The fin velocity is represented by its quarter-chord point, which is at about $\frac{1}{4}$ of the chord length behind the leading edge [30, 86]. This assumption minimizes the fluid-structure interaction and reduces the complexity of model.

It is worth noticing that EBT uses the relative velocity between the caudal fin and surrounding flow to derive thrust. The relative velocity calculation needs both velocity of caudal fin flapping and fish's surge speed. The velocity of caudal fin is normally analyzed in body-fixed coordinate of the robotic fish, and the surge speed is reflected in robot's kinematics. Thus, the thrust derivation is coupled with the robot's kinematics. One way to simplify this procedure is to normalize the relative velocity; one can consider that the relative velocity has constant direction and magnitude so that thrust derivation only relies on fin dynamics. The normalization of the relative velocity is based on two assumptions; (1) The surrounding flow is considered closely attached to the fish's body and has not been heavily disturbed by the undulatory motion. This assumption is concluded from several works that study the boundary layer flow of an undulatory flapping. Although the vortices are also produced by the head shaking or undulatory locomotion of the anterior body, they are small and not strong enough to persist in the flow that will interact with the caudal fin [87]. The undulatory motion helps the flow layer keep attached to the body [83], as long as the velocity of the undulatory is smaller than flow velocity [88, 16]. Since the robotic fish propelled by DSC has the characteristics of undulatory locomotion and low-amplitude head movement, its surrounding flow can be assumed to be parallel to the peduncle before entering oscillating foils. (2) Fish speed can be recognized as a time-invariant factor. This assumption is inspired by a hydrodynamic study analyzing the force from oscillating foil in different ambient flows. It concluded that flow speed has little impact on propulsion [89].

The model of the DSC-driven robotic fish can be simplified by using a normalized relative velocity. When the fish is operated in 'two-joint flapping,' it has a small lateral head oscillation and the servo motor does not disturb the surrounding flow. The surrounding flow can be considered stable in passing through the fish body and then participate in the oscillating foil, even under a small bias angle from the servo motor. Regardless of fish's surge speed and bias angle, thrust forces can be derived in the DSC frame, in which the only variable is flapping frequency. In the thrust calculation, a constant flow speed is added into the quarter-chord point to compensate the ignorance of surge speed. The entire approach basically minimizes the impact of the environmental factors in the thrust derivation.

Concentrate on the body-fix coordinate with an individual coordinate for the DSC mechanism. The thrust T is derived in the DSC frame, which is defined as $O_d - x_d y_d$. Its origin O_d is located at the second joint of the fish tail, as shown in Fig. 3.4. The servo motor produces the angle δ between the DSC frame $O_d - x_d y_d$ and robot body-fixed frame $G - xy$, to add a bias angle to the oscillating foil.

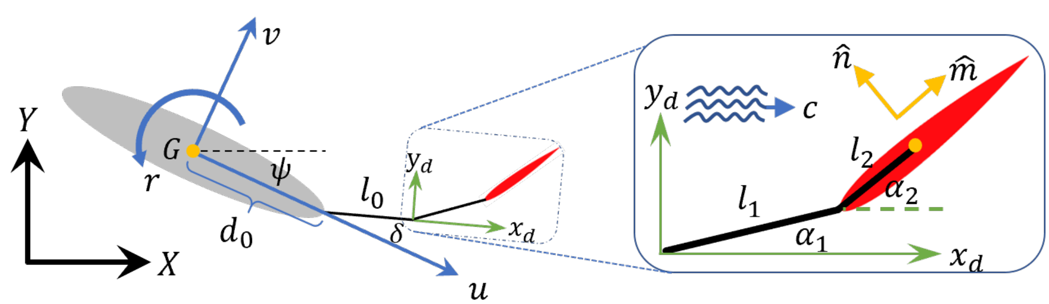


Figure 3.4: Kinematic diagram of robotic fish and DSC.

The angle of DSC's first and second joint in $O_d - x_d y_d$ are written as

$$\begin{aligned} \alpha_1 &= A_1 \sin(\omega), \\ \alpha_2 &= A_2 \sin(\omega + 0.5\pi). \end{aligned} \tag{3.11}$$

The quarter-chord point of caudal fin, which is represented as $[x_d, y_d]$ in the DSC coordinate, is calculated by

$$\begin{aligned} x_d &= l_1 \cos\alpha_1 + l_2 \cos\alpha_2, \\ y_d &= l_1 \sin\alpha_1 + l_2 \sin\alpha_2. \end{aligned} \tag{3.12}$$

The caudal fin's relative velocity, which is considered as the absolute velocity at the inertial frame in EBT, is represented as

$$V = \begin{bmatrix} \dot{x}_d - V_c \\ \dot{y}_d \end{bmatrix}, \tag{3.13}$$

where V_c is a scalar that represents a constant flow added in parallel to x_d direction, regardless of fish's speed. In this term, the magnitude of component velocity are

$$\begin{aligned} V_n &= -\dot{x}_d \sin\alpha_2 + \dot{y}_d \cos\alpha_2 + V_c \sin\alpha_2, \\ V_m &= \dot{x}_d \cos\alpha_2 + \dot{y}_d \sin\alpha_2 - V_c \cos\alpha_2. \end{aligned} \tag{3.14}$$

The unity vectors $\hat{m}$ and $\hat{n}$ at quarter-chord point can be written by

$$\hat{m} = \begin{bmatrix} \cos\alpha_2 \\ \sin\alpha_2 \end{bmatrix}, \quad \hat{n} = \begin{bmatrix} -\sin\alpha_2 \\ \cos\alpha_2 \end{bmatrix}. \tag{3.15}$$

Therefore, the reactive force in the DSC frame is represented by a vector, in which the first element is the component force in x_d direction and the second element is the component force in y_d direction

$$\overrightarrow{F_{rf}} = \begin{bmatrix} F_{x_d} \\ F_{y_d} \end{bmatrix} = -\frac{1}{2} m_i V_n^2 \hat{m} + m_i V_n V_m \hat{n} - m_i L \frac{\partial V_n}{\partial t} \hat{n}. \tag{3.16}$$

The reactive force denoted in the DSC's frame needs to be converted to the fish's body-fixed coordinate

$$\begin{bmatrix} T_x \\ T_y \end{bmatrix} = k_F \begin{bmatrix} \cos\delta & -\sin\delta \\ \sin\delta & \cos\delta \end{bmatrix} \begin{bmatrix} F_{x_d} \\ F_{y_d} \end{bmatrix}, \tag{3.17}$$

where k_F is the force coefficient for model tuning. N_T is obtained as

$$N_T = \begin{bmatrix} d_0 + (l_0 + l_1 + l_2)\cos\delta \\ (l_0 + l_1 + l_2)\sin\delta \end{bmatrix} \times \begin{bmatrix} T_x \\ T_y \end{bmatrix}, \tag{3.18}$$

where $\times$ denotes the cross product of two vectors.

3.3 Model Reduction

In this section, a linear model that describes robotic fish's steering associated with the bias angle of the oscillating foil is derived. It is inspired by a well-known linear Namoto model [69] for ocean vessels that can predict vessel's yaw motion. In this robotic fish, the swimming speed and steering are individually controlled by the DC motor in DSC and the servo motor in the first joint. One can consider that the robotic fish is equivalent to an ocean vessel in the motion. When the robotic fish initially swims straightly at a constant speed, its velocity u, v, r can be written as

$$u = u_0 + u', \quad v = v', \quad r = r'. \tag{3.19}$$

u_0 is the initial velocity, u', v', r' are small velocity perturbations. The condition $v_0 = r_0 = 0$ is valid for straight swimming that the lateral and rotatory motions are neglectable. The time derivatives of velocity are

$$\dot{u} = \dot{u}', \dot{v} = \dot{v}', \dot{r} = \dot{r}'. \tag{3.20}$$

The bias angle δ in forward swimming is also considered as a perturbation, and so is denoted as δ'. By this definition, the T_x in Eq. 3.17 is represented as

$$T_x = k_F \begin{bmatrix} \cos(\delta_0 + \delta') \\ -\sin(\delta_0 + \delta') \end{bmatrix}^T \begin{bmatrix} F_{x_d} \\ F_{y_d} \end{bmatrix} = k_F \begin{bmatrix} \cos\delta' \\ -\sin\delta' \end{bmatrix}^T \begin{bmatrix} F_{x_d} \\ F_{y_d} \end{bmatrix}, \tag{3.21}$$

where $\delta_0 = 0$. For a small bias angle satisfying $|\delta'| \in (0, 0.35]$, one can take the first order Taylor expansion to approximate the trigonometry of bias perturbation as $\cos\delta' \approx 1$ and $\sin\delta' \approx \delta'$. Also, the amplitude of reactive force is unchanged at a constant frequency, its component force can be replaced by a mean value during a period of flapping, which means $F_{x_d} = \overline{F_{x_d}}$ and $F_{y_d} = \overline{F_{y_d}}$. Eq. 3.21 turns to be

$$T_x \approx k_F(\overline{F_{x_d}} - \delta'\overline{F_{y_d}}). \tag{3.22}$$

Similarly, the T_x and N_T turn to

$$T_y \approx k_F(\delta'\overline{F_{x_d}} + \overline{F_{yd}}),$$
$$N_T \approx k_F(d_0 + l)(\delta'\overline{F_{x_d}} + \overline{F_{yd}}) - k_F l(\delta'\overline{F_{x_d}} - \delta'\overline{F_{yd}}), \tag{3.23}$$

where $l = l_0 + l_1 + l_2$ for simplification. If consider $\overline{F_{yd}} = 0$ that the integral of lateral force is zero for periodical flapping, $T_x \approx k_F\overline{F_{x_d}}$, $T_y \approx k_F\overline{F_{x_d}}\delta'$ and $N_T \approx k_F d_0\overline{F_{x_d}}\delta'$. The mean thrust $\overline{F_{x_d}} = -0.035$ N is obtained through simulation. Under this assumption, a small bias angle has minimal effect on the thrust magnitude in x direction, one can assume that

$$T_x = T_{x0},\;\; T_y = T'_y, \text{ and } N_T = N'_T. \tag{3.24}$$

The dynamic function Eq. 3.8 is rewritten as

$$\begin{aligned} (m - X_{\dot{u}})\dot{u}' + X_{|u|}u_0 + X_{|u|}u' &= T_{x0}, \\ (m - Y_{\dot{v}})\dot{v}' + (m - X_{\dot{u}})u_0 r' + Y_{|v|}v' &= T'_y, \\ (I_z - N_{\dot{r}})\dot{r}' + N_{|r|}r' &= N'_T. \end{aligned} \tag{3.25}$$

The higher-order small perturbations, such as $u'r'$, are ignored. The Coriolis angular momentum $Y_{\dot{v}} - X_{\dot{u}}$ caused by the added mass is ignored to moderate the yaw oscillation caused by the recoil head movement. The obtained perturbation model decouples the sway motion and yaw motion from the surge motion. One can extend this perturbation model to describe the steering of a robotic fish that is straightly swimming at the beginning. Given the definitions $m_1 = m - Y_{\dot{v}}$, $m_2 = m - X_{\dot{u}}$ and $m_3 = I_z - N_{\dot{r}}$, a second-order steering model can be summarized by ignoring surge velocity and has the form

$$\begin{bmatrix} \dot{v} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\frac{Y_{|v|}}{m_1} & -\frac{m_2 u_0}{m_1} \\ 0 & -\frac{N_{|r|}}{m_3} \end{bmatrix} \begin{bmatrix} v \\ r \end{bmatrix} + \begin{bmatrix} \frac{Y_\delta}{m_1} \\ \frac{N_\delta}{m_3} \end{bmatrix} \delta, \tag{3.26}$$

where $Y_\delta = k_F \overline{F_{x_d}}$ and $N_\delta = k_F d_0 \overline{F_{x_d}}$. Eq. 3.26 can be converted to a transfer function that relates the bias angle to turning speed

$$\frac{r(s)}{\delta(s)} = \frac{a_1 s + a_2}{s^2 + a_3 s + a_4}, \tag{3.27}$$

where

$$\begin{aligned} a_1 &= \frac{N_\delta}{m_3}, \\ a_2 &= \frac{Y_{|v|} N_\delta}{m_1 m_3}, \\ a_3 &= \frac{m_3 Y_{|v|} + m_1 N_{|r|}}{m_1 m_3}, \\ a_4 &= \frac{Y_{|v|} N_{|r|}}{m_1 m_3}. \end{aligned}$$

The transfer function has two negative poles given by $\{-\frac{Y_{|v|}}{m_1}, -\frac{N_{|r|}}{m_3}\}$. In the robotic fish operation, the steering speed may be heavily disturbed by the lateral oscillations of the head and position actuation. Hence, deriving a transfer function

$$\frac{\psi(s)}{\delta(s)} = \frac{1}{s} \cdot \frac{r(s)}{\delta(s)}, \tag{3.28}$$

between the bias angle and the yaw angle by adding an integrator is more feasible for implementation purposes.

3.4 Model Validation

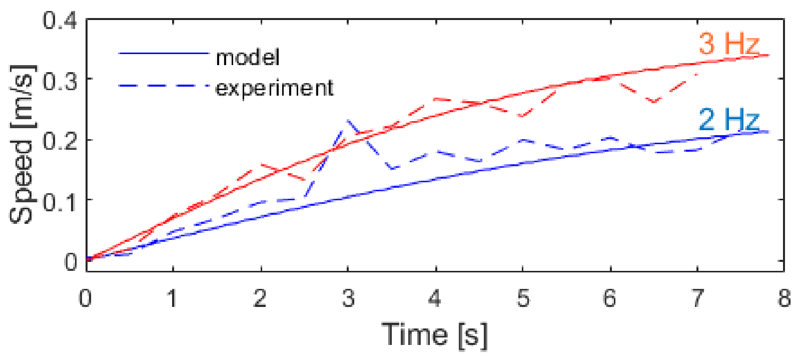
The vectorized EBT model is tuned by a least-square method to match the experimental data of forwarding speed and steering speed. In modeling the forward speed, we tune the model to match the time response of the robotic fish's swimming speed at various flapping frequencies. The Fig. 3.5a shows the model fitting result of matching the time response of velocity acquitted at 2 Hz and 3 Hz. In Fig. 3.5b, the model validation result for steady-speed, from 1.8 Hz to 3 Hz, is shown. Due to the pool size, the time period allowed for the forward swimming of robotic fish is limited. Overall, the model has a good match to the experimental data at a measured frequency. The dimensions and model parameters are shown in Table. 3.1.

Table 3.1: Parameters of dynamic model

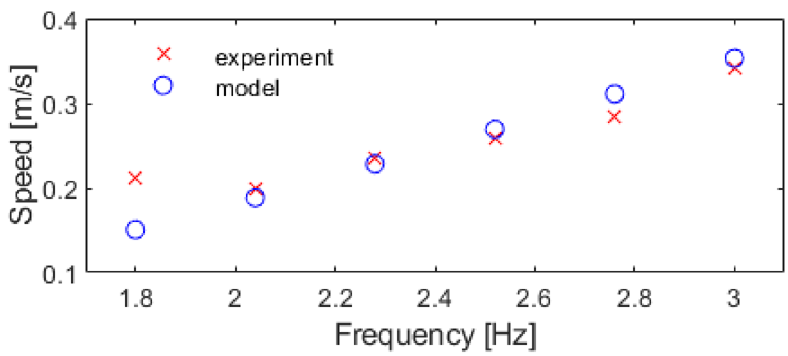
d_0[m]	l_0[m]	l_1 [m]	l_2[m]	L[m]	d[m]
0.04	0.042	0.058	0.022	0.04	0.08
c_x	c_y	c_r	s_x[m^2]	s_y[m^2]	V_c[m/s]
0.39	2.2	0.0055	0.0025	0.02	0.08
k_1	k_2	k_3	k_F	I[kg m^2]	m[kg]
0.095	0.83	0.55	0.2	0.0047	0.9

The model also has a good agreement with the experimental data in steering. It matches the steering radius and steering speed at different bias angles with a considerable small error. We add an arbitrary initial fish pose in the simulation so that the simulation has a better reflection of the real experiment. The experimental result and dynamic model validation in steering radius versus bias angle shows in Fig. 3.6a, and steering speed versus bias angle shows in Fig. 3.6b

Eq. 3.28 indicates the robotic fish's steering can be approximated into a third-order transfer function that has one zero and three poles. However, obtaining the

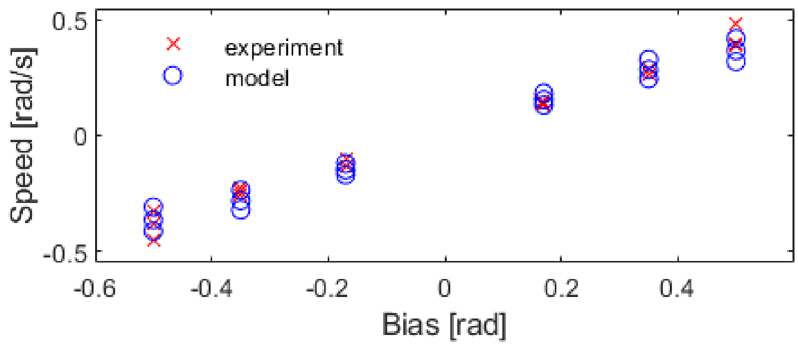


(a) Experimental results and model prediction for time response of forward speed at 2 Hz and 3 Hz.

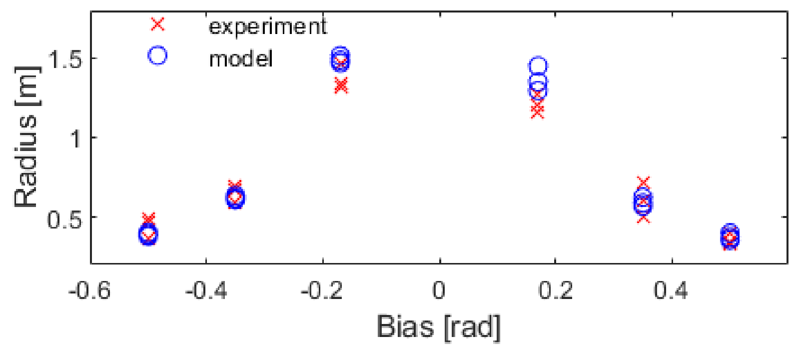


(b) Experimental results and model prediction for steady-speed from 1.8 Hz to 3 Hz.

Figure 3.5: Model validation result in forward swimming.



(a) Experimental results and model prediction for steering radius from 1.8 Hz to 2.4 Hz.



(b) Experimental results and model prediction for steering speed from 1.8 Hz to 2.4 Hz.

Figure 3.6: Model validation result in steering.

transfer function from the model's parameters could cause significant errors due to the model has been highly simplified. A more feasible way to obtain the linear model is approximating a model from input-output data. A subspace state-space system identification (MOESP)[90] method was adopted to obtain a discretized linear model through analyzing the excitation of a series of random pulse inputs. Applying MOESP through experiments is unrealistic because the identification process requires a large amount of data. The whole procedure needs the robot to move for a long period, and the experiment needs a large area.

An alternative method for MOESP is that generates data through a simulation based on the vectorized EBT model. In the simulation, fish dynamics is calculated by a 2nd-order Runge-Kutta method with following equations:

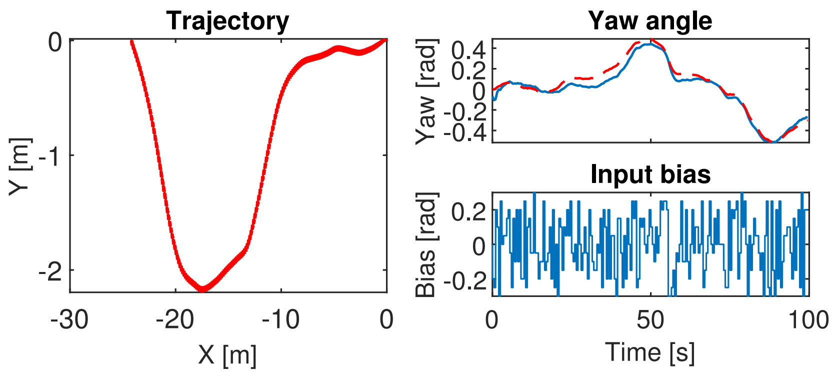
$$\dot{\xi} = \begin{cases} \xi_2 \cos \xi_5 - \xi_4 \sin \xi_5 \\ [(T_x - X_{|u|}\xi_2) + m_1 \xi_4 \xi_6]/m_2 \\ \xi_2 \sin \xi_5 + \xi_4 \cos \xi_5 \\ [(T_y - Y_{|v|}\xi_4) + m_2 \xi_2 \xi_6]/m_1 \\ \xi_6 \\ (N_r - Y_{|r|}\xi_6)/m_3 \end{cases}, \tag{3.29}$$

where $\xi = [X, u, Y, v, \psi, r]^T$ is the state vector that represents the position and velocity of the robotic fish . The simulation considers the sensing feedback as the robotic fish's positions obtained periodically through GPS. The positions is represented as $[X_i, Y_i]^T$ with added artificial Gaussian noise for $i = 1, 2, \cdots$ which is a sample in time sequence. Because MOESP requests the system be fully excited with all possible input value. The robotic fish is simulated with a serial of random bias angle sent every 0.5 s (2 Hz). The position data were collected at every 0.2 s (10 Hz) with ±1 cm Gaussian noise to approximate the robot's heading angle.

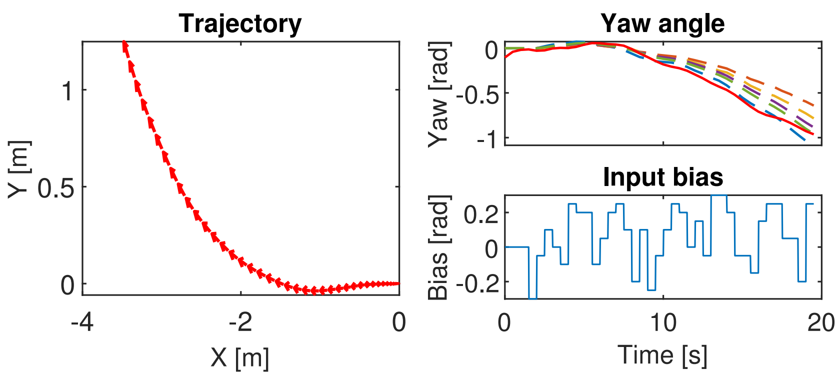
In model identification in Fig. 3.7a, the simulation was running for 100 seconds to collect sufficient position data. The robotic fish started with a straightly swimming at a constant speed. Then a serial of random bias input, varying between -0.3 rad to 0.3 rad, was sent to the robot to make the steering. The control input was sent every 0.5 s (2 Hz), which is also the frequency of discrete transfer function. Both input bias angle and output yaw angle were sent to MOESP to approximate a linear model that can fit the approximated heading angle. Because the input was random, the maneuvering in each simulation was different. A total of 20 sets of linear models were collected as a model bank for further analysis. In reality, the model-based prediction and control focus on guiding the robot to finish an individual turn. A complicated robot motion will be divided into several small maneuverings. Although MOESP requires a large amount of data for accurate identification, the validation only needs a much shorter period. In model validation, a 20-second simulation was conducted with random input bias. In Fig. 3.7b, the simulation results were compared with all models from the model bank (only five shown in the figure) to evaluate models' accuracy. The validation was repeated 50 times for all models. Their mean squared errors (MSEs) were collected and compared, and the model with the lowest MSE was selected. Its identified transfer function was

$$\frac{\psi(z)}{\delta(z)} = \frac{-0.03081z^2 + 0.02642z}{z^3 - 2.832z^2 + 2.67z - 0.8381}. \tag{3.30}$$

In comparison, the discrete form of analytical transfer function was calculated



(a) Identification.



(b) Validation.

Figure 3.7: Linear steering model identification and validation.

based on Eq. 3.27 and the parameters shown Table. 3.1, which is shown below

$$\frac{\psi(z)}{\delta(z)} = \frac{-0.0046z^2 - 0.0004z + 0.004}{z^3 - 2.768z^2 + 2.55z - 0.781}. \tag{3.31}$$

The identified transfer function Eq. 3.30 has a close denominator to analytical transfer function, which ensures that they have similar stability. Although their numerator is slightly different, their response can be unified by adding a control gain.

The linear model in Eq. 3.30 was also examined by comparing it with the time response of the experiment. The experiment yaw angle was obtained by a category of step bias input ranging from -0.5 rad to 0.5 rad. The same step input was sent to the linear model, and the linear model output (in dashed line) was compared to the experimental yaw angle (in solid line) in Fig. 3.8. One can notice that the experimental data is heavily disturbed by fish shaking and computer vision error. From observation, the linear model has an impressive accuracy in various bias angles. This time response result indicates that the linear model's identification and validation through simulations can achieve a good agreement with the experimental data. Even the vectorized EBT model is not control-orientated, it is still a reliable source to obtain a linear model through model reduction and system identification.

3.5 Chapter Summary

In this chapter, the derivation of the dynamic model and linear model are introduced. The kinematics of robotic fish is described in inertial coordinate. The rigid-body dynamics and hydrodynamic are analyzed in robotic fish's body-fixed coordinate. A vectorized EBT method that combines added-mass effect and quasi-static

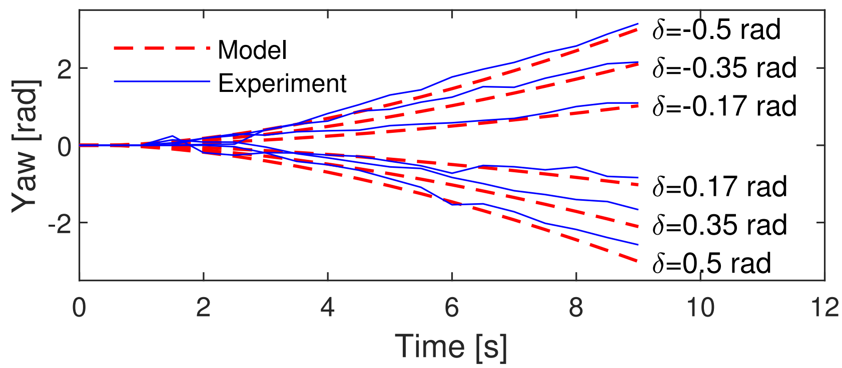


Figure 3.8: The step response comparison between experimental data and linear model output.

lift is applied when calculating the thrust of tail flapping. In addition, the thrust calculation is concentrated in an individual DSC frame because of the special character of the DSC mechanism that can achieve undulatory locomotion. Thus, the thrust calculation is decoupled from the robotic fish's kinematics, and the robot can be modeled from the vessel's perspective. Then the nonlinear dynamic model is simplified, following the idea of the Namoto model, to a linear transfer function that describes the relationship between input bias and output yaw motion. Both nonlinear dynamic model and linear model are validated in experiments and simulations. The nonlinear dynamic model is further used in developing a simulation environment, and the linear model is used for linear controller design.

Chapter 4

Robotic Fish Control

In this chapter, steering control of robotic fish is discussed. The observed-state feedback controller (OSFC) using the linear model derived in Chapter 4 is firstly introduced. Simulation and experiment results demonstrate the accuracy and effectiveness of the linear model in steering the robotic fish. After that, the event-trigger steering control of robotic fish is discussed to address the communication and computing power challenge in robotic fish's application. We combine the ideas of linear predictor and event trigger rule to ensure the stability of robotic fish in inconsistent communication situations. Experimental results show that the event-trigger controller (ETC) has the same performance as the error-based controller, and it is able to maintain the performance when the communication frequency is dropped.

4.1 Observed-state Feedback Control

The OSFC demonstrates the effectiveness of the linear model in steering the robotic fish. The transfer function in Eq. 3.30 is rewritten in a state-space form, where $x^*(k) \in \mathbb{R}^{3\times1}$ is the state variable, $y^*(k) \in \mathbb{R}$ is the output yaw angle ,and

$u^*(k) \in \mathbb{R}$ is the input bias angle

$$\begin{aligned} x^*(k+1) &= Ax^*(k) + Bu^*(k), \\ y^*(k) &= Cx^*(k). \end{aligned} \tag{4.1}$$

The system matrices A, B and C are obtained as

$$A = \begin{bmatrix} 2.832 & -1.335 & 0.8381 \\ 2 & 0 & 0 \\ 0 & 0.5 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0.25 \\ 0 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} -0.1232 & 0.05285 & 0 \end{bmatrix}.$$

One can conclude that the system with (A, B, C) is fully controllable and observable.

To provide a full state feedback, a discrete state observer is designed to use $\bar{x}^*(k) \in \mathbb{R}^{3\times1}$ to estimate $x^*(k)$, with an observer gain of $L \in \mathbb{R}^{3\times1}$

$$\bar{x}^*(k+1) = A\bar{x}^*(k) + Bu^*(k) + L[\psi(k) - C\bar{x}^*(k)]. \tag{4.2}$$

The control input is calculated based on the state feedback gain $K \in \mathbb{R}^{1\times3}$ and a scalar $K_d \in \mathbb{R}$ for reference $\psi_d(k) \in \mathbb{R}$

$$u^*(k) = K\bar{x}^*(k) + K_d\psi_d(k). \tag{4.3}$$

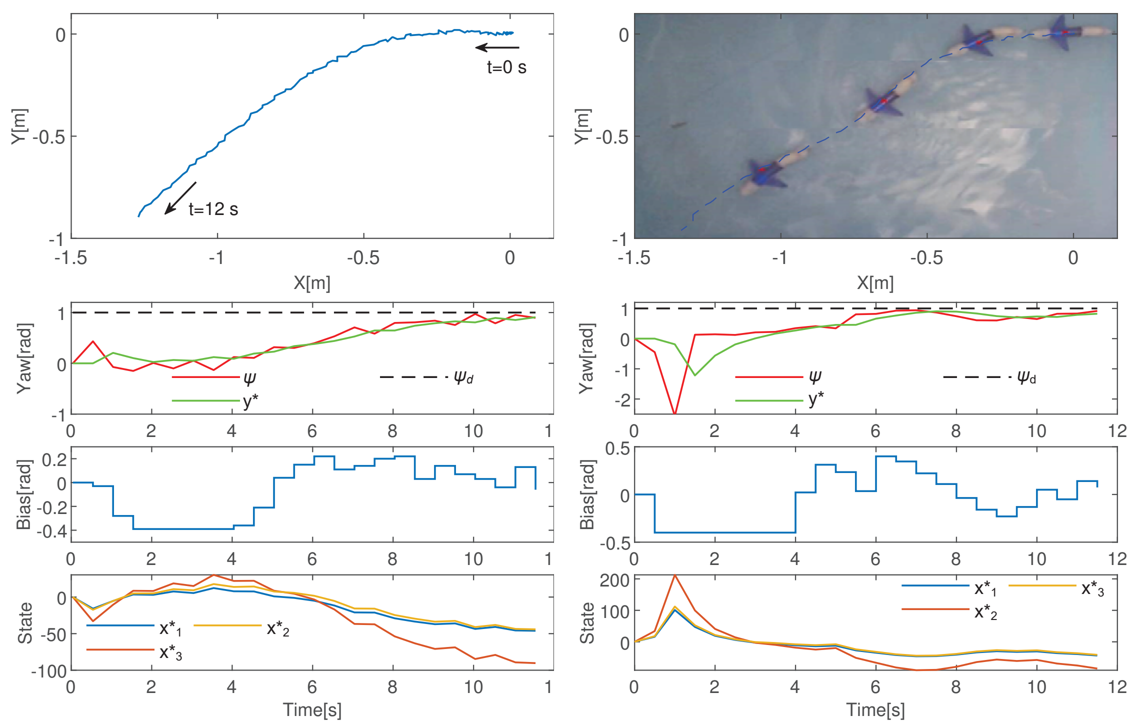
The OSFC was firstly developed and calibrated in simulations. K_z and K_d were tuned in simulations to archive a critical damping condition that drove the system rapidly to the desired value without overshooting and oscillation. The reference was selected at $\psi_d(k)$=1 rad to take the best of the pool area and stretch control time in simulations. The simulation time was 12 seconds which was sufficient for the controller to left-steering the robotic fish from stationary. In Fig. 4.1a, the bias of servo firstly fully turned to the left, then back to normal. The turning process was smooth, and the yaw angle was stabilized to the desired value. The controller gains are

$$L = \begin{bmatrix} -36.238 \\ -76.058 \\ -40.044 \end{bmatrix}, \quad K = \begin{bmatrix} -4.173 & 3.614 & -3.115 \end{bmatrix}, \quad K_d = -3.344.$$

The same gain value was used in experiments, and robotic fish's initial position and pose were the same as in the simulation. As shown in Fig. 4.1b, the bias angle has the same trend as in Fig. 4.1a. Because the robotic fish was starting from stationary, the yaw angle calculation was heavily disturbed since the displacement was small. Despite that, the control results were very close, from 2 seconds to 12 seconds between simulations and experiments. The experimental results demonstrated that the linear model had high accuracy and model-based steering control was feasible.

4.2 Event Trigger Control

The robotic fish have many advantages in mobile underwater sensor networks. However, challenges also arise when controlling robotic fish in low visibility and GPS-denied underwater environments. Due to the limited space, robotic fish is not an appropriate carrier for some external sensors, such as scanning sonar. In most cases, localization sensors are equipped in mother-ship, and the mother-ship transmits the location information remotely through an underwater acoustic network. Acoustic underwater communication has limited bandwidth, which could cause a reduced sampling rate in feedback control. Due to the under-actuated dynamics of robotic fish,



(a) Simulation

(b) Experiment

Figure 4.1: Observed-state feedback steering control.

wireless control of robotic fish with such a drop in sampling rate has become a challenge. Hence, a control method that is less communication-intensive and robust to drops in communication rates is highly desired to control robotic fish.

Standard feedback controls call for changing the values of controls at sampling times that are independent of the state of the dynamics. However, such feedback controls are not computationally efficient, because they could result in unnecessarily frequent computation of new control values [91]. In many applications (especially networked systems), such inefficiencies have been addressed using ETC that calls for only changing control values when a significant enough event is detected, and then the control is triggered [91, 92, 93]. Such events are usually modeled as instances when a measurement from the system deviates from a prescribed value by more than some prescribed amount. ETCs can significantly save bandwidth in communication networks by reducing the number of communication times when the external sensor must communicate with the local actuators. Since ETC takes a varying-sampling-rate (VSR), which is less communication-intensive, ETC has a great potential in network steering control of robotic fish for coastal monitoring. In this paper, we adopt a trigger rule that evaluates the error of the hold state to define an event [94], and add a linear-model-based predictor to form a periodic event-triggered control (PETC) [95].

4.2.1 Event trigger rule

A discrete state-space model for steering of robotic fish is obtained in Eq. 4.1. Its error dynamic representation obtained as

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k), \end{aligned} \tag{4.4}$$

where x, u, and y are defined as

$$x = x^* - x^*_{ss}, \quad u = u^* - u^*_{ss}, \text{ and } y = y^* - \psi_d.$$

A, B, and C are system matrices from Eq. 4.1. x^*_{ss} is the steady-state of x^*. u^*_{ss} is the steady-state of u^*. And ψ_d is a non-zero desired yaw angle.

The integration $q \in \mathbb{R}$ that integrals output error is defined as

$$q(k+1) = q(k) + Cx(k). \tag{4.5}$$

Define $x'(k) = [x(k), q(k)]^T$, the full dynamics of the robotic fish with the integrator can be written as

$$x'(k+1) = \underbrace{\begin{bmatrix} A & 0 \\ C & I \end{bmatrix}}_{A'} x'(k) + \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{B'} u(k), \tag{4.6}$$

where (A', B') is fully controllable. The state feedback controller is chosen as

$$u(k) = \underbrace{\begin{bmatrix} -K & H \end{bmatrix}}_{K'} \underbrace{\begin{bmatrix} x(k) \\ q(k) \end{bmatrix}}_{x'(k)} \tag{4.7}$$

with a state feedback gain $K \in \mathbb{R}^{1\times 3}$ and an integral gain $H \in \mathbb{R}$. Define $\Lambda = A' + B'K'$. Since (A', B') is fully controllable, a pole assignment approach can be used to select the gain K', such that Λ is a Hurwitz stable matrix.

Define $V(x) = x'^T P x'$, where P is a symmetric positive definite matrix. Then, the difference $\delta_V(x'(k)) = V(x'(k+1)) - V(x'(k))$ satisfies

$$\delta_V(x'(k)) = x'^T(k)(\Lambda^T P \Lambda - P)x'(k) < 0 \tag{4.8}$$

for all nonzero states $x'(k)$ of Eq. 4.7, because $\Lambda^T P \Lambda - P$ is a negative definite. Thus, x and y will ultimately approach to zero.

On the contrary, the state variables in ETC are not sent to the robot at every time instant. For each integer $k \geq 0$, we define k^- to be the previous event-triggering time. The rule for computing the event triggering times is specified below. The state variables are sent at the triggering time. We assume that the state variables are sent at time $k = 0$. At any time $k > 0$, when the state variables are not sent, the control input is obtained through a zero-order-hold, meaning

$$u(k) = K'x'(k^-). \tag{4.9}$$

Let $e(k)$ denote the error $e(k) = x'(k^-) - x'(k)$ between the value of the held state and the real-time state. Eq. 4.6 can be written as

$$\begin{aligned} x'(k+1) &= A'x'(k) + B'K'(e(k) + x'(k)) \\ &= \Lambda x'(k) + B'K'e(k). \end{aligned} \tag{4.10}$$

In this ETC case, δ_V satisfies

$$\begin{aligned} \delta_V(x'(k)) = & -x'^T(k)Qx'(k) + 2x'^T(k)\Lambda^T PB'K'e(k) \\ & + e^T(k)K'^T B'^T PB'K'e(k), \end{aligned} \tag{4.11}$$

where $Q = -\Lambda^T P\Lambda + P$ is a positive definite matrix.

For a given constant $\sigma > 0$, the event triggering times t_i for $i \geq 1$ are chosen, such that the inequality

$$\|e(k)\| \leq \sigma \|x'(k)\| \tag{4.12}$$

holds for all $k \geq 0$, where the constant $\sigma \geq 0$ will be specified and $\|\cdot\|$ is the usual Euclidean norm defined in [96]. Combining Eq. 4.11 and Eq. 4.12 gives

$$\begin{aligned} \delta_V(x(k)) &\leq -x'^T(k)Qx'(k) + 2\|x'(k)\| \left\|\Lambda^T PB'K'\right\| \|e(k)\| + \left\|K'^T B'^T PB'K'\right\| \|e(k)\|^2 \\ &\leq -\lambda_{\min}[Q] \|x(k)\|^2 + 2\sigma \left\|\Lambda^T PBK\right\| \|x(k)\|^2 + \sigma^2 \left\|K^T B^T PBK\right\| \|x(k)\|^2 \\ &\leq -\left\{\lambda_{\min}[Q] - 2\sigma \left\|\Lambda^T PB'K'\right\| - \sigma^2 \left\|K'^T B'^T PB'K'\right\| \right\} \|x'(k)\|^2, \end{aligned} \tag{4.13}$$

where $\lambda_{\min}[Q]$ is the smallest eigenvalue of Q. Since $\lambda_{\min}[Q] > 0$, there is a value $\varepsilon > 0$ such that the quantity in curly braces in Eq. 4.13 is positive for each $\sigma \in (0, \varepsilon]$, which provides an asymptotic stability decay estimate on δ_V. Thus, the system achieves asymptotic stability, and reduces the sampling frequency, which decreases the need for high bandwidth communication.

4.2.2 ETC design

To implement event trigger rule in control of robotic fish. The PETC is introduced. There are two modes of operations in PETC: ‘triggered’ mode and ‘non-triggered’ mode. In ‘triggered’ mode, communication is available between the robotic fish and remote controller. The controller inside the robot works as a traditional state feedback control using state variables observed in remote controller. In ‘non-triggered’ mode, the communication between the remote controller and robot is unavailable, leaving the controller to work without a feedback.

The overall control design can be described by the diagram as shown in Fig. 4.2, where the green block represents the dynamics of robot with its local controller and the orange block represents the dynamics of the remote sensor. Inside the robot, Predictor-1 is used to estimate the state variable when communication is not triggered. Integrator-1 is used to integrate the predicted heading angle tracking error. Control input u combines both the state feedback control and the integration of tracking error with different gains K and H. On the remote sensor side, a computer vision sensor is used to calculate heading angle y. y is sent to Observer to estimate state variable x. Predictor-2 is the mirrored Predictor-1. Integrator-2 is the mirrored Integrator-1. Integrator-3 is used to integrate the actual heading angle tracking error. When an event is triggered, the communication between robot and remote sensor will be established. Variables z_1 and z_2 in Predictor-1 and Predictor-2, respectively, will be replaced by estimated state variable x. Variables q_1 and q_2 in Integrator-1 and Integrator-2, respectively, will be replaced by q_3 in Integrator-3.

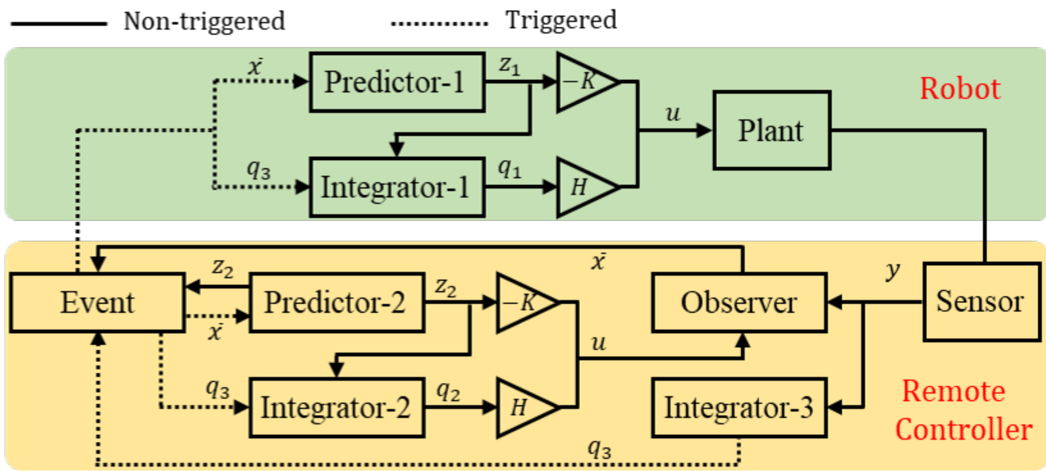


Figure 4.2: Control diagram of PETC.

In the remote controller, a sensor measures robotic fish's yaw angle ψ and calculates y for observer. The observer is defined as

$$\bar{x}(k+1) = A\bar{x}(k) + Bu(k) + L[y(k) - C\bar{x}(k)], \tag{4.14}$$

where $\bar{x}(k) \in \mathbb{R}^{3\times1}$ is the estimate of $x(k)$ and $L \in \mathbb{R}^{3\times1}$ is the observer gain that ensures the convergence of observer.

In 'non-triggered' mode, we follow the idea of embedded predictor proposed by [95] to have an open-loop predicted state variables z. z can prevent the zero-order-hold of control input u when $\bar{x}$ is unavailable for the robot. The dynamics of the predictor is defined by

$$z_i(k+1) = \begin{cases} Az_i(k) + Bu(k) & \text{(Non-triggered)} \\ A\bar{x}(k) + Bu(k) & \text{(Triggered)}, \end{cases} \tag{4.15}$$

where $z_i \in \mathbb{R}^{3\times1}$, and $i = 1, 2$. z_1 represents Predictor-1 whose dynamics are defined in Eq. 4.15. Predictor-2, defined as z_2, has the same dynamics as z_1. These two predictors run simultaneously and work at the same sampling rate as the remote controller. These two predictors use the same system matrices (A, B) as the ones used in the observer to estimate the state variables.

Through integration of output error, Integrator-3 is used to estimate the unknown disturbance caused by the modeling error and external turbulence. Then the disturbance is actively rejected by adding an extra term in control input u. Integrator-1, represented by $q_1 \in \mathbb{R}$, runs inside the robot. Integrator-2, $q_2 \in \mathbb{R}$, runs in remote controller. q_1 and q_2 have the same dynamics as

$$q_i(k+1) = \begin{cases} q_i(k) + Cz_i(k) & \text{(Non-triggered)} \\ q_3(k+1) & \text{(Triggered)}, \end{cases} \tag{4.16}$$

for $i = 1, 2$. Integrator-3 is running in the remote controller and has the dynamics

$$q_3(k+1) = q_3(k) + y(k). \tag{4.17}$$

It is updated using y to represent the 'true' error, similar to the observer representing the 'true' state variables. With $\bar{x}$, z and q are defined, control input u discussed in Eq. 4.9 is rearranged as

$$u(k) = \begin{cases} -Kz_1(k) + Hq_1(k) & \text{(Non-triggered)} \\ -K\bar{x}(k) + Hq_3(k) & \text{(Triggered)}. \end{cases} \tag{4.18}$$

$H \in \mathbb{R}$ is an integral gain.

u is calculated using the 'true' error and the 'true' state variables from the remote controller in 'triggered' mode. In 'non-triggered' mode, u is calculated using the error and state variables estimated inside the robot. The quantities z_2 and q_2 can mirror the calculation of u for the observer because they have the same dynamics as z_1 and q_1. Thus, the observer can clone the u, which is calculated inside the robot in 'non-triggered' mode, to calculate $\bar{x}(k)$.

The event triggering rule Eq. 4.12 is interpreted as

$$\begin{cases} \|z_2(k) - \bar{x}(k)\| \leq \sigma \|\bar{x}(k)\| \Rightarrow \text{(Non-triggered)} \\ \|z_2(k) - \bar{x}(k)\| > \sigma \|\bar{x}(k)\| \Rightarrow \text{(Triggered).} \end{cases} \tag{4.19}$$

A constant $\sigma \geq 0$ is specified as a threshold. The trigger rule is running concurrently with the state dynamics and determines which mode the system operates in at each time $k > 1$. The condition $\|z_2(k) - \bar{x}(k)\| > \sigma \|\bar{x}(k)\|$ is defined as the 'event', indicating that predictor's estimation accuracy is unsatisfied.

Advantages of adding predictors can be summarized into two aspects. For the first aspect, compared to Eq. 4.12, $x(k^-)$ is replaced by $z_2(k)$ to prevent the zero-order-hold in 'non-triggered' mode. Thus, instead of monitoring the error between the value of held state and real-time state, the triggering rule evaluates the state variables error between the predictor and the observer. Otherwise, the controller keeps using the last stored $x(k^-)$ to calculate u, and u is unchanged until 'triggered' mode. For the second aspect, since u can be calculated by $z_2(k)$ in 'non-triggered' mode, it indicates that the communication is not necessary to be triggered at every moment. In other words, the predictors help save the communication resources.

A task divider is added to divide a complicated task into multiple sub-tasks that each sub-task runs in an individual sub-frame. As shown in Fig. 4.3, one sub-task starts when the robotic fish is swimming straightly. A regulation control makes the robotic fish turn once to narrow the tracking error down. When the tracking error is close to zero, as the robotic fish is swimming straightly again, a new sub-task is assigned for the next maneuvering. Because the fish model is based on velocity perturbations in Eq. 3.19 during straight swimming, the model in Eq. 3.26 is considered to be deviated from the nonlinear model Eq. 3.8 at the equilibrium condition that $\Delta r = 0$ and $\Delta\delta = 0$. The robotic fish is controlled to approach the equilibrium condition in each sub-task. When a new sub-task is assigned, $\bar{x}$ and z_i is reset because $\Delta r \approx 0$ and $\Delta\delta \approx 0$. Thus, the error caused by the unmodeled dynamics and disturbance will not be inherited to the next sub-task.

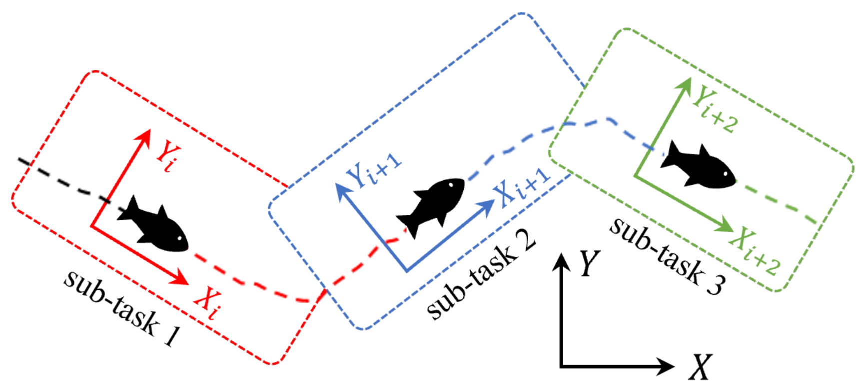


Figure 4.3: The illustration of task divider.

4.2.3 Simulations

In the simulation, the nonlinear fish dynamics is simulated using a 2nd-order Runge-Kutta method described in Eq. 3.29. The simulation assumes that robotic fish's position is obtained through an external sensor at a constant sampling rate. Yaw angle $\psi(k)$ is calculated using position data. The position acquisition in real circumstances may contain sensing errors, and the yaw angle calculation is highly disturbed by fish head's swing movement. A Gaussian noise with $\pm$ 0.02 m magnitude is added to the obtained fish position. Control objective is to control robotic fish's swimming direction to a desired reference yaw angle ψ_d. In simulations, the robotic fish starts from standstill and swims with a constant flapping frequency of 2 Hz. After 3 seconds of straight swimming, the controller starts to steer the robot. A constant communication rate (CCR) of 2 Hz is the default sampling rate. The controller design is based on the linear model Eq.4.4 and observer Eq.4.14, with

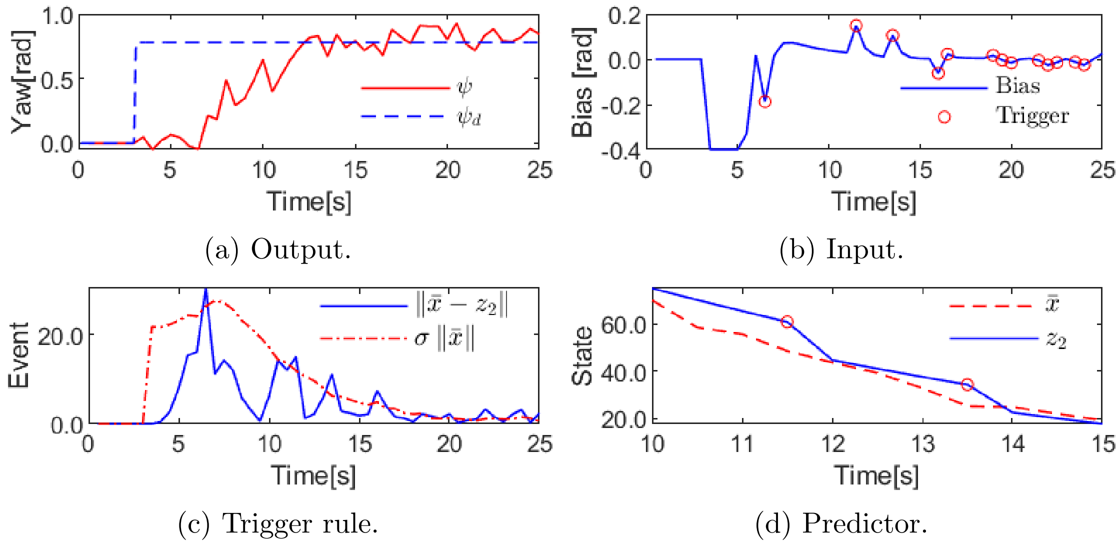
$$\begin{aligned} &L = [-36.23, -76.05, -40.04]^T, \\ &K = [3.45, -3.04, 2.67], \quad \text{and} \quad H = 0.05. \end{aligned} \tag{4.20}$$

as the coefficient matrices.

In Fig. 4.4a, the PETC with $\sigma = 0.2$ to track the ψ_d =0.25π rad is shown. Its bias control input u(k) is plotted as a solid line in Fig. 4.4b. We view the reception of $\bar{x}(k)$ as a triggered communication from the remote controller to the robot. We note that the PETC has a total of 13 triggers during the 25 second simulation. The 'event' plot Fig. 4.4c shows how the event is triggered in terms of (4.19) in Section 4.2.2. When the trigger is applied, the value of $\|z_2(k) - \bar{x}(k)\|$ is immediately lowered. More specifically, the state estimate comparison (the second state) between the observer and predictor from 10 s to 15 s is shown in Fig. 4.4d. It shows predictors' estimation is corrected when the trigger is applied.

Fig. 4.5 shows how the PETC behaves with various values of σ. It is clear that σ=0.1 causes the most triggers, from Fig. 4.5a. A smaller σ leads to not only a lower threshold, but also more frequent communication to constrain the $\|z_2(k) - \bar{x}(k)\|$. In the current circumstance, the external disturbance is not dramatic, σ value can be selected based on how much communication resources need to be saved.

While the PETC can reduce the number of triggers and save communication resources, its performance can also be maintained when the communication sampling



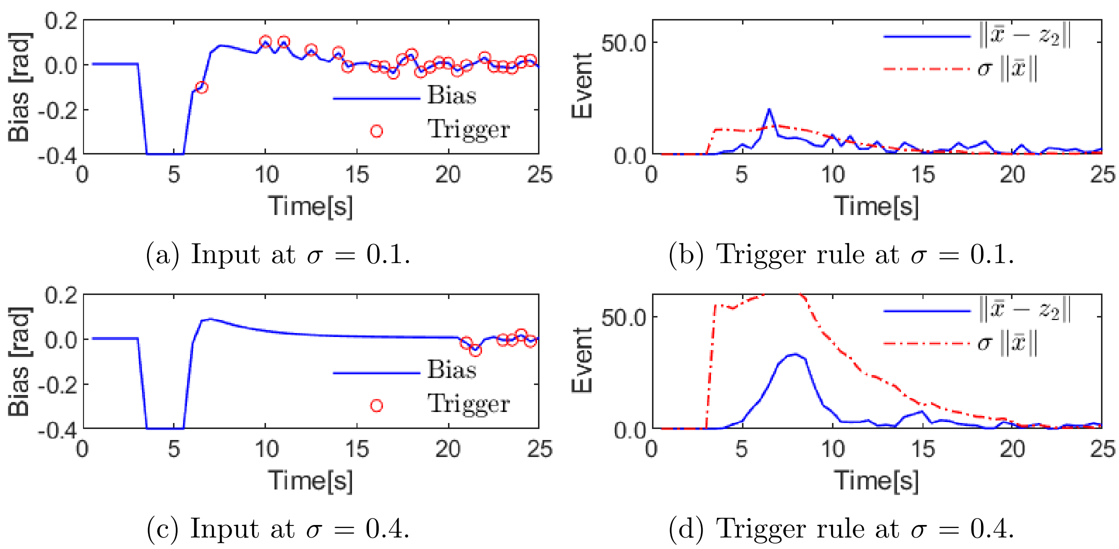
(a) Output.

(b) Input.

(c) Trigger rule.

(d) Predictor.

Figure 4.4: Simulation result of PETC for a step reference.



(a) Input at $\sigma = 0.1$.

(b) Trigger rule at $\sigma = 0.1$.

(c) Input at $\sigma = 0.4$.

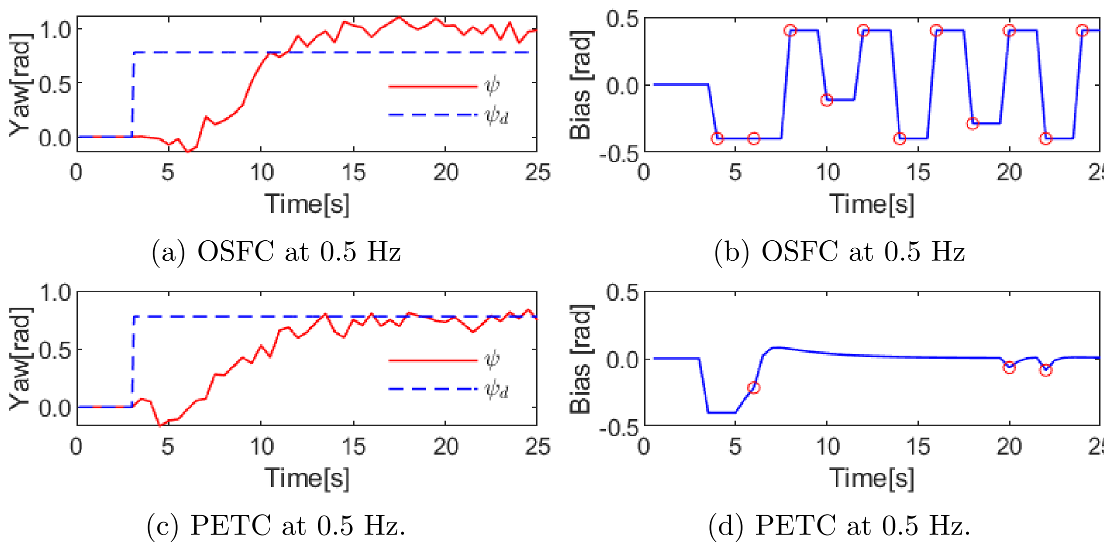
(d) Trigger rule at $\sigma = 0.4$.

Figure 4.5: Simulation result of PETC with different σ.

rate drops. The controller is tested at a reduced communication rate (RCR) case, in which the communication sampling rate drops to 0.5 Hz. An OSFC is designed for comparison. When communication is unavailable, we consider the situation is equivalent to a non-triggered mode. Hence, the control law for OSFC can be represented as

$$u(k) = \begin{cases} -K\bar{x}(k^-) + Hq_3(k^-) & \text{(Non-triggered)} \\ -K\bar{x}(k) + Hq_3(k) & \text{(Triggered)}, \end{cases} \tag{4.21}$$

using the same definition of (k^-) in Eq. 4.9. Fig. 4.6a shows how the OSFC behaves in the RCR case. Because only 11 triggers are available and $u(k)$ is held constant between triggers, the OSFC has a large error in ψ and $u(k)$ exhibits oscillations. In Fig. 4.6c, the PETC in the RCR case is shown. A much smaller steady-state error in ψ is exhibited, and $u(k)$ has only 3 triggers with no obvious oscillation.



(a) OSFC at 0.5 Hz (b) OSFC at 0.5 Hz

(c) PETC at 0.5 Hz. (d) PETC at 0.5 Hz.

Figure 4.6: Simulation results of OSFC and PETC at 0.5 Hz communication rate.

4.2.4 Experiments

In experiments, the robot swam in an above-ground swimming pool that was 10 meters in length and 5 meters in width. The observer and trigger-rule ran on a laptop computer which communicated with the robotic fish through Wi-Fi. Robotic fish's position was obtained through computer vision using a camera installed 5 m above the pool. The source of sensing noises are errors in computer vision and waves in the swimming pool. The feedback gain was adjusted to $K = [3.7, -3.33, 2.93]$ to account for the model uncertainty. The controller was tested by a comprehensive task. The first sub-task requires the robotic fish to steer to 0.25π rad in 15 seconds with a 2 Hz communication rate. The second sub-task requires the robotic fish to steer to 0.5π rad with a 0.3 Hz communication rate. Besides PETC and OSFC, we implemented a PID controller with $K_p = 0.6$, $K_i = 0.005$, and $K_d = 0.4$ for comparison.

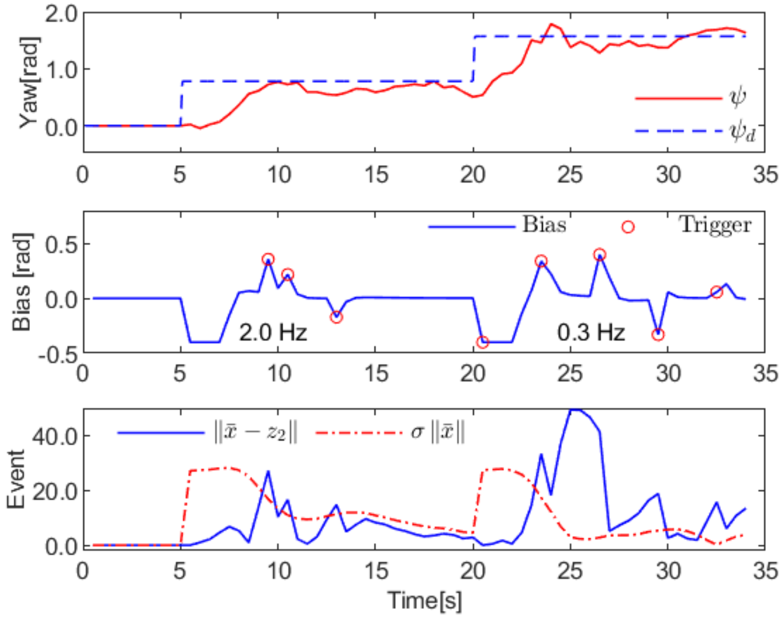
In Fig. 4.7a, the PETC with $\sigma = 0.2$ is shown. The desired and achieved yaw angles are shown in the first plot, and the bias control input $u(k)$ is plotted as a solid line in the second plot. Red circles represent each trigger. The PETC needs 8 triggers, and ψ response has a negligible overshoot with an average of 9 s settling time. The 'event' subplot shows that the triggering rule makes the prediction error converge to zero in CCR case. When the second sub-task begins, the prediction error is reset to zero. In the second sub-task, the trigger rule has less capability to constrain the prediction error. Hence, the output shows more overshoots and oscillations. However, the PETC still makes $u(k)$ converge to zero. The experimental results using OSFC are shown in Fig. 4.7b. The controller received $\bar{x}(k)$ from the observer at every communication instance after 5 seconds of straight swimming. We consider the reception of $\bar{x}(k)$ as a triggered communication from the remote controller to the robot, and also use a red circle to represent each trigger. In CCR case, we note that the OSFC triggers 30 times at 2 Hz communication rate and has a good performance. After the communication rate dropping to 0.3 Hz, the OSFC needs only 5 triggers, and $u(k)$ is held constantly between two triggers. As a consequence, there is a more obvious oscillation in the yaw plot after 20 seconds. The PID control exhibits a similar result in Fig. 4.7c.

Experimental trajectories for PETC, OSFC, and PID are plotted in Fig. 4.8a. The robotic fish starts from an initial yaw angle of 0 degrees. The value of ψ_d is selected so that the robotic fish is firstly required to track a 45-degree heading angle, followed by a 90-degree heading angle. The yaw oscillation of the OSFC and PID in the RCR case are shown as green and blue lines. Snapshots of the robot using PETC are in Fig. 4.8b.

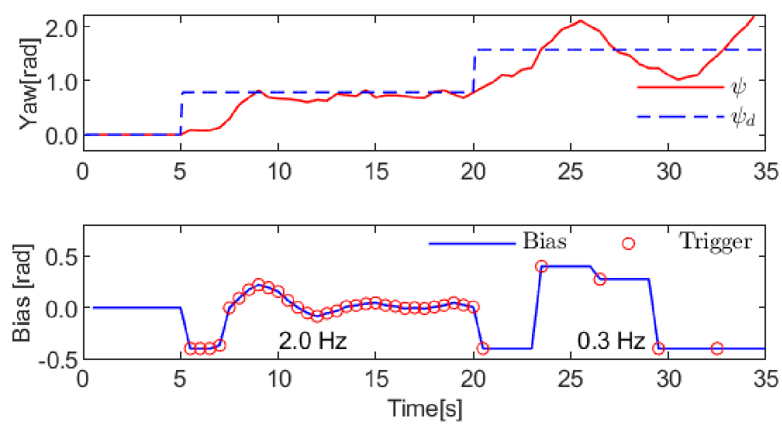
The control performances in terms of numbers of triggers and mean-square-error at steady-state are listed in Table. 4.1. Each control is tested in 5 repeated experiments to collect the data. Table. 4.1 shows that PETC sacrifices some tracking accuracy to save communication resources, and its performance is consistent when the communication sampling rate drops. The experiments exhibit consistency with the simulation.

Table 4.1: Performance of controllers

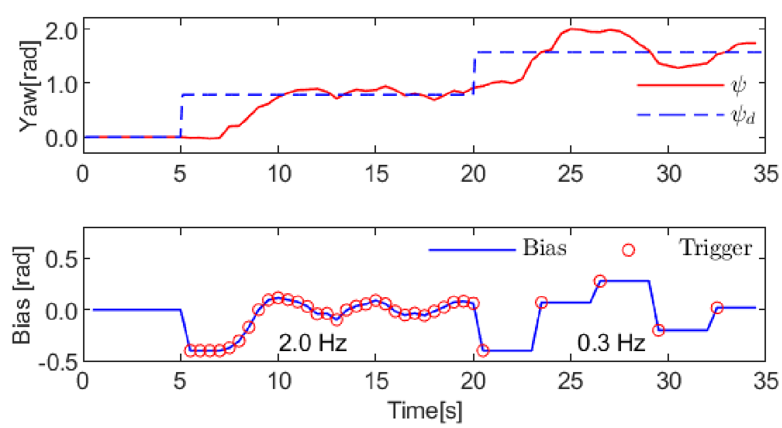
	Sub-task 1 (2 Hz)		Sub-task 2 (0.3 Hz)		Total	
	#Triggers	MSE	#Triggers	MSE	#Triggers	MSE
PETC	3	0.115	5	0.121	8	0.118
OSFC	30	0.051	5	0.476	35	0.263
PID	30	0.083	5	0.336	35	0.209



(a) Experimental result of PETC.

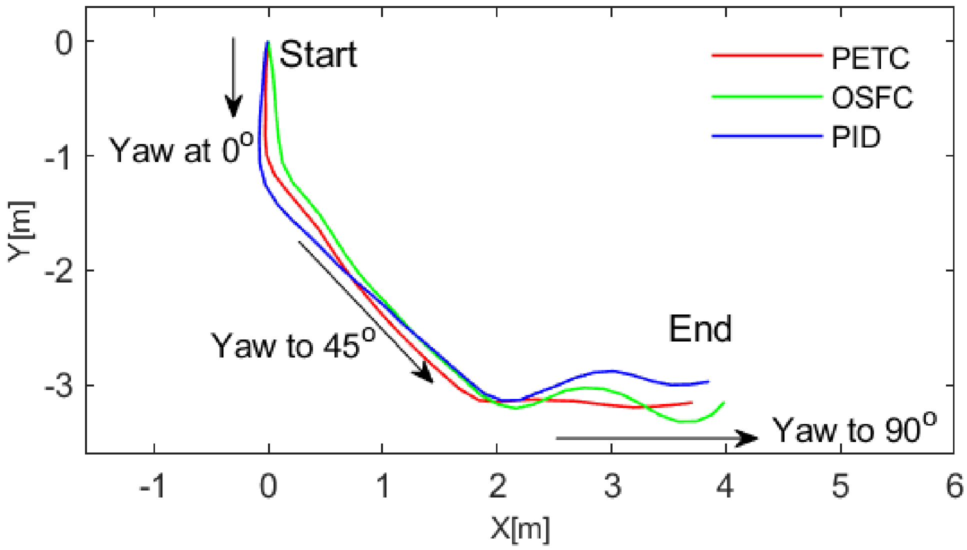


(b) Experimental result of OSFC.



(c) Experimental result of PID.

Figure 4.7: Experimental results.



(a) Trajectory of robot.

(b) Snapshots for PETC

Figure 4.8: Experimental trajectory and snapshots of robotic fish.

4.3 Chapter Summary

This chapter discusses the steering control of robotic fish based on a linear model. A transfer function is proposed in Chapter 3 to model robotic fish's yaw motion with respect to bias, which is controlled by a servo motor. A state-space model is derived from the transfer function to develop a state observer that could estimate the internal state vector using bias input and yaw output. Two types of controllers are designed: one is a traditional state-feedback controller with full state feedback from an observer, and the other is an event-trigger controller with predictors. The development of ETC addresses the challenge of communication inconsistency in robotic fish's deployment. The performance of error-based control and traditional OSFC is highly sabotaged when feedback information is not provided on time. The proposed ETC has predictors and integrators to estimate state-vector and output error without feedback information. An event trigger law monitors the error of state-vector estimation to determine the timing of providing feedback. Using ETC benefits the deployment of robotic fish in saving communication resources and increasing robustness. Under ETC, the feedback information is not necessary to be sent to the robot at every sampling time. Thus the communication resource is saved. When communication frequency is inconsistent, the bias of robotic fish is kept adjusted instead of zero-order-hold so that the steering control performance is maintained.

Chapter 5

Three-dimensional Robotic Fish

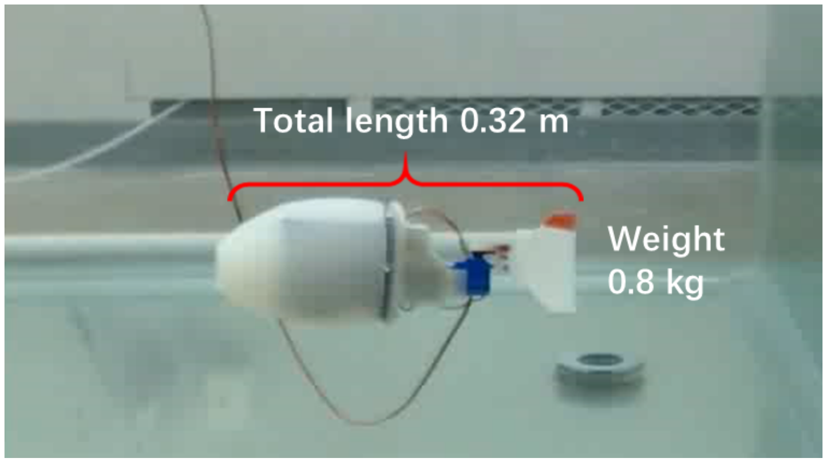
Three-dimensional (3D) maneuverable robotic fish is highly favorable because of its abilities in underwater maneuvering. The most existing robotic fish lack vertical maneuverability because the size of depth control mechanism is too big to fit in a small underwater robot. Well-developed underwater robots usually adopt the dynamics control method because this method changes depth quickly. The buoyancy control method should be complementary for extreme circumstances, such as weight changes and emergency ascent. Some robotic fish employ dynamic control methods using pectoral fin and center mass devices, and they have received good results. However, due to its compact size and limited energy storage capability, adopting the buoyancy control method is still challenging for robotic fish. In this chapter, a compact 3D maneuverable robotic fish focusing on the buoyancy control method is developed. It equips with an onboard water electrolyzer that generates gases for depth change. The design concept, dynamic model, and experiments are discussed to demonstrate the maneuverability of robotic fish in the vertical direction. The robotic fish's major component, buoyancy control device (BCD), is discussed with a state-of-art dynamic analysis to demonstrate its novelty and potential. In this chapter, the BCD design, modeling, and experiments were conducted by my labmate Alicia Keow. Based on her model and identified parameters, I developed the optimal trajectory planning considering input constraint and velocity constraint and integrated the BCD with robotic fish to achieve 3D maneuvering capability. I also conducted modeling and testing of 3D maneuverable robotic fish.

5.1 3D Robotic Fish

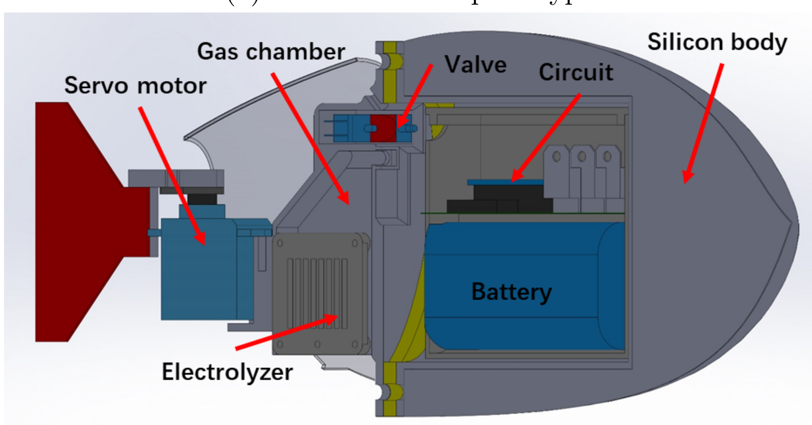
5.1.1 Overall structure of robotic fish

The overall design of the fish is shown in Fig. 5.1a. It has a total length of 0.32 m and a weight of 0.8 kg. The head is covered with silicon rubber to balance the weight. A single-joint caudal fin design is adopted because of its simplicity and reliability for two-dimensional swimming. The caudal fin is actuated by a servo motor which could output a sinusoidal pattern flapping with 30 degrees amplitude. The embedded circuit consists of a micro-controller and a 7.4 V Li-ion battery. The swimming speed of robotic fish is changed by fishtail's flapping frequency, and the steering is governed by the bias angle of flapping. An BCD is used for depth control.

BCD consists of three components: a gas chamber, a solenoid valve, and a IPMC-enabled electrolyzer. The IPMC-enabled electrolyzer is at the bottom of the gas chamber and submerged in water. IPMC is made by a Nafion membrane compressed by two acrylic frames. Two pieces of titanium, on both sides of the membrane, act as a pair of electrodes (Fig. 5.2). The Nafion membrane is an electron and anion insulator but is permeable to cations. When a voltage is applied across the electrodes, the accumulation of charge occurs across the electrodes. The permeable membrane in the IPMC allows the hydrogen ions to penetrate and combine with free electrons on



(a) 3D robotic fish prototype.



(b) 3D robotic fish assembly.

Figure 5.1: 3D robotic fish.

the cathode to form hydrogen gas. On the other side, oxygen gas is produced by the anode. When going up is needed, the electrolyzer will keep generating hydrogen and oxygen gas, and the gas is stored in the gas chamber. The gas displaces the volume occupied by water so that the overall system's volume increases. Therefore, the buoyancy is increased. The volume of the gas chamber is $9 \times 10^{-6} m^3$. The gas occupies only a slight chamber space. Therefore it does not impact the robotic fish's front-rear weight balance. When going down is needed, the stored gas will be released by the solenoid valve so that the volume is surrendered to water. Therefore and buoyancy is decreased. Overall, BCD uses a gas chamber and a solenoid valve to change the density of the entire system so that net vertical accelerations are produced.

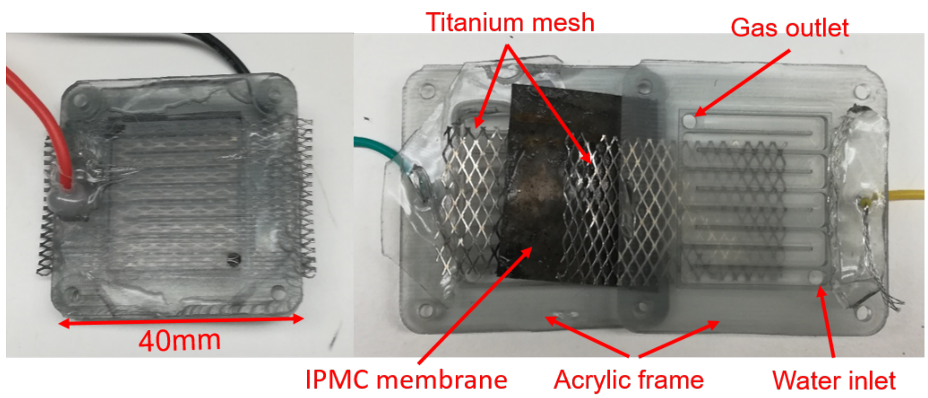


Figure 5.2: Electrolyzer assembly.

5.1.2 Dynamic model of robotic fish

A four-dimensional dynamic model is developed based on the Cartesian coordinate system with $O - XYZ$, standing for the inertial coordinate frame, and $G - xyz$, standing for body-fixed coordinate frame [85]. The following assumptions are given to simplify the model. The roll and pitch motions are not considered because they are unchanged and uncontrollable. Also, the body axis origin coincides with the center of mass, which is also the center of rotation. Let u, v, w be the linear velocity components and r be the z-axis angular velocity in body-fixed coordinate. ψ be the angle between coordinates (Fig. 5.3). The coordinates transformation equation is

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 & 0 \\ \sin\psi & \cos\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ r \end{bmatrix}. \tag{5.1}$$

In body-fixed coordinate, a simple rigid-body dynamics model is adopted , consid-

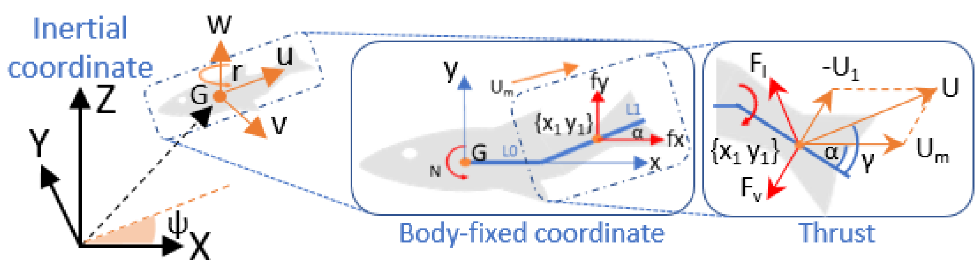


Figure 5.3: Cartesian coordinate system of 3D robotic fish.

ering F_x, F_y, F_z to be the force component at xyz-axis, and N to be the rotational moment

$$\begin{aligned} m(\dot{u} - vr) &= F_x, \\ m(\dot{v} + ur) &= F_y \\ m\dot{w} &= F_z, \\ I\dot{r} &= N. \end{aligned} \tag{5.2}$$

Because BCD doesn't change pitch angle, the robotic fish's vertical motion is separately analyzed from two-dimensional motion.

In two-dimensional motion, the external forces and moments in Eq. 5.2 can be sorted into two parts, namely, the thrust $f_{(\cdot)}$ from the fishtail and the drag force $D_{(\cdot)}$ on the fish body

$$F_{(\cdot)} = f_{(\cdot)} + D_{(\cdot)}. \tag{5.3}$$

Here we neglect the viscous resistance by assuming the fluid surrounding the fish is inviscid. The drag force is calculated based on the body's shape. Because the robotic fish belong to the category of single-joint robotic fish, thrust is obtained from the caudal fin's pure oscillating. A quasi-state model [29] that ignores the elasticity of caudal fin is adopted. In this method, only the force produced from the caudal fin is considered. The thrust produced by the caudal fin comes from two aspects: the lift force generated when the fin goes through the flow, and the fluid inertial force caused by added mass phenomenon.

In the body-fixed frame, as shown in Fig. 5.3, α represents the tail's flapping angle. Since the tail is made from a high stiffness plastic board, the thrust is considered generated at the center of mass of the caudal fin, whose coordinate $[x_1, y_1]$ is given by:

$$\begin{aligned} x_1 &= l_0 + l_1 \cos\alpha, \\ y_1 &= l_1 \sin\alpha. \end{aligned} \tag{5.4}$$

The fin's center point has velocity $U_1 = [\dot{x}_1, \dot{y}_1]$. It is assumed that the robotic fish is steady in a constant flow of U_m, which is regarded as the relative flow caused by the fish's forward swimming. Let U be the relative velocity to the flow at the center point of the tail (Fig. 5.3). The angle of attack γ of the tail is obtained as follows

$$\begin{aligned} U &= \sqrt{\dot{y_1}^2 + (\dot{x}_1 - U_m)^2}, \\ \gamma &= \alpha + \arctan\Big(\frac{\dot{y}_1}{\dot{x}_1 - U_m}\Big). \end{aligned} \tag{5.5}$$

In Fig. 5.3, the lift force f_l is perpendicular to U. Based on the added mass theory [97], the fluid inertia force f_v is perpendicular to the fin. These forces are as

$$\begin{aligned} f_l &= \pi\rho LCU^2 \sin\gamma, \\ f_v &= -0.5\pi\rho LC^2 \frac{\mathrm{d}(U\sin\gamma)}{\mathrm{d}t}, \end{aligned} \tag{5.6}$$

where ρ is the fluid density, L and C are the span and chord of the caudal fin, respectively. The resultant forces f_x and f_y acting on the body are obtained as

$$\begin{aligned} f_x &= (f_l \cos\gamma + f_v)\sin\alpha, \\ f_y &= (f_l \cos\gamma + f_v)\cos\alpha. \end{aligned} \tag{5.7}$$

We only consider the drag force from fish body. A_x and A_y are the projected body areas along the x and y axes. C_{Dx} and C_{Dy} are the respective drag coefficients. Thus the drag force components along the x and y axes are

$$\begin{aligned} D_x &= \frac{1}{2} C_{Dx}\, \rho\, A_x\, u^2, \\ D_y &= \frac{1}{2} C_{Dy}\, \rho\, A_y\, v^2. \end{aligned} \tag{5.8}$$

When the servo motor outputs a constant bias angle, the unbalanced lateral force will generate a torque N acting about the point G, as follows:

$$N = f_y x_1 - f_x y_1. \tag{5.9}$$

In the vertical direction, the displacement $Z = 0$ is defined at the water surface, and vertical velocity is denoted as w. While ignoring changes in pitch angle, the equation of motion is

$$\begin{aligned} m\dot{w} &= f_G - f_B - D_z, \\ f_B &= \rho g(V_1 + V_2), \\ D_z &= \frac{1}{2}\rho A_z C_{Dz} w|w|. \end{aligned} \tag{5.10}$$

f_G represents the gravity. f_B represents the buoyancy. Drag force D_z acts along the direction opposite to w, in which C_{Dz} is the drag coefficient, A_z is the projected body areas along the vertical direction. V_1 is the constant volume of the device, which includes all the in-compressible rigid parts. V_2 is the gas volume in the gas chamber. From experiments, following parameters are obtained.

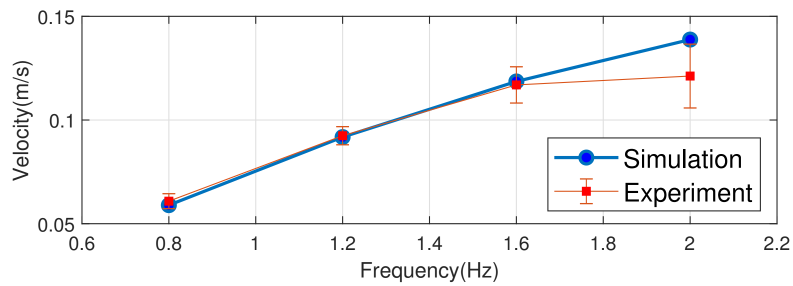
Table 5.1: Parameters of 3D robotic fish

l_0	l_1	L	C
0.05 m	0.04 m	0.09 m	0.05 m
V_1	$V_{chamber}$	A_x	A_y
$7.96 \times 10^{-4}\text{m}^3$	$9 \times 10^{-6}\text{m}^3$	0.01 m^2	0.03 m^2
A_z	C_{Dx}	C_{Dy}	C_{Dz}
0.015 m^2	0.4	2.2	0.2
Q_{in}	Q_{out}	I	m
$1 \times 10^{-7}\text{m}^3/\text{s}$	$1 \times 10^{-7}\text{m}^3/\text{s}$	0.025 kg/m^2	0.8 kg

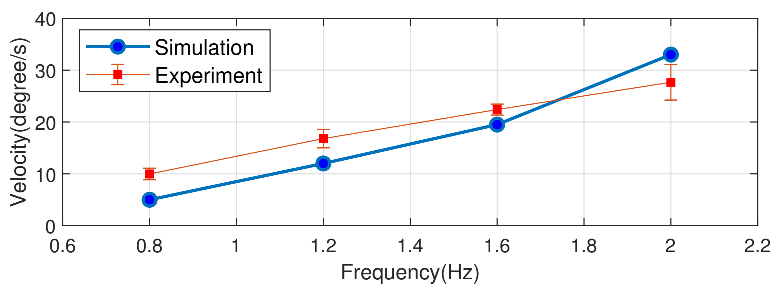
5.1.3 Forward swimming and turning test

While two-dimensional swimming is not the focus of 3D robotic fish, basic tests were conducted at the water surface to collect data for model validation. The swimming speed was tested in various flapping frequencies without bias angle. The steering test kept the bias angle at ±30 degrees and measured the angular velocity at various flapping frequencies. The simulation results, derived through the parameters from

Tab. 5.1, are compared with experimental data in Fig. 5.4a for forward swimming, and in Fig. 5.4b for steering.



(a) Forward velocity.



(b) Turning velocity.

Figure 5.4: Experimental and simulation data of robotic fish's two-dimensional swimming.

5.1.4 Three-dimensional maneuverability test

The 3D maneuverability test focuses on investigating the robotic fish's vertical maneuvering performance. Two types of tests were conducted to exhibit the fish's vertical maneuverability; the spiral motion combines the fish's turning motion with vertical motion. It requires the robotic fish to perform steering while descending or ascending. In Fig. 5.5, the robotic fish started from the bottom to rise while keep turning clockwise. In experiments, the voltage applied to the IPMC electrolyzer is 3 V, and the flapping frequency of the tail is 1 Hz. With a lower gas generation rate, it takes 44 s for fish to reach the surface while finishing a 900 ° turn. The snapshot presents a clear spiral path made by fish.

Figure 5.5: Experimental snapshots of robotic fish's spiral motion.

A vertical motion test examines the robotic fish's performance in changing and maintaining its depth. During that test, buoyancy needs to be adjusted by the corporation of electrolyzer and valve. In an open-loop control experiment in Fig. 5.6, snapshots record the fish making six depth changes in 3 minutes. The fish starts from the tank bottom and electrolyzer spends 30 s to surmount the volume insufficiency and start to lift the fish. Then solenoid valve releases a small amount of gas and lets the fish remain steady at 15 cm for 20 s. The electrolyzer then lifts the fish to 30 cm and makes it stay for another 20 s. After that, the fish moves up and down by keeping the electrolyzer running and periodically releases gas. This experiment proves the maneuverability of the robotic fish in the vertical direction and the effectiveness of this design.

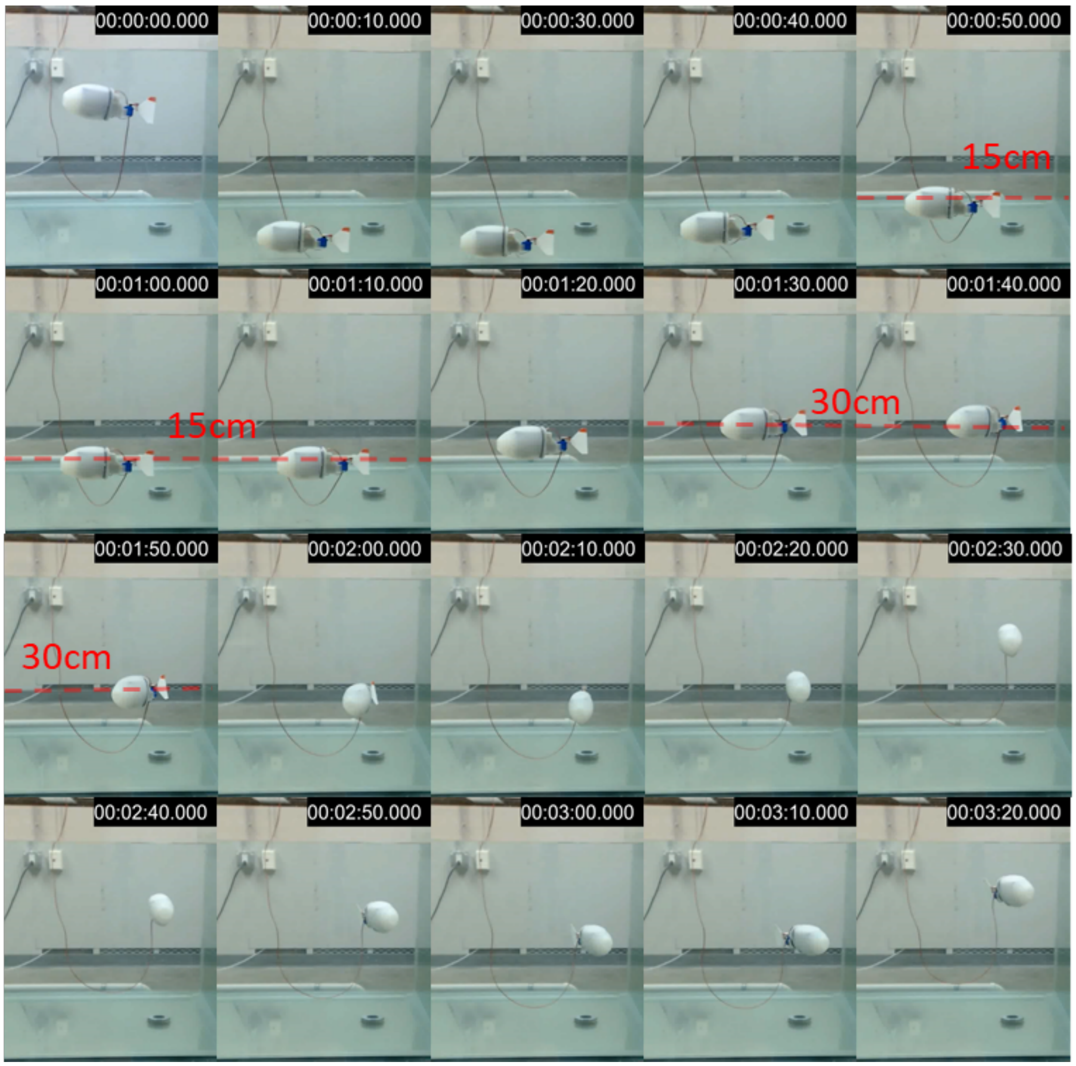


Figure 5.6: Experimental snapshots of robotic fish's vertical motion.

5.2 Buoyancy Control Device

BCD is the core component of 3D robotic fish. Its design and performance determine the mobility of the robotic fish in the vertical direction. Previously, a proportional-integral-derivative (PID) controller was developed by my labmate Alicia Keow for BCD to maintain depth, and a pressure sensor provided the depth feedback [3]. Due to the slow gas generation rate, BCD's maneuvering is restricted in order to avoid control saturation. This section discusses detailed BCD dynamics, as well as a one-dimensional trajectory planner. The trajectory planner is based on optimal control theory. It considers the constraint of system input and system state. The goal of this trajectory planner is to find the quickest path for BCD while staying bounded within an allowable input range.

BCD has three major components: a solenoid valve, a gas chamber, and a water electrolyzer. Water electrolyzer produces hydrogen and oxygen gases. It sits below a gas chamber and is immersed in the surrounding fluid. The gas chamber is designed to store the gas produced from the electrolyzer. Gases are accumulated at the top of the chamber due to gravity, displacing water and increasing buoyancy. The inlet of the solenoid valve locates at the top of the gas chamber, while its outlet connects to the exterior. Turning on the solenoid valve allows the gas to escape and decreases the buoyancy. The schematic of BCD is shown in Fig. 5.7.

The electrolyzer is actuated by voltage, which is interpreted by pulse-width modulation (PWM) to change the speed of gas generation. The gas generation rate is linearly related to applied voltage. At a steady-state, the flow of charge from the voltage source to the IPMC is proportional to the movement of hydrogen ions across the membrane, which also represents the rate of hydrogen gas generation. Experiments that run at the atmospheric pressure record the time of the electrolyzer filling a gas

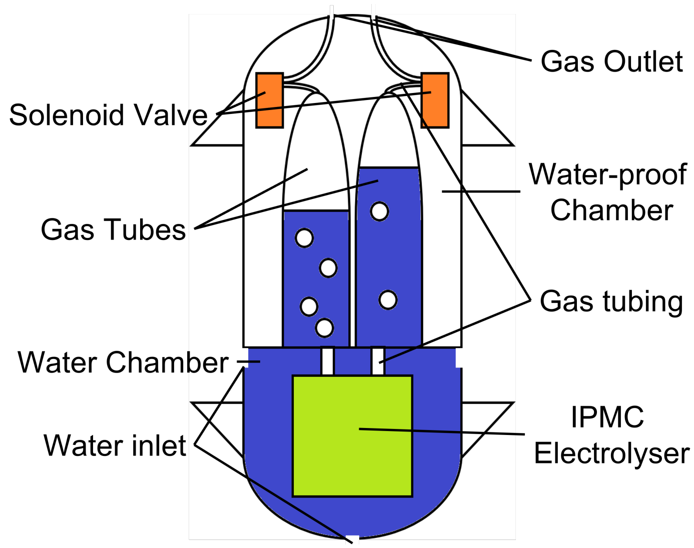
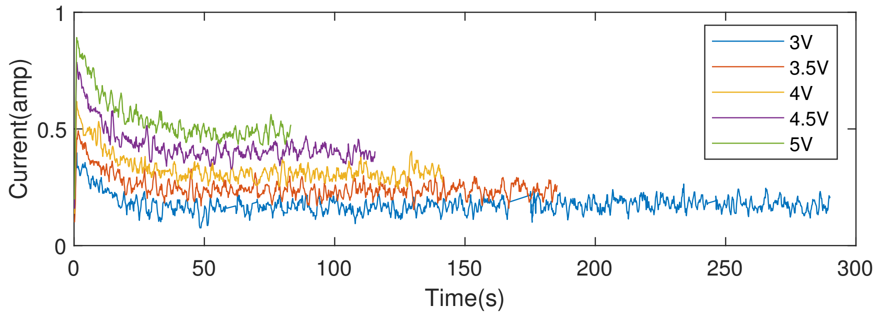
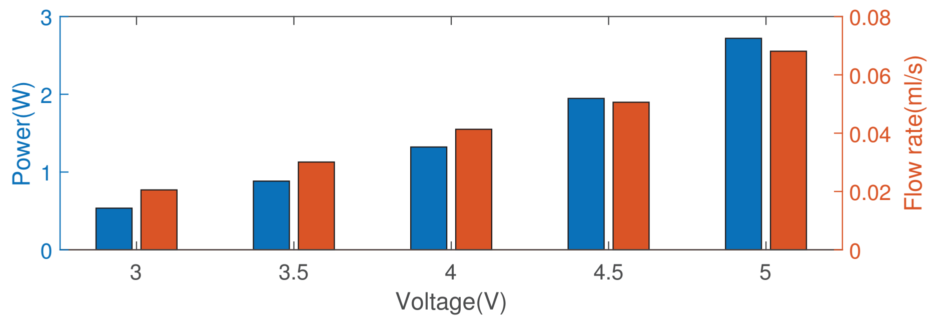


Figure 5.7: Schematic of BCD [3].

tube in various voltage conditions. The electrolyzer is tested from 3 V to 5 V at a 0.5 V step while recording the current and time needed for the electrolyzer to generate 6 ml gas. The result in Fig. 5.8a shows that at 3 V, electrolyzer needs 290 s to collect the gas, which denotes a gas generation rate of $2 \times 10^{-8} \mathrm{m}^3/\mathrm{s}$. Respectively, at 5 V, the electrolyzer needs 85 seconds to collect the gas, which denotes a gas generation rate of $7 \times 10^{-8} \mathrm{m}^3/\mathrm{s}$. Based on repeated measurements, the average gas generation rate is $5 \times 10^{-8} \mathrm{m}^3/\mathrm{s}$ at 4.5 V and the average gas release rate is $3.5 \times 10^{-6} \mathrm{m}^3/\mathrm{s}$. The power consumption at each voltage is displayed in Fig. 5.8b.



(a) Current strength versus voltage



(b) Power consumption and gas generation rate

Figure 5.8: Gas generation test result

5.2.1 BCD model

The model of BCD is similar to Eq. 5.10 in Z-axis with the vertical velocity defined as $\dot{Z}$ and acceleration defined as $\ddot{Z}$. The origin is $Z = 0$. The equation of motion is

$$
\begin{aligned}
m\ddot{Z} &= mg - f_B - D_z, \\
f_B &= \rho g(V_1 + V_2), \\
D_z &= C'_D \dot{Z}|\dot{Z}|,
\end{aligned}
\tag{5.11}
$$

where C'_D is a generalized drag coefficient. From the equation, the buoyancy is adjusted by changing the volume V_1 and V_2. The difference between V_1 and V_2 as following: V_1 includes all the rigid volume of the device, which is in-compressible and time-invariant; V_2 represents the volume occupied by gas, which is compressible and time-variant. It is well known that the water pressure is increased by depth, so the same amount of gas cannot provide constant buoyancy with the variation of depth. A necessary step in modeling the BCD is finding the instantaneous gas volume inside the chamber. From Chen's work [98], V_2 can be written as

$$V_2 = \frac{P_{atm}(V_{in} + V_3)}{P_{atm} + \rho g Z}, \tag{5.12}$$

where P_{atm} represents the atmospheric pressure, V_{in} is an arbitrary initial volume at the surface, and V_3 is the gas volume increment from the actuators if under atmospheric pressure. Combine Eq. 5.11 and Eq. 5.12 leads to

$$\ddot{Z} = g - \frac{\rho g}{m}[V_1 + \frac{P_{atm}(V_{in} + V_3)}{P_{atm} + \rho g z}] - \frac{C_D'}{m}\dot{Z}|\dot{Z}|. \tag{5.13}$$

Define V_0 as the standard volume needed to keep the system in neutrally buoyant condition. From Eq. 5.11, with zero acceleration and drag force, V_0 will be

$$V_0 = \frac{m}{\rho} - V_1. \tag{5.14}$$

Note that V_0 is a constant value. Whenever V_2 equals to V_0 in underwater, the system will be neutrally buoyant. Here the mass added to the system from the liquid-gas conversion is neglected. Inserting V_0 into Eq. 5.13, it can be reorganized as

$$\ddot{Z} = \frac{c_1 V_3 + c_2 Z + c_3}{P_{atm} + \rho g Z} - \frac{C_D'}{m}\dot{Z}|\dot{Z}|, \tag{5.15}$$

$$c_1 = -\frac{P_{atm}\rho g}{m}, \quad c_2 = \frac{V_0 \rho^2 g^2}{m}, \quad c_3 = \frac{\rho g P_{atm}(V_0 - V_{in})}{m}.$$

Define state variables as $x_1 = Z$, $x_2 = \dot{Z}$, and $x_3 = V_3$. The nonlinear dynamic function can be written as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = f(.) = \begin{bmatrix} x_2 \\ \frac{c_1 x_3 + c_2 x_1 + c_3}{P_{atm} + \rho g x_1} - \frac{C'_D}{m} x_2 |x_2| \\ u \end{bmatrix}. \tag{5.16}$$

Control input $u = dV_3/dt$ is the rate of change of the volume. At $f(.) = 0$, the equilibrium point x_{1eq} can be selected at any depth. For example, If the equilibrium point is selected at water surface, then $\dot{x}_1 = \dot{x}_2 = \dot{x}_3 = 0$, and $V_{in} = V_0$. If the equilibrium point x_{1eq} is not on water surface, V_{eq} denotes the required surface volume that makes system to be neutrally buoyant at x_{1eq}. In this case, V_{in} is

$$V_{in} = V_{eq} = V_0 + \frac{V_0 \rho g}{P_{atm}} x_{1eq}. \tag{5.17}$$

The explicit system behavior is analyzed around the equilibrium point, $[x_{1eq}, 0, 0]$, $V_{in} = V_{eq}$. Firstly, the input u is eliminated from the equation, thus $x_3 = 0$ and the c_1 can be removed from the equation. Let $x'_1 = x_1 - x_{1eq}$ represent the depth variation regarding to the equilibrium depth. Hence, the system is modified to be

$$\begin{bmatrix} \dot{x}'_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{c_2 x'_1}{P_{x_{1eq}} + \rho g x'_1} - \frac{C'_D}{m} x_2 |x_2| \end{bmatrix}, \tag{5.18}$$

and modified equilibrium point changes to $x_0 = [0, 0]$.

The equilibrium x_0 is unstable if existing a decrescent function $V : R^+ \times R^n \to R$

such that for initial time t_0.

(i)$\dot{V}(t,x)$ is a locally positive definite function

(ii)$V(t,0) = 0 \ \ \forall t \geq t_0$

(iii)$\exists$ a point $x_0 \neq 0$ arbitrarily close to 0 s.t.$V(t_0,x_0) \geq 0$

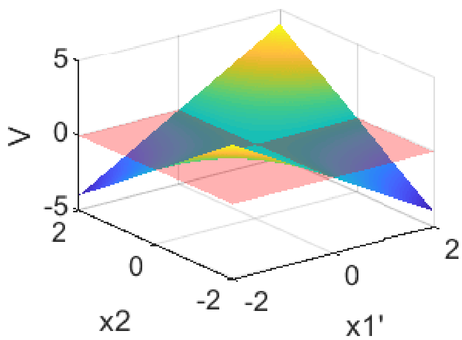
In this analysis, the system is in time invariant. We define a function $V = x_1' x_2$ that has $V(0) = 0$ and $V(x) > 0$ for some x which are arbitrarily close to 0. Take the time derivative on V and we have

$$\dot{V} = \frac{c_2 x_1'^2}{P_{x_{1eq}} + \rho g x_1'} + x_2^2 (1 - \frac{C_D'}{m} x_1' \text{sign}(x_2)). \tag{5.19}$$

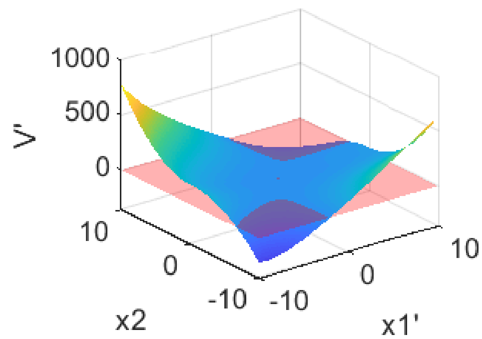
If $|x_1'| < \min\left\{\frac{P_{x_{1eq}}}{\rho g}, \frac{m}{C_D'}\right\}$, then $P_{x_{1eq}} + \rho g x_1' > 0$ and $1 - \frac{D}{m} x_1' \text{sign}(x_2) > 0$. One can conclude that $\dot{V}$ is locally positive definite over the ball B_r and $r \in (0, \min\left\{P_{x_{1eq}}/\rho g, \frac{m}{C_D'}\right\})$. As shown in Fig. 5.9b, $\dot{V}$ is locally positive definite around the origin. This instability conclusion also can be checked by the phase plot, for $x_{1eq} = 5$ as shown in Fig. 5.9c. Any initial state located in the second and fourth quadrant will try to approach the equilibrium point. After that, the state will shift to the first and third quadrant and move to infinity. This instability analysis indicates that any disturbance will lead the stable BCD to be either sinking to the bottom or jumping out of the surface.

5.2.2 Velocity constraint

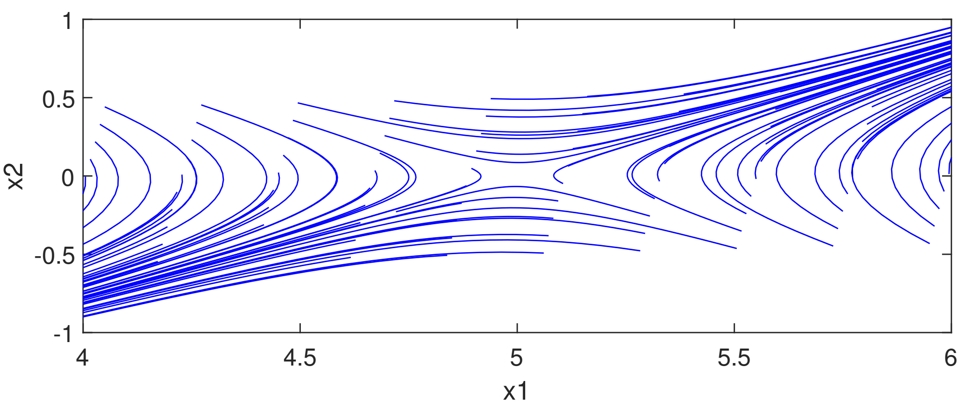
Eq. 5.19 identifies the system Eq. 5.16 is unstable. A control saturation will lead the system to an irreversible failure, especially during diving. Identifying the system constraint is an essential step in designing the controller. The input constraint can



(a) Plot of V



(b) Plot of $\dot{V}$



(c) Phase portrait diagram at 5m

Figure 5.9: Stability analysis at equilibrium point.

be represented by u_{max}, which is the limitation of the actuator. When BCD is diving, the instantaneous gas volume is decreasing due to water pressure increasing, thus the electrolyzer needs to keep generating gas resisting volume loss. Diving velocity needs to be bounded to accommodate the input constraint. With considering the drag force, x_{2max} is reached at

$$\exists x_{2max} \in \mathbb{R} : |u| = u_{max} \text{ and } \dot{x}_2 = 0. \tag{5.20}$$

When diving, $x_2 > 0$, apply $\ddot{Z} = \dot{x}_2 = 0$ at Eq. 5.15, it turns out to be

$$\frac{c_1 x_3 + c_2 x_1 + c_3}{P_{atm} + \rho g x_1} = \frac{C'_D}{m} x_2^2. \tag{5.21}$$

Move x_3 to the left hand side

$$x_3 = \frac{C'_D}{P_{atm}} x_2^2 x_1 + \frac{C'_D}{\rho g} x_2^2 - \frac{\rho g V_0}{P_{atm}} x_1 - (V_{in} - V_0). \tag{5.22}$$

The control input u can be denoted by taking the time derivative of x_3

$$u = \frac{C'_D}{P_{atm}} x_2^3 - \frac{\rho g V_0}{P_{atm}} x_2 + 2\dot{x}_2 x_2 (\frac{C'_D}{P_{atm}} x_1 + \frac{C'_D}{\rho g}). \tag{5.23}$$

When $\dot{x}_2 = 0$, the third term of the equation can be removed, which leads to u_{max}

$$\left| \frac{C'_D}{P_{atm}} x_{2max}^3 - \frac{\rho g V_0}{P_{atm}} x_{2max} \right| = u_{max}. \tag{5.24}$$

With u_{max}, x_{2max} can be found by solving the cubic equation. x_{2max} sets up a boundary to guarantee device's safety. Gas generation still could stop BCD from descending when velocity within boundary. Whenever the velocity exceeds this boundary, it will drop to the bottom.

We use a simulation to explain the velocity boundary x_{2max}. The simulation is based on the model parameters from previous Keow's work [3]. The parameters are list in Tab. 5.2. In this simulation, velocity constraint is obtained at the initial depth at 5m. Based on Eq. 5.24, the maximum diving velocity x_{2max} is 0.0208 m/s. In testing the velocity constraint, the model is simulated at the an initial velocity range 0.0208 ± 0.005m/s. The position-velocity trajectories of this initial velocity range are shown in Fig. 5.10, in which a red line highlights the trajectory of the initial velocity $x_2(0) = x_{2max}$. From the phase plot, the upward trajectory $x_2(0) < x_{2max}$ (in blue) indicates the gas generation is able to recover the BCD from sinking, while the downward trajectory $x_2(0) > x_{2max}$ (in green) indicates the BCD will sink to the bottom. The trajectory of $x_2(0) = x_{2max}$ sits in between, indicating the BCD can temporarily maintain its velocity.

Table 5.2: Parameters of BCD

P_{atm}	g	m	ρ	C'_D
101325 Pa	9.8 N/kg	0.5 kg	997 kg/m^3	0.25
V_1	V_0	K_s	$\hat{u}_{max}$	Z_{eq}
4×10^{-6}m^3	5×10^{-4}m^3	0.3	1×10^{-6}m^3/s	5 m

5.2.3 Optimal jerk trajectory

The trajectory of BCD is designed by optimal control theory [99]. A virtual system is introduced and applied into a typical newton's second law system, which indicates the vertical maneuvering of the BCD. A virtual state is defined as $\hat{x}_3 = \ddot{z}$ to represent acceleration. Thus, a linear representation of the virtual system is given as

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = \hat{x}_3, \quad \dot{\hat{x}}_3 = \hat{u}. \tag{5.25}$$

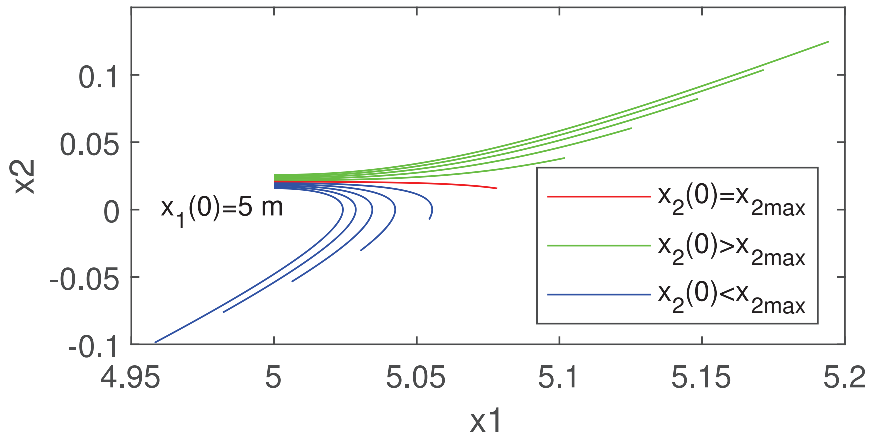


Figure 5.10: Phase plot near constraint.

Virtual control $\hat{u}$ is a jerk, which is the third time derivative of the displacement. One can take another time derivative in (5.16) to find the jerk. When acceleration is very small, $\hat{u}$ can be treated linearly to u close to initial position $x_1(t_0)$

$$\begin{aligned}\hat{u} &= \frac{c_1 u}{P_{atm} + \rho g x_1} + \frac{(c_2 P_{atm} - c_1 \rho g x_3)\hat{x}_3}{(P_{atm} + \rho g x_1)^2} + \frac{2C_D'}{m}\hat{x}_3 x_2 \\ &\approx \frac{c_1 u}{P_{atm} + \rho g x_1(t_0)}.\end{aligned} \tag{5.26}$$

The input constraint is transferred from u_{max} to $\hat{u}_{max}$ by Eq. 5.26. A safety coefficient $K_s \in (0,1)$ is chosen to make sure that there is enough margin before the system exceeding the control boundary. The constraint of the jerk $\hat{u}_{max}$ can be approximately represented as

$$|\hat{u}(t)| \leq \hat{u}_{max} K_s \approx \frac{c_1 u_{max}}{P_{atm} + \rho g x_1(t_0)} K_s. \tag{5.27}$$

The jerk trajectory aims to make BCD accelerate from zero and avoid unrealistic changes in velocity. The control target is minimizing the total time. The performance index is defined by

$$J = \int_{t_0}^{t_f} 1 dt = t_f - t_0. \tag{5.28}$$

Applying Lagrange multiplier method to adjoin the performance index and the state function, Hamiltonian equation H can be formed as

$$H(x(t), \lambda(t), \hat{u}(t)) = 1 + \lambda_1(t)x_2(t) + \lambda_2(t)\hat{x}_3(t) + \lambda_3(t)\hat{u}(t), \tag{5.29}$$

where λ_i is a Lagrange multiplier. Assuming x^*, λ^* and u^* are the optimal value of the state and control, according to the Ponryagin principle, the minimization of the Hamiltonian equation can be shown as

$$\begin{aligned} H(x^*(t), \lambda^*(t), \hat{u}^*(t)) &\leq H(x^*(t), \lambda^*(t), \hat{u}(t)) \\ &= \min_{|\hat{u}|<\hat{u}_{max}} H(x^*(t), \lambda^*(t), \hat{u}(t)). \end{aligned} \tag{5.30}$$

It leads to

$$\lambda_3^*(t)\hat{u}^*(t) \leq \lambda_3^*(t)\hat{u}(t), \tag{5.31}$$

where $|\hat{u}| < \hat{u}_{max}$, implying the minimum of the right side is

$$\min_{|\hat{u}|<\hat{u}_{max}} \lambda_3^*(t)\hat{u}(t) = -\hat{u}_{max}|\lambda_3^*(t)|. \tag{5.32}$$

Thus, the control law can be applied as

$$\hat{u}^*(t) = \begin{cases} -\hat{u}_{max}K_s & \lambda_3^*(t) > 0 \\ \varnothing & \lambda_3^*(t) = 0 \\ \hat{u}_{max}K_s & \lambda_3^*(t) < 0 \end{cases} . \tag{5.33}$$

It is clear that the control law follows the bang-bang method, which is one of the approaches to deal with the input constraint problem [100]. The sign of λ_3 is related to the boundary condition. The boundary conditions are $x(t_0) = [x_1(t_0), 0, 0]$, $x(t_f) =$

$[x_1(t_f), 0, 0]$, and $x_1(t_f) > x(t_0)$. The acceleration's direction needs to be changed twice, which requires jerk's direction to be changed three times. Define three change timings as T_1, T_2, T_3 and initial time as T_0. Hence, the acceleration, velocity, and displacement need be integrated from the jerk, along with three time periods

$$\hat{u}^*(t) = \begin{cases} \hat{u}_{max}K_s & T_0 \leqslant t < T_1 \\ -\hat{u}_{max}K_s & T_1 \leqslant t < T_2 \\ \hat{u}_{max}K_s & T_2 \leqslant t < T_3 \end{cases}$$
$$x(T_3) = (\int_{T_2}^{T_3} + \int_{T_1}^{T_2} + \int_{T_0}^{T_1})\dot{x}(t)dt + x(T_0). \tag{5.34}$$

Equation Eq. 5.34 is solved numerically by Broyden's method [101], which is

$$r_{n+1} = r_n - A_n^{-1}F(r_n), \tag{5.35}$$

where A is Jacobian matrix of r_n, then n_{th} is iteration of $[T_1, T_2, T_3]$. $F(r)$ is defined as

$$A_n = A_{n-1} + \frac{F(r_n)}{\|s_n\|_2^2}s_n^T,$$
$$F(r_n) = f(r_n) + f^{'}(r_n)(r_{n+1} - r_n) = 0,$$
$$s_n = r_n - r_{n-1},$$

where $f(r)$ is the function of $[T_1, T_2, T_3]$ after n iterations.

For a small displacement, the transit velocity may not exceed the constraint. In this case, a bang-bang method with two switching times is optimal. If the velocity is outside of the constraint due to the big maneuvering, the bang-bang method needs to be modified to the bang-bang-off-bang-bang (BBFBB) method. State constraint is

$$|x_2(t)| \leqslant x_{2max}K_s. \tag{5.36}$$

The jerk will be turned off when $|x_2(t)|$ reaches the $x_{2max}K_s$, which brings an off period and a pair of on-off timing. Therefore, a total of six control actions are needed: start-end timing T_0 and T_5, two switch timing T_1 and T_4, as well as on-off timing T_2 and T_3. The whole track is divided into three sections, two individual bang-bang trajectories and one off period. The boundary condition is

$$\begin{aligned}
&\text{Bangbang1}\begin{cases}\text{Initial} & = [x_1(T_0), 0, 0] \\ \text{Final} & = [x_1(T_2), x_{2max}K_s, 0]\end{cases} \\
&\text{Off}\begin{cases}\text{Initial} & = [x_1(T_2), x_{2max}K_s, 0] \\ \text{Final} & = [x_1(T_3), x_{2max}K_s, 0]\end{cases} . \\
&\text{Bangbang2}\begin{cases}\text{Initial} & = [x_1(T_3), x_{2max}K_s, 0] \\ \text{Final} & = [x_1(T_5), 0, 0]\end{cases}
\end{aligned}$$

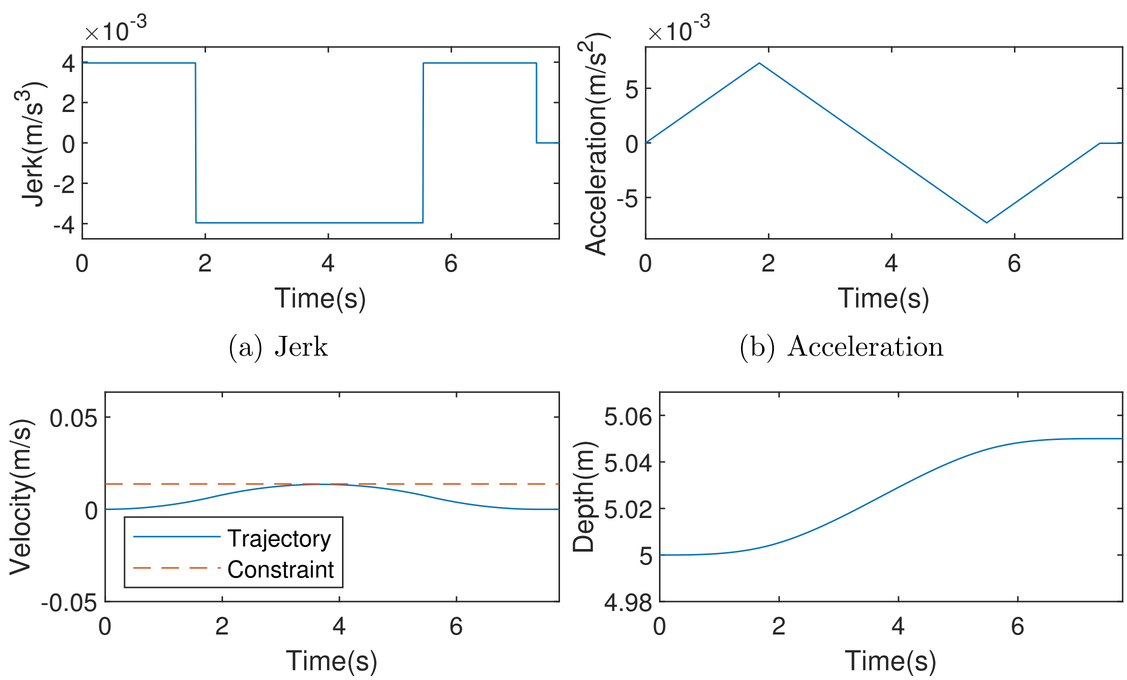
The control law is

$$\hat{u}^*(t) = \begin{cases} \hat{u}_{max}K_s & T_0 \leqslant t < T_1 \\ -\hat{u}_{max}K_s & T_1 \leqslant t < T_2 \\ 0 & T_2 \leqslant t < T_3 \\ -\hat{u}_{max}K_s & T_3 \leqslant t < T_4 \\ \hat{u}_{max}K_s & T_4 \leqslant t < T_5 \end{cases} . \tag{5.37}$$

T_{1-5} and the corresponding optimal trajectory $x^*(t)$ are solved from the numerical solution. T_1, T_2 and $T_5 - T_3$ can be found by integrating jerk to velocity $x_{2max}K_s$, also the displacement of bang-bang section. With the constant velocity during the off control, we calculate the time $T_3 - T_2$ and solve T_3, T_4 and T_5. The existing of on-off timing T_2 and T_3 increases the time consumption, resulting the trajectory are not global optimal anymore. But it is still the conditional optimal trajectory with the consideration of safety. All the state signals can be referred for controller design.

5.2.4 Trajectory planner

Here we demonstrate two types of trajectory. In the first demonstration, the bang-bang method plans the trajectory for BCD to dive from depth 5m to 5.05m, as shown in Fig. 5.11. At that time, the velocity does not exceed the velocity boundary, which is indicated by the red dashed line in the figure, and the jerk switches twice. In Fig. 5.12, the trajectory planner switch to BBFBB method to dive from depth 5m to 6 m. The trajectory is smooth and velocity is bounded. The acceleration and depth signal are the reference signal for the tracking controller.



(a) Jerk

(b) Acceleration

(c) Velocity

(d) Depth

Figure 5.11: Bang-Bang method

Fig. 5.13 shows a BCD experiment using the optimal trajectory [4]. The experiment was done by my labmate Alicia Keow. The trajectory was planned for

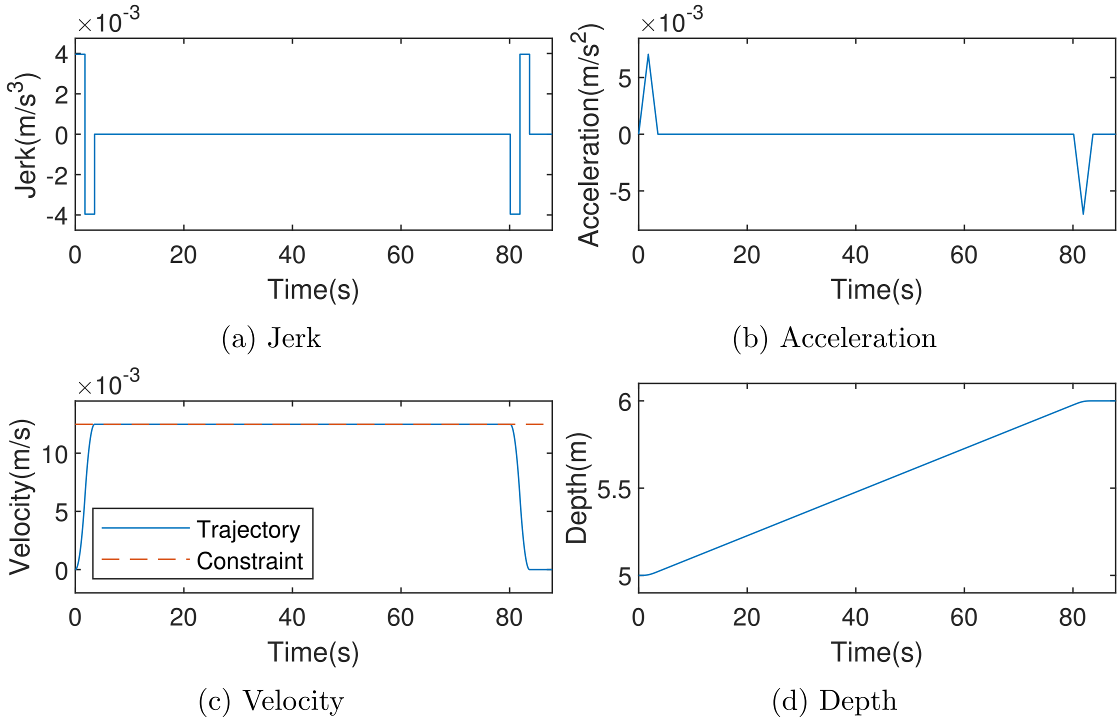


Figure 5.12: BBFBB method

BCD to move from $x_1(t0)$= 0.74 m to $x_1(tf)$= 0.3 m, with the input constraint of $|\hat{u}_{max}| = 1.2 \times 10^{-7}\mathrm{m}^3/\mathrm{s}$. Alicia Keow designed a PDA control to track the optimal trajectory. The experiment results show no overshoot or steady-state oscillation, and BCD traverses smoothly between two depths. The diving speed of BCD had not reached the velocity boundary because the displacement was small, that is why the bang-bang trajectory was applied.

5.3 Chapter Summary

This chapter discusses the study of robotic fish's maneuverability in the vertical direction. The proposed 3D maneuverable robotic fish equips with a BCD to adjust depth. The BCD has an IPMC enabled water electrolyzer that generates hydrogen and oxygen inside the robot. The robotic fish adjusts its overall density by storing

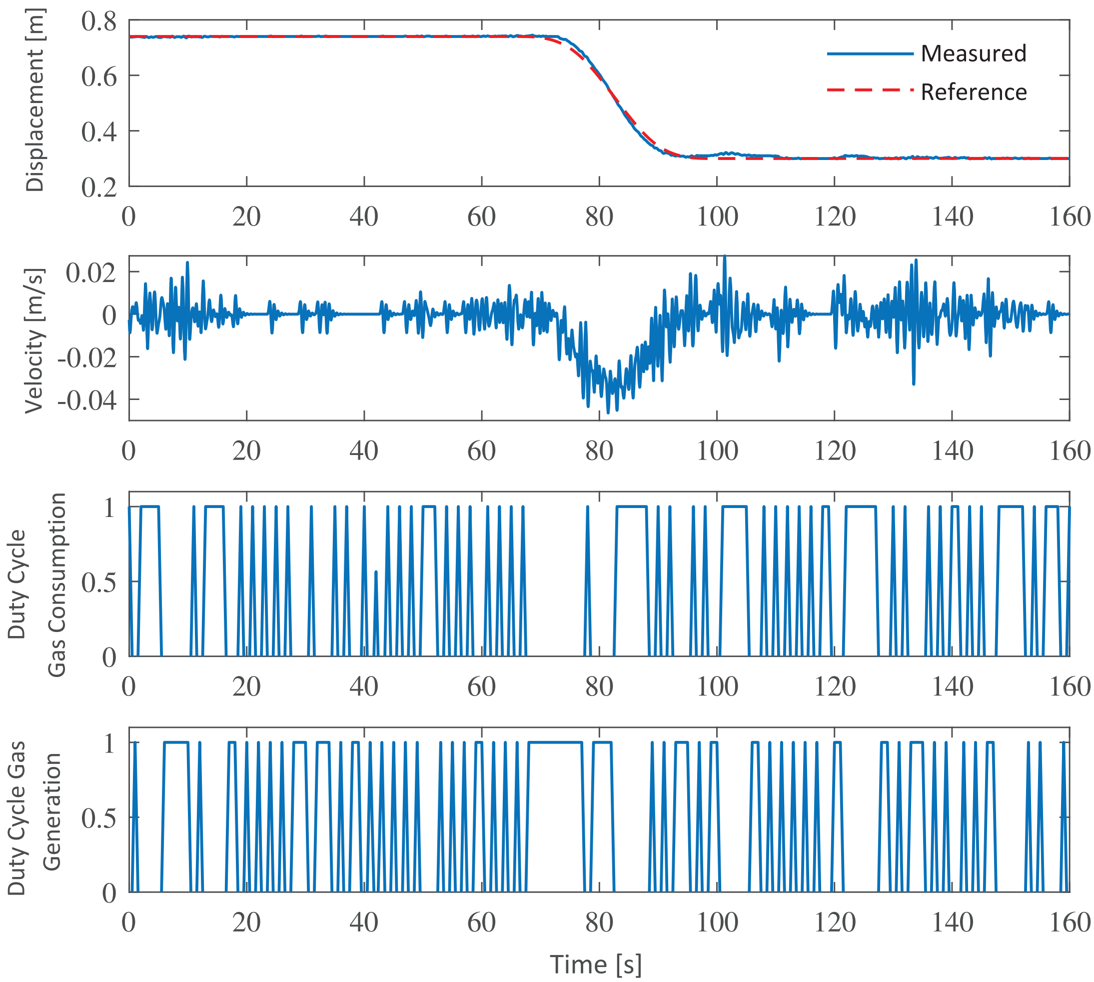


Figure 5.13: Experimental result of BCD with an optimal trajectory [4].

and releasing the gas to produce a net buoyancy to maneuver vertically. The robotic fish also equips a servo motor for two-dimensional maneuvering. In experiments, the robotic fish demonstrates diving to tank bottom and floating to water surface. It also demonstrates a spiral motion that steers the robotic fish while rising.

Considering the importance of BCD, BCD's dynamic model is further investigated to analyze BCD's controllability and stability. The BCD is identified as an unstable system with input constraint and state constraint. To ensure safety in vertical maneuvering, an optimal trajectory planner is proposed to prevent breaking the speed limit and saturating control input. The trajectory can slowly guide BCD to adjust depth smoothly, and the concept is tested in an experiment.

Chapter 6

Robotic Fish Enabled Motion Tomography

It is obvious that the ocean current plays an important role in the operation of marine vehicles. When marine vehicles are surveying in a confined area, the perception of the flow field of this area greatly helps the vehicles in path planning and improve energy efficiency. Traditional flow observations rely on information obtained through buoys and satellites. It is expensive and time-inefficient. Therefore, using the position and velocity information of the vehicles to predict the flow field can significantly improve work efficiency. Motion tomography is a technique that uses vehicles' navigation information to estimate the flow field. By collecting data from multiple vehicles, the estimation accuracy can be improved, and a larger flow field area can be observed. This technology provides a time-efficient and convenient way to monitor ocean currents.

Robotic fish is an ideal agent for environment sensing tasks due to its maneuverability and multi-functional. Using robotic fish to sense the flow field can greatly benefit transportation and environment study. To qualify the work, we add an active heading control (AHC) to moderate the passive heading change caused by the flow field. With the position and direction data collected from multiple trips, a vectorized flow map could estimate the flow field with considerable accuracy.

6.1 Mechanism

Motion tomography uses the trajectory of a moving vehicle to recover the flow field of a specific area and describe the field using a map of vectors. Since ocean currents affect marine vehicle's motion, the vehicle's trajectory is a combined effort of vehicle's propulsion thrust and flow-induced forces and moments. The predicted trajectory is a trajectory caused by vehicle's thrust only without surrounding flow. If there is no flow in the sensing area, the actual trajectory is equal to the predicted trajectory. The motion-integration error, denoted as d, is the shift between the actual trajectory and predicted trajectory. It is a result of flow-induced forces and moments with the integration of time. In motion tomography, the key step is to obtain the motion-integration error d. Thus, a first-order particle model is used to describe the kinematics of the moving vehicle with a constant velocity s_h. Define $\tilde{\mathbf{r}} \in \mathbb{R}^{2\times 1}$ as the position of the predicted trajectory, then vehicle's velocity is represented as

$$\dot{\tilde{\mathbf{r}}}(t) = s_h \begin{pmatrix} \cos\psi \\ \sin\psi \end{pmatrix}, \tag{6.1}$$

where ψ is the heading angle of the vehicle. It is easy to find that the predicted trajectory is derived by considering the vehicle is moving straightly and without affecting the flow field. Respectively, the vehicle's actual velocity is represented as a combined effect of predicted velocity and flow [102]

$$\dot{\mathbf{r}}(t) = s_h \begin{pmatrix} \cos\psi \\ \sin\psi \end{pmatrix} + \mathbf{f}(\mathbf{r}, t), \tag{6.2}$$

where $\mathbf{r} \in \mathbb{R}^{2\times 1}$ is the position of actual trajectory, and $\mathbf{f} \in \mathbb{R}^{2\times 1}$ represents the flow vector at position $\mathbf{r}$ during the period $t \in (t^0, t^f)$. Thus, the motion-integration error is denoted by subtracting the predicted trajectory from the actual trajectory

$$\mathbf{d} = \int_{t^0}^{t^f} \left(\dot{\mathbf{r}}(\tau) - \dot{\tilde{\mathbf{r}}}(\tau)\right) \mathrm{d}\tau = \int_{t^0}^{t^f} \mathbf{f}(\mathbf{r}, \tau)\mathrm{d}\tau. \tag{6.3}$$

Because of flow $\mathbf{f}(\mathbf{r}, t)$ is position-dependent and time-dependent, the actual trajectory of moving vehicle is high likely a irregular curve. In real circumstances, the vehicle takes time to go through the sensing field, and the motion-integration error is only accessible at $t = t^f$. The flow field cannot be observed in a time-variant manner. Two assumptions are made to simplify the motion tomography problem.

For the first assumption, the flow field is considered time-invariant in $[t^0, t^f]$, hence the flow field can be represented as $\mathbf{f}(\mathbf{r}, t) \to \mathbf{f}(\mathbf{r})$. The other assumption is that the vehicle velocity is considered to be s_h all the time. Therefore, the actual velocity of the vehicle, represented by s_{tr}, is a combination of vehicle velocity and the flow, and it is position-dependent,

$$s_{tr}(\mathbf{f}(\mathbf{r})) = \|\mathbf{r}\| = \left\| s_h \begin{pmatrix} \cos\psi \\ \sin\psi \end{pmatrix} + \mathbf{f}(\mathbf{r}) \right\|. \tag{6.4}$$

However, the actual trajectory, in most cases, is shown as a curve and defined by γ.

An arc length parameter ℓ is introduced for the actual trajectory that

$$\mathrm{d}\ell = s_{tr}(\mathbf{f}(\mathbf{r}))\mathrm{d}t. \tag{6.5}$$

Substituting Eq. 6.5 into Eq. 6.3, the integration error obtained through one trajectory is

$$\mathbf{d} = \int_{\gamma} \frac{\mathbf{f}(\mathbf{r})}{s_{tr}(\mathbf{f}(\mathbf{r}))}\mathrm{d}\ell. \tag{6.6}$$

To cover a large area, a feasible way is to discretize the domain D (sensing area) into P grid cells, with R columns and S rows, $P = RS$. A cell index $j = \{1 \cdots P\}$ is defined to indicate jth cell D_j. When a vehicle passes through a the sensing area, its position and velocity in each grid cell will be recorded, and the flow field within this cell is normalized by a vector started from the center of the cell. The assumptions mentioned previously are also validated in each cell, both the flow in each cell and vehicle's heading are constant. Thus, vehicle's velocity in jth cell is

$$s_{tr}^{j}(\mathbf{f}_j) = \left\| s_h \begin{pmatrix} \cos\psi_j \\ \sin\psi_j \end{pmatrix} + \mathbf{f}_j \right\|. \tag{6.7}$$

When the vehicle goes through jth cell, the length of trajectory within the cell is denoted as L_j that

$$L_j = \int_{\gamma[D_j]} \mathrm{d}\ell = \left\| \mathbf{r}_j^f - \mathbf{r}_j^0 \right\|. \tag{6.8}$$

The above equation indicates that the actual trajectory curve within one cell is approximated into a straight line, which is defined by the entering position $\mathbf{r}_j^0$ and the exiting position $\mathbf{r}_j^f$. Hence the entire actual trajectory in D is reconstructed by a polyline. As shown in Fig. 6.1, the entering position $\mathbf{r}_j^0$ is the exiting position $\mathbf{r}_{j-1}^f$ of the past cell, while the length is calculated as the distance between the entering and exiting positions within the cell.

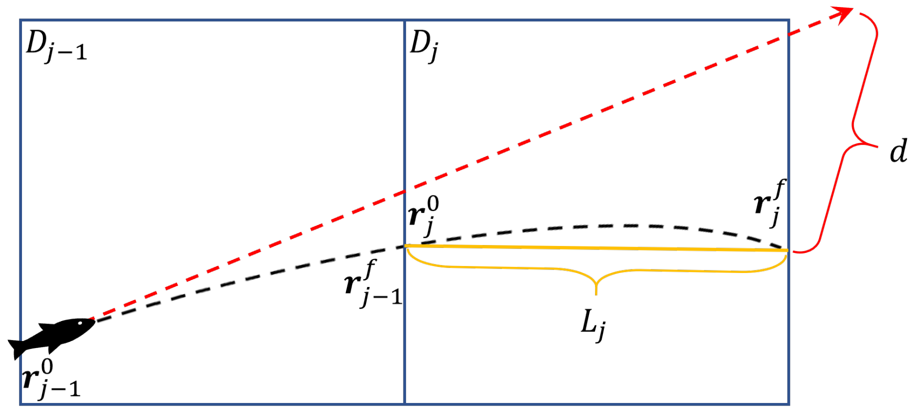


Figure 6.1: The trajectory of a vehicle in two connecting cells.

Based on the discretization setting, the integration error Eq. 6.6 is rewritten as

$$\mathbf{d} = \sum_{j=1}^{P} \frac{L_j(\bar{\mathbf{f}})}{s_{tr}^j(\mathbf{f}_j)} \mathbf{f}_j, \tag{6.9}$$

where $\bar{\mathbf{f}} = [\bar{\mathbf{f}}_x, \bar{\mathbf{f}}_y]$. $\bar{\mathbf{f}}_x = [f_{x,1}, f_{x,2}, \cdots, f_{x,P}]^T$ is x component of flow vector in each cell, and $\bar{\mathbf{f}}_y = [f_{y,1}, f_{y,2}, \cdots, f_{y,P}]^T$ is y component of flow vector for each cell. More specifically, the integration error is also separately accumulated in x direction and y direction,

$$\begin{aligned} d_x &= \sum_{j=1}^{P} \frac{L_j(\bar{\mathbf{f}})}{s_{tr}^j(\mathbf{f}_j)} f_{x,j}, \\ d_y &= \sum_{j=1}^{P} \frac{L_j(\bar{\mathbf{f}})}{s_{tr}^j(\mathbf{f}_j)} f_{y,j}. \end{aligned} \tag{6.10}$$

An accurate flow field estimation requires multiple trajectories to cover as many grid cells as possible. Assuming K trajectories are collected, and $i = \{1 \cdots K\}$ is the index for each trajectory, the Eq. 6.10 is rewritten as

$$\mathbf{d}_x = \mathbf{L}(\bar{\mathbf{f}})\bar{\mathbf{f}}_x, \mathbf{d}_y = \mathbf{L}(\bar{\mathbf{f}})\bar{\mathbf{f}}_y, \tag{6.11}$$

where $\mathbf{d}_x = [d_{x,1}, d_{x,2}, \cdots, d_{x,K}]^T$ and $\mathbf{d}_y = [d_{y,1}, d_{y,2}, \cdots, d_{y,K}]^T$. $\mathbf{L}(\bar{\mathbf{f}})$ is a memory matrix that collects the trajectory length in each grid cell for each trajectory

$$\mathbf{L}(\bar{\mathbf{f}}) = \begin{bmatrix} \frac{L_{(1,1)}(\bar{\mathbf{f}})}{s_{tr}^{(1,1)}(\mathbf{f}_1)} & \cdots & \frac{L_{(1,P)}(\bar{\mathbf{f}})}{s_{tr}^{(1,P)}(\mathbf{f}_P)} \\ \vdots & \ddots & \vdots \\ \frac{L_{(K,1)}(\bar{\mathbf{f}})}{s_{tr}^{(K,1)}(\mathbf{f}_1)} & \cdots & \frac{L_{(K,P)}(\bar{\mathbf{f}})}{s_{tr}^{(K,P)}(\mathbf{f}_P)} \end{bmatrix}.$$

Based on the knowledge of each trajectory and the motion-integration error, the motion tomography problem turns to an optimization problem in which a nonlinear function

$$\mathbf{L}(\bar{\mathbf{f}})\bar{\mathbf{f}} = \mathbf{d}, \tag{6.12}$$

is approximated with an optimal estimation of $\bar{\mathbf{f}}$ so that the estimation has the best match of the actual flow field. The Kaczmarz method [80] is employed to solve the Eq. 6.12 by iterating the following optimization process

$$\begin{aligned} \bar{\mathbf{f}}^{k+1} &= \underset{\bar{\mathbf{f}}}{\operatorname{argmin}} \frac{1}{2} \left\| \bar{\mathbf{f}} - \bar{\mathbf{f}}^k \right\|^2, \\ \text{subject to} \quad & d_i = \mathbf{L}_i(\bar{\mathbf{f}}^k)\bar{\mathbf{f}}, \end{aligned} \tag{6.13}$$

where k denotes the number of iterations.

In Fig. 6.2, a simulation demonstrates how motion tomography works using a moving vehicle driven by a first-order particle model. The sensing area is a 10 meter $\times$ 10 meter domain, and it can be divided in to 36 cells. An artificial flow field is added at the center of the domain, and it is defined by the following function

$$\mathbf{f}(\mathbf{r}) = h_1 e^{-h_2(X-X_0)^2 - h_2(Y-Y_0)^2}, \tag{6.14}$$

where h_1 and h_2 are parameters that define the magnitude and density of the field, X_0 and Y_0 are the centers of the field. The function builds an artificial vector field similar to the electrical field caused by a positive charge; the vector radiates out from the center with decreasing magnitude. When the vehicle moves within the domain, the artificial vector field pushes the vehicle away from the center. When the vehicle has been pushed out of the domain, in which the strength of flow is too weak to disturb the vehicle, the vehicle continues to move in its original direction. The predicted trajectory, indicated by a green arrow in Fig. 6.2, is calculated using the vehicle's velocity before entering the domain and the time vehicle stays in the domain, following by Eq. 6.1. The actual trajectory in every cell, indicated by black arrows, is calculated through Eq. 6.8. The motion-integration error is the difference between two trajectories.

Estimating the flow field of the entire domain needs the actual trajectory covering every cell. In Fig. 6.3a, a simulation demonstrates a vehicle traveling the flow field 24 times to cover every cell in the domain. It is worth noticing that the vehicle barely passes the center because the repulsion is strong in the center of the flow field. Using the motion-integration errors from all trajectories, the flow field is estimated through Eq. 6.13 and shown in Fig. 6.3b. The flow field within a cell is concentrated at the center of the cell and is represented by a vector. The entire flow field is represented by 24 vectors indicated by blue arrows, and the estimation result is very close to the artificial flow field (red arrows) using Eq. 6.14.

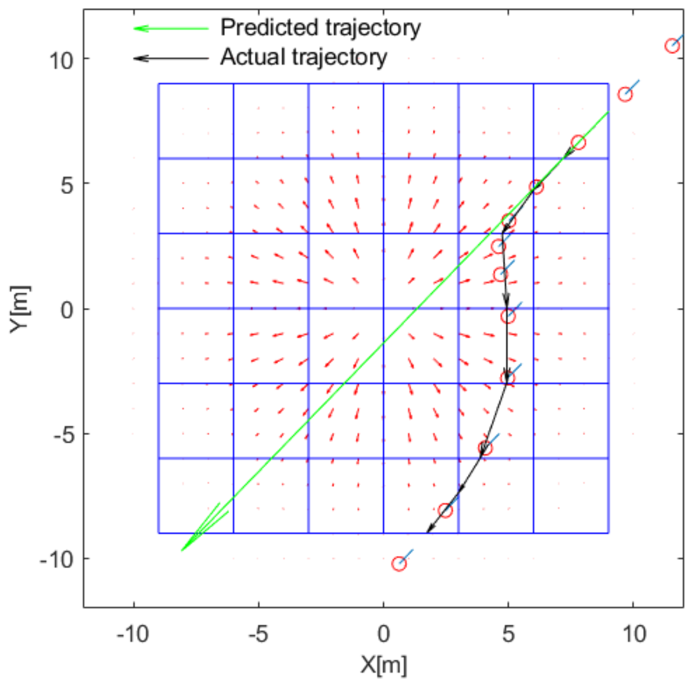
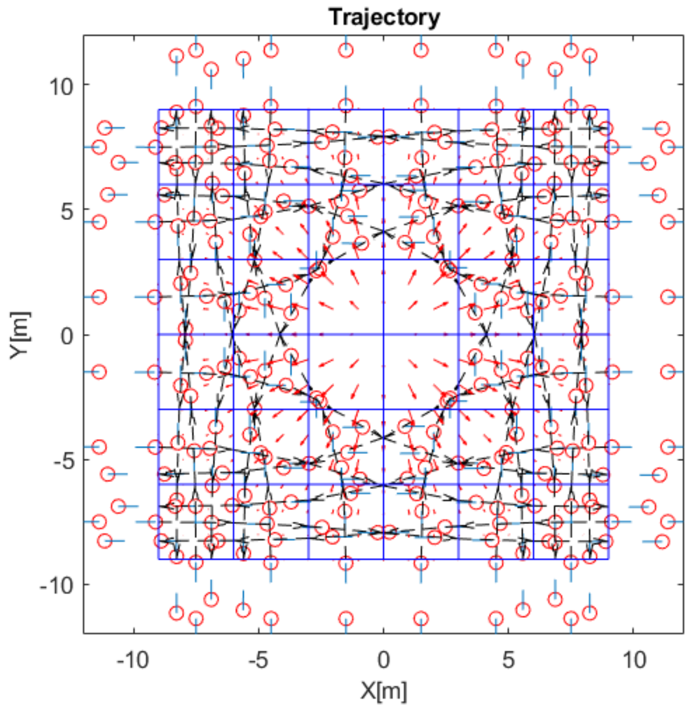


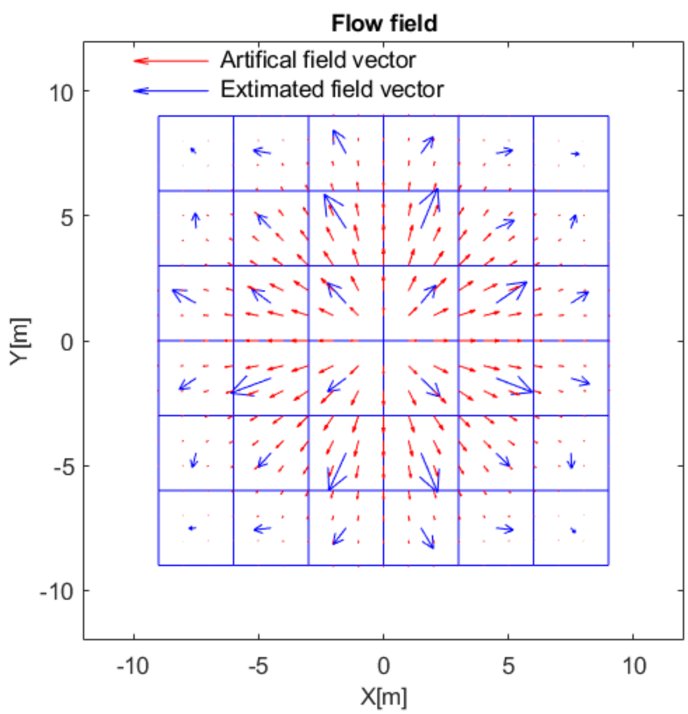
Figure 6.2: Illustration of the actual trajectory and the predicted trajectory.

6.2 Active Heading Control

In motion tomography, the predicted trajectory is derived through a first-order particle model, which ignores the rigid-body rotation. The first-order particle model is applicable if the sensing field is much larger than the vehicle. In this case, the flow field around the vehicle can be considered uni-directional, and the flow-induced force is uniformly acting on the vehicle. Thus, the flow would not lead to a rotational force, and the vehicle only drifts without heading change. However, in some environments, such as inland rivers and lakes, the sensing area is small, and the flow field is chaotic. The flow-induced force may not uniformly act on the vehicle, and the flow-body interaction can cause a rotational force that turns the vehicle. In this scenario, the assumption of actual trajectory in Eq. 6.13 is no longer standing. Also, the heading change of vehicle can cause an extra motion-integration error and deteriorate the



(a) Illustration of the multiple trajectories.



(b) Result of flow field estimation.

Figure 6.3: Motion tomography using 24 trajectories.

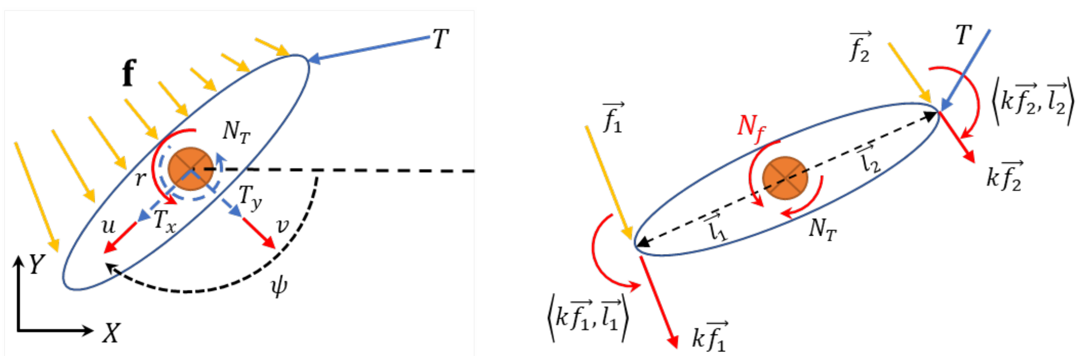
accuracy of motion tomography. Therefore, the vehicle's rigid-body dynamics should be considered when the sensing area is small or the flow field is irregular.

Considering that we use a robotic fish to map the flow field of a confined area, the robot's motion of dynamics can be represented by a simple three-degree-of-freedom rigid body dynamic model as shown in the following form

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} X \\ u \\ Y \\ v \\ \psi \\ r \end{bmatrix} = \begin{bmatrix} u\cos(\psi) - v\sin(\psi) \\ \frac{T_x - X_{|u|}u + mvr}{m} \\ u\sin(\psi) + v\cos(\psi) \\ \frac{T_y - Y_{|v|}v - mur}{m} \\ Tr \\ \frac{N_T + N_f - N_{|r|}r}{I} \end{bmatrix}. \tag{6.15}$$

The indexes follow the definition in Fig. 3.1. The rigid body dynamics considers fluid drag, Coriolis force, and rotational force. Assuming that the flow applied to fish body is not uniformly distributed, as shown in Fig. 6.4a, the flow-induced forces bring not only a displacement, but also a moment about the fish inertia. Two assumptions are made to establish a simple model for the dynamics of nonuniform flow toward a rigid body. The first one is that, the nonuniform flow is summarized as two major flow vectors that separately work at the fish head and tail. The net force induced by these two flows causes a rotational force N_f that leads to the fish rotation. The other assumption is to assume the flow-induced force has the same direction as the flow vector, and its strength is proportional to the vector's magnitude. With these two assumptions, we can model the flow-induced rotation in Fig. 6.4b. $\vec{f_1}$ indicates the flow vector applied to the fish head. Respectively, the force vector corresponding to $\vec{f_1}$ is represented by $k\vec{f_1}$. In addition, $\vec{f_2}$ and $k\vec{f_2}$ represent the flow velocity vector and force vector acting on fishtail. One can assume that the acting spot in head and tail has the same distance to the center of mass, $\vec{l_1}$ is the position vector from the center of mass to head, and $\vec{l_2}$ is the position vector from the center of mass to tail. $\vec{l_1}$ and $\vec{l_2}$ have equal magnitude and opposite directions. Hence the moment caused by $\vec{f_1}$ is $\left\langle k\vec{f_1}, \vec{l_1} \right\rangle$, where $\langle\cdot\rangle$ is a cross product. So as $\left\langle k\vec{f_2}, \vec{l_2} \right\rangle$ is caused by $\vec{f_1}$. The net moment acting at the center of mass is

$$N_f = \left\langle k\vec{f_1}, \vec{l_1} \right\rangle + \left\langle k\vec{f_2}, \vec{l_2} \right\rangle. \tag{6.16}$$



(a) Nonuniform flow.

(b) Flow-induced moment.

Figure 6.4: Flow-induced moment in the body-fixed frame of robotic fish.

The flow-induced rotation exaggerates the motion-integration error by including the displacement caused by heading change. In other words, predicting the trajectory as a straight line is not valid anymore. In order to make the robotic fish become capable of motion tomography, an active control aiming to correct the flow-induced rotation could be added. As shown in Fig. 6.4b, the flow-induced moment N_f is offset by N_T, which is the moment generated from thrust. The active control makes the following relation stand

$$N_T + N_f - N_{|r|}r \approx 0. \tag{6.17}$$

This equation indicates that AHC aims to counteract the moment caused by the flow, allowing the robotic fish to behave like a particle model.

During implementation, AHC uses the robotic fish's pose as feedback. The control target is maintaining the pose of robotic fish unchanged when swimming inside the sensing area. Consider the measured heading angle of robotic fish is $\bar{\psi}_j^0$, the controller is designed as

$$u(t) = K_p(\bar{\psi}_1^0 - \bar{\psi}(t)) + K_i \int (\bar{\psi}_1^0 - \bar{\psi}(t))\mathrm{dt}. \tag{6.18}$$

It is clear that a proportional-integral control law is adopted with K_p as proportional gain and K_i as integral gain. $u(t)$ is the control effort that steers robotic fish. $\bar{\psi}_1^0$ is the heading at the moments the robotic fish enters the sensing area. The control target is maintaining the pose of robotic fish unchanged when swimming inside the sensing area. The controller design skips derivative control because the pose keeps oscillating due to the robotic fish's head motion.

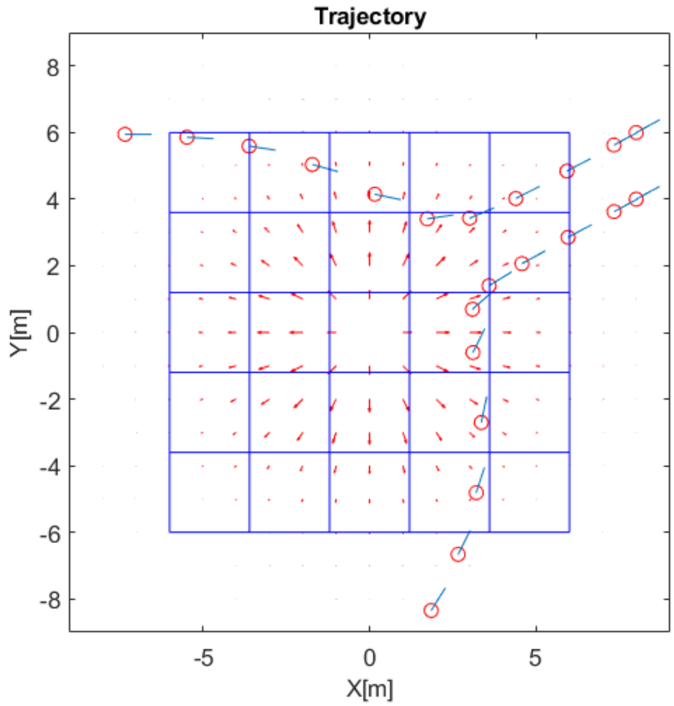
The simulation in Fig. 6.5 demonstrates how a robotic fish driven by a rigid-body model behaves in a flow field with and without AHC. In Fig. 6.5a, the non-AHC case shows that the flow field leads to the heading change of robotic fish and eventually results in a huge integration error. It is easy to observe that the robotic fish leaving the field has a different heading compared to the second it enters the sensing area. In Fig. 6.5b, AHC corrects the heading during the swimming, making sure the heading is consistent from the moment entering to the moment exiting the field. Therefore, the integration error is much smaller than the case in Fig. 6.5a, and the result is very close to the simulation that uses a first-order particle model for demonstration in Fig. 6.2.

Overall, adding AHC significantly reduces the impact of the flow-induced rotation, making the collected position and velocity data reasonable for motion tomography

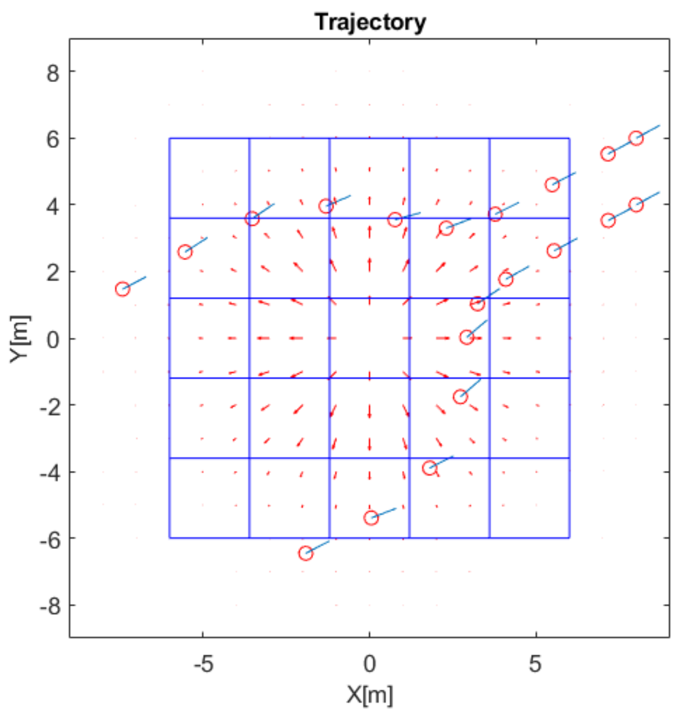
6.3 Experiment and Simulation

In experiments, motion tomography was performed in an above-ground swimming pool that was 10 meters long and 5 meters wide, shown in Fig. 6.6. The flow field was realized by installing a 3/4 horsepower submersible pump to generate a water jet toward the water surface, making a radiation-like flow field similar to Eq. 6.14. The pump was placed at the center of the pool. The sensing area was defined as a 5-meter long, 3-meter wide rectangle. The robotic fish used in this experiment was developed from previous project. It has 0.3 meters in length and 0.9 kilograms in weight. It swam at a constant speed of 0.25 m/s in experiments, and its location and pose were captured by a camera installed 5 meters above the pool. The computer recognized robotic fish's position and pose through a computer vision algorithm. The control signals were calculated in MATLAB and transmitted wirelessly through Wi-Fi.

We first examined the effectiveness of AHC using the robotic fish. The controller in Eq. 6.18 is reorganized into discrete-time form with proportional gain is $K_p = -1.2$ and integral gain is $K_i = -0,005$. The negative value is because the steering direction is opposite to the control direction for robotic fish. As shown in Fig. 6.7a, the pump was placed at the corner of the pool. The robotic fish was tested three times, starting from the same position with the same initial pose. In the first run (trajectory on the left), the pump was turned off. The fish can swim straightly without external flow. In the second run (trajectory on the right), the pump was turned on, and the fish swam without AHC. The flow from the water jet dramatically changed the robotic fish's heading, and the fish was quickly being pushed away. In the third



(a) Without AHC.



(b) With AHC.

Figure 6.5: The simulation of AHC.

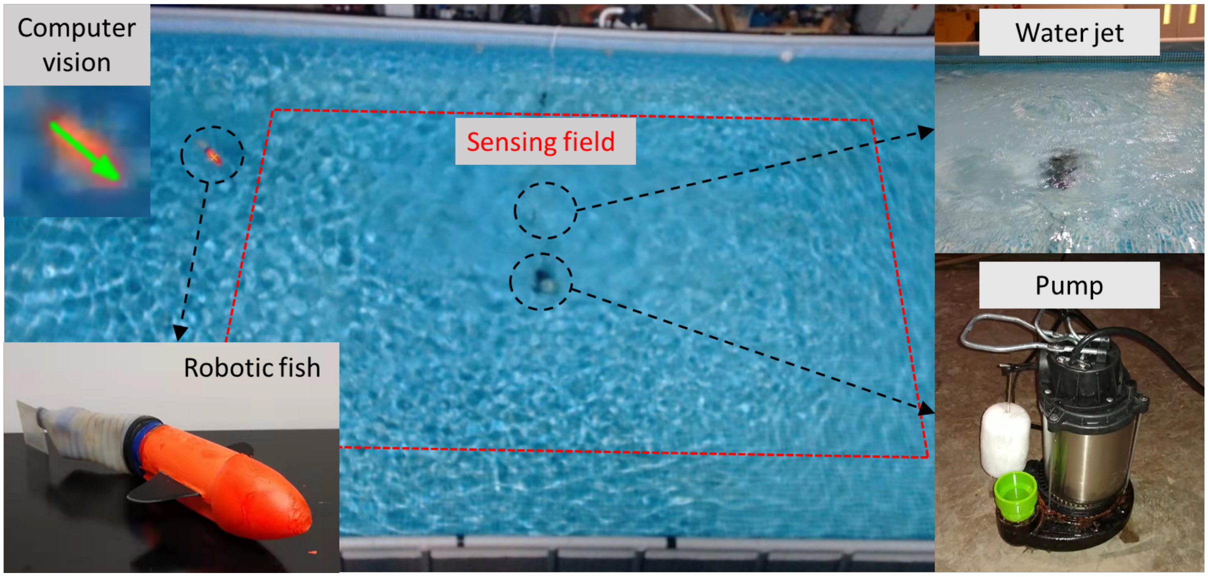
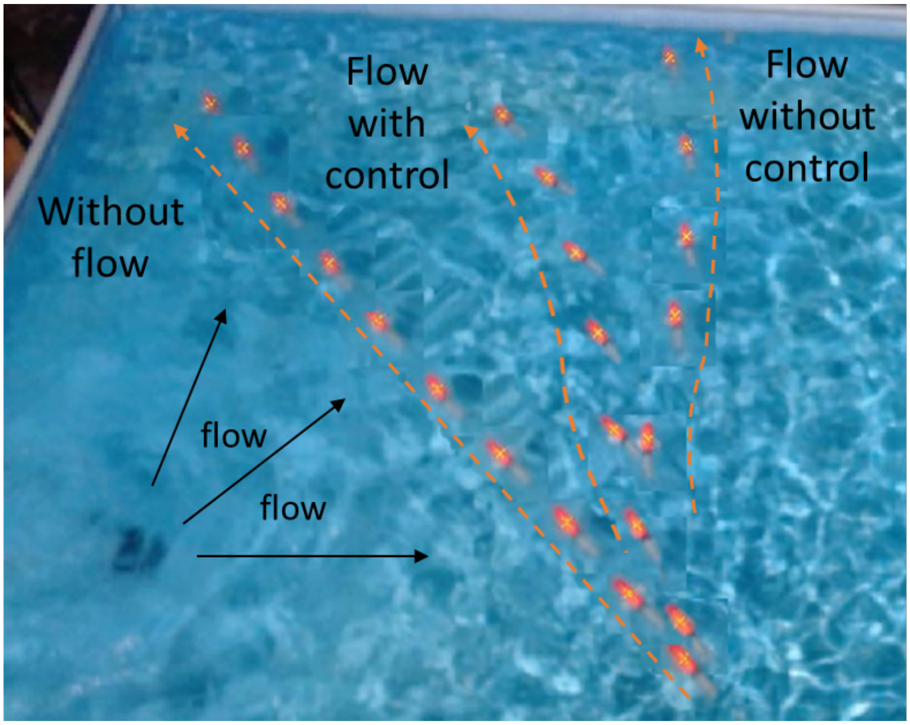


Figure 6.6: Experiment configuration.

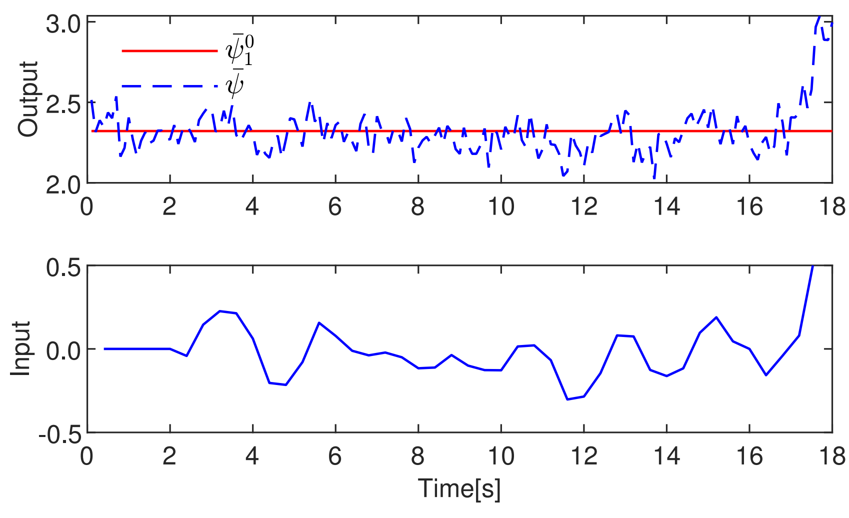
run (trajectory in the middle), we kept the pump turned on and enabled AHC for robotic fish. The AHC effectively adjusts fish's heading angle to limit its variation, and the recorded snapshot exhibits a consistent pose. The performance of AHC is demonstrated in Fig. 6.7b. The robotic fish's heading is bounded within ± 0.3 rad of the initial heading $\bar{\psi}_1^0$ when the robotic fish is entering the sensing area. The large error after t=16 s is due to the robotic fish hitting the wall after leaving the sensing area. The control input demonstrates a quick controller response, which provides a robust adjustment to reject the flow-induced moment.

These experiments obtained a similar result from Fig. 6.2, in which the simulation was driven by a first-order particle model. One can consider the first run was the predicted trajectory mentioned in Eq. 6.1, which neglects the external flow, and the third run was an actual trajectory. With these two trajectories, a reasonable motion-integration error can be obtained. Thus, the robotic fish was qualified to perform motion tomography with the help of AHC.

The next experiment collected 24 trajectories of robotic fish with the pump turned



(a) Experimental trajectories.



(b) Experiment result of AHC.

Figure 6.7: Experimental illustration of flow-induced rotation and AHC.

on. The entire sensing area was divided into 60 cells, and the pump was placed close to the center of sensing area. In order to ensure that the robotic fish swam through each cell, the robotic fish was started from the corners of the pool and swam towards the pump. All collected trajectories are shown in Fig. 6.8, where four numbers indicate the starting position of robotic fish. The flow field was estimated from these trajectories and the estimated results are shown in Fig. 6.9a. Similar to Fig. 6.3b, a blue vector started from the center of the cell represents the estimated flow in this cell. The entire vectorized flow field exhibits a radiating distribution from the center area. Because the flow field at the center of the sensing has the highest strength, none of the trajectories can touch that region. Therefore, the estimation for the center of the field appears blank. A denser grid improves the resolution of the flow field estimation. In Fig. 6.9b, the estimation using 240 cells exhibits more details of the flow field. The streamline and trend of the flow field are much clearer than the result shown in Fig. 6.9a. Overall, experimental results successfully were used to reproduce the map of the flow field. With AHC's help suppress the heading change effectively, the validity of the data collected by the robotic fish for motion tomography appears to be solid.

6.4 Chapter Summary

In this chapter, the motion tomography enabled by robotic fish is discussed. The mechanism of motion tomography relies on a first-order particle model that neglects the rigid-body dynamics of marine vehicles. This assumption raises a limitation that the rigid-body dynamics cannot be ignored when performing motion tomography in a confined area with a chaotic flow field. To address this problem, an AHC is developed for the robotic fish to offset the rotational moment caused by flow-induced

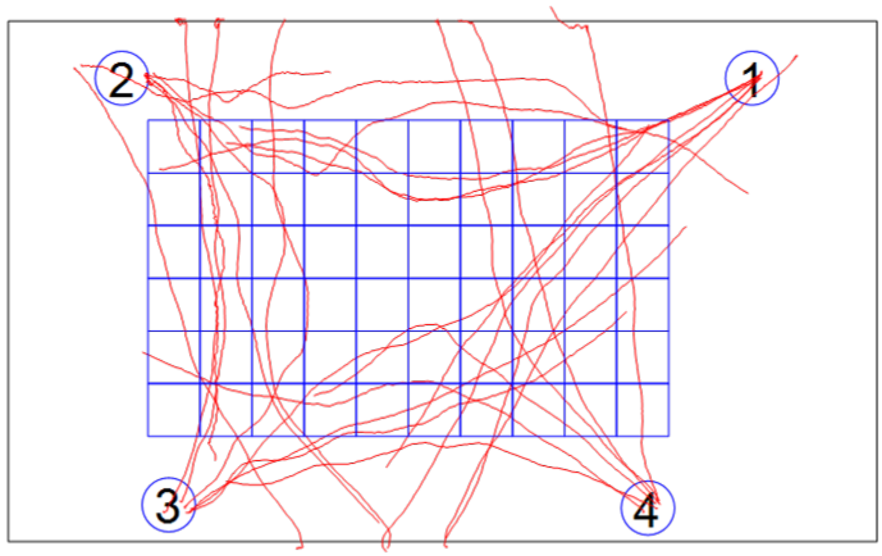
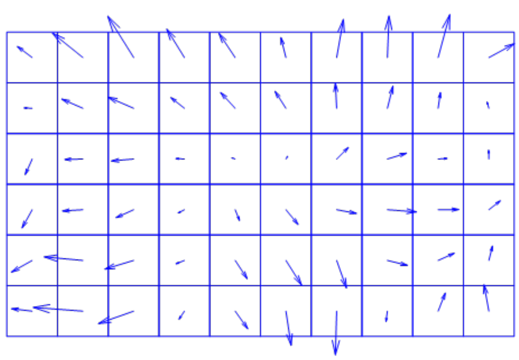
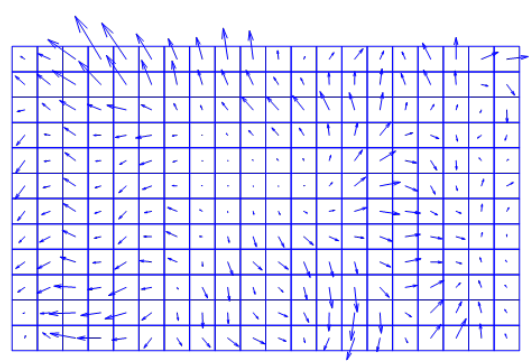


Figure 6.8: Collected trajectories using robotic fish.



(a) Flow field estimation in 60 cells.



(b) Flow field estimation in 240 cells.

Figure 6.9: Flow field estimation using experimental data.

force. The concept is studied in a simulation using a simplified rigid-body dynamic model. To further prove the effectiveness of AHC and the feasibility of using robotic fish to perform motion tomography, experiments were conducted using a giant indoor pool and a submersible pump. Multiple trajectories of robotic fish were collected, seeking to recover the flow field generated by a water jet. These experiments lead to a promising result that the robotic fish is able to perform a good flow field estimation work with the help of AHC.

Chapter 7

Summary and Future Work

This dissertation addresses the challenges associated with the design, modeling, control, and application of robotic fish. From a system perspective, the conclusion of this dissertation can be summarized as follows.

- Chapter 2 focuses on demonstrating the maneuverability of the robotic fish driven by the DSC mechanism. The novel design of the DSC mechanism ensures that the fishtail flaps like an oscillating foil. In robotic fish design, a servo motor is added in front of the DSC to form a hybrid propulsion system that enables robotic fish to swim on the water surface freely. The proposed robotic fish is tested in experiments to examine its swimming speed and steering capability.

- In Chapter 3, the derivation of the dynamic model is benefited from the DSC design which is guided by a traveling wave equation. Additionally, the thrust calculation is concentrated in an individual DSC frame because DSC can achieve undulatory locomotion. Thus, the thrust calculation is decoupled from the robotic fish's kinematics, and the robot can be modeled from the vessel's perspective. The nonlinear dynamic model is simplified to a linear transfer function following the idea of the Namoto model that describes the relationship between input bias and output yaw motion. The nonlinear dynamic model is further used in developing a simulation environment, and the linear model is used for linear controller design.

- In Chapter 4, the steering control of robotic fish based on the linear model is discussed. A state-space model is derived from the transfer function proposed in Chapter 3 to develop a state observer that estimates the system state vector using bias input and yaw output. Two types of controllers are designed. One is a traditional state-feedback controller OSFC, and the other is PETC. PETC addresses the challenge of communication inconsistency in robotic fish's deployment. It has a model-based predictor which estimates state-vector and output error without feedback information. Under PETC, the feedback information is not necessarily to be sent to the robot at every sampling time. Thus the communication resource is saved. When communication frequency is inconsistent, the bias of robotic fish is continuously adjusted instead of zero-order-hold so that the steering control performance is maintained. The advantage of PETC with respect to OSFC and PID are demonstrated in experiments and simulations.

- Chapter 5 discusses the study of robotic fish's maneuverability in the vertical direction. The proposed 3D maneuverable robotic fish equips a BCD to adjust depth. The BCD has an IPMC-enabled water electrolyzer that can generate hydrogen and oxygen inside the robot. The robotic fish adjust its overall density by storing and releasing the gas to produce a net buoyancy to maneuver vertically. The robotic fish also equips a servo motor for two-dimensional maneuvering. In experiments, the robotic fish demonstrates a spiral motion that steers the robotic fish while rising. Considering the importance of BCD, its dynamic model is further investigated to analyze BCD's controllability and stability. BCD is identified as an unstable system with input constraints and state constraints. To ensure BCD's safety during vertical maneuvering, an optimal trajectory planner is proposed to prevent BCD from breaking the speed limit and saturating control input. The trajectory can slowly guide BCD to adjust depth smoothly, and the concept is tested in an experiment.

- In Chapter 6, the motion tomography enabled by robotic fish is discussed. The mechanism of motion tomography relies on a first-order particle model that neglects the rigid-body dynamics of marine vehicles. This assumption raises a limitation that the rigid-body dynamics is not ignorable when performing motion tomography in a confined area with a chaotic flow field. To address this problem, an AHC is developed for the robotic fish to offset the rotational moment caused by flow-induced force. This concept is studied in a simulation using a simplified rigid-body dynamic model. To further prove the effectiveness of AHC and the feasibility of using robotic fish to perform motion tomography, experiments were conducted using a giant indoor pool and a submersible pump. Multiple trajectories of robotic fish were collected, expecting to recover the flow field generated by a water jet. Experimental results have validated that the robotic fish can perform a good flow field estimation work with the help of AHC.

7.1 Current Limitation and Future Works

We also considered the limitation of the current work. The critical thinking and future work can be summarized as following:

- Firstly, The study of linear models in time domain hasn't been sufficiently addressed due to the limitations of experimental facilities; the model validation should include experimental data from different swimming patterns. In model reduction, many variables are assumed time-invariant or change in a small range. Although these assumptions benefit model derivation, they limit the model's generality and the robustness of the model-based controller. Because the model's parameters will inevitably migrate in practical application, errors caused by unmodeled dynamics and model uncertainties could be amplified with time. In future work, we will focus on online model identification and adaptive control research. Adding an observer that can adaptively monitor the model's parameters will improve the model's accuracy and the controller's robustness.

- Next, the study of 3D maneuverable robotic fish requires more efforts in combining dynamic depth control and buoyancy control. Current work is only focused on the development of BCD using an onboard water electrolyzer. Although it demonstrates vertical maneuvering with low energy cost, its shortage, such as slow response speed and low gas generation rate, should not be ignored. Our ongoing project uses both depth control methods, a vertically installed propeller and a Proton-exchange membrane enabled BCD, to control an ROV with the time-varying load. It is believed to be the most advanced depth control approach for ROV. We hope this project can benefit our robotic fish design by developing a compact dual-method BCD specifically for robotic fish. Moreover, we have separately studied the robotic fish in the planar maneuvering and vertical maneuvering, but these two concepts have not yet been combined. Subsequent work includes investigating the model and control of a robotic fish with five DOFs with a newly developed BCD. By that, the maneuverability of robotic fish would be significantly enhanced, and the application value of robotic fish would be greatly expanded.

- Last, developing applications for robotic fish is also the direction of our future focus. For now, good preliminary results in robotic fish's application study are achieved with an ingenious setup for the lab-scale experiments to reflect complex real-world environments. We still need to move our scope from the laboratory to the real outdoor environment, where the robotic fish's capabilities will ultimately be challenged. When testing the robotic fish, we have tried the field test of robotic fish swimming in the ocean and lake, and the experiment results prove the design's reliability. Therefore, our goal is to add a more comprehensive positioning and perception device to the robotic fish so that we can collect more valuable data from experiments in nature.

References

[1] C. C. Lindsey. Form, function and locomotory habits in fish. *Locomotion*, 1978.

[2] Hassan Ghassemi and Ehsan Yari. The added mass coefficient computation of sphere, ellipsoid and marine propellers using boundary element method. *Polish maritime research*, 18(1):17–26, 2011.

[3] Alicia Keow and Zheng Chen. Modeling and control of artificial swimming bladder enabled by ipmc water electrolysis. In *Dynamic Systems and Control Conference*, volume 51890, page V001T04A009. American Society of Mechanical Engineers, 2018.

[4] Alicia Keow, Wenyu Zuo, Fathi Ghorbel, and Zheng Chen. Reversible fuel cell enabled underwater buoyancy control. *Mechatronics*, 2022 under review.

[5] JW Nicholson and AJ Healey. The present state of autonomous underwater vehicle (auv) applications and technologies. *Marine Technology Society Journal*, 42(1):44–51, 2008.

[6] Jinhyun Kim and Wan Kyun Chung. Accurate and practical thruster modeling for underwater vehicles. *Ocean Engineering*, 33(5-6):566–586, 2006.

[7] George V Lauder. Fish locomotion: recent advances and new directions. *Annual Review of Marine Science*, 7:521–545, 2015.

[8] George V Lauder and Eric D Tytell. Hydrodynamics of undulatory propulsion. *Fish Physiology*, 23:425–468, 2005.

[9] 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.

[10] Örjan Ekeberg. A combined neuronal and mechanical model of fish swimming. *Biological Cybernetics*, 69(5):363–374, 1993.

[11] Carl J Walters. A generalized computer simulation model for fish population studies. *Transactions of the American Fisheries Society*, 98(3):505–512, 1969.

[12] C M. jr. Breder. Locomotion of fishes. *Zoologica*, 4, 07 2015.

[13] M. Sfakiotakis, D. M. Lane, and J. B. C. Davies. Review of fish swimming modes for aquatic locomotion. *IEEE Journal of Oceanic Engineering*, 24(2):237–252, Apr 1999.

[14] 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.

[15] D. S. Barrett, M. S. Triantafyllou, D. K. P. Yue, M. A. Grosenbaugh, and M. J. Wolfgang. Drag reduction in fish-like locomotion. *Journal of Fluid Mechanics*, 392:183–212, 1999.

[16] Lian Shen, Xiang Zhang, Dick K. P. Yue, and Michael S. Triantafyllou. Turbulent flow over a flexible wall undergoing a streamwise traveling wavy motion. *Journal of Fluid Mechanics*, 484:197–221, 2003.

[17] 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.

[18] Robert K Katzschmann, Andrew D Marchese, and Daniela Rus. Hydraulic autonomous soft robotic fish for 3d swimming. In *Experimental Robotics*, pages 405–420. Springer, 2016.

[19] Yong Zhong, Zheng Li, and Ruxu Du. A novel robot fish with wire-driven active body and compliant tail. *IEEE/ASME Transactions on Mechatronics*, 22(4):1633–1643, 2017.

[20] C Rossi, J Colorado, W Coral, and A Barrientos. Bending continuous structures with smas: a novel robotic fish design. *Bioinspiration & Biomimetics*, 6(4):045005, 2011.

[21] 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, 2009.

[22] L Cen and A Erturk. Bio-inspired aquatic robotics by untethered piezohydroelastic actuation. *Bioinspiration & biomimetics*, 8(1):016006, 2013.

[23] Richard James Clapham and Huosheng Hu. isplash: Realizing fast carangiform swimming to outperform a real fish. In *Robot Fish*, pages 193–218. Springer, 2015.

[24] Ziyu Ren, Xingbang Yang, Tianmiao Wang, and Li Wen. Hydrodynamics of a robotic fish tail: effects of the caudal peduncle, fin ray motions and the flow speed. *Bioinspiration & Biomimetics*, 11(1):016008, 2016.

[25] Jamie M Anderson, K Streitlien, DS Barrett, and Michael S Triantafyllou. Oscillating foils of high propulsive efficiency. *Journal of Fluid Mechanics*, 360:41–72, 1998.

[26] Michael S. Triantafyllou, Franz S. Hover, Alexandra H. Techet, and Dick K.-P. Yue. Review of hydrodynamic scaling laws in aquatic locomotion and fishlike swimming. *Applied Mechanics Reviews*, 58:226–237, 2005.

[27] Daniel Floryan, Tyler Van Buren, Clarence W. Rowley, and Alexander J. Smits. Scaling the propulsive performance of heaving and pitching foils. *Journal of Fluid Mechanics*, 822:386–397, 2017.

[28] Michael J. Lighthill. Aquatic animal propulsion of high hydromechanical efficiency. *Journal of Fluid Mechanics*, 44(2):265–301, 1970.

[29] Motomu Nakashima, Norifumi Ohgishi, and Kyosuke Ono. A study on the propulsive mechanism of a double jointed fish robot utilizing self-excitation control. *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, 46(3):982–990, 2003.

[30] Richard Mason and Joel W. Burdick. Experiments in carangiform robotic fish locomotion. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 428–435. IEEE, 2000.

[31] Saurab Verma and Jian-Xin Xu. Analytic modeling for precise speed tracking of multilink robotic fish. *IEEE Transactions on Industrial Electronics*, 65(7):5665–5672, 2017.

[32] Junzhi Yu, Chen Wang, and Guangming Xie. Coordination of multiple robotic fish with applications to underwater robot competition. *IEEE Transactions on Industrial Electronics*, 63(2):1280–1288, 2015.

[33] KH Low and A Willy. Biomimetic motion planning of an undulating robotic fish fin. *Journal of Vibration and Control*, 12(12):1337–1359, 2006.

[34] Paul Phamduy, Jayhwan Cheong, and Maurizio Porfiri. An autonomous charging system for a robotic fish. *IEEE/ASME Transactions on Mechatronics*, 21(6):2953–2963, 2016.

[35] Zongshuai Su, Junzhi Yu, Min Tan, and Jianwei Zhang. Closed-loop precise turning control for a bcf-mode robotic fish. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 946–951. IEEE, 2010.

[36] Christian Meurer, Ashutosh Simha, Ülle Kotta, and Maarja Kruusmaa. Nonlinear orientation controller for a compliant robotic fish based on asymmetric actuation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4688–4694. IEEE, 2019.

[37] Liang Li, Anquan Liu, Wei Wang, Sridhar Ravi, Rubin Fu, Junzhi Yu, and Guangming Xie. Bottom-level motion control for robotic fish to swim in groups: modeling and experiments. *Bioinspiration & biomimetics*, 14(4):046001, 2019.

[38] Saurab Verma, Dong Shen, and Jian-Xin Xu. Motion control of robotic fish under dynamic environmental conditions using adaptive control approach. *IEEE Journal of Oceanic Engineering*, 43(2):381–390, 2017.

[39] Hai Wang, Chunlong Mi, Zhenwei Cao, Jinchuan Zheng, Zhihong Man, Xiaozheng Jin, and Hao Tang. Precise discrete-time steering control for robotic

fish based on data-assisted technique and super-twisting-like algorithm. *IEEE Transactions on Industrial Electronics*, 67(12):10587–10599, 2020.

[40] 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.

[41] 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.

[42] 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.

[43] Pichet Suebsaiprom and Chun-Liang Lin. Maneuverability modeling and trajectory tracking for fish robot. *Control Engineering Practice*, 45:22–36, 2015.

[44] Yongnan Jia and Long Wang. Leader–follower flocking of multiple robotic fish. *IEEE/ASME Transactions on Mechatronics*, 20(3):1372–1383, 2014.

[45] Vishwamithra Sunkara, Animesh Chakravarthy, Xiongfeng Yi, Wenyu Zuo, and Zheng Chen. Cooperative optimal collision avoidance laws for a hybrid-tailed robotic fish. *IEEE Transactions on Control Systems Technology*, 2019.

[46] Robert K. Katzschmann, Joseph DelPreto, Robert MacCurdy, and Daniela Rus. Exploration of underwater life with an acoustically controlled soft robotic fish. *Science Robotics*, 3(16), 2018.

[47] Vladislav Kopman, Jeffrey Laut, Giovanni Polverino, and Maurizio Porfiri. Closed-loop control of zebrafish response using a bioinspired robotic-fish in a preference test. *Journal of the Royal Society Interface*, 10(78):20120540, 2013.

[48] Osama Ennasr, Christopher Holbrook, Darryl W Hondorp, Charles C Krueger, Demetris Coleman, Pratap Solanki, John Thon, and Xiaobo Tan. Characterization of acoustic detection efficiency using a gliding robotic fish as a mobile receiver platform. *Animal Biotelemetry*, 8(1):1–13, 2020.

[49] Adam Aili and Erik Ekelund. Model-based design, development and control of an underwater vehicle, 2016.

[50] Adrian Manzanilla, Sergio Reyes, Miguel Garcia, Diego Mercado, and Rogelio Lozano. Autonomous navigation for unmanned underwater vehicles: Real-time experiments using computer vision. *IEEE Robotics and Automation Letters*, 4(2):1351–1356, 2019.

[51] Shuo Wang, Yu Wang, Qingping Wei, Min Tan, and Junzhi Yu. A bio-inspired robot with undulatory fins and its control methods. *IEEE/ASME Transactions on Mechatronics*, 22(1):206–216, 2016.

[52] Gonca Ozmen Koca, Mustafa Can Bingol, Cafer Bal, Zuhtu Hakan Akpolat, Mustafa Ay, and Deniz Korkmaz. Design and control of diving mechanism for the biomimetic robotic fish. In *International Conference Mechatronics*, pages 662–670. Springer, 2017.

[53] Sang-Ki Jeong, Hyeung-Sik Choi, Jae-Hyun Bae, Sam-Sang You, Hyeon Seung

Kang, Shin-Je Lee, Joon-Young Kim, Dong-Hee Kim, and Yong-Kuk Lee. Design and control of high speed unmanned underwater glider. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 3(3):273–279, 2016.

[54] Paul Phamduy, Raymond LeGrand, and Maurizio Porfiri. Robotic fish: Design and characterization of an interactive idevice-controlled robotic fish for informal science education. *IEEE Robotics & Automation Magazine*, 22(1):86–96, 2015.

[55] 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, 2015.

[56] Michail Makrodimitris, Ioannis Aliprantis, and Evangelos Papadopoulos. Design and implementation of a low cost, pump-based, depth control of a small robotic fish. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1127–1132. IEEE, 2014.

[57] KS Wasserman, JL Mathieu, MI Wolf, A Hathi, SE Fried, and AK Baker. Dynamic buoyancy control of an rov using a variable ballast tank. In *Oceans 2003. Celebrating the Past... Teaming Toward the Future (IEEE Cat. No. 03CH37492)*, volume 5, pages SP2888–SP2893. IEEE, 2003.

[58] Shawn A Woods, Robert J Bauer, and Mae L Seto. Automated ballast tank control system for autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 37(4):727–739, 2012.

[59] Mahdi Choyekh, Naomi Kato, Ryan Dewantara, Hidetaka Senga, and Hajime Chiba. Depth and altitude control of an auv using buoyancy control device. *Journal of Electrical Engineering*, 4:133–149, 2016.

[60] Zhengxing Wu, Junzhi Yu, Jun Yuan, and Min Tan. Towards a gliding robotic dolphin: design, modeling, and experiments. *IEEE/ASME Transactions on Mechatronics*, 24(1):260–270, 2019.

[61] Deniz Korkmaz, Gonca Ozmen Koca, Guoyuan Li, Cafer Bal, Mustafa Ay, and Zuhtu Hakan Akpolat. Locomotion control of a biomimetic robotic fish based on closed loop sensory feedback cpg model. *Journal of Marine Engineering & Technology*, pages 1–13, 2019.

[62] Enrico Petritoli, Fabio Leccese, and Marco Cagnetti. High accuracy buoyancy for underwater gliders: The uncertainty in the depth control. *Sensors*, 19(8):1831, 2019.

[63] Shuxiang Guo, Toshio Fukuda, and Kinji Asaka. A new type of fish-like underwater microrobot. *IEEE/ASME Transactions on Mechatronics*, 8(1):136–141, 2003.

[64] Colin G Cameron and Jeffrey H Smith. Wet buoyancy engine, March 2010.

[65] Tae I Um, Zheng Chen, and Hilary Bart-Smith. A novel electroactive polymer buoyancy control device for bio-inspired underwater vehicles. In *2011 IEEE International Conference on Robotics and Automation*, pages 172–177. IEEE, 2011.

[66] Zheng Chen, Tae I Um, and Hilary Bart-Smith. Modeling and control of artificial bladder enabled by ionic polymer-metal composite. In *2012 American Control Conference (ACC)*, pages 1925–1930. IEEE, 2012.

[67] Han Zhang, Wei Wang, Yingzheng Qu, Chen Wang, Ruifeng Fan, and Guangming Xie. Model identification for the yaw motion of a tail-actuated robotic fish. In *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 313–318. IEEE, 2016.

[68] Irfan Ariyanto, Taesam Kang, and Young Jae Lee. Dynam ics of a fish-like robot and it's controller design. In *Proceedings of the SICE-ICASE International Joint Conference*, pages 4297–4301. IEEE, 2006.

[69] Kensaku Nomoto, T. Taguchi, Keinosuke Honda, and Susumu Hirano. On the steering qualities of ships. *International Shipbuilding Progress*, 4(35):354–370, 1957.

[70] Frank E. Fish. Advantages of aquatic animals as models for bio-inspired drones over present auv technology. *Bioinspiration & Biomimetics*, 15(2):025001, 2020.

[71] Frank E Fish. Bio-inspired aquatic drones: overview. *Bioinspiration & Biomimetics*, 15(6), Paper 060401, 2020.

[72] Bingxi Li, Brian R Page, Barzin Moridian, and Nina Mahmoudian. Collaborative mission planning for long-term operation considering energy limitations. *IEEE Robotics and Automation Letters*, 5(3):4751–4758, 2020.

[73] Hongyi Li, Ziran Chen, Ligang Wu, and Hak-Keung Lam. Event-triggered control for nonlinear systems under unreliable communication links. *IEEE Transactions on Fuzzy Systems*, 25(4):813–824, 2016.

[74] Joo P Hespanha, Payam Naghshtabrizi, and Yonggang Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162, 2007.

[75] Shayok Mukhopadhyay, Chuanfeng Wang, Mark Patterson, Michael Malisoff, and Fumin Zhang. Collaborative autonomous surveys in marine environments affected by oil spills. In *Cooperative Robots and Sensor Networks 2014*, pages 87–113. Springer, 2014.

[76] Ki Myung Brian Lee, Chanyeol Yoo, Ben Hollings, Stuart Anstee, Shoudong Huang, and Robert Fitch. Online estimation of ocean current from sparse gps data for underwater vehicles. In *2019 International conference on robotics and automation (ICRA)*, pages 3443–3449. IEEE, 2019.

[77] KY Cadmus To, Chanyeol Yoo, Stuart Anstee, and Robert Fitch. Distance and steering heuristics for streamline-based flow field planning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1867–1873. IEEE, 2020.

[78] Anastasios M Lekkas and Thor I Fossen. Trajectory tracking and ocean current estimation for marine underactuated vehicles. In *2014 IEEE Conference on Control Applications (CCA)*, pages 905–910. IEEE, 2014.

[79] Mohammadreza Bayat, Naveena Crasta, Antonio Pedro Aguiar, and Antonio M Pascoal. Range-based underwater vehicle localization in the presence of unknown ocean currents: Theory and experiments. *IEEE Transactions on Control Systems Technology*, 24(1):122–139, 2015.

[80] Dongsik Chang, Wencen Wu, Catherine R Edwards, and Fumin Zhang. Motion tomography: Mapping flow fields using autonomous underwater vehicles. *The International Journal of Robotics Research*, 36(3):320–336, 2017.

[81] Meriam Ouerghi, Sean Maxon, Mengxue Hou, and Fumin Zhang. Improved trajectory tracing of underwater vehicles for flow field mapping. *International Journal of Intelligent Robotics and Applications*, 6(1):69–85, 2022.

[82] Daniele Costa, Giacomo Palmieri, Matteo-Claudio Palpacelli, David Scaradozzi, and Massimo Callegari. Design of a carangiform swimming robot through a multiphysics simulation environment. *Biomimetics*, 5(4):46, 2020.

[83] Audrey P. Maertens, Amy Gao, and Michael S. Triantafyllou. Optimal undulatory swimming for a single fish-like body and for a pair of interactingswimmers. *Journal of Fluid Mechanics*, 813:301345, 2017.

[84] Yong-Jai Park, Useok Jeong, Jeongsu Lee, Seok-Ryung Kwon, Ho-Young Kim, and Kyu-Jin Cho. Kinematic condition for maximizing the thrust of a robotic fish using a compliant caudal fin. *IEEE Transactions on Robotics*, 28(6):1216–1227, 2012.

[85] Jianxun Wang and Xiaobo Tan. A dynamic model for tail-actuated robotic fish with drag coefficient adaptation. *Mechatronics*, 23(6):659–668, 2013.

[86] 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.

[87] Seth A. Brooks and Melissa A. Green. Experimental study of body-fin interaction and vortex dynamics generated by a two degree-of-freedom fish model. *Biomimetics*, 4(4):67, 2019.

[88] Sadatoshi Taneda. Visual study of unsteady separated flows around bodies. *Progress in Aerospace Sciences*, 17(4):287–348, 1977.

[89] Tyler Van Buren, Daniel Floryan, Nathaniel Wei, and Alexander J. Smits. Flow speed has little impact on propulsive characteristics of oscillating foils. *Physical Review Fluids*, 3(1):013103, 2018.

[90] Michel Verhaegen and Patrick Dewilde. Subspace model identification part 2. analysis of the elementary output-error state-space model identification algorithm. *International Journal of Control*, 56(5):1211–1241, 1992.

[91] W.P.M.H. Heemels, K.H. Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 3270–3285, 2012.

[92] Anton Selivanov and Emilia Fridman. Event-triggered H_∞ control: A switching approach. *IEEE Transactions on Automatic Control*, 61(10):3221–3226, 2016.

[93] Anton Selivanov, Emilia Fridman, and Alexander Fradkov. Event-triggered adaptive control of minimum-phase systems. *IFAC-PapersOnLine*, 50(1):4276–4281, 2017.

[94] Paulo Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, 2007.

[95] WPMH Heemels and MCF Donkers. Model-based periodic event-triggered control for linear systems. *Automatica*, 49(3):698–711, 2013.

[96] Hassan K Khalil. *Nonlinear Systems, Third Edition.* Prentice Hall, Upper Saddle River, NJ, 2002.

[97] John Elie Sader. Frequency response of cantilever beams immersed in viscous fluids with applications to the atomic force microscope. *Journal of Applied Physics*, 84(1):64–76, 1998.

[98] Zheng Chen, Tae I. Um, and Hilary Bart-Smith. Modeling and control of artificial bladder enabled by ionic polymer-metal composite. In *Proc. of 2012 American Control Conference (ACC)*, pages 1925–1930. IEEE, 2012.

[99] D Subbaram Naidu. *Optimal control systems.* CRC press, 2002.

[100] Sonja Macfarlane and Elizabeth A. Croft. Jerk-bounded manipulator trajectory planning: design for real-time applications. *IEEE Transactions on Robotics and Automation*, 19(1):42–52, 2003.

[101] Charles G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19(92):577–593, 1965.

[102] Thor I Fossen. *Guidance and control of ocean vehicles*, volume 199. Wiley New York, 1994.