MODELING, CONTROL AND LOCOMOTION PLANNING OF AN ANGUILLIFORM FISH ROBOT

XUELEI NIU

(B. Eng.), Harbin Institute of Technology, China

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
2013

DECLARATION

I hereby declare that this thesis is my original work and it has been written by me in its entirety.

I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Xuelei Niu

Xuelei Niu

1 July, 2013

Acknowledgments

I would like to express my deepest gratitude to Prof. Jian-Xin Xu, my main supervisor, for his inspiration, excellent guidance, support and encouragement. His erudite knowledge, the deepest insights on the fields of control theory and robotics have been the most inspirations and made this research work a rewarding experience. Here I express my gratitude to him for giving me the curiosity about the learning and research in the domains of control, robotics and biomimetics. Also, his rigorous scientific approach and endless enthusiasm have influenced me greatly. The progress of this PhD program would not be possible without his guidance. I think I am quite fortunate to work under his supervision, which has made the past four years such an enjoyable and rewarding experience.

Also, I would like to express my gratitude to Prof. Qing-Guo Wang, my co-supervisor, for the quite useful and inspiring discussions.

Thanks also go to Electrical & Computer Engineering Department in National University of Singapore and China Scholarship Council, for the financial support during my pursuit of a PhD.

I would like to thank my Thesis Advisory Committee, Prof. Ben M. Chen and Prof. Sanjib K. Panda of National University of Singapore, who provided me a lot of suggestive questions for my research.

I am also grateful to all my friends in Control and Simulation Lab, National University of Singapore. Their kind assistance and friendship have made my life in Singapore easy and colorful.

Contents

Summary

In this thesis, mathematical model, control law design, different locomotion patterns, and locomotion planning are presented for an Anguilliform robotic fish. The robotic fish, consisted of links and joints, are driven by torques applied to the joints. Considering kinematic constraints, Lagrangian formulation is used to obtain the mathematical model of the robotic fish. The model reveals the relation between motion of the fish and external forces. Computed torque control method is first applied, which can provide satisfactory tracking performance for reference joint angles. To deal with parameter uncertainties, sliding model control is adopted. Three locomotion patterns – forward locomotion, backward locomotion, and turning locomotion – are realized by assigning appropriate reference angles to the joints, and the three locomotions are verified by experiments and simulations. Relations among swimming speed, turning radius, and related parameters are also investigated. Based on the relations, a motion library is built, from which the robotic fish can choose suitable parameters to achieve desired speed and turning radius. Based on the motion library, a motion planning strategy is designed, which can handle different tasks. The motion of robotic fishes with different number of links are investigated, and their performances are compared. By using feedback of camera, an experiment is conducted in which the robotic fish is able to track a predefined curve. A new form of central pattern generator (CPG) model is presented, which consists of three-dimensional coupled Andronov-Hopf oscillators, artificial neural network (ANN), and outer amplitude modulator. By using this CPG model, swimming pattern of a real Anguilliform fish is successfully applied to the robotic fish in an experiment.

List of Tables

List of Figures

Nomenclature

Symbol	Meaning or Operation
N	number of links of the robotic fish
i	index number of the i-th link
x_i, y_i	position of the link i
ϕ_i	orientation angle of link i
τ_i	torque exerted between link i and link $i+1$
v_i	velocity of link i
$v_{i\perp}$	perpendicular component of the velocity v_i
$v_{i\|}$	parallel component of the velocity v_i
v_{ix}	projection of the velocity v_i on x-axis
v_{iy}	projection of the velocity v_i on y-axis
f_i	water resistance coefficient of link i
$f_{i\perp}$	perpendicular component of the water resistance coefficient of link i
$f_{i\|}$	parallel component of the water resistance coefficient of link i
w_i	hydrodynamic force on link i
$w_{i\perp}$	perpendicular component of w_i
$w_{i\|}$	parallel component of w_i
w_{ix}	projection of the hydrodynamic force w_i on x-axis
w_{iy}	projection of the hydrodynamic force w_i on y-axis
$\mathbf{p}$	coordinates vector
l_i	length of link i
$\mathbf{g}(\mathbf{p})$	constraints in the system
$L(\mathbf{p}, \dot{\mathbf{p}})$	total energy of the system

Symbol	Meaning or Operation
$K(\mathbf{p}, \dot{\mathbf{p}})$	kinetic energy of the system
$V(\mathbf{p})$	potential energy of the system
$J(\mathbf{p})$	Jacobian of the constraints matrix $\mathbf{g}(\mathbf{p})$
Γ	internal force of the system
λ	vector of relative magnitudes of the constraint forces
$\mathbf{w}$	external force vector
M	mass matrix of the system
m_i	mass of link i
I_i	moment of inertia of link i
ϕ_{ir}	reference angle of link i
A_m	amplitude of ϕ_{ir}
ω	oscillation frequency
θ	phase difference
$\mathbf{e}$	angular error vector
σ	sliding surface
C	a diagonal matrix associated with the sliding surface
τ_0	one term of τ, which is used to handle nominal model
τ_s	one term of τ, which is used to handle uncertainties
ρ	parameter in the sliding mode control law
η	parameter in the sliding mode control law
τ_{eq}	equivalent control of the sliding mode control law
α	uncertainty coefficient in the mass matrix
β_1, β_1	uncertainty coefficient in the water resistance coefficients
ϵ	parameter of the saturation function in the modified sliding mode control law
h_c	height of the camera
h_w	depth of the water
x_c	position of the camera
x_a	actual position of the fish

Symbol	Meaning or Operation
x'_o	the position where the extension line of the camera's line-of-sight and the bottom of the water meet
α_a	angle of incidence
α_w	angle of refraction
n_a	refraction index of air
n_w	refraction index of water
$\gamma(j)$	deflection angle on link i
v_s	steady speed of the fish
$\gamma_{\max}$	the maximum deflection angle
$\mathbf{z} = [\mathbf{m}, \mathbf{n}]^{\mathbf{T}}$	state vector of oscillator
$\mathbf{c} = [c_1, c_2]^T$	oscillation center
a_i	amplitude of the oscillator i
β	attraction rate of the oscillator
k	constant coupling strength
w_{ij}	weight of connection between two oscillators
g_{ij}	amplitude ratio between two oscillators
α_{ij}	desired phase difference between two oscillators
$S(\alpha_{ij})$	rotation transformation matrix
K	spacial scaling matrix
μ	decay rate

Chapter 1

Introduction

1.1 Background and Motivation

In the past three decades, there has been a tremendous surge of activity in robotics, both in terms of academic research and practical application [2]. The general public have already witnessed its seemingly endless and diverse possibilities in different areas of our life. This period has been accompanied by a technological maturation of robots as well, from the simple pick and place and painting and welding robots, to more sophisticated assembly robots for inserting integrated circuit chips onto printed circuit boards, to mobile carts for parts handling and delivery. Whether we notice them or not, robots exist everywhere in our daily life. As pointed by Bill Gates [3], in the near future, robots will appear in every home, just like the popularization of personal computers years ago.

Among all kinds of robots, bio-inspired robots are the most special and attractive kind. Different from industrial robots, which always do some repetitive tasks in industrial applications, bio-inspired robots are made from inspiration from animals or human beings. The idea of producing this kind of robots is inspired by mimicking behaviors of animals in natural world or human beings ourselves. The most famous example of bio-inspired robots is ASIMO, as shown in Fig. 1.1, a humanoid robot made by the company of Honda. ASIMO has the ability to recognize moving objects, postures, gestures, its surrounding environment, sounds and faces, which enable it to interact with humans. Another quite famous example of bio-inspired robots is the BigDog, as shown in Fig. 1.2, which is built for military applications. The BigDog is capable of traversing difficult terrain, running at 4 miles per hour (6.4 km/h), carrying 340 pounds (150 kg), and climbing a 35 degree incline. With such capability, BigDog is designed to serve as a robotic pack mule to accompany soldiers in terrain too rough for conventional vehicles.

Other bio-inspired robots include snake robot which resembles the body structure and locomotions of snakes, flapping wing robot which can fly like a bird by flapping its wings, ant robot, spider robot, etc (as shown in Fig. 1.3). Because most bio-inspired robots are autonomous, which means the supervision of human beings is not needed when this kind of robot is in operation, bio-inspired robot can execute many intelligent tasks, such as surveillance, looking for survivals after accidents or natural disasters. Moreover, they are able to work in hazardous environments such as high radiation field or high toxic environment. Without these robots, people have to do these things personally, which will generate a huge cost on money and human resource.

Figure 1.1: The ASIMO robot.

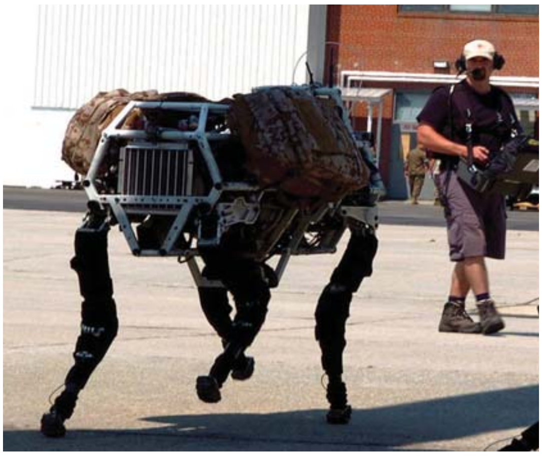


Figure 1.2: The BigDog robot.

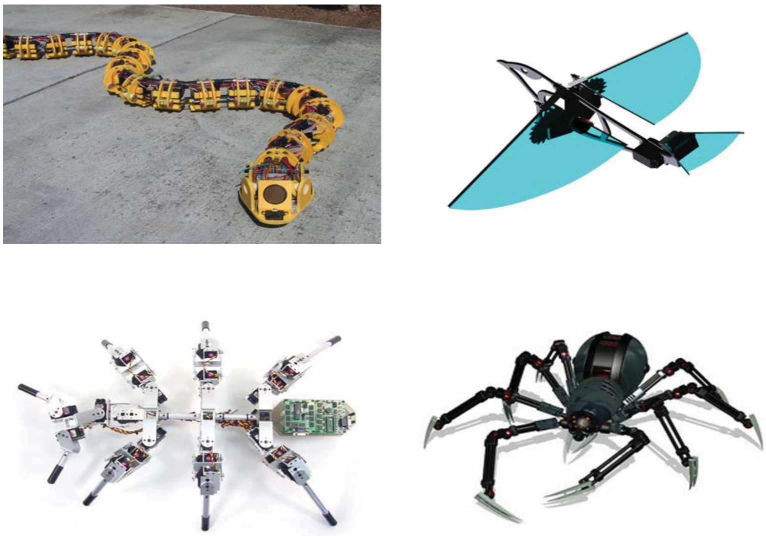


Figure 1.3: Bio-inspired robots: snake robot, flapping wing robot, ant robot, spider robot.

One representative example of bio-inspired robots is fish-like robot. In recent years, with increasing underwater activities and research work, such as underwater archaeology, oil pipe leakage detection, military activity [4], Autonomous Underwater Vehicle (AUV) is receiving more and more attention [5]. Traditional AUV, usually thrusted by rotatory propellers, may not be satisfactory in efficiency, maneuverability and noise control. Thus, new type of AUV is needed. During the long period time of nature selection, fishes have evolved body structures and swimming patterns that highly adapt to aquatic environments [6]. Some fishes are power-efficient, thus consume fewer energy when in a long distance journey. Some fishes are highly maneuverable and flexible, which is useful when conduct a complex task. Moreover, the noiseless propulsion is another advantage in military applications [7]. Actually, they are more advanced swimming machines with higher efficiency, more remarkable maneuverability and less noise than conventional AUV.

Attracted by the appealing merits that real fishes possess, such as power efficiency, maneuverability, flexibility, and noiseless propulsion, a lot of efforts have been spent on studying how real fishes move [8–10]. In these works, different theories are developed to investigate the mechanism of fish swimming, and numerous prototypes of robotic fishes (as shown in Fig. 1.4) are made to verify whether those theories are effective.

On the one hand, robotic fish is a topic related to robotics, a traditional field where modeling work and control method are needed. On the other hand, robotic fish is related to biology, from where new concepts of generating signals and implementing actuators are borrowed. Thus, research topics about fish-like robots include: mathematical modeling of the motion dynamics of the robotic fish; general control issues of robots - what kind of control approach will be applied to robots considering surroundings, such as environmental uncertainties; locomotion generation - how to coordinate the body movement,

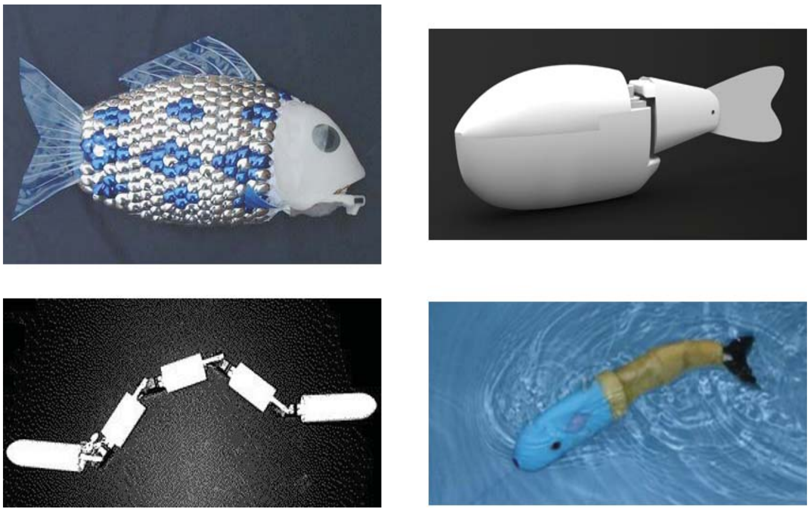


Figure 1.4: Different kinds of robotic fishes.

in order to mimic the pattern that real fishes move; path planning - let the robot move along a desired path to accomplish specific task; etc. In the following, some general literature review about the above contents is given.

Mathematical modeling is important to analyze the characters of the robotic fish. By conducting necessary geometric abstract and omitting subordinate factors, a mathematical formulation will be given to the fish and a model will be obtained. With the model, it can be investigated of the underlying motion mechanism of the fish, and design appropriate control laws on it. One of the earliest and the most famous modeling work for fishes is elongated body theory (EBT) [11]. EBT, assuming sinusoidal motion of the fish body, was first applied to Anguilliform fishes. EBT investigated the relation among several variables which involve mean speed of the fishes, velocity of lateral pushing of a vertical water slice, velocity of a traveling wave. By calculating the rate of fish doing work under different frames of reference, the thrust was obtained. EBT was extended in [12], which was called large-amplitude elongated body theory, to better suit to Carangiform locomotion. However, EBT and its extended version were principally used to study steady state propulsion, involving no dynamics. Following EBT [11, 12], researchers have developed many other robotic fish models, which will be elaborated in next chapters. However, in these mathematical models, the relation between motion of the fish and efforts of actuators are not explicitly given, but the relation is critical for control law design.

After mathematical model of the fish is obtained, control laws need to be designed, so that the robotic fish can be manipulated to perform desired motions. In [13–20], many control approaches, either open-loop or closed-loop, are given. These control approaches include PID control, fuzzy logic control, geometric nonlinear control, etc. It can be found that in a large proportion of papers, simple sinusoidal signals are applied to the control signals. Although it is quite an easy way to implement the control signals, the control performance may not be good.

In order to achieve complicated tasks, the robotic fish need to swim in different locomotion patterns, which can be obtained by assigning different control laws to the robotic fish. The most common locomotion patterns include forward locomotion, backward locomotion, and turning locomotion, which are extensively presented in existing works [21–25]. Except for the above three patterns, some new locomotion patterns are also investigated, such as spinning pattern and sideways pattern [26], which are not usually seen in natural world.

In practical application, the robotic fish will encounter all kinds of complicated scenarios, where the three basic locomotion patterns are not competent. To achieve complex tasks, the fish need to combine and organize the basic locomotion patterns. Since there are many parameters contained in the robotic fish system, such as the amplitude of each joint angle, the oscillation frequency, the phase difference between two connecting links, and the deflection angle, how to choose appropriate parameters in different conditions, is an important issue to discuss. Also, it is important to choose when to conduct each individual locomotion, and in this case, it is necessary to add feedback to make decision. The core principle to generate complicated locomotion patterns is that, we have to always relate the physical meaning of the useful parameters with the characters of the locomotions. In another way, we can say that we need to always think in a biomimetic way. Concerning the issues of parameter study and motion planning in the robotic fish system, there are a lot of works that have been done [6, 24, 27–32]. However, these works are confined to the study of part of the parameters in the system, a more detailed investigation needs to be conducted.

Apart from traditional ways of producing control signal for robotic fishes, some new approaches have been developed by researchers, and central pattern generator (CPG) is one of them. Central pattern generators are neural circuits found in both invertebrate and vertebrate animals that can produce rhythmic patterns of neural activity without receiving rhythmic inputs. Some neurobiological findings [33] concerning locomotor CPG include: (i) locomotion rhythms are generated centrally without requiring sensory information; (ii) CPGs are distributed networks made of multiple coupled oscillatory centers; (iii) While sensory feedback is not needed for generating the rhythms, it plays a very important role in shaping the rhythmic patterns. Some properties of CPG involve: (i) The purpose of CPG models is to exhibit limit cycle behavior; (ii) CPGs are well suited for distributed implementation; (iii) CPG models typically have a few control parameters that allow modulation of the locomotion; (iv) CPGs are ideally suited to integrate sensory feedback signals; (v) CPG models usually offer a good substrate for learning and optimization algorithms.

Other than traditional servo motors, new materials are also adopted in the robotic fish design. In [34], by mimicking the sea lamprey, a biologically based underwater autonomous vehicle is developed. The undulation of the fish robot is actuated by artificial muscles composed of shape memory alloy. In [35], shape memory alloy is also used to actuate the backbone of the robotic fish, that is, to change the curvature of the body, so that the fish can swim. The robot is motor-less and gear-less and is able to swim in some standard patterns. In [36], a physics-based model was proposed for a biomimetic robotic fish propelled by an ionic polymerCmetal composite (IPMC) actuator. The model incorporated both IPMC actuation dynamics and the hydrodynamics, and predicts the steady-state cruising speed of the robot under a given periodic actuation voltage. Also by using IPMC, [37] gave both an analytical model and a computational fluid dynamics (CFD) model of the robotic fish, where the analytical model was developed to compute the thrust force generated by a two-link tail and the resulting moments in the active joints, and CFD modeling was also adopted to examine the flow field, the produced thrust, and the bending moments in joints. It showed agreement of the two models when comparing the thrust forces. In [38], a modeling framework of biomimetic underwater vehicles propelled by vibrating IPMC was developed. The motion of the vehicle body was described using rigid body dynamics in fluid environments. Hydrodynamic effects, such as added mass and damping, are included in the model to enable a thorough description of the vehicles surge, sway, and yaw motions.

1.2 Contributions

The contributions of this thesis are summarized as follow:

First, we present the mathematical model of a robotic fish. Through this model, the analytical relation between the motion of the fish and the external forces/torques can be obtained. Compared with previous works, the major superiority of our work is that: Unlike [11], [12] and [17], which treat the fish body as a smooth and continuous curve, we construct a mathematical model for the robotic fish that consists of joints and links, which is more of practical concern. The model reveals the explicit relation between torques added on the robotic fish and the corresponding motion of the fish.

Second, based on the previously derived mathematical model of the robotic fish, two different control approaches are developed. In computed torque control method, torques are calculated by using joint angle positions, joint angle velocity, and their references. To deal with parameter uncertainty and external disturbance, which always arise in practical circumstance, sliding mode control is adopted. Compared with previous work, the major superiority of our work is twofold: (i) The control torques are derived analytically by our model, which contains the information of reference inputs, position feedback and velocity feedback, thus reference joint angles can be accurately tracked, while the control signals in [14–16] are simple sinusoidal signals; (ii) In our model, the parameter uncertainty in the model is handled by using sliding mode control, thus the control law is still effective in the case of existence of uncertainty, which is inevitable in the model. While to the best of our knowledge, this problem is not mentioned in other models.

Third, we present the relations among speed, turning radius and related parameters for the four-link robotic fish. Based on the relations, we build a motion library, from which the robotic fish can choose suitable parameters according to various scenarios. We give elaborated tasks to show the application of the motion library to motion planning of the robotic fish. Also, a motion planning experiment which contains visual feedback of camera is presented. Compared with other works, the major superiority of our work is: A motion library, that contains the relations between speed, turning radius of the fish and parameters of undulation frequency, amplitude, phase difference, deflections, is constructed. Although some works [24] [34] cover part of the contents, to the best of our knowledge, the motion library presented in this chapter contains the most detailed and the most elaborated relations in existing works.

Fourth, we present a new form of CPG model, which consists of coupled Andronov-Hopf oscillators, an artificial neural network (ANN), and an outer amplitude modulator. By using this model, we successfully applied swimming data of a real fish to our Anguilliform robotic fish, and the robotic fish is able to swim forward and backward as predicted. Compared with other works, the major superiority of our work is threefold: (i) Unlike previous works that use only coupled oscillators therefore can only generate fixed-pattern waveforms, we add artificial neural network and an outer amplitude modulator to the CPG structure, which makes it possible to generate different kinds of waveforms. Specifically, the CPGs in our work can generate swimming pattern of a real fish, while to the best of our knowledge, other works do not possess such capability; (ii) Three-dimensional topology is used in structure design of the coupled oscillators, and faster contraction rate can be achieved compared with those use traditional one-dimensional or two-dimensional topologies. Also, the three-dimensional topology is more robust under perturbations; (iii) By using different parameters, both forward and backward locomotion patterns can be realized within one CPG structure.

1.3 Organization of Thesis

The thesis is organized as follows.

In Chapter 2, the mechanical model of the robotic fish and its Lagrangian formulation are given, then we obtain dynamics of the system and the relation between the motion of the fish and its external forces/torques.

In Chapter 3, analytical control torques are first given by using computed torque method. Due to the fact that the number of actuators is less than the number of the control input, the reference is redesigned after analyzing the equilibrium point of the system. To deal with parameter uncertainties in the system, sliding mode control is proposed.

In Chapter 4, three common locomotion patterns of Anguilliform fish are obtained by assigning different reference angles to each joint of the fish, and corresponding experiments are given.

In Chapter 5, the relations among the speed of the fish, oscillation frequency, angle amplitude, and phase difference are investigated. Based on the relations, a motion library is built. By choosing appropriate parameters from the motion library, the robotic fish can achieve different tasks.

In Chapter 6, the CPG approach is applied to the robotic fish such that it is able to conduct locomotion learning from a real fish. Experiments are conducted to verify the effectiveness of the CPG approach.

In Chapter 7, conclusion of the thesis is given.

Chapter 2

Modeling of the Anguilliform Fish Robot

2.1 Introduction

In this chapter, first locomotion classification for different types of fishes is given, and illustrates the character of each type. Then, some literature review about mathematical modeling of fishes is given.

Since there are so many types of fishes in the world, it is necessary for us to know the particular character of each type of fishes, then select the most suitable one. According to different body structures and locomotion patterns, fishes are usually classified into two categories: the first is called body and/or caudal fin (BCF) locomotion, and the second is called median and/or paired fin (MPF) locomotion. [7] The most remarkable characteristic of BCF locomotion is that, when the fish is moving forward, there is a body wave traveling backward from the fish's head to its tail, and the thrust is generated by undulation of their bodies. In MPF locomotion, the bodies of fishes mainly stay rigid or have unobservable movement, thus the thrust is produced by oscillation of their median and paired fins instead of their bodies. Generally speaking, BCF locomotion is more efficient than MPF locomotion considering energy consumption, while MPF locomotion excels in maneuverability compared with BCF locomotion. It is estimated that only 15% of fishes use non-BCF locomotion as their routine propulsive style, while others rely on BCF mode. It can be seen that BCF is a more common locomotion mode that fishes adopt, thus we mainly consider BCF type in this work.

In BCF locomotion, there are three main types of fishes: Anguilliform, Carangiform, and Thunniform, as shown in Fig. 2.1-2.3. Anguilliform fishes, which are typical of eels, lampreys, have long and flexible bodies. When an Anguilliform fish moves, the whole body participates in large amplitude undulation. Carangiform fishes, which include mackerel and snapper, have narrow peduncles and tall forked caudal fins. Carangiform locomotion also involves undulation of the whole body, but large amplitude undulation is mainly confined to the last one third part of the body, and the thrust is produced by the rather stiff caudal fin [39]. Carangiform fishes usually swim faster than Anguilliform fishes, but slower than Thunniform fishes. Thunniform fishes, including tuna and some sharks, have very low-drag streamline body shapes, narrow peduncles, and tall lunate caudal fins. In Thunniform, the undulation proportion on the body is even less than that in Carangiform, and most part of the body remain stiff. Their unique body structures lead to their high cruising speed. In this paper, we mainly focus on Anguilliform fish, because it has higher maneuverability and more locomotion patterns compared with the other two swimming modes [7].

Inspired by the appealing merits that real fishes possess, such as power efficient, maneuverable, flexible, and noiseless propulsion, researchers have developed many theories and numerous robotic fish prototypes to study and mimic the way that real fishes move.

Apart from EBT [11, 12], many other mathematical models are established. In [17],

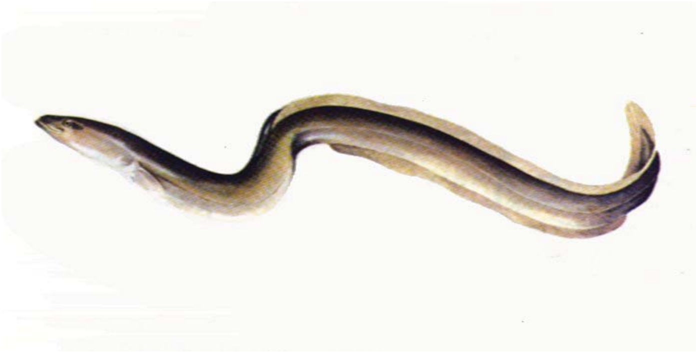


Figure 2.1: Anguilliform fish.

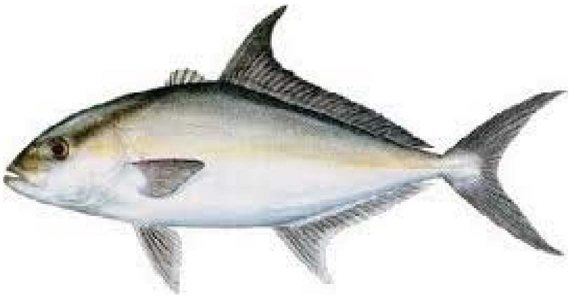


Figure 2.2: Carangiform fish.

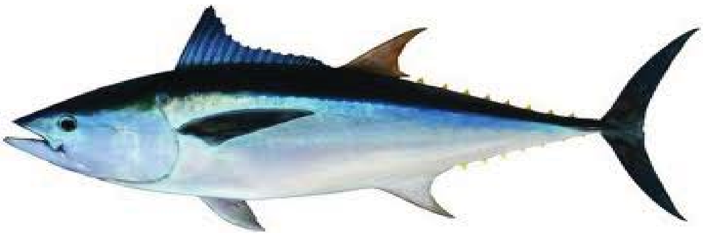


Figure 2.3: Thunniform fish.

the authors presented the dynamic modeling of a continuous three-dimensional swimming eel-like robot. The modeling approach was based on the geometrically exact beam theory and on Newton-Euler formulation. The proposed algorithm was used to compute the robots Galilean movement and the control torques as a function of the expected internal deformation of the eel's body. In [40], modeled after the ostracion meleagris, a dynamic model is presented for a robotic fish driven by its pectoral fins. In [41], a planar model for the swimming of certain marine animals was proposed based on reduced Euler-Lagrange equations for the interaction of a rigid body and an incompressible fluid. This model assumed the form of a control-affine nonlinear system with drift; preliminary accessibility analysis suggested its utility in predicting efficacious gaits for piscimimetic robots. In [23], the authors presented a simplified dynamic model and open-loop control routines for Anguilliform fishes, and compared experimental results to analytically derived, but approximated expressions for proposed gaits for forward/backward swimming, circular swimming, sideways swimming and turning in place. In [26], the authors investigated some issues of momentum generation for a class of eel-like swimming robots, and issues of control and motion planning for it. In [18] considered a biologically inspired sensor-based "centering" behavior for undulatory robots, which could traverse corridor-like environments. [42], the authors presented a neuronal model and a mechanical model of fish swimming, and combined the two models together by the transformation of the motoneuron activity to mechanical forces and feedback of fish movements to stretch receptors. In [21], the dynamic model of a multi-joint robotic fish is given. The effects of trailing vortex, leading-edge suction force are considered, and central pattern generators are used to produce the swimming data. In [16], the effects of added mass, quasi-steady lift, and drag are considered, then a system model is built in a control-affine structure.

By using geometric nonlinear control theory, a trajectory tracking algorithm is developed for a free-swimming underwater vehicle. In [14], based on quasi-steady fluid flow theory, the modeling, control design and experimental trajectory tracking results for a planar Carangiform robotic fish are presented. However, in these modeling methods, the precise relation between the torques added on the robotic fish and the motion of the fish is lacking, even though the relation is compulsory for control method design.

In this chapter, a links-and-joints based robotic fish model is presented. Considering the constraints existing in this mechanical model, Lagrangian method is adopted to analyze its dynamics, and the analytical relation between the motion of the fish and the external forces/torques is obtained. Due to the fact that the number of actuators is less than the number of the control input, reference planning method is adopted to obtain appropriate reference inputs. Compared with previous works, the major superiority of our work is that: Unlike [11], [12] and [17], which treat the fish body as a smooth and continuous curve, a mathematical model for the robotic fish is constructed which consists of joints and links, which is more of practical concern. The model gives the relation between torques added on the robotic fish and the corresponding motion of the fish. According to this model, control torques can be given analytically.

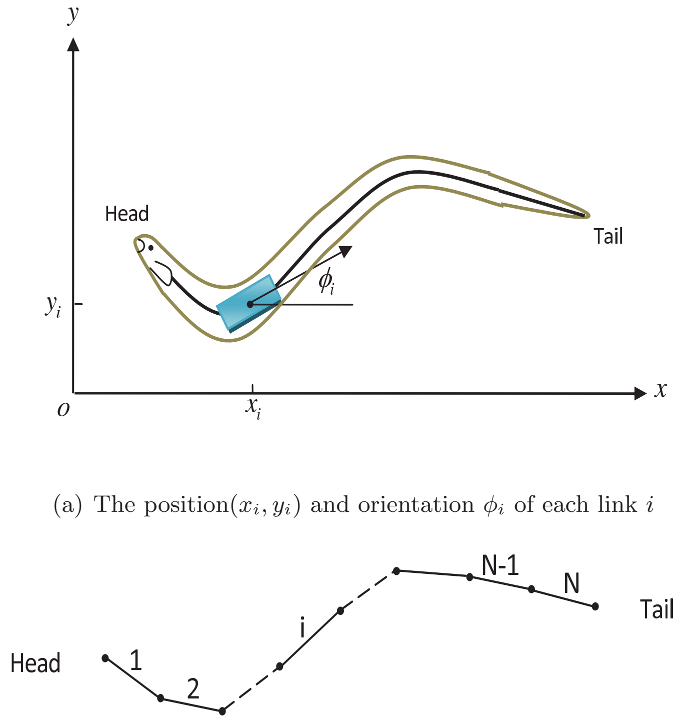
2.2 Fish Body Sketch

From a biological perspective, we recall that in Anguilliform swimming mode, the whole body of the fish, from head to tail, participates in large amplitude undulations. Every part of the fish's body contributes to its motion, which is different from the pattern that Carangiform or Thunniform fish moves. The most remarkable characteristic in Anguilliform fish moving process, is that there exists a body wave, traveling from head to tail [11]. Obviously, the traveling direction of the body wave in the fish is backwards, which is opposite to the direction that fish moves forward.

In [11], the authors gave the propulsive model of real Anguilliform fish, mainly from a mathematical point of view. This model gave a basic principle when design the robotic fish, however, considering the implementation of the fish by using links and joints, the model may not be applicable because the fish is abstracted into a smooth curve. Another limitation in the model of [11] is that only steady state motion of the fish was considered, while the dynamics of the fish motion was not handled. In our work, we will construct a dynamic model of the Anguilliform fish, and construct the relations between the motion of the fish and the control input (the external torques) added on it.

In nature, the geometry feature of Anguilliform fish, such as eels or lampreys, is complicated to describe in mathematical functions. For simplicity, we use links and joints to mimic the shape of Anguilliform fish. As shown in Fig. 2.4, we select the central line, which locates at the center of the fish body and stretches from head to tail, to represent the Anguilliform fish. The fish consists of N links and $N-1$ joints, where two connective links are connected by one joint. There is one motor on each joint, and it exerts torque to its neighboring links.

Fig. 2.4 shows the top view of the central curve of the Anguilliform fish. xoy is the world coordinates system. The position and orientation of each link i are described by three coordinates x_i, y_i and ϕ_i: x_i and y_i denote the position of the midpoint of link i, while ϕ_i denotes the angle from $+x$-axis to link i. The links are numbered from head to tail (see Fig. 2.4B). Each link i is impacted by two types of external forces: hydrodynamic forces w_i and torques τ_i, τ_{i-1} (see Fig. 2.5).



(a) The position(x_i, y_i) and orientation ϕ_i of each link i

(b) Numbering of links

Figure 2.4: Sketch of the Anguilliform robotic fish model. (a) Position and orientation representation. (b) Link numbering.

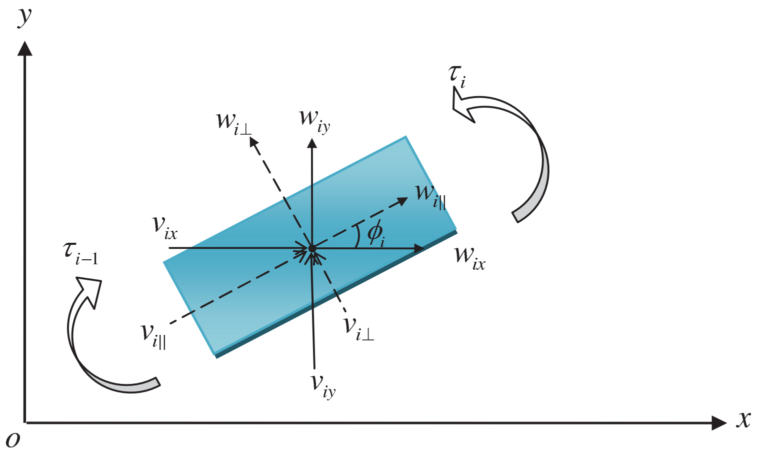


Figure 2.5: External forces acting on link i.

2.3 Hydrodynamic Force

When there is relative motion between the fish and the surrounding fluid, fluid is displaced and hydrodynamic force arise. The force can be obtained through surface integrals of vector force per area around the fish body. Since this force is related with the geometry of the object immersed in water and relative velocity between the object and water, in principle, the exact force distribution can be obtained by solving the Navier-Stokes equation. However, the calculation is quite complicated and time consuming [43]. As shown in Fig. 2.5, we adopt a simplified approximation of this force as (2.1) and (2.2) indicate

$$w_{i\perp} = -f_{i\perp}(v_{i\perp})^2 sgn(v_{i\perp}) \tag{2.1}$$

$$w_{i\|} = -f_{i\|}(v_{i\|})^2 sgn(v_{i\|}) \tag{2.2}$$

where $v_{i\perp}$, $v_{i\|}$ are perpendicular component and parallel component of the velocity v_i, and $f_{i\perp}$, $f_{i\|}$ are the water resistance coefficients in corresponding directions. The notation $sgn(\cdot)$ represents $+1$ if the element in the parentheses is positive or -1 if negative. Based on the geometric relationship (refer to Fig. 2.5), we have

$$\begin{aligned} v_{i\perp} &= -v_{ix}\sin\phi_i + v_{iy}\cos\phi_i \\ v_{i\|} &= v_{ix}\cos\phi_i + v_{iy}\sin\phi_i \\ w_{ix} &= -w_{i\perp}\sin\phi_i + w_{i\|}\cos\phi_i \\ w_{iy} &= w_{i\perp}\cos\phi_i + w_{i\|}\sin\phi_i \end{aligned}$$

where v_{ix}, v_{iy} are projection of the velocity v_i on x-axis and y-axis; w_{ix}, w_{iy} are projection of the hydrodynamic force w_i on x-axis and y-axis. All of them are scalars.

Hydrodynamic forces experienced by all the links can be calculated the same way.

Since the link velocity v_i can be possibly in any direction, it is arduous to find each water resistance coefficient f in corresponding direction. Fortunately, f remains unchanged in the direction of parallelling the link, as well as in the perpendicular direction. Thus, the hydrodynamic forces are calculated in such a way that the need of the value of f in arbitrary direction is avoided.

2.4 Lagrangian Formulation of the Mechanical Model

In dynamic analysis of robotics, two approaches dominate: Newton-Euler formulation and Lagrangian formulation. The major difference between them is that Newton-Euler formulation is a force balance based approach to dynamics, while Lagrangian formulation is a energy based approach to dynamics [44]. From energy perspective, Lagrangian formulation regards a mechanical system as a whole, thus usually has a neat form. Additionally, Lagrangian formulation can handle internal forces in a much easier way. In this part, Lagrangian formulation will be applied to the fish model.

First, we define coordinates vector $\mathbf{p} \in \Re^{3N}$ as

$$\mathbf{p} = [x_1, y_1, \phi_1, x_2, y_2, \phi_2, \cdots, x_N, y_N, \phi_N]^T \tag{2.3}$$

where the notation $(\cdot)^T$ denotes transpose of a vector or a matrix $(\cdot)$. The robotic fish has $N + 2$ freedom, but there are $3N$ coordinates totally. Thus there are more than necessary number of variables to fully describe the system, which means these variables are not independently to each other. Due to this, some constraints will be added to describe their internal relationship. Two consecutive links are constrained by the same joint, forcing them jointed together. By use of the position of each joint, the constraints can be expressed as

$$\begin{aligned} x_i + \frac{l_i}{2}\cos\phi_i &= x_{i+1} - \frac{l_{i+1}}{2}\cos\phi_{i+1} \\ y_i + \frac{l_i}{2}\sin\phi_i &= y_{i+1} - \frac{l_{i+1}}{2}\sin\phi_{i+1} \end{aligned} \tag{2.4}$$

where $i \in \{1, 2, \cdots, N-1\}$, l_i is the length of link i. The above constraints can be reformulated in matrix form

$$\mathbf{g}(\mathbf{p}) = \begin{bmatrix} x_1 + \frac{l_1}{2}\cos\phi_1 - x_2 + \frac{l_2}{2}\cos\phi_2 \\ y_1 + \frac{l_1}{2}\sin\phi_1 - y_2 + \frac{l_2}{2}\sin\phi_2 \\ x_2 + \frac{l_2}{2}\cos\phi_2 - x_3 + \frac{l_3}{2}\cos\phi_3 \\ y_2 + \frac{l_2}{2}\sin\phi_2 - y_3 + \frac{l_3}{2}\sin\phi_3 \\ \vdots \\ x_{N-1} + \frac{l_{N-1}}{2}\cos\phi_{N-1} - x_N + \frac{l_N}{2}\cos\phi_N \\ y_{N-1} + \frac{l_{N-1}}{2}\sin\phi_{N-1} - y_N + \frac{l_N}{2}\sin\phi_N \end{bmatrix} = 0$$

Note that the number of total constraints is $2(N-1)$, thus $\mathbf{g}(\mathbf{p}) \in \Re^{2(N-1)}$

Next, define the Lagrangian

$$L(\mathbf{p}, \dot{\mathbf{p}}) = K(\mathbf{p}, \dot{\mathbf{p}}) - V(\mathbf{p}) \tag{2.5}$$

where K is the kinetic energy, V is the potential energy and L is the total energy of the system, all written in the coordinates we just defined, and they can be calculated as follows

$$\begin{aligned} K(\mathbf{p}, \dot{\mathbf{p}}) &= \sum_{i=1}^{N}\{\frac{1}{2}m_i(\dot{x}_i^2 + \dot{y}_i^2) + \frac{1}{2}I_i\dot{\phi}_i^2\} \\ V(\mathbf{p}) &= 0 \end{aligned}$$

here $V(\mathbf{p})$ is zero because our fish model is a two-dimensional model in a horizontal plane.

Define $J(\mathbf{p})$ as the Jacobian of the constraints matrix $\mathbf{g}(\mathbf{p})$

$$J(\mathbf{p}) = \frac{\partial \mathbf{g}(\mathbf{p})}{\partial \mathbf{p}}$$

where $J(\mathbf{p}) \in \Re^{2(N-1)\times 3N}$. Here we give the expressions of $J(\mathbf{p})$ and $\dot{J}(\mathbf{p})$ for future use.

$$J(\mathbf{p}) = \begin{bmatrix}
1 & 0 & -\frac{l_1}{2}\sin\phi_1 & -1 & 0 & -\frac{l_2}{2}\sin\phi_2 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 1 & \frac{l_1}{2}\cos\phi_1 & 0 & -1 & \frac{l_2}{2}\cos\phi_2 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & -\frac{l_2}{2}\sin\phi_2 & -1 & 0 & -\frac{l_3}{2}\sin\phi_3 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & \frac{l_2}{2}\cos\phi_2 & 0 & -1 & \frac{l_3}{2}\cos\phi_3 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 & -\frac{l_{N-1}}{2}\sin\phi_{N-1} & -1 & 0 & -\frac{l_N}{2}\sin\phi_N \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & \frac{l_{N-1}}{2}\cos\phi_{N-1} & 0 & -1 & \frac{l_N}{2}\cos\phi_N
\end{bmatrix}$$

$$\dot{J}(\mathbf{p}) = \begin{bmatrix}
0 & 0 & -\frac{l_1}{2}\dot{\phi}_1\cos\phi_1 & 0 & 0 & -\frac{l_2}{2}\dot{\phi}_2\cos\phi_2 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & -\frac{l_1}{2}\dot{\phi}_1\sin\phi_1 & 0 & 0 & -\frac{l_2}{2}\dot{\phi}_2\sin\phi_2 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -\frac{l_2}{2}\dot{\phi}_2\cos\phi_2 & 0 & 0 & -\frac{l_3}{2}\dot{\phi}_3\cos\phi_3 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -\frac{l_2}{2}\dot{\phi}_2\sin\phi_2 & 0 & 0 & -\frac{l_3}{2}\dot{\phi}_3\sin\phi_3 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & -\frac{l_{N-1}}{2}\dot{\phi}_{N-1}\cos\phi_{N-1} & 0 & 0 & -\frac{l_N}{2}\dot{\phi}_N\cos\phi_N \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & -\frac{l_{N-1}}{2}\dot{\phi}_{N-1}\sin\phi_{N-1} & 0 & 0 & -\frac{l_N}{2}\dot{\phi}_N\sin\phi_N
\end{bmatrix}$$

It is found that

$$J(\mathbf{p})\dot{\mathbf{p}} = \frac{\partial \mathbf{g}(\mathbf{p})}{\partial \mathbf{p}}\dot{\mathbf{p}} = \frac{d}{dt}(\mathbf{g}(\mathbf{p})) = 0 \tag{2.6}$$

since $\mathbf{g}(\mathbf{p}) = 0$. Differentiating (2.6) yields

$$J(\mathbf{p})\ddot{\mathbf{p}} + \dot{J}(\mathbf{p})\dot{\mathbf{p}} = 0 \tag{2.7}$$

Then write the constraint force, i.e., internal force, as

$$\Gamma = J(\mathbf{p})^T\lambda \tag{2.8}$$

where $\lambda \in \Re^{2(N-1)}$ is the vector of relative magnitudes of the constraint forces, and is commonly known as Lagrange multipliers. Afterwards, we give the external forces vector which acts on individual coordinate of $\mathbf{p}$

$$\mathbf{w} = [w_{1x}, w_{1y}, \tau_1, w_{2x}, w_{2y}, \tau_2 - \tau_1, \cdots, w_{Nx}, w_{Ny}, -\tau_{N-1}]^T \tag{2.9}$$

where w_{ix}, w_{iy} $(i = 1, \cdots, N)$ represent the horizontal component and vertical component of the hydrodynamic force w_i, $\tau_i - \tau_{i-1}$ represents the total torque exerted on link i. It should be noted that $\tau_0 = \tau_N = 0$, since there is no torques at the endpoints.

The equations of motion are formed by considering the constraint forces as an additional force which affects the motion of the system, as well as the external forces. Hence, the dynamics of the system can be written as

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{p}}} - \frac{\partial L}{\partial \mathbf{p}} = \mathbf{w} + \Gamma \tag{2.10}$$

By substituting (2.5), (2.8) and (2.9) into (2.10), we get

$$M\ddot{\mathbf{p}} = \mathbf{w} + J(\mathbf{p})^T \lambda \tag{2.11}$$

where M is the mass matrix and it can be written as

$$M = diag\{m_1, m_1, I_1, m_2, m_2, I_2, \cdots, m_N, m_N, I_N\}$$

where m_i is the mass and I_i is the moment of inertia of link i. The notation $diag\{\cdots\}$ represents that M is a diagonal matrix, and the diagonal elements are in the braces.

Considering (2.7), (2.11) can be transformed into the following form

$$\begin{aligned} J(\mathbf{p})M^{-1}J(\mathbf{p})'\lambda &= J(\mathbf{p})\ddot{\mathbf{p}} - J(\mathbf{p})M^{-1}\mathbf{w} \\ &= -\dot{J}(\mathbf{p})\dot{\mathbf{p}} - J(\mathbf{p})M^{-1}\mathbf{w} \end{aligned}$$

The matrix $JM^{-1}J'$ is full rank since the constraints are independent [2]. Hence the Lagrange multipliers is obtained

$$\lambda = (J(\mathbf{p})M^{-1}J(\mathbf{p})')^{-1}(-\dot{J}(\mathbf{p})\dot{\mathbf{p}} - J(\mathbf{p})M^{-1}\mathbf{w}) \tag{2.12}$$

Using this equation, the Lagrange multipliers is computed as a function of the current state $\mathbf{p}$, $\dot{\mathbf{p}}$ and external force $\mathbf{w}$. The information of $\ddot{\mathbf{p}}$ can be obtained by substituting λ back to (2.11), then we get

$$\ddot{\mathbf{p}} = A(\mathbf{p})\dot{\mathbf{p}} + B(\mathbf{p})\mathbf{w} \tag{2.13}$$

where $A(\mathbf{p}) = -M^{-1}J'(JM^{-1}J')^{-1}\dot{J}$, $B(\mathbf{p}) = M^{-1}[I - J'(JM^{-1}J')^{-1}JM^{-1}]$, I is identity matrix with the same dimension as M. Therefore, the motion of the robotic fish is determined.

(2.13) contains all the acceleration terms, of which we are more interested in angular acceleration terms $\ddot{\phi}_i$. By partitioning (2.13), we get equations that only contain angular acceleration terms

$$\ddot{\phi} = A_1(\mathbf{p})\dot{\mathbf{p}} + B_1(\mathbf{p})\mathbf{w}_x + B_2(\mathbf{p})\mathbf{w}_y + B_3(\mathbf{p})B_\tau\tau \tag{2.14}$$

where

$$\begin{aligned}
\phi &= [\phi_1, \phi_2, \cdots, \phi_N]^T \\
\mathbf{w}_x &= [w_{1x}, w_{2x}, \cdots, w_{Nx}]^T \\
\mathbf{w}_y &= [w_{1y}, w_{2y}, \cdots, w_{Ny}]^T \\
\tau &= [\tau_1, \tau_2, \cdots, \tau_{N-1}]^T
\end{aligned}$$

$$B_\tau = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -1 & 1 & \ddots & \vdots \\ 0 & -1 & \ddots & 0 \\ \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & -1 \end{bmatrix}$$

and $A_1(\mathbf{p}) \in \Re^{N\times 3N}, B_1(\mathbf{p}) \in \Re^{N\times N}, B_2(\mathbf{p}) \in \Re^{N\times N}, B_3(\mathbf{p}) \in \Re^{N\times N}$ are corresponding coefficient matrices obtained from matrix $A(\mathbf{p})$, $B(\mathbf{p})$ in (2.13). It is worth noting that the dimension of τ is $N-1$, one less than the total number of links N.

2.5 Conclusion

From a biomimetic point of view, this chapter focuses on the modeling of an Anguilliform robotic fish.

In the beginning, a simplified mechanical model of an Anguilliform robotic fish is established, which is based on links and joints. Next, hydrodynamic forces are formulated to describe the interaction forces between the fish and the water analytically. Then, through Lagrangian formulation considering constraints of connecting links, the dynamic equation of the robotic fish is derived, which construct the relations between the motion of the fish and the torques added on it. At last, we partition the complete dynamic equation and obtain equation that only contain angular acceleration terms. This model reveals the relation between torques added on the fish and corresponding motion of the fish. Also, the model is critical for simulating dynamic motion of the fish and developing appropriate control method which will be introduced later.

Chapter 3

Control Law Design

3.1 Introduction

In the previous chapter, we have derived a mathematical model for the robotic fish. In this chapter, control approaches will be given based on the model.

The robotic fish is substantially a robot, and we need to apply manipulation to it, so that the robotic fish can achieve our desired behavior. Therefore, it is necessary for us to find an appropriate way to control the fish. In this chapter, two controllers design are given, computed torque control and sliding mode control.

Many works have been done on the subject of the control of robotic fish. In [13], the authors designed a four-link robotic fish, and implemented a PID controller and a fuzzy logic controller to control its speed and orientation respectively. In the experiment, a point-to-point control algorithm was implemented and an overhead vision system was adopted to provide real-time visual feedback. In [14], the modeling, control design and experimental trajectory tracking results for a planar Carangiform robotic fish was presented. The model for the fish's propulsion was based on quasi-steady fluid flow theory. Using this model, the paper proposed gaits for forward and turning trajectories and analyzed system response under such control strategies. [15] considered the task of trajectory stabilization for a fish-like robot by means of feedback. Authors used oscillatory control inputs and applied correction signals at the endpoints of each periodic input signal. Such a strategy was proven to cause the system to converge to a desired trajectory and experiment results verified stabilization. In [16], techniques from geometric mechanics and geometric nonlinear control theory, were applied to modeling and construction of trajectory tracking algorithms for a free-swimming underwater vehicle, that locomotions and maneuvers using a two-link actuated tail and independently actuated pectoral fin bow planes. Restricting consideration of fluid forces to the simple effects of added mass and quasi-steady lift and drag, the resulting system model can be expressed in a control-affine structure. In [17], dynamic modeling of a continuous three-dimensional swimming eel-like robot is presented. The proposed algorithm is able to to compute the robot's Galilean movement and the control torques as a function of the expected internal deformation of the eel's body. In [18], the authors considered a biologically inspired sensor-based centering behavior for undulatory robots traversing corridor-like environments. The biomimetic centering behavior has been implemented, both by explicit body shape control and by neuromuscular control of body undulations. In [19], grounded on an optimized kinematic and dynamic model, a free-swimming multi-link robotic fish and its motion control are designed. Employing top-down design approach, a hierarchical architecture is proposed for the system which consists of five different levels. In [20], a closed-loop maneuvering control method is proposed to enhance the turning precision and turning response speed of a robotic fish. In the method, the turning maneuver is divided into three individual phases: the bending, holding, and unbending phases. In [45], a fish-mimetic underwater robot is developed with good dynamics performance. Control system is given and three turning modes are discussed for the fish robot that uses tail swing.

In this chapter, based on the previously derived mathematical model of the robotic fish, different control approaches are developed. First, computed torque control method is presented. In this method, torques are calculated by using joint angle positions, joint angle velocity, and their references. To deal with parameter uncertainty and external disturbance, which always arise in practical circumstance, sliding mode control is adopted. Compared with previous work, the major superiority of our work is twofold: (i) The control torques are derived analytically by our model, which contains the information of reference inputs, position feedback and velocity feedback, thus reference joint angles can be accurately tracked, while the control signals in [14–16] are simple sinusoidal signals; (ii) In our model, the parameter uncertainty in the model is handled by using sliding mode control, thus the control law is still effective in the case of existence of uncertainty, which is inevitable in the model. While to the best of our knowledge, this problem is not mentioned in other models.

3.2 Computed Torque Control

Forward motion is the most common locomotion pattern of Anguilliform fish. One of the characters of Anguilliform fish swimming is that, if the fish moves forward, there exists a body wave traveling backwards. Since the wave travels from its head to tail, the head is preceding the tail affected by the wave. More generally, the movement of the former part of the body has a phase lead than the latter one, and it is reflected in the phase difference among the link orientation angle ϕ_j $(j = 1, \cdots, N)$. Considering the fact that all parts of the body participate in large amplitude movement in Anguilliform fish, it is reasonable for us to let the amplitude of ϕ_j be the same. For the reason that the backward moving wave has the same oscillating frequency at different places, it is supposed that ϕ_j follow the same angular frequency. Following these considerations, we let the reference angles ϕ_{jr} assume the following form

$$\phi_{jr} = A_m \cdot \sin[\omega t + (2 - j)\theta] \tag{3.1}$$

where $j = 1, 2, \cdots, N$. t denotes time instant. A_m, ω are the amplitude and angular frequency of ϕ_{jr} respectively, and θ is the phase lag of link i comparing with its former one.

Given the desired ϕ_{jr}, our control objective is to let ϕ_j follow ϕ_{jr}, concerning the dynamics in (2.14). There are many control methods in the field of robotics can achieve this objective, and one of widely used methods is computed torque method. [46] The advantage of this method is that it can convert a nonlinear problem to a linear one, which is easier to handle. Based on computed torque method, we design torque τ. In (2.14), since the input matrix $B_{33}B_0$, which is associated with τ, is not a square matrix, we cannot solve τ directly by using the inverse of $B_{33}B_0$. By multiplying the transpose of B_0 on both sides of (2.14) and adding proportional-derivative feedback terms, we derive τ as

$$\tau = (B_\tau^T B_3 B_\tau)^{-1} B_\tau^T [\ddot{\phi}_r + k_1(\phi_r - \phi) + k_2(\dot{\phi}_r - \dot{\phi}) - (A_1 \dot{\mathbf{p}} + B_1 \mathbf{w}_x + B_2 \mathbf{w}_y)] \tag{3.2}$$

where $\phi_r = [\phi_{1r}, \phi_{2r}, \cdots, \phi_{Nr}]^T$, ϕ_{jr} $(j = 1, 2, \cdots, N)$ is reference angle of link j. Position feedback and velocity feedback are added to compensate error between ϕ_j and ϕ_{jr}, and k_1, k_2 are corresponding coefficients.

In our model, we select $N = 4$, i.e., the robotic fish consists of 4 links. Table 3.1 shows mechanical parameters of the links, where l_i, m_i, I_i are the length, mass and moment of inertia of link i respectively, $f_{i\perp}$ and $f_{i\|}$ are water resistance coefficients that are identified through experiments (the experiment will be introduced in Section 4.2.3). Their SI units are $m(meter)$, kg, $kg \cdot m^2$, Ns^2/m^2, Ns^2/m^2.

Table 3.1: Mechanical parameters of the links.

Link #	l_i	m_i	I_i	$f_{i\perp}$	$f_{i\|}$
1.	0.22	0.313	1.260×10^{-3}	3.75	1.11
2.	0.12	0.171	2.052×10^{-4}	2.05	0.61
3.	0.12	0.171	2.052×10^{-4}	2.05	0.61
4.	0.20	0.285	9.500×10^{-4}	3.41	1.01

Based on (3.1), we give the reference angles ϕ_{jr}. In the first scenario, parameters are chosen as $A_m = 0.45$, $\omega = 2\pi$, $\theta = 1.6$, and the feedback coefficients $k_1 = 10$, $k_2 = 1$. At time $t = 0$, the fish is still, and its four links are aligned on x-axis with its head on the origin, which means $\mathbf{p} = [0.11\ 0\ 0\ 0.28\ 0\ 0\ 0.40\ 0\ 0\ 0.56\ 0\ 0]'$. By applying control torque in (4.1), we get simulation results of the actual angles ϕ, angular errors, torques trajectory, and x_1 trajectory, as shown in Fig. 3.1 ∼ Fig. 3.4.

3.3 Sliding Mode Control

Modeling inaccuracies always exist and have strong adverse effects on control systems. Thus, any practical design must address them explicitly [47]. Otherwise, the control law may lose effect since the actual parameters deteriorate the performance of the whole system. Here we adopt sliding mode control (SMC), which belongs to robust controllers. In this robotic fish system, it is obvious that the number of actuators is less than that of reference input. Therefore, we have some considerations on that when design the sliding mode control law.

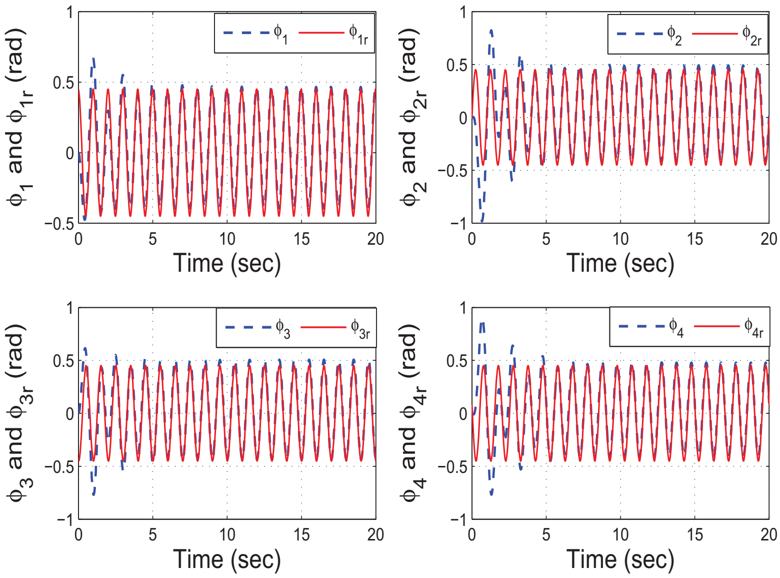


Figure 3.1: Scenario 1: Actual angle ϕ and reference angle ϕ_r trajectory, with parameters $A_m = 0.45, \omega = 2\pi, \theta = 1.6$.

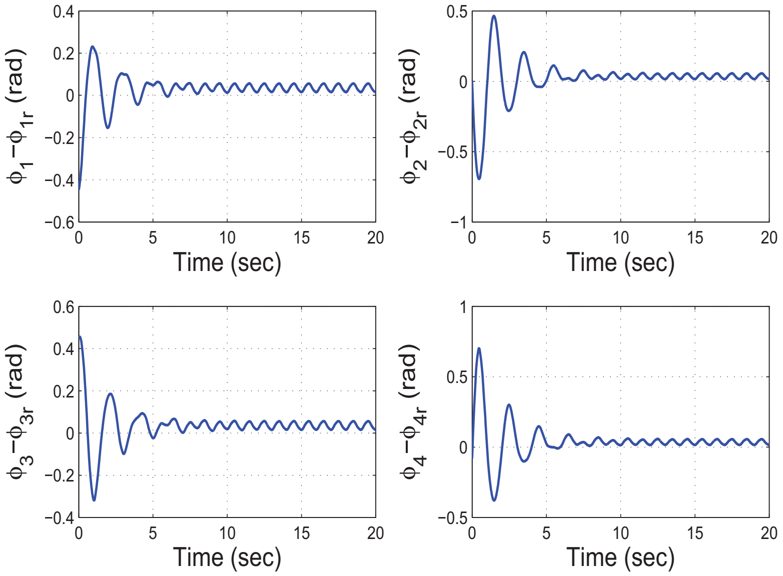


Figure 3.2: Scenario 1: Angular errors, with parameters $A_m = 0.45, \omega = 2\pi, \theta = 1.6$.

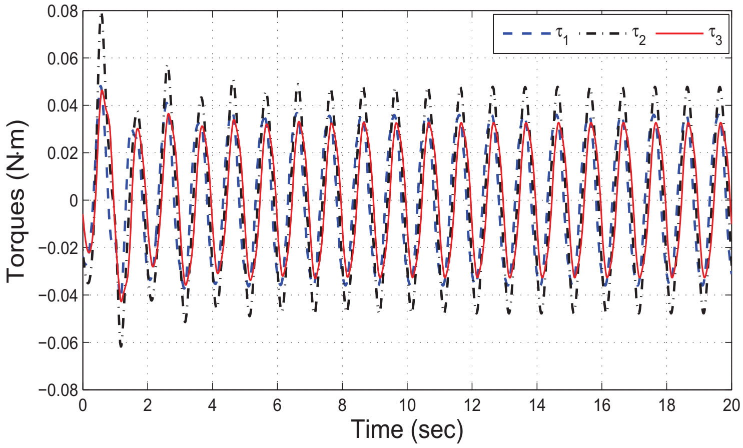


Figure 3.3: Scenario 1: Torques trajectory, with parameters $A_m = 0.45, \omega = 2\pi, \theta = 1.6$.

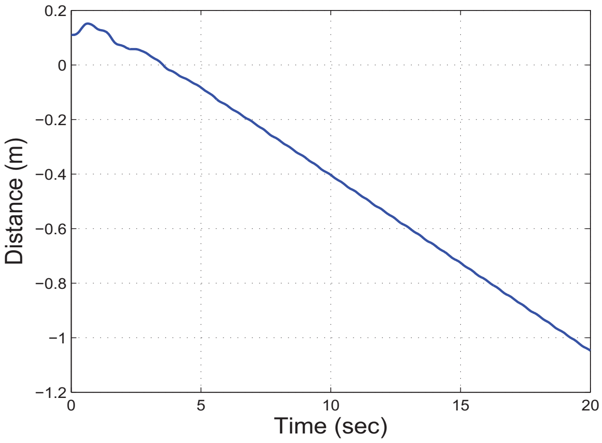


Figure 3.4: Scenario 1: x_1 trajectory, with parameters $A_m = 0.45, \omega = 2\pi, \theta = 1.6$.

3.3.1 Parameter uncertainty

In the robotic fish model we construct, many parameters involve uncertainties. These uncertainties either come from inaccuracy in the modeling, or come from unpredictable influence from surroundings. The water resistance coefficient f can be affected by many factors, such as different velocities of fish with respect to the environment. Thus, it is rather an estimated parameter than an accurate one. The mass matrix M can be measured accurately on ground, but when the fish comes into water, M becomes inaccurate because of added mass effect. Though we cannot know the exact information of the interested parameters due to complex factors, those parameters always change in a predictable range. This is reasonable because every parameter has its own physical meaning, thus it neither blows up to infinity nor becomes too small. Then, an upper bound and a lower bound can always be given for each parameter.

We define that

$$\begin{aligned} F_{\perp} &= diag\{f_{1\perp}, f_{2\perp}, \cdots, f_{N\perp}\} \\ F_{\parallel} &= diag\{f_{1\parallel}, f_{2\parallel}, \cdots, f_{N\parallel}\} \end{aligned}$$

Assume that there exist parameter uncertainties on M, $F_{\perp}$ and $F_{\parallel}$, and their norms are bounded.

For derivation convenience, define that

$$\begin{aligned} \mathbf{v}_{\perp\sin} &= [v_{1\perp}^2 sgn(v_{1\perp}) \sin\phi_1, \cdots, v_{N\perp}^2 sgn(v_{N\perp}) \sin\phi_N]^T \\ \mathbf{v}_{\perp\cos} &= [v_{1\perp}^2 sgn(v_{1\perp}) \cos\phi_1, \cdots, v_{N\perp}^2 sgn(v_{N\perp}) \cos\phi_N]^T \\ \mathbf{v}_{\parallel\sin} &= [v_{1\parallel}^2 sgn(v_{1\parallel}) \sin\phi_1, \cdots, v_{N\parallel}^2 sgn(v_{N\parallel}) \sin\phi_N]^T \\ \mathbf{v}_{\parallel\cos} &= [v_{1\parallel}^2 sgn(v_{1\parallel}) \cos\phi_1, \cdots, v_{N\parallel}^2 sgn(v_{N\parallel}) \cos\phi_N]^T \end{aligned}$$

Then, hydrodynamic forces $\mathbf{w}_x$ and $\mathbf{w}_y$ can be written as

$$\mathbf{w}_x = F_{\perp}\mathbf{v}_{\perp\sin} - F_{||}\mathbf{v}_{||\cos} \tag{3.3}$$

$$\mathbf{w}_y = -F_{\perp}\mathbf{v}_{\perp\cos} - F_{||}\mathbf{v}_{||\sin} \tag{3.4}$$

where $\mathbf{w}_{x0}, \mathbf{w}_{y0}$ are nominal values of $\mathbf{w}_x, \mathbf{w}_y$. From a practical point of view, the coordinate $\mathbf{p}$ and its derivative $\dot{\mathbf{p}}$ are always bounded, which indicates that both $A(\mathbf{p})$ and $B(\mathbf{p})$ are bound, because $A(\mathbf{p})$ and $B(\mathbf{p})$ are functions of $\mathbf{p}$ and $\dot{\mathbf{p}}$. Together with the fact that $F_{\perp}$ and $F_{||}$ are bounded, it is known that $\mathbf{w}_x$ and $\mathbf{w}_y$ are bounded from (3.3) and (3.4).

3.3.2 Sliding mode control law design

Generally, there are two standard steps in sliding mode control design: 1) a sliding surface is given such that system on it manifests desired behavior; 2) a discontinuous control law is utilized to drive the system states into that surface and stay on it for all future time [48].

The sliding mode control law is composed of two parts. The first part is used to handle the nominal model, while the second is used to handle system uncertainties. Since there are not enough number of actuators to track all the reference inputs, we have to make a trade-off when setting the control objective, i.e., tracking the same number of reference inputs as that of actuators. In this model, the number of actuators, i.e., the number of torques, applied on the fish is $N-1$. Thus, we make the dimension of the sliding surface be $N-1$.

Before designing the sliding surface, we first define angular error and its derivative

$$\begin{aligned}\mathbf{e} &= \phi_{New} - \phi_{rNew} \\ \dot{\mathbf{e}} &= \dot{\phi}_{New} - \dot{\phi}_{rNew}\end{aligned}$$

where $\phi_{New} = [\phi_1, \phi_2, \cdots, \phi_{N-1}]^T$, representing the first $N-1$ actual joint angles, and $\phi_{rNew} = [\phi_{1r}, \phi_{2r}, \cdots, \phi_{(N-1)r}]^T$, representing the first $N-1$ reference joint angles. The dynamics of ϕ_{New} is

$$\begin{aligned}\ddot{\phi}_{New} &= A_2(\mathbf{p})\dot{\mathbf{p}} + B_4(\mathbf{p})\mathbf{w}_x + B_5(\mathbf{p})\mathbf{w}_y + B_6(\mathbf{p})B_\tau\tau \\ &= A_{2n}(\mathbf{p})\dot{\mathbf{p}} + B_{4n}(\mathbf{p})\mathbf{w}_{xn} + B_{5n}(\mathbf{p})\mathbf{w}_{yn} + B_{6n}(\mathbf{p})B_\tau\tau + \mathbf{d}(\mathbf{p}, \tau, t) \qquad (3.5)\end{aligned}$$

where $A_2(\mathbf{p})$, $B_4(\mathbf{p})$, $B_5(\mathbf{p})$, $B_6(\mathbf{p})$, are submatrices obtained from matrix $A(\mathbf{p})$, $B(\mathbf{p})$ in (2.13), corresponding to $\ddot{\phi}_{New}$, and $A_{2n}(\mathbf{p})$, $B_{4n}(\mathbf{p})$, $B_{5n}(\mathbf{p})$, $B_{6n}(\mathbf{p})$ are their nominal values. $\mathbf{d}(\mathbf{p}, \tau, t) = (A_2 - A_{2n})\dot{\mathbf{p}} + (B_4\mathbf{w}_x - B_{4n}\mathbf{w}_{xn}) + (B_5\mathbf{w}_y - B_{5n}\mathbf{w}_{yn}) + (B_6B_\tau\tau - B_{6n}B_\tau\tau)$, represents the difference between the actual terms and nominal terms. In the expression of $d(\mathbf{p}, \tau, t)$, all the terms are bounded, thus we assume that the norm of it has an upper bound

$$||\mathbf{d}(\mathbf{p}, \tau, t)|| \leq d_{\max}$$

From the definition of $\mathbf{e}$, we set the control objective as tracking the first $N-1$ reference inputs. Next, define the sliding surface as

$$\sigma = C\mathbf{e} + \dot{\mathbf{e}} \qquad (3.6)$$

where C is a diagonal matrix whose entries are positive scalars.

Assume that information of the coordinate vector $\mathbf{p}$ and its velocity $\dot{\mathbf{p}}$ is available by means of vision or other measurement system. Now, we give the control law. As stated before, the control law consists of two parts

$$
\begin{aligned}
\tau &= \tau_0 + \tau_s && (3.7)\\
\tau_0 &= (B_{6n}B_\tau)^{-1}[\ddot{\phi}_{rNew} - C(\dot{\phi}_{New} - \dot{\phi}_{rNew}) - (A_{2n}\dot{\mathbf{p}} + B_{4n}\mathbf{w}_{xn} + B_{5n}\mathbf{w}_{yn})] && (3.8)\\
\tau_s &= -\rho(B_{6n}B_\tau)^{-1}\frac{\sigma}{||\sigma||} && (3.9)
\end{aligned}
$$

where $\rho = d_{\max} + \eta$, η is a positive constant. τ_0 is used to handle nominal model, τ_s is used to handle the uncertainties.

Then, we have the following theorem.

Theorem 3.1. *Consider the nonlinear system (3.5) associated with the chosen sliding surface $\sigma = 0$. Under the control law (3.7)-(3.9), the sliding surface will be reached in finite time.*

Proof. First define the Lyapunov function

$$V = \frac{1}{2}\sigma^T\sigma$$

Differentiating it, we obtain

$$
\begin{aligned}
\dot{V} &= \sigma^T\dot{\sigma} = \sigma^T(C\dot{\mathbf{e}} + \ddot{\mathbf{e}})\\
&= \sigma^T[C(\dot{\phi}_{New} - \dot{\phi}_{rNew}) - \ddot{\phi}_{rNew} + (A_2\dot{\mathbf{p}} + B_4\mathbf{w}_x + B_5\mathbf{w}_y + B_6B_\tau\tau)]\\
&= \sigma^T[C(\dot{\phi}_{New} - \dot{\phi}_{rNew}) - \ddot{\phi}_{rNew} + (A_{2n}\dot{\mathbf{p}} + B_{4n}\mathbf{w}_{xn} + B_{5n}\mathbf{w}_{yn} + B_{6n}B_\tau\tau) + \mathbf{d}]
\end{aligned}
$$

Substituting (3.7), (3.8) and (3.9) into $\dot{V}$, one obtains

$$\begin{aligned}\dot{V} &= \sigma^T[-\rho\frac{\sigma}{||\sigma||}+\mathbf{d}] \\ &= -\rho||\sigma||+\sigma^T\mathbf{d} \\ &\leq -\rho||\sigma||+||\sigma||\cdot||\mathbf{d}|| \\ &\leq -\rho||\sigma||+d_{\max}||\sigma|| \\ &= -\eta||\sigma||\end{aligned}$$

It is obvious that $\dot{V}$ is negative definite. By Lyapunov theorem for stability [47], the equilibrium at the origin $\sigma = 0$ is asymptotically stable. If $\phi_{New}(t = 0)$ is off $\phi_{rNew}(t = 0)$ in the beginning, the sliding surface is reached in a finite time $t_{reach} \leq \|\sigma(t = 0)\|_\infty/\eta$, where $\|(\cdot)\|_\infty$ denotes the $\infty-$norm of $(\cdot)$. After the system reaches the sliding surface $\sigma = 0$, it stays there. In the sliding mode, $\sigma(t) = 0$, $\dot{\sigma}(t) = 0$, the equivalent control is $\tau_{eq} = (B_6B_\tau)^{-1}[\ddot{\phi}_{rNew}-C(\dot{\phi}_{New}-\dot{\phi}_{rNew})-(A_2\dot{\mathbf{p}}+B_4\mathbf{w}_x+B_5\mathbf{w}_y)]$. (3.6) gives the dynamics of $\mathbf{e}$, which contains the first three angular errors. Since all the entries of the diagonal matrix C are chosen to be positive scalars, it is easy to show that on the sliding surface, each single element of $\mathbf{e}$ always converges to 0, thus yielding the result that the first three reference inputs can be well tracked.

□

3.3.3 Numerical examples

In this part, we also use the four-link robotic fish model, with the same mechanical parameters as shown in subsection 3.3.

We adopt forward locomotion of the fish here to illustrate the performance of the SMC method. Based on (3.1), the reference angles ϕ_{jr} are given. Since this robotic fish is essentially an underactuated system, tracking of arbitrary number of reference inputs is impossible. However, by reference planning approach, which conducts equilibrium analysis at the neighborhood of the equilibrium point, this problem can be handled. Parameters are chosen as $A_m = 0.45$, $\omega = 2\pi$, $\theta = 1.6$. For simplicity, it is supposed that the uncertainty is in the following form: $M = (1+\alpha)M_0$, $F_\perp = (1+\beta_1)F_{\perp 0}$, $F_\parallel = (1+\beta_2)F_{\parallel 0}$, where $\alpha = 0.2$, $\beta_1 = 0.2$, $\beta_2 = 0.2$.

In the second scenario, sliding mode control is used, and select the parameters as $C = I_3$ (a 3 × 3 identity matrix), $d_{\max} = 10$, $\eta = 0.1$. At time $t = 0$, the fish is still, and its four links are aligned on x-axis with its head at the origin. The actual angles and reference angles are shown in Fig. 3.5, and the control torques are shown in Fig. 3.6. Fig. 3.7 shows the distance that the fish has traveled, and it can be seen that the fish can move forward normally.

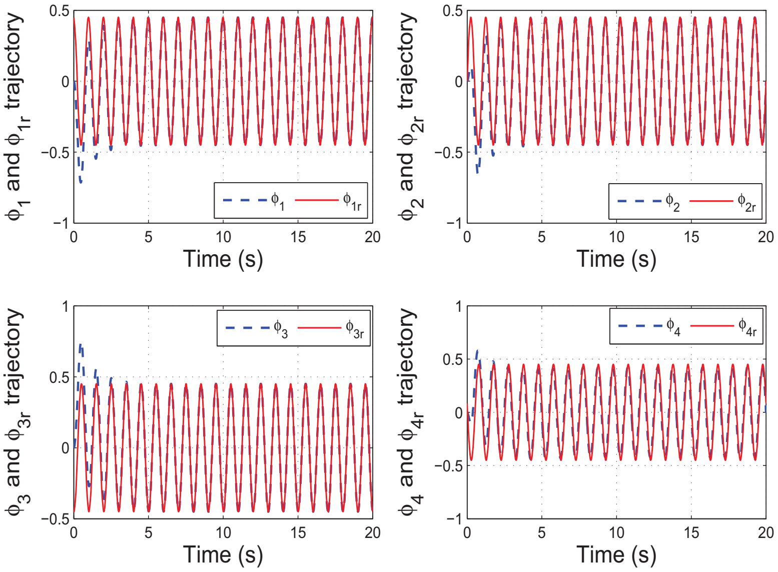


Figure 3.5: Scenario 2: Actual angle ϕ and reference angle ϕ_r trajectory, with parameters $A_m = 0.45, \omega = 2\pi, \theta = 1.6$.

Under the existence of parameter uncertainties, now we compare the performance of sliding mode control and the performance of computed torque control. In this case, the

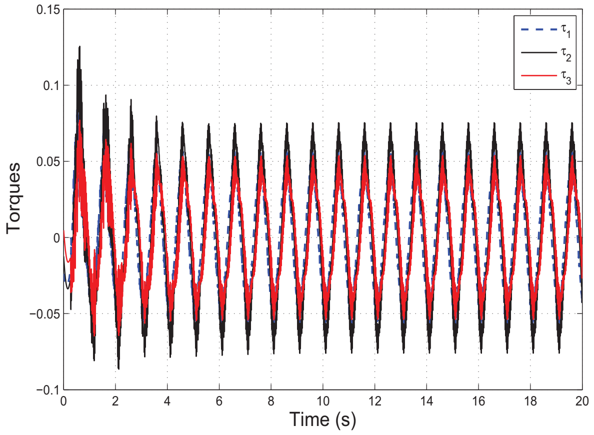


Figure 3.6: Scenario 2: Torques trajectory (sliding mode control using sign function, with parameters $A_m = 0.45, \omega = 2\pi, \theta = 1.6$).

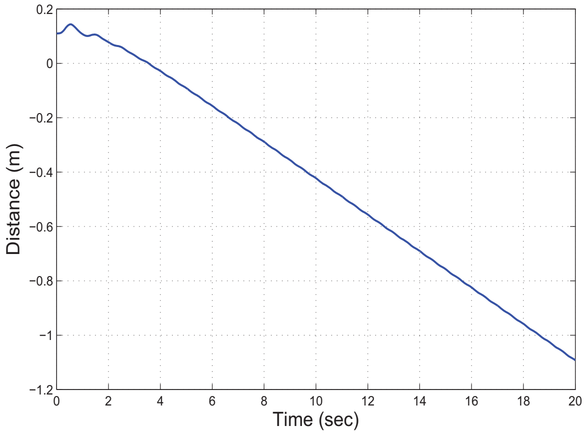


Figure 3.7: Scenario 2: x_1 trajectory, with parameters $A_m = 0.45, \omega = 2\pi, \theta = 1.6$.

parameters and initial condition of computed torque control are the same as those of the first scenario. The only difference is that, parameter uncertainties exist.

The comparison of angular error between sliding mode controller and computed torque controller is shown in Fig. 3.8. It is obvious that by SMC method, the first three joint angular errors quickly converge to 0 after a short period of time, but the fourth joint angular error can not converge to 0 since the system is an underactuated system. While by computed torque method, the first three joint angular errors are much larger than those obtained from SMC method, and none of them converge to 0 in the end.

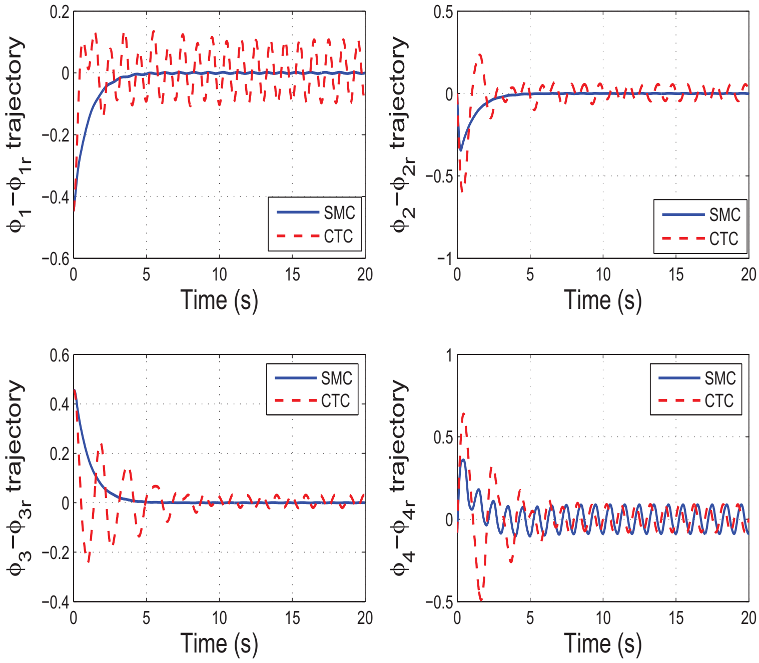


Figure 3.8: Comparison of angular error between sliding mode control (SMC) and computed torque control (CTC), under the existence of parameter uncertainties.

Note that chattering phenomenon exists in the second scenario, which is a character of sliding mode control. The reason of chattering is that $\frac{\sigma}{\|\sigma\|}$ in (3.9) is not a continuous function. In the third scenario, in order to have a smoother control signal and to benefit the virtue of sliding mode control, we replace $\frac{\sigma}{||\sigma||}$ with a saturation function $sat(\sigma)$

$$sat(\sigma) = \begin{cases} \dfrac{\sigma}{||\sigma||}, & \text{if } ||\sigma|| > \epsilon_1, \\ \dfrac{\sigma}{\epsilon_1}, & \text{otherwise.} \end{cases}$$

and here we choose $\epsilon_1 = 0.1$, with other parameters and initial condition are the same as the second scenario. The control torques are shown in Fig. 3.9, and the comparison of angular errors between using saturation function and sign function is shown in Fig. 3.10.

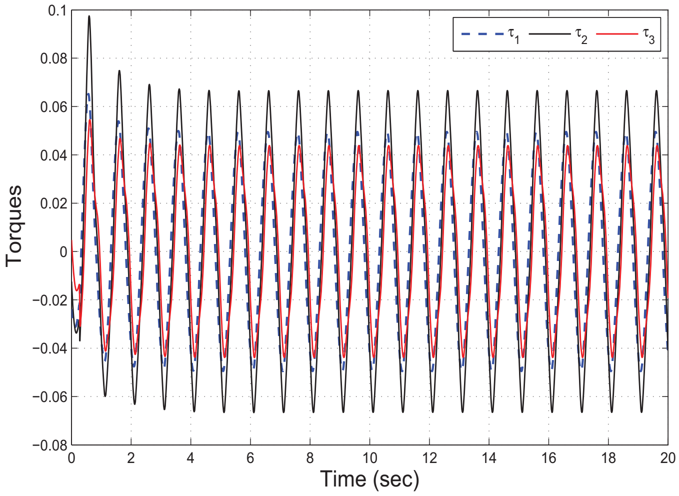


Figure 3.9: Scenario 3: Torques trajectory (sliding mode control using saturation function, with parameters $A_m = 0.45, \omega = 2\pi, \theta = 1.6, \epsilon_1 = 0.1$).

Comparing the three scenarios, it is found that when there exist parameter and unmodeled uncertainties, computed torque control cannot work very well, because the error between the actual joint angles and reference joint angles are large and always exists. When we use sliding mode control, the first three reference inputs can be perfectly tracked, thus we achieve the goal of tracking the first three angles by designing the sliding surface. The chattering phenomenon in Scenario 2 is adverse to actuators in practical implementation. We overcome this drawback by introducing a saturation

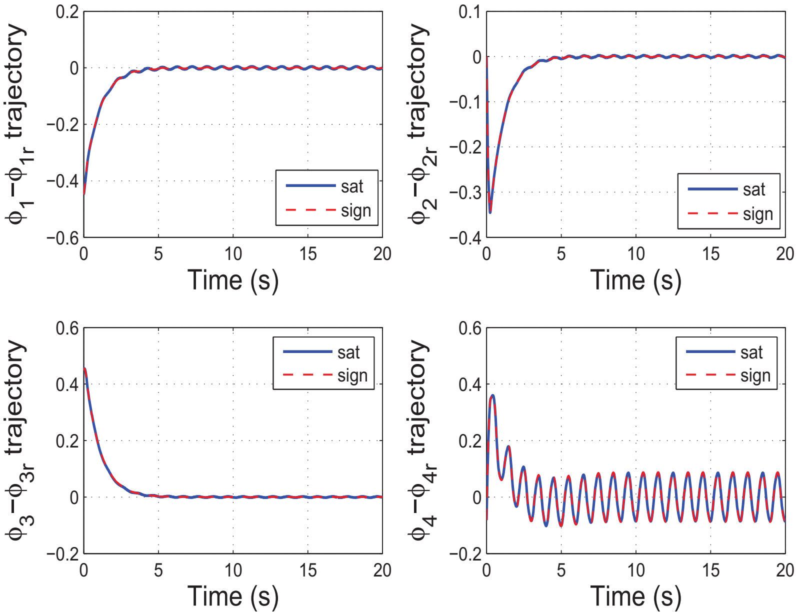


Figure 3.10: Comparison of angular error between Scenario 3: SMC with saturation function and Scenario 2: SMC with sign function.

function in Scenario 3, in which the control torques become much smoother, and the tracking performance has no noticeable change.

In Scenario 2 and 3, though ϕ_{1r}, ϕ_{2r} and ϕ_{3r} can be tracked accurately, perfect tracking of ϕ_{4r} cannot be promised theoretically. The fundamental reason is that the number of actuators in the system is fewer than the number of independent physical variables, which means arbitrary number of trajectory tracking is impossible.

3.4 Conclusion

This chapter mainly focuses on the control law design for the Anguilliform robotic fish, which applies on the previously derived mathematical model of the robotic fish.

Given the motion dynamics of the fish, torques are developed by using computed torque control method firstly, where position feedback and velocity feedback of joint angles are used. Numerical results show the effectiveness of the computed torque control law, and the robotic fish can move forward as predicted. Aiming at the practical circumstance where parameter uncertainties exist, sliding mode control is proposed to handle the actual system. Proof is given to show that the angular error involved in the sliding surface will converge to zero. By SMC approach, the first $N-1$ joint angles can be well tracked. Numerical results show that the effectiveness of SMC to resist uncertainties, and better tracking performance is obtained comparing with that uses computed torque control. Considering the chattering phenomenon, a saturation function is used in order to smoothen the control signals, and its performance makes no difference.

Chapter 4

Locomotion Generation

4.1 Introduction

In order to achieve practical tasks, the robotic fish need to move in different ways. Only forward moving, as presented in previous chapter, is not enough. After deriving the mathematical model of the robotic fish and investigating its control law, now we present how to generate different locomotion patterns for the robotic fish.

Since in the robotic fish system, there are many parameters that can be modulated, such as the amplitude of each joint angle A_m, the oscillation frequency ω, the phase difference between two connecting links θ, and the deflection angle for each link γ. How to tune or organize these parameters, such that the robotic fish can move in our desired pattern, is the problem we will solve in this chapter. The core principle to generate these locomotion patterns is that, we have to always relate the physical meaning of the useful parameters with the characters of the locomotions. In another way, we can say that we need to always think in a biomimetic way.

There are a lot of research work concerning the locomotion generation of robotic fish. In [21], the dynamic model of a multi-joint robotic fish is given. The effects of trailing vortex, leading-edge suction force are considered. By using the dynamic model, backward swimming pattern is investigated as a case study. In [17], the dynamic modeling of a continuous three-dimensional swimming eel-like robot is presented. By using the proposed control algorithm, forward locomotion and turning locomotion are detailed for the eel-like robot. In [26], the authors investigate basic issues of momentum generation for a class of dynamic mobile robots, focusing on eel-like swimming robots. For the three-link eel robot, forward gait and turning gait are generated. For the five-link eel robot, spinning gait and sideways gait are presented, which are not common in natural world. In [22] and [23], experimental verification of open-loop motion planning for a biomimetic robotic system is conducted, and different locomotion patterns including forward motion, circular path and turning in place, are investigated. In [15], the task of trajectory stabilization is considered for a robotic fish. By using feedback correction signal, the system is made to converge to a desired trajectory. In [24], the authors focus on turning control of a multi-link robotic fish in free swimming, and research on the parameters that determine turning performance, including magnitude, position, and time of the deflections applied to the links. In [25], by learning from different species of fish, the mechanism design and the motor control of swimming machines are shaped in different forms. From an engineering viewpoint, two different forms of design – serial open-chain design and parallel mechanism design – are given.

In this chapter, we present a real Anguilliform robotic fish and provide the details of its mechanical and electrical design. Then the robotic fish is tested in water. Three locomotion patterns of the Anguilliform robotic fish – forward moving, backward moving and turning – are investigated in simulations and experiments respectively, and comparisons are made. Since we mainly focus on the relation between the locomotion patterns and the reference input added on it, we simply adopt the control torques τ derived from computed torque control method in (4.1).

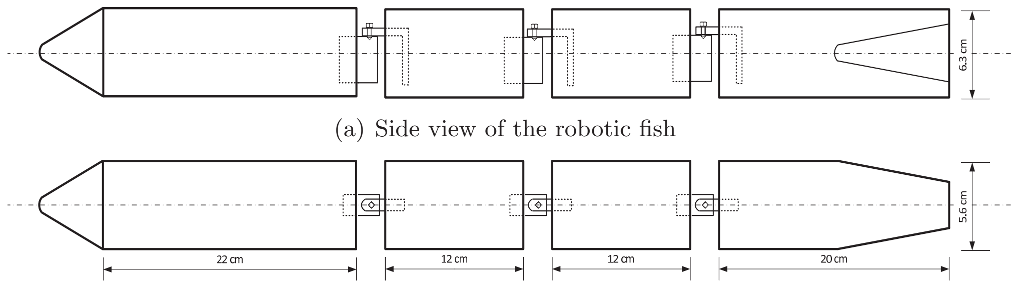
4.2 Experimental Setup

4.2.1 Robotic fish prototype and hardware description

To validate the effectiveness of the biomimetic model that has been developed, we build an Anguilliform type robotic fish as a test platform.

As shown in Fig. 4.1, the robotic fish body consists of four links, and every two neighboring links are connected by a servo motor and a metal connector, thus there are three motors in total. Basically, the shape of the middle part of the fish body is a cylinder, the shape of the fish head is a cone with a round nose for the purpose of reducing drag forces of water, and the shape of the fish tail is designed as a trapezoid (top view) in order to increase the area of interacting with water. The dimensions of the fish is $72 \times 5.6 \times 6.3$ cm^3 (Length×Width×Height). The robotic fish is made water proof to make sure that the electronic components can work normally. The density of the fish is lower than but approaching to 1000 kg/m^3, so when the fish swims in water, its position is near the water surface rather than at the bottom of the water. The distribution of mass along the whole fish is finely adjusted to ensure majority of the fish body mass is located in the lower part of it. Hence, the center of gravity is below the center of buoyancy to ensure the vertical stability of the robot. As a result, when put in water, the fish can erect itself without external supports.

We use an Atmega128 chip (Atmel Corporation) as the central microprocessor for processing all the calculations, a wireless communication module (bluetooth) to receive command from host computer, a Li-ion rechargeable battery to provide power for both the chips and the servo motors, and a toggle switch to switch on/off the battery. A



(a) Side view of the robotic fish

(b) Top view of the robotic fish

Figure 4.1: Sketch of the Anguilliform robotic fish.

battery indicator is used to show the battery status. The motors installed on the robotic fish are servo motors, by using which we can realize precise angle position control. The built-inside position feedback mechanism is one of the characteristics of servo motors. Therefore, the servo motors can follow the reference angles given by the host computer. Due to mechanical restriction, the angle range for the servo motors is about $-60° \sim 60°$. As shown in Fig. 4.2, the main electronics board, the bluetooth module, and the toggle switch are compacted within a small plastic box. All the electronic devices, including the servo motors, are made waterproof.

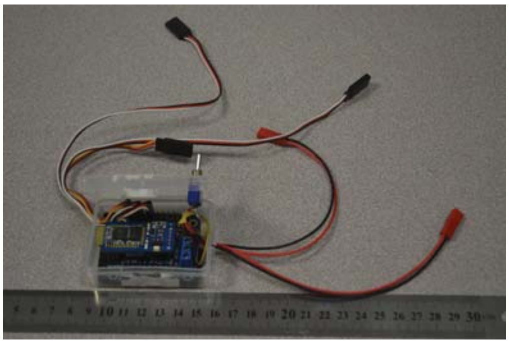


Figure 4.2: Electronics devices in a plastic box.

As shown in Fig. 4.3, the experiments in this paper are performed in a water tank of the size about 3×1.8 m^2 (length×width). Fig. 4.3 is a snapshot of the robotic fish swimming. It can be seen that, two yellow rulers, placed orthogonally at the bottom of the water, are used to indicate the position coordinates of the robotic fish. Therefore, the position of the fish is known at any time instant. In the experiment, a camera is used to record the movement of the Anguilliform robotic fish.

First, the fish is put at one end of the tank, and the fish maintains still. Next, reference angles are sent to the robotic fish from the host computer. After receiving the signals of reference angles through the wireless module, the processor transform them to pulse width modulation (PWM) signals to drive the servo motors. Then, the motors begin to work correspondingly. Corresponding locomotion patterns will be conducted by the fish according to the signals that it received. A camera is used to record the trajectories of the fish. The entire hardware configuration is shown in Fig. 4.4.

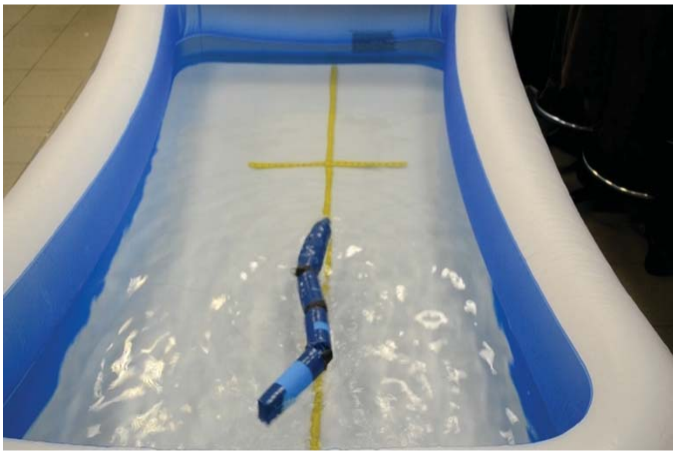


Figure 4.3: Snapshot of the robotic fish swimming.

4.2.2 Identification of water resistance coefficients

Before using the mathematical model of the robotic fish, we need to identify the water resistance coefficients. The parameters $f_{i\perp}$ and $f_{i\|}$ depend on the shape of the fish, the surface area and the material properties of the fish and the fluid. Since the robotic fish has an unique and specific shape, its water resistance coefficients need to be identified experimentally.

Equation (2.1) shows the relation between the water resistance coefficients and the

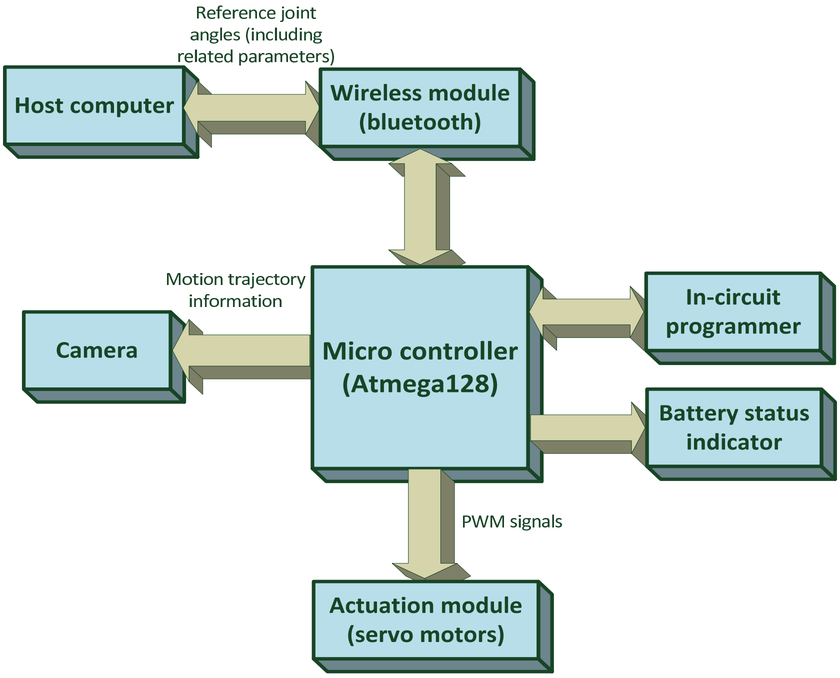

Reference joint angles (including related parameters)
Host computer
Wireless module (bluetooth)
Motion trajectory information
Camera
Micro controller (Atmega128)
In-circuit programmer
Battery status indicator
PWM signals
Actuation module (servo motors)

Figure 4.4: Block diagram of the hardware configuration.

drag force that the fish experiences. By this equation, the water resistance coefficients can be derived. The identification process is shown in Fig. 4.5

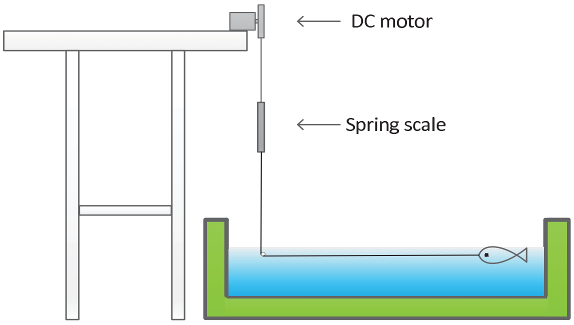


Figure 4.5: Identification of water resistance coefficients.

In the experiment set up as shown in Fig. 4.5, a DC motor and a spring scale are used. The DC motor rotates at a constant speed and the speed is set manually so it is known. Thus, the speed of the fish is constant and known. Since the fish moves straightly at a constant speed, the drag force is equal to the water resistance force. Because the drag force can be measured by the spring scale, the water resistance force can be obtained.

By using equation (2.1), we can calculate the parameters. Table 3.1 shows the identified water resistance coefficients.

4.3 Locomotion Generation for the Robotic Fish

In this section, three locomotion patterns of Anguilliform fish – forward moving, backward moving and turning, are investigated. Since we focus on the relations between the locomotion patterns and the reference input angles, we simply adopt the control torques τ derived from computed torque control method,

$$\tau = (B_\tau^T B_3 B_\tau)^{-1} B_\tau^T [\ddot{\phi}_r + k_1(\phi_r - \phi) + k_2(\dot{\phi}_r - \dot{\phi}) - (A_1\dot{\mathbf{p}} + B_1\mathbf{w_x} + B_2\mathbf{w_y})], \quad (4.1)$$

where $\phi_r = [\phi_{1r}, \phi_{2r}, \cdots, \phi_{Nr}]^T$, ϕ_{jr} $(j = 1, 2, \cdots, N)$ is reference angle of link j, and k_1, k_2 are coefficients related to feedback terms. Here we choose $k_1 = 10$, $k_2 = 1$.

In this model, we select $N = 4$, i.e., the robotic fish consists of four links. Mechanical parameters of the links are the same as shown in Table 3.1.

Note that in all the simulations and experiments, the initial condition is, the robotic fish is still, and its four links are aligned straightly on x-axis with its head on the origin. The "Distance" in the following Distance-Time graphs is expressed by x_1($x-$coordinate of the mid-point of the first link), for simulation convenience.

4.3.1 Forward locomotion

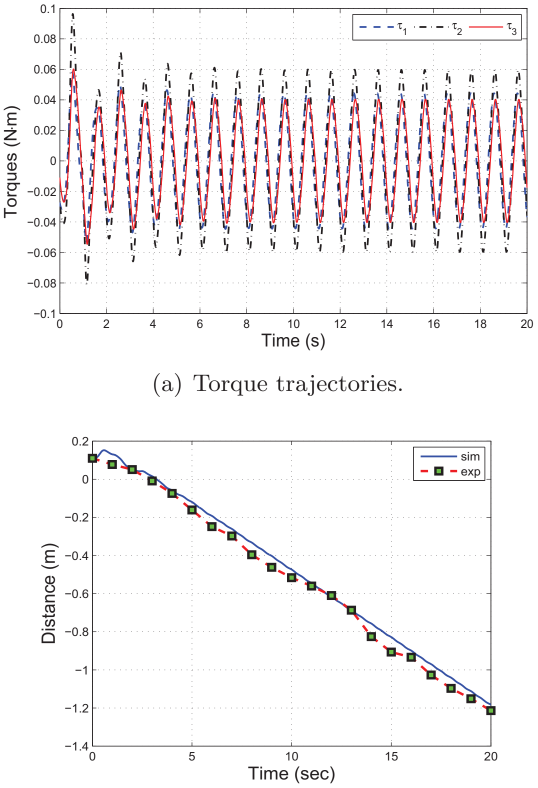
Forward locomotion is the most common locomotion pattern of Anguilliform fish. One of the characters of Anguilliform fish swimming is that, if the fish moves forward, there exists a body wave traveling backwards, which is the opposite direction of the fish moving. Since the wave travels from fish head to tail, the movement of the former part of the fish body has a phase leading the movement of the latter part, and this phenomenon is reflected in the phase difference among the link orientation angles ϕ_j $(j = 1, \cdots, N)$. Considering the fact that all parts of the body participate in large amplitude movement in Anguilliform fish, it is reasonable that the amplitudes of all the different ϕ_j be the same. For the reason that the backward traveling wave has the same oscillating frequency at different places, it is supposed that all of ϕ_j have the same angular frequency. Following these considerations, the reference ϕ_{jr} takes the following form

$$\phi_{jr} = A_m \cdot \sin[\omega t + (2 - j)\theta], \tag{4.2}$$

where $j = 1, 2, \cdots, N$. t denotes time instant. A_m, ω are the amplitude and angular frequency of ϕ_{jr} respectively, and θ is the phase lead of link i comparing with its latter one.

In this case, parameters are chosen $A_m = 0.45$ (rad), $\omega = 2\pi$ (sec^{-1}), $\theta = 1.5$ (rad). The trajectories of the three torques are shown in Fig. 4.6(a), and we see that the torque curves basically follow the sinusoidal pattern, as the reference angles in (4.2). That is because the joints are driven by the torques that exert on them, their waveforms generally follow the similar pattern. The distance-time graph of forward locomotion is shown in Fig. 4.6(b), where the legend “sim” represents simulation result and the legend “exp” represents the experimental result. We see that the simulation result and the experimental result are consistent with each other, and here we use percent relative mean-square error (RMSE%) [49] in (4.3) to evaluate the similarity between simulation data and experimental data. A smaller RMSE% implies higher similarity between the two data sets. In (4.3), s_i and e_i represents simulation data and experimental data at time $t = i$ sec, and we choose $n = 20$ here. In forward locomotion, RMSE% is 0.46%.

$$\text{RMSE\%} = 100 * \frac{\sum_{i=0}^{n}(s_i - e_i)^2}{\sqrt{\sum_{i=0}^{n} s_i^2}\sqrt{\sum_{i=0}^{n} e_i^2}} \tag{4.3}$$

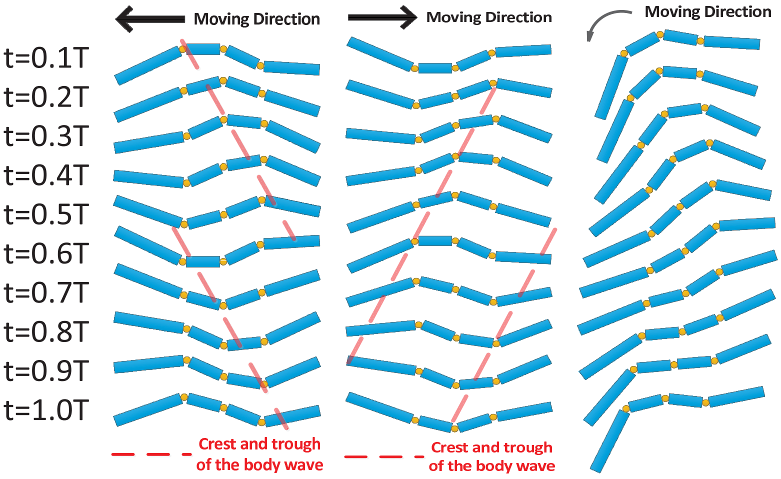


(a) Torque trajectories.

(b) Distance(x_1)-Time graph.

Figure 4.6: Distance(x_1)-Time graph and torque trajectories of forward locomotion, with parameters $A_m = 0.45$, $\omega = 2\pi$, $\theta = 1.5$.

Fig. 4.7(a) shows discretization of the forward locomotion of the robotic fish in a single complete cycle, where the period is 1 sec and the discretized time interval is 0.1 sec. From the figure we see that, when the fish moves forward, there is a body wave traveling backward.



(a) Forward. (b) Backward. (c) Turning.

Figure 4.7: Discretization of the three locomotions of the robotic fish in a single complete cycle.

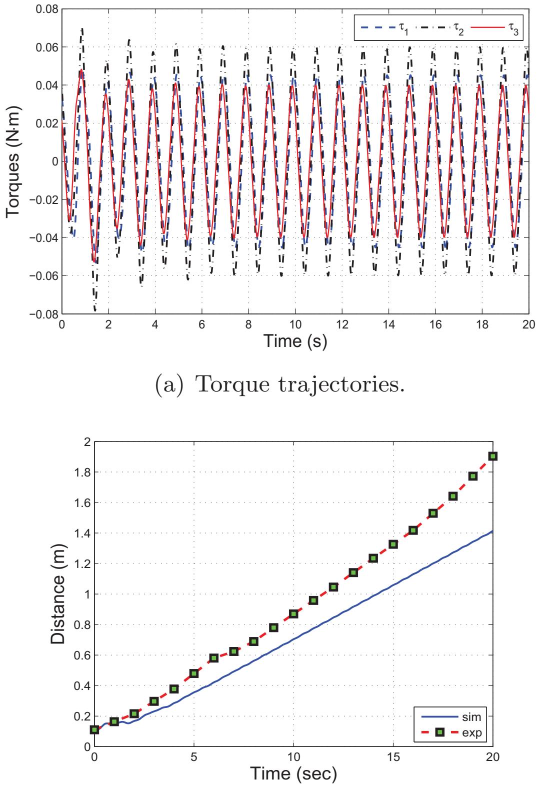
4.3.2 Backward locomotion

One of the unique locomotion patterns of Anguilliform fish, which differs from other types of fish such as Carangiform fish or Thunniform fish, is that Anguilliform fish can also move backwards. In this case, the direction of the body wave is opposite to that of the forward locomotion case, which means the body wave moves forward while the fish moves backward. Thus, the movement of the former part of the body has a phase lag compared with the latter part. As a result, we define the reference ϕ_{jr} as below

$$\phi_{jr} = A_m \cdot \sin[\omega t - (2 - j)\theta]. \tag{4.4}$$

Same as the forward case, parameters are chosen as $A_m = 0.45$ (rad), $\omega = 2\pi$ (sec^{-1}), $\theta = 1.5$ (rad). The trajectories of the three torques are shown in Fig. 4.8(a), and it is seen that the torque curves basically follow the sinusoidal pattern as the reference angles in (4.4). That is because the joints are driven by the torques that exert on them, their waveforms generally follow the similar pattern. The distance-time graph of backward locomotion is shown in Fig. 4.8(b). We see that the results of simulation and experiment are consistent. In backward locomotion, we calculate that RMSE% is 6.23%.

Fig. 4.7(b) shows discretization of the backward locomotion of the robotic fish in a single complete cycle, where the period is 1 sec and the discretized time interval is 0.1 sec. From the figure it is seen that, when the fish moves backward, there is a body wave traveling forward.



(a) Torque trajectories.

(b) Distance(x_1)-Time graph.

Figure 4.8: Distance(x_1)-Time graph and torque trajectories of backward locomotion, with parameters $A_m = 0.45$, $\omega = 2\pi$, $\theta = 1.5$.

Because the fish starts from still, it will experience an accelerating phase before swimming in a steady speed, when conducting forward locomotion or backward locomotion. As shown in Fig. 4.6(b) and Fig. 4.8(b), in the starting phase (about 0–4 sec), the fish accelerates itself, and the speed gradually increases. In this period, the fish travels a comparatively small distance due to the low average speed. After 4 sec, the fish is in the phase of steady speed, and the fish moves at a higher speed and maintains the speed at the current set of parameters. Thus, in the steady speed phase, the distance-time curves in Fig. 4.6(b) and Fig. 4.8(b) are approximately linear.

Remark 4.1. The significance of backward locomotion is that we can use it in spatially constrained scenarios. For instance, when the robotic fish swims forward into a narrow opening and wants to move back, there is no enough space for the fish to turn itself. At this time, backward locomotion is needed, by using which, the fish is able to move out of the opening without making a turn.

Remark 4.2. Note that though the results of simulation and experiments are consistent, there exist discrepancy between the simulation data and the experimental data. The reason is two fold. The first reason is that the robotic fish is inevitably affected by the reflection waves from the border of the swimming tank, due to the size limitation of the tank. The second reason is that in the robotic fish model we construct, many parameters involve uncertainties, which either come from inaccuracy in the modeling, or come from unmodeled dynamics. As a result, discrepancy arises between simulation and experiments.

4.3.3 Turning locomotion

In previous two cases, the time integral or average value of any reference angle is zero. Thus the movement of the fish neither deflects to the left side nor to the right, namely maintains in a straight line. In turning locomotion, the time integral or average value of any reference angle is a non-zero value, which represents an offset or a deflection. Thus, the reference angles take the following form:

$$\phi_{jr} = A_m \cdot \sin[\omega t + (2-j)\theta] + \gamma(j), \tag{4.5}$$

where $\gamma(j)$ represents angle deflection added on different joints. By using computed torque control, the fish can achieve turning movement by following the given reference angles. Similar to preceding two cases, parameters are chosen as $A_m = 0.45$ (rad), $\omega = 2\pi$ (sec^{-1}), $\theta = 1.5$ (rad), and the deflection $\gamma = [\frac{\pi}{4}\ \frac{\pi}{6}\ \frac{\pi}{12}\ 0]$. The trajectories of the three torques are shown in Fig. 4.9, and we see that the torque curves basically follow the sinusoidal pattern as the reference angles in (4.5). Note that the torques in both forward locomotion and backward locomotion are symmetric about 0. However, in turning case, the torques are not symmetric about 0, but have an offset. In this particular case, the offset is negative. The reason is that, the reference angles in (4.5) are not symmetric about 0 but have a deflection. Thus, the torques, which generate the joint angles, have deflections themselves.

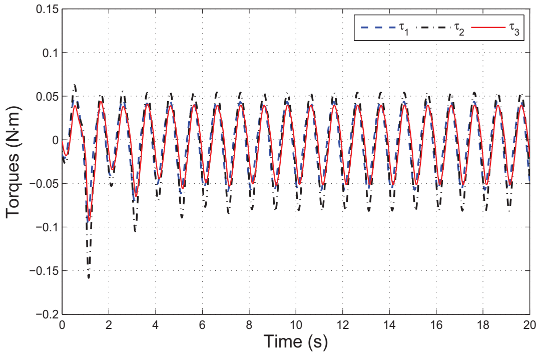
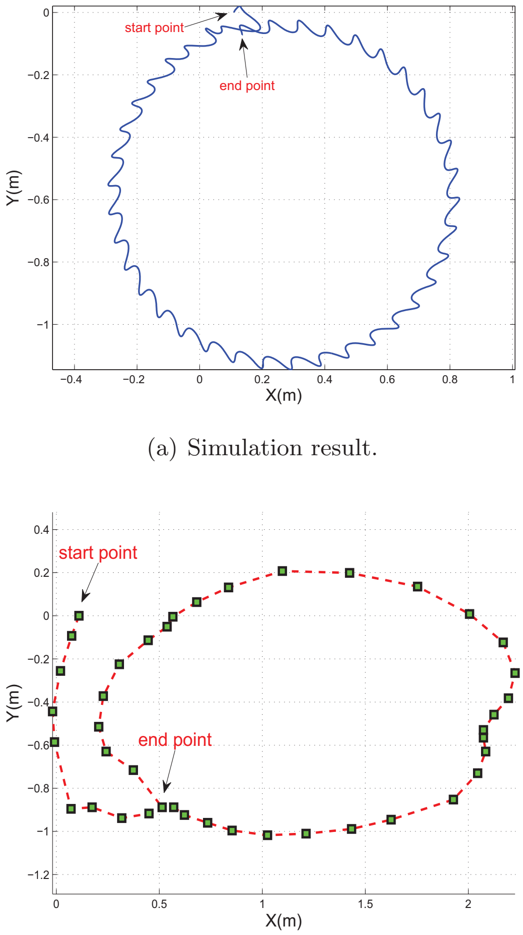


Figure 4.9: Torque trajectories of turning locomotion, with parameters $A_m = 0.45$, $\omega = 2\pi$, $\theta = 1.5$, $\gamma = [\frac{\pi}{4}\ \frac{\pi}{6}\ \frac{\pi}{12}\ 0]$.

Fig. 4.7(c) shows discretization of the turning locomotion of the robotic fish in a single complete cycle, where the period is 1 sec and the discretized time interval is 0.1 sec. From the figure we see that, when the fish makes a turn, there exist angle deflections



(a) Simulation result.

(b) Experiment result.

Figure 4.10: $x - y$ trajectory of turning locomotion.

between different joints. Since the reference angles in turning locomotion consist of two terms as shown in (4.5), and the second term assign different values for different links, thus it is not easy to identify the traveling body wave compared with the forward and backward locomotion cases.

$x - y$ trajectory of the fish is shown in Fig. 4.10. From Fig. 4.10, we see that the results of simulation and experiment are basically consistent, each trajectory can form a closed circle, which indicates that the fish is doing a turning locomotion. However, the path in the experiment is not a circle but an ellipse. There are several factors for the discrepancy. First of all, in the experiment, because of the size limitation of the water tank, there is inevitably some reflection waves from the boundary of the tank. These reflection waves continuously push the fish, and gradually, the fish deviates from its original trajectory. Since the water tank is rectangular, the turning trajectory is near to the boundaries in one direction but far from the boundaries in the other direction. Hence, along the direction (short axis of the ellipse in our case) that is nearer to the boundaries, the effect from reflection waves is larger. While along the other direction, the effect is smaller. The different effects of waves along two directions make the final turning trajectory not circular. Second, in the simulation work, we mainly focus on the dominant reason how the robotic fish can swim, i.e., the generation of body wave and interaction with surrounding water. However, some unmodeled dynamics, such as wake, are still there and they will affect the system in some degree. Third, in the simulation, the coefficients of hydrodynamic forces may not be the same as the ones that the robotic fish actually experience.

In Fig. 4.10 (a), we note that there exists fluctuation in the turning trajectory. The reason is that when the fish is moving, there exists a body wave traveling on the fish.

Thus, the undulation of the body leads to the yaw motion of fish. Hence, fluctuation appears in the trajectory of the fish.

Note that all the elements of the deflection vector γ are non-negative in this case, and the fish will turn in a counter-clockwise direction. If all the elements of the deflection vector γ are non-positive, then the fish will turn in a clockwise direction.

In (4.5), the fish is turning while moving forward. Similarly, if the reference angles are as in (4.6), the fish will turn while move backward.

$$\phi_{jr} = A_m \cdot \sin[\omega t - (2 - j)\theta] + \gamma(j). \tag{4.6}$$

4.4 Conclusion

This chapter mainly focuses on the different locomotions generation and experimental verification for an Anguilliform robotic fish.

In the beginning, we present that a robotic fish that imitates the shape of an Anguilliform fish. We give the detailed mechanical design of the robotic fish, including the dimensions, the shapes, the mass distribution of all the four links. Then, we described the control box, by using which, we can let the robotic fish receive command signals from the host computer, as well as supply PWM signals to the servo motors of the robotic fish. Next, a video recording system is set up to record the trajectory of the fish. Based on the dynamic model we derived, the relations between reference joint angles and the three locomotion patterns of the fish are investigated. We explore three most useful locomotion patterns of the Anguilliform fish: forward locomotion, backward locomotion, and turning locomotion. We find that when the former joint has a phase lead compared with the latter one, the fish moves forward; when the former joint has a phase lag, the fish moves backward; when there exist deflections on the reference angles, the fish makes a turn. The three basic locomotion patterns serve as cornerstones for more complicated motion, and they are all verified by simulations and experiments, where the results are consistent with each other.

Chapter 5

Motion Library Design and Motion Planning

5.1 Introduction

In the previous chapter, three basic locomotion patterns – forward locomotion, backward locomotion and turning locomotion – have been investigated. Serving as corner stones for more complex locomotions, the three basic locomotions are indispensable. However, in order to complete some complicated tasks, it is far from enough for the robotic fish to swim only in these basic locomotion patterns. To achieve complex tasks, the fish need to combine and organize the basic locomotion patterns organically, and when to conduct each individual locomotion is important as well. Detailedly, when encountering some specific environment, the robotic fish should have some corresponding actions. For example, when the fish is swimming in a narrow pipe with circular parts of different radii, the fish need to select different turning parameters to successfully pass the pipe. How to choose appropriate parameters in different conditions, is what we will discuss in this chapter.

The robotic fish system involves a set of parameters that can be manipulated, such as angle amplitude A_m, frequency ω, phase difference θ, and deflection angles γ. It is important to explore the explicit relations between the locomotion patterns and those parameters, i.e., explore how the parameters affect the locomotions of the robotic fish. For convenience of later use, a motion library can be built. With such a motion library, desired locomotion characters, such as desired swimming speed or desired turning radius, can be well planned by selecting appropriate values of these parameters.

Apart from the four-link robotic fish that we have developed, it is interesting to investigate how a robotic fish, with different number of links, can swim. Also, it is interesting to see the difference between the motion libraries of robotic fishes with different number of links, i.e., the discrepancy that parameters take effect. Moreover, it is important to explore the effect of body wave traveling on fish body on the speed of it, which is indicated in Lighthill's theory. Thus, in this chapter, we present an eight-link robotic fish, whose length is the same as that of the four-link fish. Since the simulation result and the experimental result consist quite well in previous chapters, we conduct only simulations to investigate the performance of the eight-link fish.

Considering complex tasks that the robotic fish need to achieve, motion planning is another important issue to investigate. Motion planning involves getting a robot to automatically determine how to move while avoiding collisions with obstacles [50]. Traditional motion planning methods are used for rigid-body mobile robot. However, the robotic fish does not have a rigid body, but have several articulated parts jointed together. Thus, we have to design a custom motion planning strategy specially for the robotic fish.

To conduct a motion planning experiment, the robotic fish needs to know its current position and surrounding information. Thus, feedback needs to be used. In lab environment, it is quite a convenient and practical way to use web camera as the feedback sensor. Based on unique markers that individually labeled on the fish and surroundings, useful information can be extracted to identify the position of the fish.

For some robots/manipulators, there exists a mapping between joint space and task space. Given a desired task space trajectory, the corresponding joint space trajectory can be derived by using inverse kinematic technique, if applicable. However, comparing with these robots, the robotic fish does not have a fixed base. Thus, the mapping relation between joint space and task space is not that simple. This complicated relation directly matters whether exact trajectory tracking in task space can be realized. In this chapter, we will also give some discussions on that.

There are many works that have been done on the subject of motion library, and motion planning, and target tracking. In [34], quantitative analysis of various forms of behavior such as swimming, crawling, burrowing, withdrawal, and turning has been conducted for the robotic fish. A look-up table is generated from these behavior for the purpose of offering appropriate data to the controller. In [27], considering the inherent kinematic constraints of the robotic fish, a new control law is proposed to stabilize the robotic fish on a specified position. Furthermore, limit-cycle approach is employed to deal with the collision avoidance problem among multiple robotic fish, resulting that the robotic fish can avoid one another smoothly and efficiently. In [28], to deal with the collision-free motion planning problem, a novel approach based on numerical flow field is proposed. Referring to the idea in computational fluid dynamics, a feasible velocity vector field is generated instead of a pre-determined path. An algorithm called tangent circle method, is developed in [29], to let the fish robots play water polo game in the water based on local vision information. In [16], the effects of added mass, quasi-steady lift, and drag are considered, then a system model is built in a control-affine structure. By using geometric nonlinear control theory, a trajectory tracking algorithm is developed for a free-swimming underwater vehicle. In [51], a maneuvering control strategy is proposed for an aquatic vehicle that uses an oscillating foil as a propulsor. The complete motion plan is obtained by concatenating time-scaled copies of the primitives. In [6], cooperative control for trajectory tracking of multiple biomimetic robotic fish is presented by using neural network based sliding mode control method. The robotic fish can receive information of itself and target point, then makes decisions autonomously to track the planned trajectory in a decentralized way. In [24], turning control in free swimming is presented for a multi-link biomimetic robotic fish. The effect of magnitude, position, and time of the deflections applied to the links are investigated. In [30], a cooperative underwater box-pushing scenario is presented, in which three autonomous robotic fish can move an elongated box from some initial location to a goal location. Considering the complexity of the underwater environment and the limited capability of a single robotic fish, the original task is decomposed into three subtasks and each subtask is assigned to the most capable robotic fish. In [31], the authors presented a vision-based autonomous robotic fish capable of 3D locomotion. A decentralized control method is investigated in target-tracking and collision-avoidance task for two autonomous robotic fish. The decentralized control, which is based on situated-behavior, is employed on each robotic fish according to its visual data. In [32], genetic algorithm and modified dynamic programming are applied to path planning for a robotic fish. By using the method, an optimal or sub-optimal path can be obtained. In [52], the proposed multi-physics model, a swimming dynamic model of a fish-like robot, and an electric model of an embedded electro-location sensor are combined together, and applied to the task of obstacle avoidance. In [53], a coordination method for two biomimetic robotic fish in transporting box-like object task is proposed. A situated-behavior design method is employed to divide the environment into a set of complete and exclusive situations, and for each situation, a specific behavior is designed.

This chapter is organized as follows. In Section 2, we first present the relations among speed, turning radius and related parameters for the four-link robotic fish. In Section 3 and Section 4, we investigate the motion of the eight-link robotic fish, and the relations same as that of the four-link fish. Based on the relations explored, we build a motion library, from which the robotic fish can choose suitable parameters according to various scenarios. In Section 5, we give elaborated tasks to show the application of the motion library on motion planning for the robotic fish. Three tasks –pipe task, tunnel task, and irregular-shape pipe task are assigned to the two robotic fishes, and corresponding control strategy and simulation results are given. In Section 6, a motion planning experiment which contains visual feedback of camera is presented. In the experiment, the fish can follow a “U” shape desired trajectory and conduct corresponding locomotions at specific points. In Section 7, some discussions on trajectory planning is give, and the reason why we control the fish through joint space is presented. In Section 8, a brief conclusion is presented.

Compared with other works, the major superiority of our work is: A motion library, that contains the relations between speed, turning radius of the fish and parameters of undulation frequency, amplitude, phase difference, deflections, is constructed. Although some works [24] [34] cover part of the contents, to the best of our knowledge, the motion library presented in this chapter contains the most detailed and the most elaborated relations in existing works.

5.2 Relations among Speed, Turning Radius and Related Parameters (Four-Link Fish)

The robotic fish system involves a set of parameters that can be manipulated, such as angle amplitude A_m, frequency ω, phase difference θ, and deflection angles γ. We need to explore the explicit relations between the locomotion patterns and those parameters, namely build a motion library. With such a motion library, we can plan desired motions, such as desired swimming speed or desired turning radius, by selecting appropriate values of these parameters.

Based on the results from the previous section, the model-based simulation results are fairly consistent with experimental results. Thus, in this section, we will investigate the above mentioned relations through only simulations, then build the motion library.

5.2.1 Relations among steady speed v_s and the parameters ω, A_m, θ (four-link fish)

In this part, the relations among steady speed v_s of the fish and the parameters ω, A_m and θ, are investigated. Here straight-line moving cases are mainly focused on, and forward moving case is used as an example.

First, the relationship between the steady speed of the fish and the angular frequency ω is investigated. Here parameters are chosen as $A_m = 0.45$ (rad), $\theta = 1.5$ (rad), and then run simulations with parameter ω that varies from low to high. The obtained different steady speed v_s under different ω, are as shown in Fig. 5.1.

From Fig. 5.1, we see that the speed v_s increases linearly as the angular frequency ω increases. The reason is that, the more frequently the fish interacts with water, the more hydrodynamic force that the fish will experience during the same period of time, thus

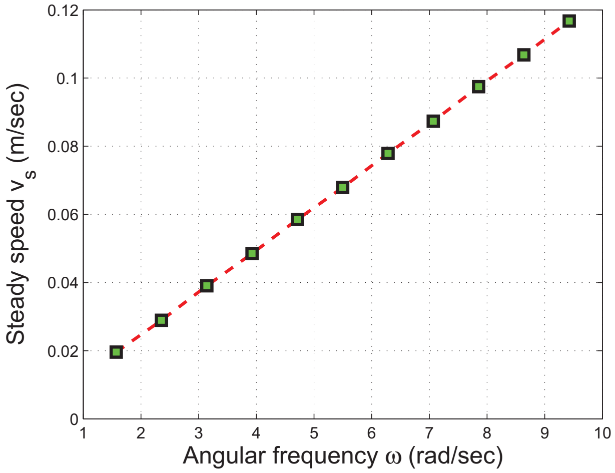


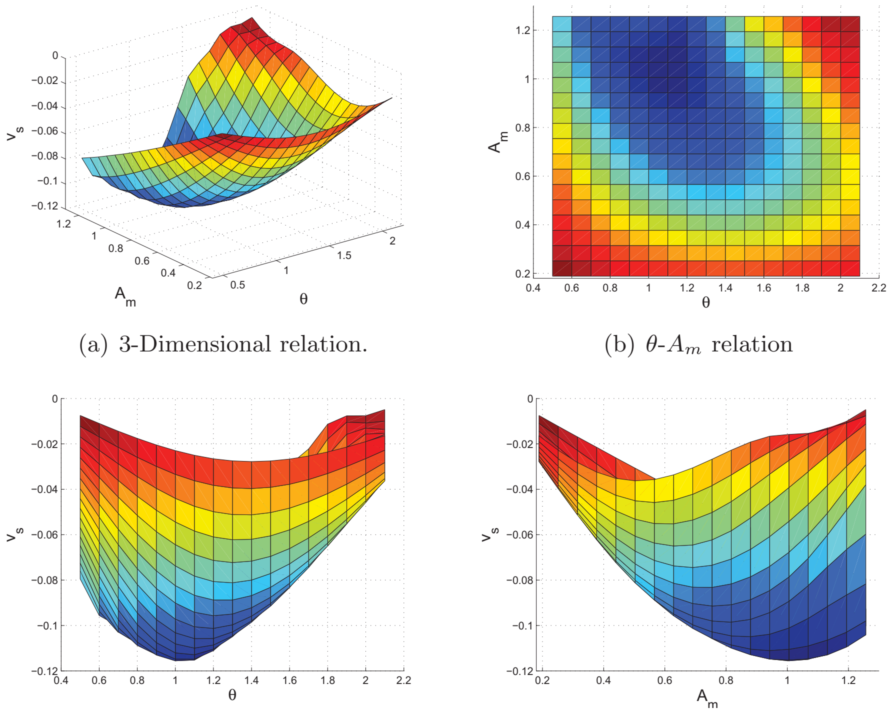
Figure 5.1: Steady speed v_s under different angular frequency ω.

the faster that the fish can move. However, there exists an angular frequency limitation for the servo motors. With other parameters being the same, the maximum speed is limited by the maximum angular frequency.

The amplitude A_m and the phase difference θ have a more complicated effect on speed. Next, we investigate their relations.

Set $\omega = 2\pi$ (rad/sec). θ is chosen from 0.5 to 2.1 (rad). A_m is chosen from 0.06π to 0.40π (rad). The simulation time is 40 seconds. The 3-D relation is shown in Fig. 5.2(a). Fig. 5.2(b) is the top view of the 3-D relation, Fig. 5.2(c) and Fig. 5.2(d) are the side views of the 3-D relation. It can be seen that the shape of the surface is approximately parabolic, and there is a minimum point in the valley. From Fig. 5.2(b), the coordinates of the minimum point can be obtained: $\theta = 1.0$, $A_m = 0.32\pi$, and v_s=-0.116 (m/sec). Here it should be noted that, because the direction that the fish moves forward is selected as negative x-axis, this "minimum point" actually represents the "maximum speed" of the fish.

From Fig. 5.2(c), we know that for a specific value of A_m, there exists a value of θ such that v_s has a minimum value, and similarly, for a specific value of θ, there exists



(a) 3-Dimensional relation.

(b) θ-A_m relation

(c) θ-v_s relation.

(d) A_m-v_s relation

Figure 5.2: Relations among v_s and the parameters A_m, θ.

a value of A_m such that v_s has a minimum value. This is reasonable because: If A_m is less than the specific value, the interaction between the fish and water is not enough to make the fish get its maximum speed; While if A_m is more than the specific value, the undulation amplitude of the body wave on the fish is too large such that this wave introduces more drag than thrust. Similarly, θ also has a most appropriate value such that the speed of the fish is maximized.

From the simulation, it can be observed that the speed of the fish relies on the composition of the parameters A_m and θ, rather than relies on each of them independently. That is, the effects of A_m and θ on the speed v_s are coupled. This phenomenon is also observed in our experiments. One of our observations in the experiments is that, for the same value of A_m, when apply different θ, v_s could be either quite high, or as low as 0 (the fish stays in the original place and can hardly move). Another observation is that, for the same value of speed v_s, there may exist multiple pairs of parameters A_m and θ that yield the same speed.

5.2.2 Relationship between turning radius and the parameter γ (four-link fish)

In this part, the relationship between turning radius and related parameters is mainly focused on. Although parameters ω, A_m and θ have effects in some way, the essential factor that results in turning movement of the fish is the deflection angle γ. Therefore, we focus on the effect of deflection γ on the turning performance of the robotic fish.

The fish is at rest at time $t = 0$, and the parameters are selected as $A_m = 0.45$ (rad), $\omega = 2\pi$ (rad/sec), $\theta = 1.5$ (rad), and the deflection angles γ take the following values

$$\gamma = \gamma_{\max} \cdot [1, \frac{2}{3}, \frac{1}{3}, 0], \tag{5.1}$$

where $\gamma_{\max}$ is the maximum deflection angle. Here eight different values are chosen, 0.05π, 0.10π, 0.15π, 0.20π, 0.25π, 0.30π, 0.35π and 0.40π, for $\gamma_{\max}$, respectively. For each particular $\gamma_{\max}$, the turning radius can be oatained in the same way. All the obtained data are summarized in Fig. 5.3.

From the above cases, a conclusion can be drawn that if the deflection is larger on each link, the turning radius becomes smaller. The reason for this relation is quite straight forward: if the deflection is larger, the degree to which the fish turns is larger, which leads to a smaller turning radius.

However, we cannot obtain as small turning radius as we want, because there exists a lower bound of the turning radius. Another observation is that, when the maximum deflection angle $\gamma_{\max}$ is less than or equal to 0.15π, the change of the turning radius with the parameter γ is quite prominent. While $\gamma_{\max}$ is larger than 0.15π, the change

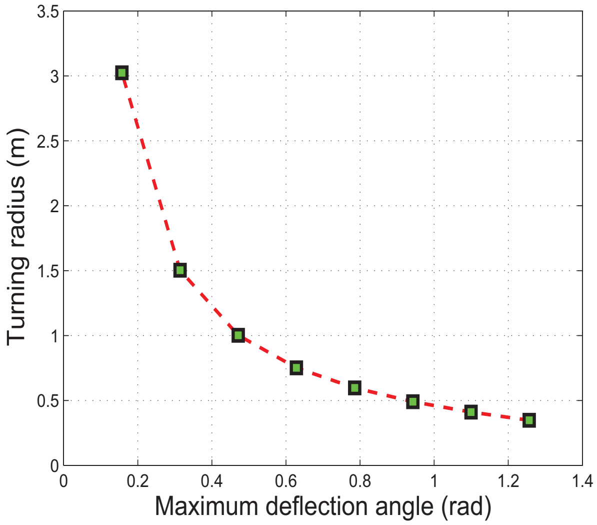


Figure 5.3: Turning radius under different maximum deflection angle γ_{max}.

becomes less significant.

Till now, we have investigated the relations among speed, turning radius and related parameters, such as oscillation frequency ω, amplitude A_m, phase difference θ, deflection angle γ. A motion library is then defined as combination of all the above information, which contains the relations that have been explored.

Remark 5.1. The significance of motion library is that it can be applied to path planning of the robotic fish. Given a task and environmental information, different reference joint angles can be sent to the robotic fish, to let it move forward, move backward or make a turn, at appropriate time. Further, based on the elaborated relations contained in the motion library, appropriate parameters can be selected to regulate the speed and turning radius of the fish. Hence, the fish is able to achieve desired motion.

5.3 Investigation of Motion of an Eight-Link Anguilliform Robotic Fish

In previous chapters, we have already done some work on the four-link Anguilliform robotic fish. In this section, we will investigate on the motion of the fish with more links, to see the performance of robotic fish with more links. Now, we select $N = 8$, i.e., we consider a eight-link robotic fish. Table 5.1 shows mechanical parameters of the links, where l_i, m_i, I_i $(j = 1, 2, \cdots, 8)$ are the length, mass and moment of inertia of link i respectively, $f_{i\perp}$ and $f_{i\|}$ are water resistance coefficients. Their SI units are $m(meter)$, kg, $kg \cdot m^2$, Ns^2/m^2, Ns^2/m^2.

Table 5.1: Mechanical parameters of the links.

Link #	l_i	m_i	I_i	$f_{i\perp}$	$f_{i\|}$
1.	0.11	0.1565	1.578×10^{-4}	1.875	0.555
2.	0.11	0.1565	1.578×10^{-4}	1.875	0.555
3.	0.06	0.0855	2.565×10^{-5}	1.025	0.305
4.	0.06	0.0855	2.565×10^{-5}	1.025	0.305
5.	0.06	0.0855	2.565×10^{-5}	1.025	0.305
6.	0.06	0.0855	2.565×10^{-5}	1.025	0.305
7.	0.10	0.1425	1.1875×10^{-4}	1.705	0.505
8.	0.10	0.1425	1.1875×10^{-4}	1.705	0.505

Note that in Table 5.1, the total length of the eight-link fish is the same as the length of previous four-link fish. Such a selection ensures that their performances are comparable under the same standard. Actually, the length of each two neighboring links of the eight-link fish is selected as half length of the corresponding link of the four-link fish. Next, some simulation results are given by using the mathematical model of the fish derived in Chapter 2.

We use forward locomotion as an example to see the performance of the eight-link fish. Based on (4.2), we give the reference angles ϕ_{jr}. The parameters are chosen as follows: $\theta = 0.75$, which is half as that of the four-link case. Other parameters are chosen as $A_m = 0.45$, $\omega = 2\pi$, and the feedback coefficients $k_1 = 10$, $k_2 = 1$, which are the same as those of the four-link case. At time $t = 0$, the fish is still, and its eight links are aligned on x-axis with its head on the origin. By applying computed torque control method in (4.1), we get the trajectories of actual and reference angles ϕ_j and ϕ_{jr}, torques, x-trajectory, as shown in Fig. 5.4 $\sim$ Fig. 5.6.

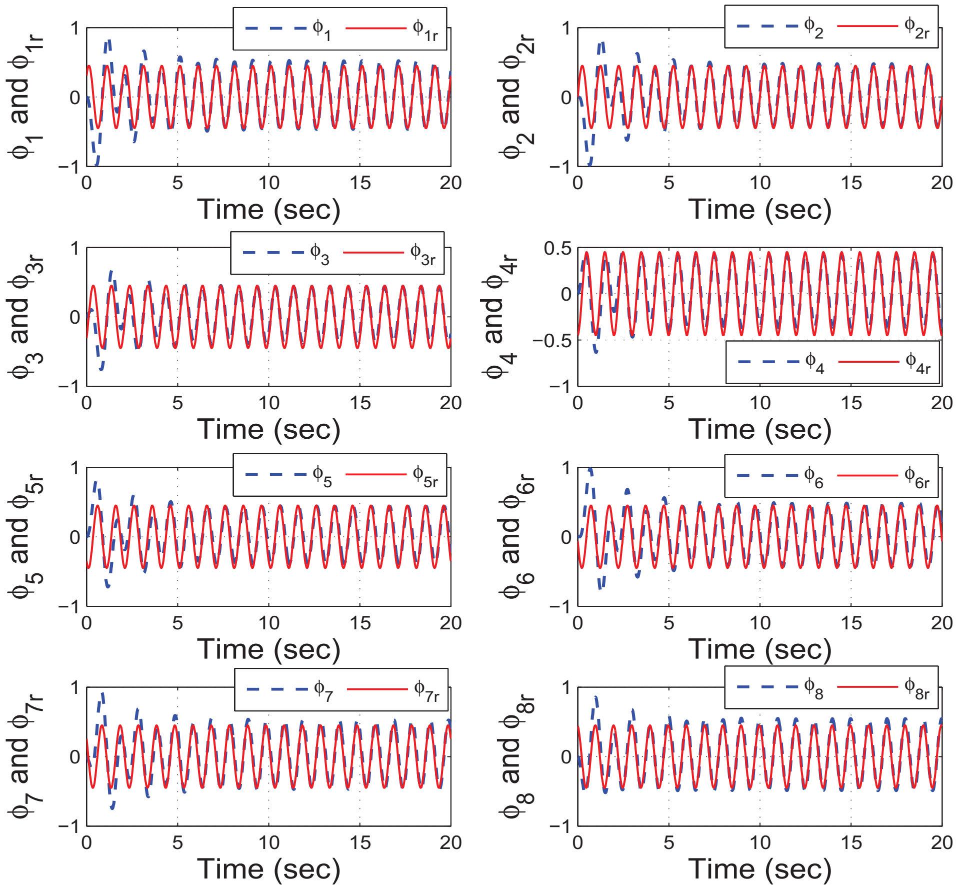


Figure 5.4: Actual angle ϕ_1 and reference angle ϕ_{1r} trajectory, with parameters $A_m = 0.45, \omega = 2\pi, \theta = 0.75$.

From the simulation results, we see that the reference joint angles can be roughly tracked, and the speed of the eight-link fish is faster than that of the four-link fish under

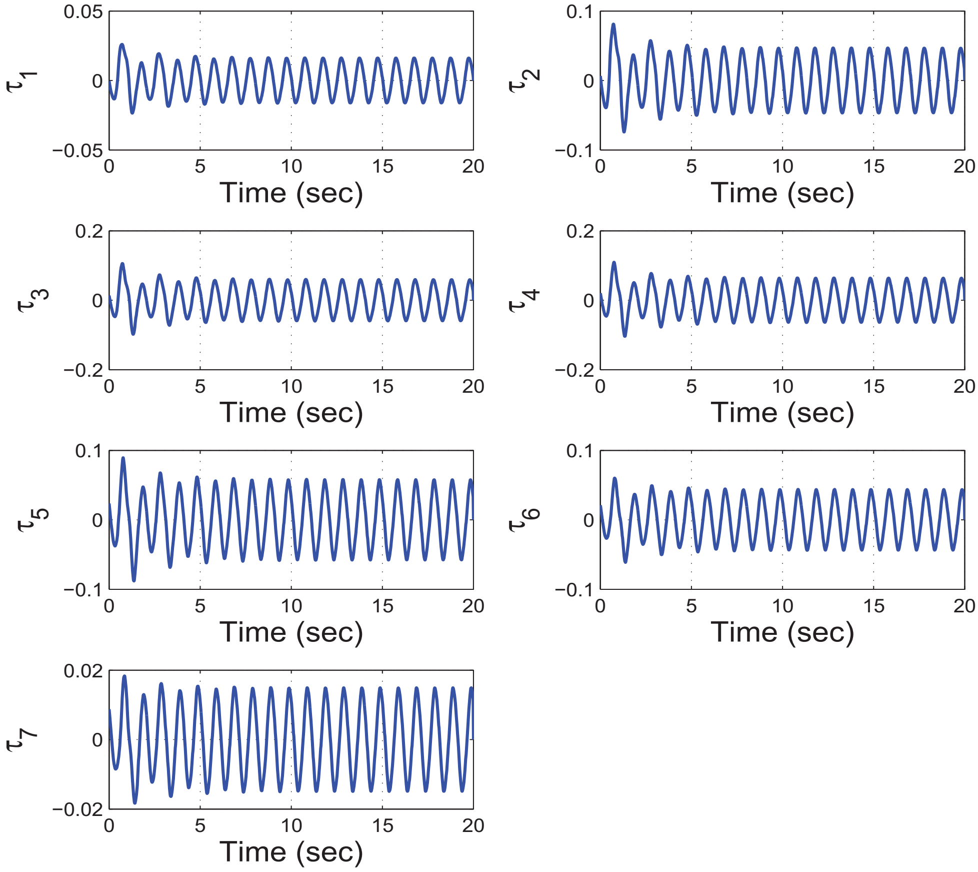


Figure 5.5: Torques trajectory, with parameters $A_m = 0.45, \omega = 2\pi, \theta = 0.75$.

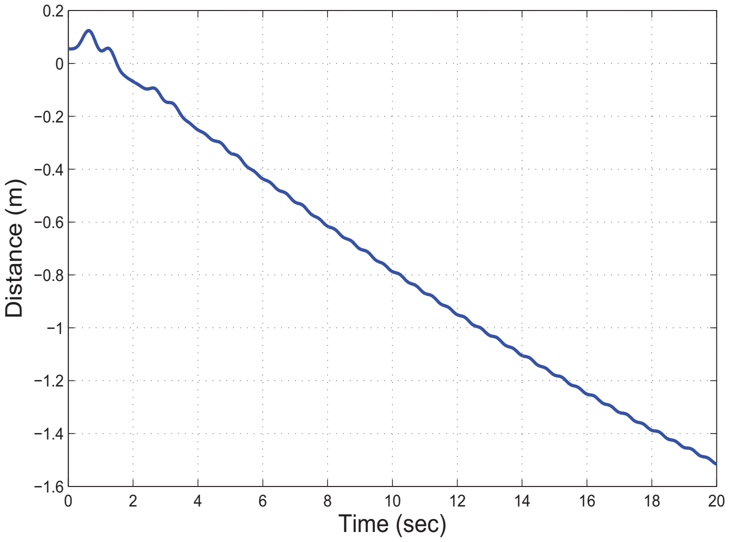


Figure 5.6: Distance (x_1) trajectory, with parameters $A_m = 0.45, \omega = 2\pi, \theta = 0.75$.

similar parameters.

Next, we will investigate whether smoother motion is generated by using more number of links. According to [13], the traveling body wave on a fish body needs to be a sinusoidal curve. In this way, the body wave on the fish can be transferred better. In this part, we use the curve fitting toolbox of Matlab to check whether the curve can be obtained. For the eight-link robotic fish, Fig. 5.7 shows that at an instant, the positions of all the eight links of the robotic fish. Fig. 5.8 shows the sinusoidal curve fitting result. While by using four links, Fig. 5.9 shows at an instant, the positions of all the four links of the robotic fish. Fig. 5.10 shows the sinusoidal curve fitting result. Note that all the original data is fitted by a single sinusoidal curve, which is based on the fact that there exists a body wave traveling along the fish body. Comparing Fig. 5.8 and Fig. 5.10, it is obvious that in the case of eight links, the fish body can be much better fitted by the sinusoidal curve. Thus, the body wave can be transferred more smoothly, which illustrates why the eight-link fish's speed is faster.

We see that under the situation that undulation amplitude and angular frequency are the same, the fish can achieve faster speed with more number of links. This is because with more links, the motion of the fish body is more approaching to a traveling wave, and the motion is much smoother, which will reduce the friction and get higher swimming efficiency, thus produce faster speed.

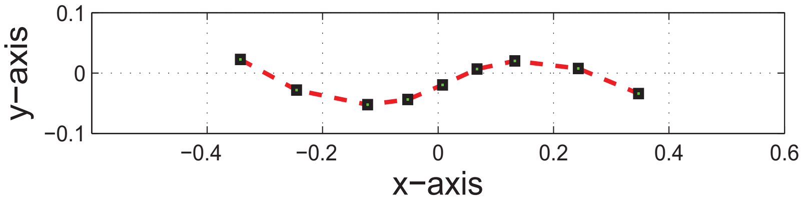


Figure 5.7: Link distribution at an instant (eight link).

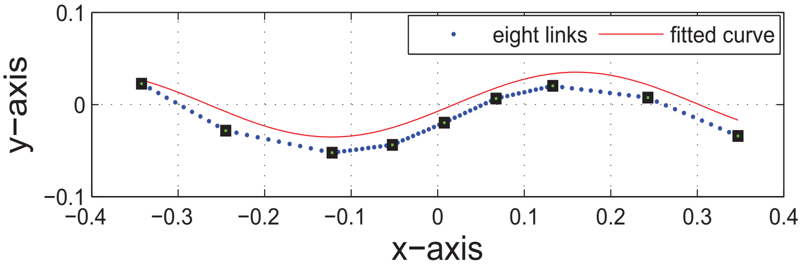


Figure 5.8: Curve fitting of all the links (eight link).

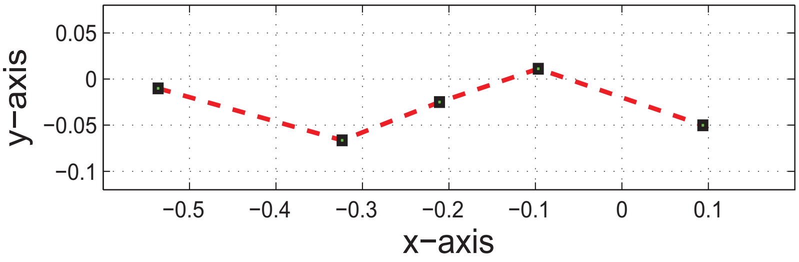


Figure 5.9: Link distribution at an instant (four link).

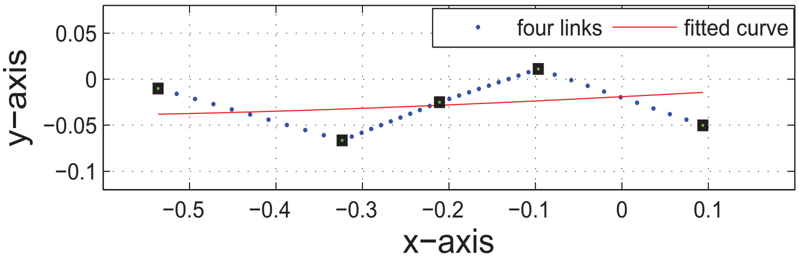


Figure 5.10: Curve fitting of all the links (four link).

5.4 Relations among Speed, Turning Radius and Related Parameters (Eight-Link Fish)

5.4.1 Relations among steady speed v_s and the parameters ω, A_m, θ (eight-link fish)

In this part, we will investigate the relation between the steady speed v_s and the angular frequency ω of robotic fish with more number of links. Here we use the eight-link fish, and choose other parameters as $A_m = 0.45$, $\theta = 0.75$. We obtain different v_s under different ω, as shown in Fig. 5.11.

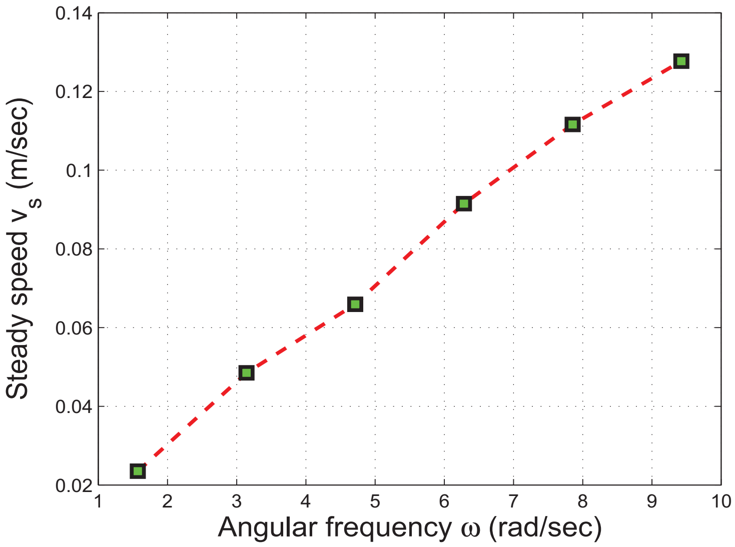
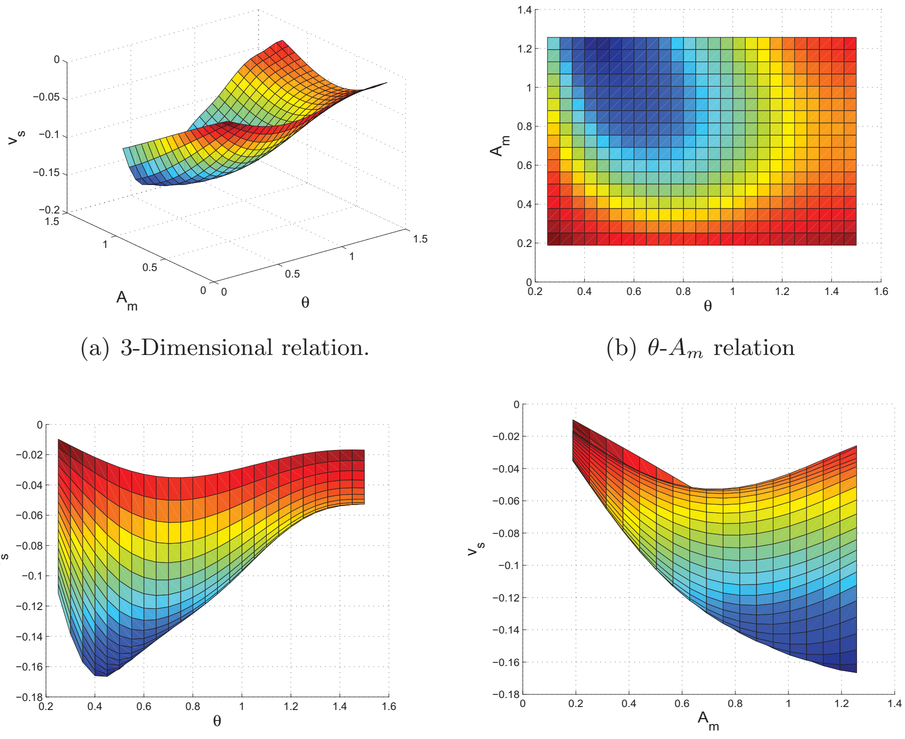


Figure 5.11: Relation between the steady speed v_s and angular frequency ω.

From Fig. 5.11, we see that the speed v_s increases approximately linearly as the angular frequency ω increases. Since the more frequently the fish interacts with water, the more hydrodynamic force that fish will experience during the same period of time, thus the faster that fish can move. However, there exists an angular frequency limitation for the servo motors. With other parameters being the same, the maximum speed is limited by the maximum angular frequency.

The relation between the speed and the oscillation frequency is quite simple and straight forward, while the amplitude A_m and the phase difference θ have a more complicated effect on speed. Next, we investigate their relations.

Here we fix the value $\omega = 2\pi$ (rad/sec). θ is chosen from 0.25 to 1.5 (rad). A_m is chosen from 0.06π to 0.40π (rad). The simulation time is 40 seconds. The 3-D relation is shown in Fig. 5.12(a). Fig. 5.12(b) is top view of the 3-D relation, Fig. 5.12(c) and Fig. 5.12(d) are side views of the 3-D relation. It can be seen that the shape of the surface is approximately parabolic, and there is a minimum point in the valley. From Fig. 5.12(b), the coordinates of the minimum point can be obtained: $\theta = 0.45$, $A_m = 0.4\pi$, and v_s=-0.167 (m/sec). Here it should be noted that, because the direction that the fish moves forward is selected as negative x-axis, this "minimum point" actually represents the "maximum speed" of the fish.



(a) 3-Dimensional relation.

(b) θ-A_m relation

(c) θ-v_s relation.

(d) A_m-v_s relation

Figure 5.12: Relations among v_s and the parameters A_m, θ.

From Fig. 5.12(c), we know that for a specific value of A_m, there exists a value of θ such that v_s has a minimum value, and similarly, for a specific value of θ, there exists a value of A_m such that v_s has a minimum value. This is reasonable because: If A_m is less than the specific value, the interaction between the fish and water is not enough to make the fish get its maximum speed; While if A_m is more than the specific value, the undulation amplitude of the body wave on the fish is too large such that this wave introduces more drag than thrust. Similarly, θ also has a most appropriate value such that the speed of the fish is maximized.

From the simulation, it can be observed that the speed of the fish relies on the composition of the parameters A_m and θ, rather than relies on each of them independently. That is, the effects of A_m and θ on the speed v_s are coupled. This phenomenon is also observed in our experiments. One of our observations in the experiments is that, for the same value of A_m, when apply different θ, v_s could be either quite high, or as low as 0 (the fish stays in the original place and can hardly move). Another observation is that, for the same value of speed v_s, there may exist multiple pairs of parameters A_m and θ that yield the same speed.

Compared the two robotic fishes, we have found an interesting phenomenon: for the eight-link fish, when achieving the fastest speed, the phase difference between the last link and the first link is $3.15(= 0.45 \times 7)$ rad, where 0.45 rad is the phase difference between two neighboring links; while for the four-link fish, when achieving the fastest speed, the phase difference between the last link and the first link is $3(= 1 \times 3)$ rad, where 1 rad is the phase difference between two neighboring links. It can be noted that, when the two fishes achieve their fastest speed, the tail of the fish will have a phase lag of π approximately. That means, if the fish wants to maximize its speed, it must let its last link have a phase lag of π compared with its first link.

The fastest speed of the four-link robotic fish is 0.116 m/sec, while the fastest speed of the eight-link robotic fish is 0.167 m/sec. We see that the eight-link fish is about 44% percent faster than the four-link fish. The reason is that body wave on the eight-link robotic fish can be transferred more smoothly, thus the drag force can be more reduced (refer to Fig. 5.8 and Fig. 5.10).

5.4.2 Relation between turning radius and the parameter γ (eight-link fish)

In previous section, we have investigated the effect of the deflection angles γ on the turning radius of the four-link robotic fish. While in this part, we will further verify this relation by using an eight-link fish.

The fish is at rest at time $t = 0$, and the parameters are selected as $A_m = 0.45$ (rad), $\omega = 2\pi$ (rad/sec), $\theta = 0.75$ (rad), and the deflection angles γ take the following values

$$\gamma = \gamma_{\max} \cdot [1, \frac{6}{7}, \frac{5}{7}, \frac{4}{7}, \frac{3}{7}, \frac{2}{7}, \frac{1}{7}, 0], \tag{5.2}$$

where $\gamma_{\max}$ is the maximum deflection angle. Here eight different values are chosen, 0.05π, 0.10π, 0.15π, 0.20π, 0.25π, 0.30π, 0.35π and 0.40π, for $\gamma_{\max}$, respectively. For each particular $\gamma_{\max}$, the turning radius can be obtained in the same way.

Fig. 5.13 shows the turning radius under different deflection angles. From the figure, we see that the turning radius decreases as the deflection angles increase, which further verifies the conclusion we obtained for the four-link robotic fish.

From the simulation result, a conclusion can be drawn that, if the deflection is larger on each link, the turning radius becomes smaller. The relation of deflection angles and turning radius is quite straight forward, because if the deflection is larger, the degree to

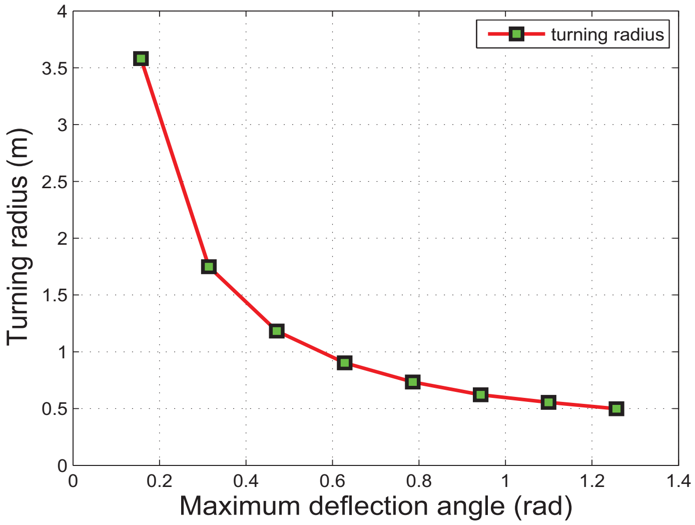


Figure 5.13: Turning radius under different deflection angle γ (eight link).

which the fish turns is larger, which leads to a smaller turning radius.

5.5 Application of Motion Library on Motion Planning for Robotic Fishes

One of the remarkable characters that Anguilliform fish possesses is its maneuverability. With highly flexible body, Anguilliform fish can achieve complex motion that is quite hard for other types of fishes. To show the performance that can be achieved by Anguilliform fish, here we give some examples.

The main idea of our motion planning method is that: Any complex shape of path is composed of two basic shapes – straight lines and circular curves with different radii. Given appropriate reference angles as discussed in previous chapter, the fish can swim straightly or circularly according to the given reference angles. By changing one of the signs in reference angles, we can change the direction that the fish swims – forward or backward. Furthermore, we have built a motion library, which contains the relation between the deflection and the turning radius, thus we can select corresponding deflection parameter according to the desired turning radius.

5.5.1 Pipe task (four-link fish)

In this part, we will let the four-link fish conduct a "pipe task". Fig. 5.14 shows a pipe with complex shape, which contains straight parts and circular parts. The given task is that, the robotic fish is required to get into the entrance and get out of the exit, without touching any part of the pipe in the whole process. The task frequently arises in scenarios such as pipe detection, irregular-shape environment exploration.

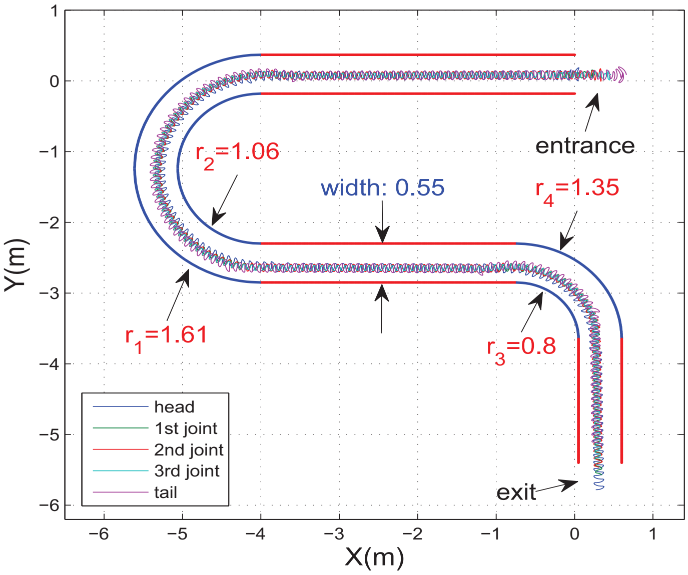


Figure 5.14: Trajectory of the fish passing through the pipe.

Based on the motion library developed above, we now give the control strategy of collision-free motion planning in a constrained space for the Anguilliform robotic fish. We assume that all the feedback information is available. Note that in the whole process of motion planning, the dynamics of the fish, as shown in (2.13), is involved. Fig. 5.15 shows flowchart of our motion planning method, basically following the procedure mentioned above.

In Fig. 5.14, the five trajectories with different colors represent the $x-y$ trajectories of

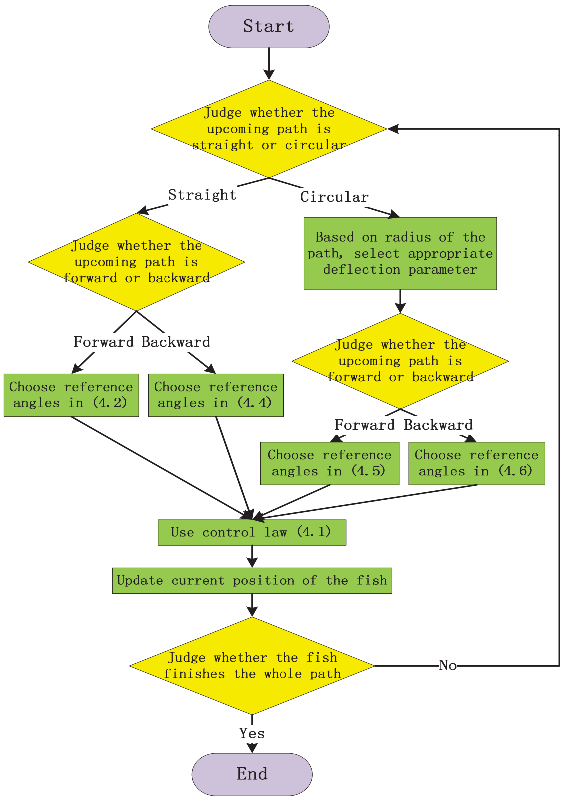

Start
Judge whether the upcoming path is straight or circular
Straight
Circular
Judge whether the upcoming path is forward or backward
Based on radius of the path, select appropriate deflection parameter
Forward Backward
Judge whether the upcoming path is forward or backward
Choose reference angles in (4.2)
Choose reference angles in (4.4)
Forward Backward
Choose reference angles in (4.5)
Choose reference angles in (4.6)
Use control law (4.1)
Update current position of the fish
Judge whether the fish finishes the whole path
No
Yes
End

Figure 5.15: Flowchart of the motion planning method.

head, 1st joint, 2nd joint, 3rd joint and tail, respectively. It can be seen that by applying the motion planning method developed above, the fish successfully passed through the pipe without touching any part of it. In the above task, the motion planning method serves as a decision making process. That is, based on currently updated situations, such as the position of the fish and the type of upcoming path, the method will choose different reference angles ϕ_{jr} from (4.2), (4.4), (4.5) or (4.6) correspondingly. As a result, under the control law (4.1) that contains the reference angles ϕ_{jr}, the robotic fish will perform by following the reference angles. The straight-line locomotion is easily realized, and a set of parameters is chosen as $A_m = 0.45$, $\omega = 2\pi$, $\theta = 1.5$ (appropriate sets of parameters are not limited to this set, other sets may also work). It can be noted that the robotic fish needs to conduct turning locomotion at two different places. At the first place, the radius of the central line of the pipe is 1.335 meters, and the fish needs to turn counter-clockwise. At the second place, the turning radius is 1.075 meters, and the fish needs to turn clockwise. Thus, from the relation between turning radius and maximum deflection angle $\gamma_{\max}$ in Fig. 5.3, appropriate deflection angle can be chosen individually. Note that when the fish makes a clockwise turn, a negative sign needs to be added on the deflection angle. The motion planning method concatenates different locomotion patterns together, so that the fish can swim continuously with the transition of different reference angles, and avoid colliding with the pipe in the task.

5.5.2 Tunnel task (eight-link fish)

In this part, we will let the eight-link fish conduct a "tunnel task". Fig. 5.16 shows a tunnel, which consists of straight parts and circular parts, and the tunnel has only one opening. The given task is that, the robotic fish is required to get into the tunnel, reach the end of the tunnel, and get out of it, without touching any part of it in the whole process. Since the tunnel has only one opening, the entrance is also the exit. From the figure, we see that, the width of the tunnel (0.3 m) is much smaller than the minimal turning diameter of the fish. Thus, it is impossible for the fish to turn itself in such a narrow space. Also, the fish can not get out of the tunnel by using only forward locomotion while it is turning. It is necessary to use both forward locomotion and backward locomotion while turning.

By using the motion library we have developed, the control strategy is given for the robotic fish, which is similar to the previous one. We assume that all the feedback information is available. Note that in the whole process of motion planning, the dynamics of the fish is involved. Aiming at this specific task, forward locomotion, backward locomotion, and turning locomotion are adopted. The whole procedures are that: get into the tunnel, move forward, turn while advancing, move forward, move backward, turn while

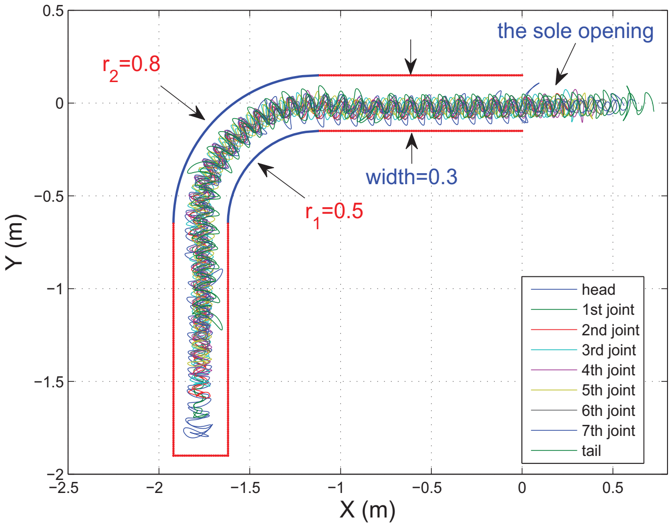


Figure 5.16: Trajectory of the fish inside the tunnel.

backing, move backward, get out of the tunnel. To avoid collision with the tunnel, the robotic fish must stop to move forward before reaching the end of the tunnel. Thus, here we define a safety distance, and its value is determined by experience. Once the distance between the fish head and the end of the tunnel reaches the safety distance, it will stop moving forward, but will move backward.

Applying the above control strategy to the eight-link robotic fish on the tunnel task, we get trajectory of the robotic fish as shown in Fig. 5.16. In the figure, the nine trajectories represent the trajectories of the head, tail and all the joints of the robotic fish. We see that the fish successfully arrived at the end of the tunnel and swam out without touching any part of it.

5.5.3 Irregular-shape pipe task (four-link fish)

In previous two motion planning examples of application of motion library, the fish either turns a 90° angle or a 180° angle. In this part, we will let the fish go through a pipe with irregular shape. Fig. 5.17 shows an irregular-shape pipe, which contains straight parts and circular parts. The given task is that, the robotic fish is required to get into the entrance and get out of the exit, without touching any part of the pipe in the whole process. Considering the size of the robotic fish, it needs to complete the task in a quite constrained space.

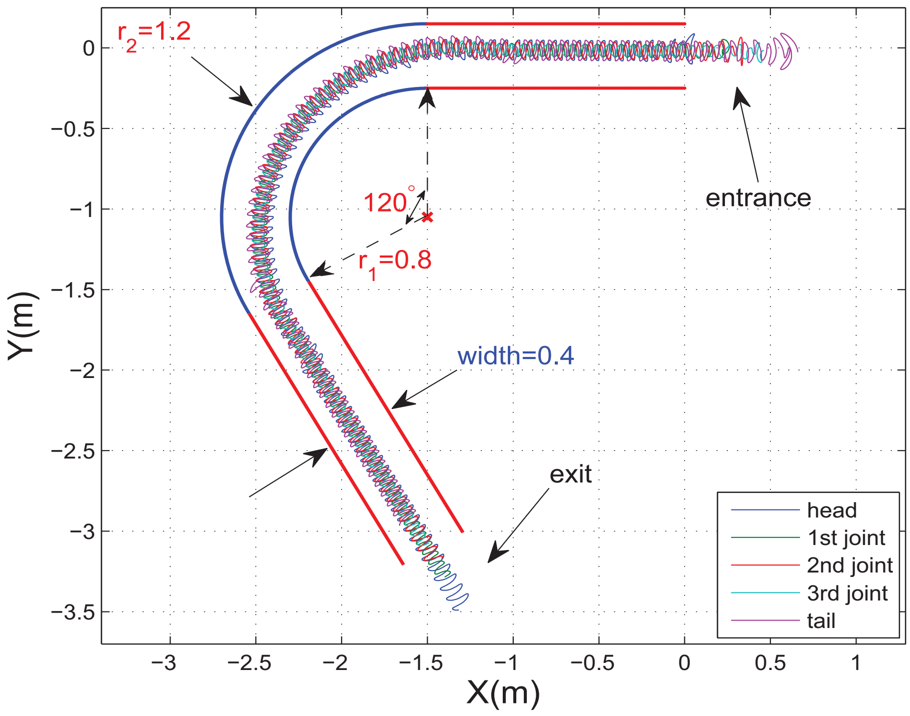


Figure 5.17: Trajectory of the fish inside the irregular-shape pipe.

By using the motion library that we construct, the control strategy given to the robotic fish is similar to the previous one. We assume that all the feedback information is available. Note that in the whole process of motion planning, the dynamics of the fish is involved. Aiming at this specific task, only forward locomotion and turning locomotion are adopted. The whole procedures are that: get into the pipe, move forward, turn while advancing, move forward, get out of the tunnel.

Applying the above control strategy to the four-link robotic fish on the irregular-shape pipe task, we get trajectory of the robotic fish as shown in Fig. 5.17. In the figure, the five trajectories represent the trajectories of the head, tail and all the joints of the robotic fish. From the figure, We see that the fish successfully passed through the pipe without touching any part of it.

5.6 Experiment of Motion Planning

Different from previous experiments, in this part, feedback from external sensor (an overhead camera) will be added to the robotic fish. Specifically, the fish will follow a prescribed trajectory according to the sensory information. Based on the different locomotion patterns of the fish that we have explored, and with the help of feedback technique, we can let the robotic fish achieve more complicated task.

5.6.1 Task description

Fig. 5.18 shows the experiment that the robotic fish will conduct. The green rectangle represents the swimming pool, and the small red rectangles are markers which can position the desired trajectory. The fish will start at the position of the "Start Point", and swims all the way along the red rectangular markers. The fish will pass by the "Change Point 1" and "Change Point 2", where locomotion of the fish needs to be changed. Finally, the fish will stop at the "End Point". In the whole process, the trajectory of the fish will form a "U" shape.

5.6.2 Control strategy

The "U" shape of the desired trajectory is shown in Fig. 5.19, where the solid line represents the desired trajectory of the robotic fish, and the two dashed lines represent the inner border and outer border of the desired trajectory, respectively. The desired trajectory is actually a curve that connects all the red markers together. The space

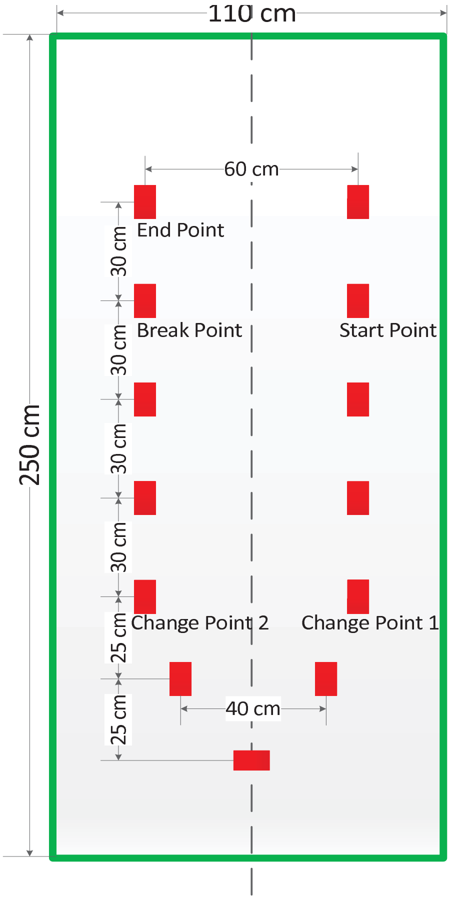


Figure 5.18: Sketch of the motion planning experiment.

between the inner border and the outer border serves as an "acceptable" area defined by us. Specifically speaking, if the fish swims inside this area, we say that the fish technically follows the desired trajectory. Otherwise, if the fish swims outside this area, we say that the fish fails to follow the desired trajectory. In such a case, the robotic fish needs to change locomotion correspondingly in order to swim back to the "acceptable" area.

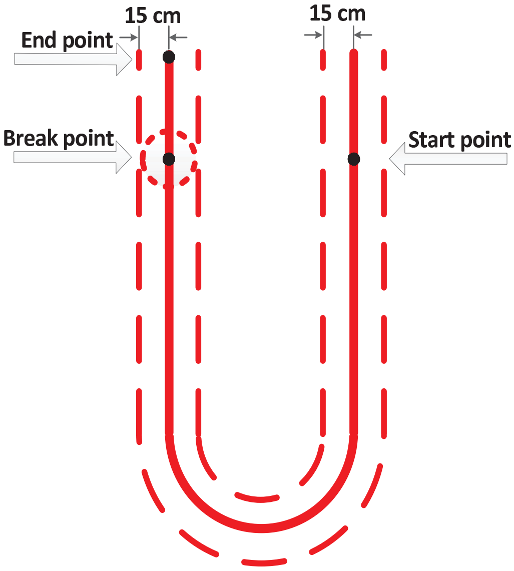


Figure 5.19: Borders of the U shape.

Setting this border area is necessary and reasonable. If we let the fish just follow the desired trajectory exactly, severe chattering phenomenon will happen. That is because disturbances and uncertainties always take place in the environment of the swimming pool, if we need the fish to track the exact trajectory, whenever the fish deviates, it has to change locomotion. Such frequent locomotion change inevitably result in chattering phenomenon. Another reason that the fish can not track the exact trajectory is that, there always exists yaw motion when the fish is moving, and this motion makes the trajectory of the fish zigzag, thus makes it impossible to track a regular straight line or a circle. To deal with the chattering phenomenon, we borrow the idea in sliding mode control. We use the border area as "threshold", by using which, better result can be obtained. In our case as shown in Fig. 5.19, the width between the desired trajectory and the inner border, and the width between the desired trajectory and the outer border, are both set as 15 cm.

At ahead of the end point, we set a "break point", where the fish stops oscillation. After this break point, the fish will drift all the way to the end point. The distance between the break point and the end point is empirically set as 30 cm. We have to note that, after the fish passes by the break point, it stops oscillation, thus the fish can not be controlled. Around the break point, we set a circle area, the radius of which is also 15cm. This circle area works similarly to the border area. As long as the fish enters this area, not necessarily reach the break point, it will stop its movement.

The flow chart in Fig. 5.20 shows the control strategy of motion planning for the robotic fish. First, the fish will check its current position, if it is in between start point and change point1 or in between change point 2 and the end point, the fish will go straight; if it is in between change point 1 and change point 2, the fish will go circularly, and the deflection angle $def_{nominal}$ can be selected based on radius of the path; if it reaches the break point circle, the fish will stop movement and drifts to the end point. Before the fish enters the break point circle, no matter it is going straightly or circularly, the fish will check whether it collides with border. If there is no collision with border, the fish will move in default mode; if there is collision with border, the fish will conduct corresponding reaction as shown in the flow chart.

From the flow chart, it can be seen that the external camera has two functions. The first function is to identify current position of the fish, the step immediately after "Start". The second function is to check if the fish collide with border. These two functions are essential feedback procedures, by using which, the robotic fish is able to know where it is and to conduct corresponding reaction when necessary.

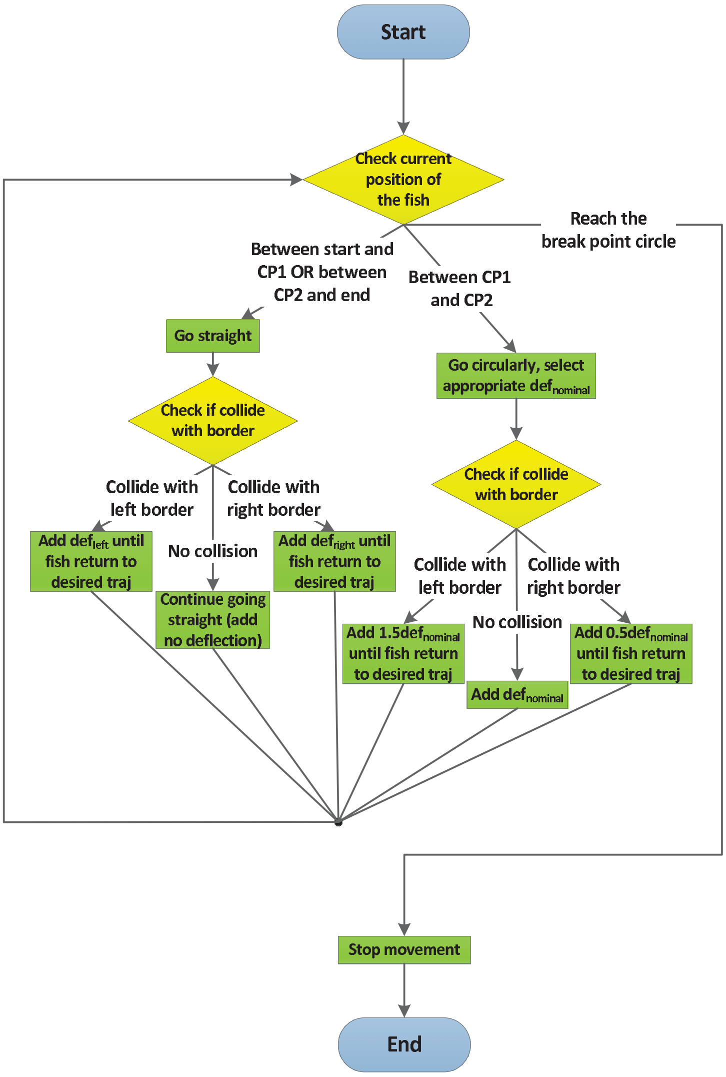

Start
Check current position of the fish
Reach the break point circle
Between start and CP1 OR between CP2 and end
Between CP1 and CP2
Go straight
Go circularly, select appropriate $def_{nominal}$
Check if collide with border
Collide with left border
Collide with right border
Add def_{left} until fish return to desired traj
No collision
Add def_{right} until fish return to desired traj
Continue going straight (add no deflection)
Check if collide with border
Collide with left border
Collide with right border
No collision
Add $1.5def_{nominal}$ until fish return to desired traj
Add $def_{nominal}$
Add $0.5def_{nominal}$ until fish return to desired traj
Stop movement
End

Figure 5.20: Flow chart of the motion planning.

5.6.3 Vision processing

As an external sensor, the camera's function is to determine the location of the fish and individual markers. Only if know where the fish and the markers are, can we use corresponding control strategy to deal with different situations.

To identify a specific object that we are interested in, an unique character of the specific object needs to be used, either its shape, its color or any other appropriate character. Note that the unique character of one object should be easy to recognize from the character of another object. In this work, color information are used to recognize objects. Note that these colors must be quite distinct compared with surrounding environment. Also, these colors can not be mixed together. In order to localize the markers and the robotic fish, red color is used to denote the markers, and yellow color is used to denote the fish in our experiment.

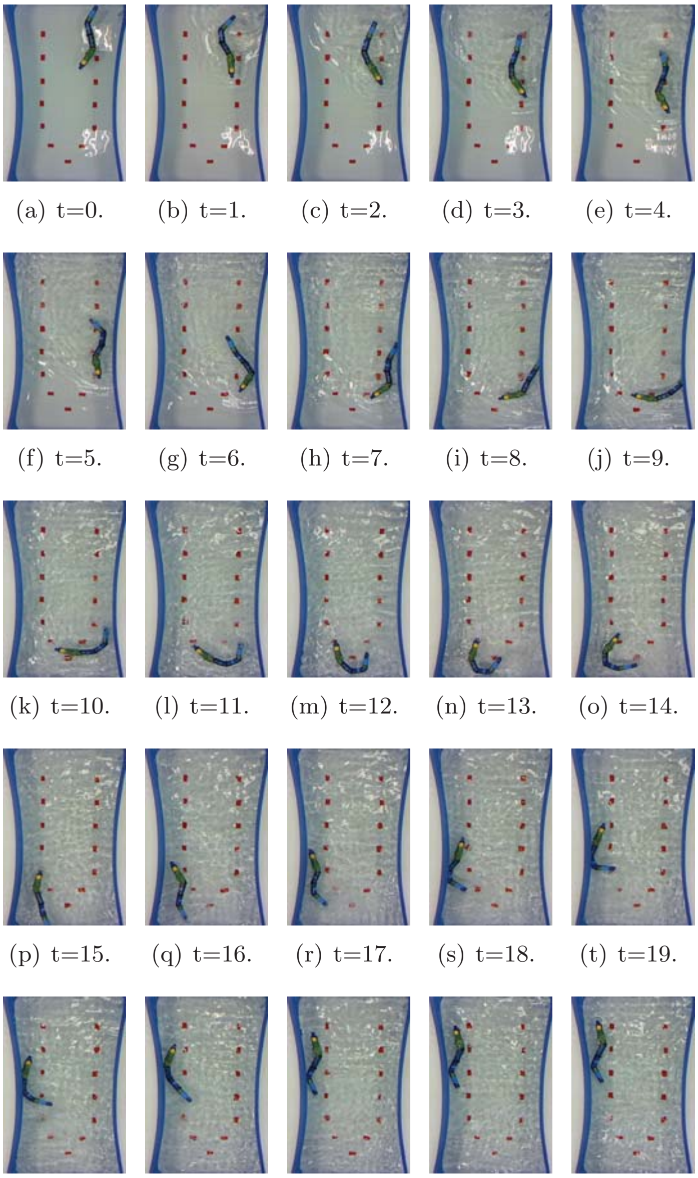
There are two steps to localize an object in our approach, and the two steps will be utilized on each frame of the video stream. The first step is to set a specific threshold for the color that we need to track. By doing this, color that falls in this threshold can be detected, while other colors that we are not interested in will be ignored. In our work, the original image will be first transformed to hue, saturation, value (HSV) format for further processing. In HSV color space, the threshold for red color is (170,160,60)–(180,255,255), while the threshold for yellow color is (80,100,100)–(150,255,255). Through this threshold process, the original image will be transformed into a binary image. In the binary image, color within the threshold shows as white, while color out of the threshold shows as black. The second step is to determine the position of the interested object. On the binary image, the contour of the interested object is represented by white color. Thus, the object can be recognized. By calculating the spacial moment and central moment of all the pixels on the binary image, the centroid of the white contour can be obtained. Thus, the position of the interested target can be identified. After finding the positions of the fish and all red the markers, and based on their distance relations, we can apply the control strategy on the fish and let it conduct corresponding actions.

5.6.4 Experimental result

By using the control strategy as shown in Fig. 5.20, we have done an experiment. In this experiment, the robotic fish started from the start point, moved all the way following the desired "U" shape trajectory, braked at the brake point, and finally stopped at the end point. In the whole process, the oscillation frequency ω is set to 2π (rad/sec), since we do not need to regulate the speed of the fish. Also, we set the oscillation amplitude and phase difference as constants: $A_m = 0.45$ (rad), $\theta = 1.5$ (rad).

The experiment result is shown in Fig. 5.21. As shown in the figure, it takes a total time of 24 seconds for the robotic fish to swim from the start point to the end point, and snapshot at each single second is presented in the figure. From the figure, it can be seen that at $t = 0 - 5$ sec, the fish moved approximately straightly. At $t = 5$ sec, the fish met change point 1, and it started to turn itself. Thus, at $t = 5 - 15$ sec, the fish moved circularly. In this period of time, it can be obviously note that the deflection on the fish body. At $t = 15$ sec, the fish met change point 2, and it started to change locomotion again, i.e., moved straightly afterwards. From $t = 15 - 21$ sec, the fish moved forward in a straight line. At $t = 21$ sec, the fish met brake point, and it stopped moving. In the following three seconds $t = 21 - 24$ sec, the fish drifted to the end point. From the last three figures, it can be seen that the fish body maintain the same gesture, which means that the fish has stopped movement and drifted forward due to inertia effect.

From the experiment result as shown in Fig. 5.21, we see that by using the control strategy we have designed, the fish can conduct locomotion change in the two change points, and can stop moving in the brake point. As a result, the fish can basically follow the desired U shape trajectory.



(a) t=0. (b) t=1. (c) t=2. (d) t=3. (e) t=4.

(f) t=5. (g) t=6. (h) t=7. (i) t=8. (j) t=9.

(k) t=10. (l) t=11. (m) t=12. (n) t=13. (o) t=14.

(p) t=15. (q) t=16. (r) t=17. (s) t=18. (t) t=19.

(u) t=20. (v) t=21. (w) t=22. (x) t=23. (y) t=24.

Figure 5.21: Snapshots of the forward locomotion.

5.7 Some Discussions on Trajectory Tracking

In Chapter 4, we see that the fish can be controlled to move forward, move backward and turn. All these movements are achieved by regulating different joints' orientation, which is a job in joint space. However, if we want the fish to arrive a desired point or follow a desired trajectory, trajectory tracking work is needed.

Remark 5.2. In this section, we discuss "exact" trajectory tracking, which is different from motion planning in previous sections, but similar to industrial manipulator's trajectory tracking.

There are several methods for the trajectory tracking work.

One possible method is to design reference trajectories of the position and orientation of the first link $x_{1r}, y_{1r}, \phi_{1r}$. Due to the fact that all the links are jointed together, if the first link can follow desired trajectory, other links would also follow. We define that $\mathbf{z}_1 = [x_1, y_1, \phi_1]^T$, and $\mathbf{z}_{1r} = [x_{1r}, y_{1r}, \phi_{1r}]^T$, where $\mathbf{z}_1$ and $\mathbf{z}_{1r}$ contain the actual values and reference values of the first link's position and orientation, respectively, then we have the dynamics of $\mathbf{z}_1$

$$\ddot{\mathbf{z}}_1 = \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{\phi}_1 \end{bmatrix} = C_1(\mathbf{p})\dot{\mathbf{p}} + D_1(\mathbf{p})\mathbf{w}_x + D_2(\mathbf{p})\mathbf{w}_y + D_3(\mathbf{p})B_\tau \tau$$

where $C_1(\mathbf{p}) \in \Re^{(N-1)\times 3N}, D_1(\mathbf{p}) \in \Re^{(N-1)\times N}, D_2(\mathbf{p}) \in \Re^{(N-1)\times N}, D_3(\mathbf{p}) \in \Re^{(N-1)\times N}$ are corresponding coefficient matrices obtained from matrix $A(\mathbf{p})$, $B(\mathbf{p})$ in (2.13).

For simplicity, we adopt the computed torque control to get the torque

$$\tau = (D_3 B_\tau)^{-1}[\ddot{\mathbf{z}}_{1r} + k_1(\mathbf{z}_{1r} - \mathbf{z}_1) + k_2(\dot{\mathbf{z}}_{1r} - \dot{\mathbf{z}}_1) - (C_1\dot{\mathbf{p}} + D_1\mathbf{w}_x + D_2\mathbf{w}_y)] \tag{5.3}$$

If the above torques τ can work, the matrix $D_3 B_\tau$ must be invertible. To verify this, we can use the result from Chapter 3, where the torques τ derived from computed torque control can work normally. From the obtained matrix A and B, we can get our interested matrix D_3, then we can plot the eigenvalues of $D_3 B_\tau$ as shown in Fig. 5.22.

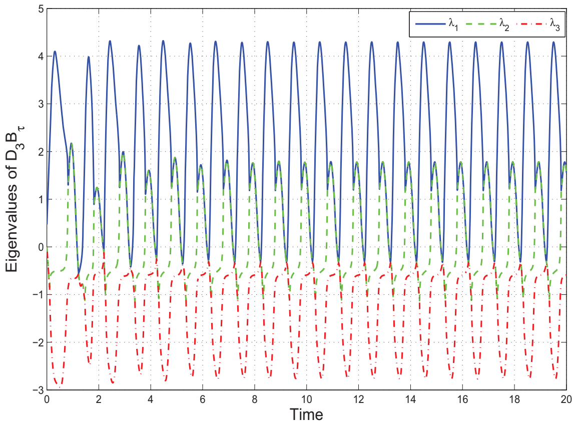


Figure 5.22: Eigenvalues of $D_3 B_\tau$.

In Fig. 5.22, we see that the eigenvalues of $D_3 B_\tau$ are sometimes 0, which indicates that $D_3 B_\tau$ is not invertible. Therefore, the torque given by (5.3) does not work.

Similarly, another way for trajectory tracking is to design desired x-trajectory for all the links. We define that $\mathbf{x} = [x_1, x_2, x_3, x_4]^T$, and $\mathbf{x_r} = [x_{1r}, x_{2r}, x_{3r}, x_{4r}]^T$, where $\mathbf{x}$ and $\mathbf{x_r}$ contain the actual values and reference values of all the links' x coordinates, respectively, then we have the dynamics of $\mathbf{x}$

$$\ddot{\mathbf{x}} = A_3(\mathbf{p})\dot{\mathbf{p}} + B_7(\mathbf{p})\mathbf{w}_x + B_8(\mathbf{p})\mathbf{w}_y + B_9(\mathbf{p})B_\tau \tau$$

where $A_3(\mathbf{p}) \in \Re^{N\times 3N}, B_7(\mathbf{p}) \in \Re^{N\times N}, B_8(\mathbf{p}) \in \Re^{N\times N}, B_9(\mathbf{p}) \in \Re^{N\times N}$ are corresponding coefficient matrices obtained from matrix $A(\mathbf{p})$, $B(\mathbf{p})$ in (2.13).

We also adopt the computed torque control to get the torques

$$\tau = (B_\tau^T B_9 B_\tau)^{-1} B_\tau^T [\ddot{\mathbf{x}}_\mathbf{r} + k_1(\mathbf{x}_\mathbf{r} - \mathbf{x}) + k_2(\dot{\mathbf{x}}_\mathbf{r} - \dot{\mathbf{x}}) - (A_3\dot{\mathbf{p}} + B_7\mathbf{w}_x + B_8\mathbf{w}_y)] \qquad (5.4)$$

If the above torques τ can work, the matrix $B_\tau^T B_9 B_\tau$ must be invertible. To verify this, we can use the result from Chapter 3, where the torques τ derived from computed torque control can work normally. From the obtained matrix A and B, we can get our interested matrix B_9, then we can plot the eigenvalues of $B_\tau^T B_9 B_\tau$ as shown in Fig. 5.23.

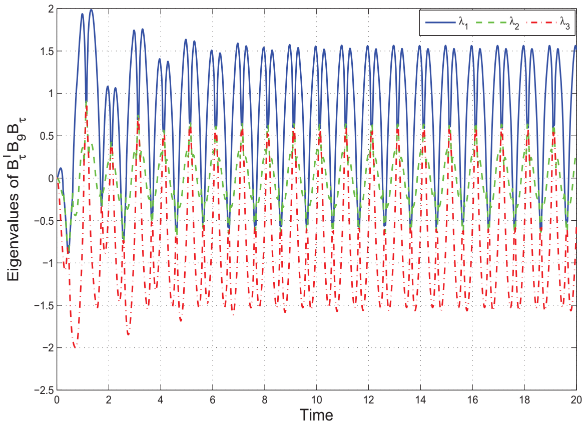


Figure 5.23: Eigenvalues of $B_\tau^T B_9 B_\tau$.

In Fig. 5.23, we see that the eigenvalues of $B_\tau^T B_9 B_\tau$ are sometimes 0, which indicates that $B_\tau^T B_9 B_\tau$ is not invertible. Therefore, the torque given by (5.4) does not work.

Remark 5.3. As discussed in the above two cases, control laws are developed individually following (5.3) and (5.4). However, the control torque τ always diverges to infinity, in both cases.

Apart from the above two methods, we have also tried to use dynamics of $[x_1, y_1]^T$, and $[x_1, \phi_1]^T$. However, neither of them works.

Till now, the only method that works is to regulate joints' orientation, as indicated in Chapter 3, where the torques are given by

$$\tau = (B_\tau^T B_3 B_\tau)^{-1} B_\tau^T [\ddot{\phi}_r + k_1(\phi_r - \phi) + k_2(\dot{\phi}_r - \dot{\phi}) - (A_1\dot{\mathbf{p}} + B_1\mathbf{w}_x + B_2\mathbf{w}_y)]$$

We can plot the eigenvalues of $B_\tau^T B_3 B_\tau$, as shown in Fig. 5.24. We see that none of the eigenvalues is 0 in the period of simulation, thus τ can be obtained accordingly from the above equation.

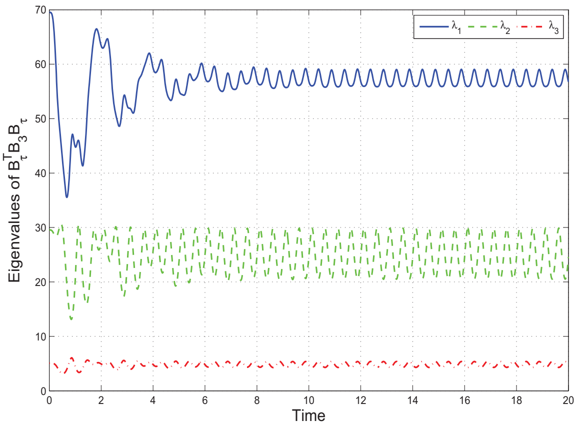


Figure 5.24: Eigenvalues of $B_\tau^T B_3 B_\tau$.

Remark 5.4. In the first case and the second case, the matrices D_3 and B_9, in which we are interested, are got from the last case. We find that even in the situation that control law τ works normally, the eigenvalues of $D_3 B_\tau$ and $B_\tau^T B_9 B_\tau$ still cross the line where the eigenvalue equals to 0. Therefore, the divergence phenomenon is inevitable.

In traditional robotic manipulator, the joint space and task space are usually split apart, and they are related by forward kinematics in (5.5)

$$r = f(q) \tag{5.5}$$

where r is the coordinates of task space, and q is the coordinates of joint space. If we want the end-effector of the manipulator to reach a point or follow a desired trajectory, we only need to solve out corresponding trajectory in joint space through method of inverse kinematics. Such a traditional manipulator usually has a stable base which is fixed, thus the relation in (5.5) exists.

However in our case, there is no such a fixed base, and the links of the fish are free to move in water. Thus, a simple relation between joint space and task space such as (5.5) does not exist. In our model, the joint space and task space are involved together, as shown in (2.3). A much more complex relation between them is in the dynamics of the fish as shown in (2.13). Indeed, the variables in task space is a consequence of the manipulation of the variables in joint space. So if we regulate the variables from both spaces at the same time, conflict may occur. Therefore, to avoid conflict, we will only manipulate variables in joint space.

Another reason that exact trajectory tracking can not be realized is, as mentioned in previous section, there always exists yaw motion in the robotic fish, no matter it conducts whichever locomotion. However, as far as the trajectory tracking task is concerned, the yaw motion along the desired trajectory is totally unnecessary.

Then, how to deal with the trajectory tracking problem? One feasible idea is that the desired trajectory can be decomposed into some basic primitives, either straight lines or arcs with different radii. In this way, the original trajectory can be represented by some simpler trajectories which are easy for the robotic fish to realize. By using the motion library, appropriate parameters can be selected so that the robotic fish can achieve these trajectories. Also, feedback can be used when the robotic fish deviates from the original trajectory, thus, tracking error can be rectified.

5.8 Conclusion

This chapter presents the investigation on a eight-link robotic fish, motion library building for the two fishes with different number of links, motion planning, corresponding simulation and experimental verification for the Anguilliform robotic fish.

First, simulation is conducted on an eight-link robotic fish. It is found that by using the given reference joint angles, the robotic fish can move forward as predicted. The successful result shows that the previously developed mathematical model and control approach can be applied on robotic fishes with different number of links. We also found that, the body wave on the eight-link fish is much smoother than that of the four-link fish, which may directly result in the higher speed of the eight-link fish. For both of the two fishes, motion libraries are built which contains the relations among the speed, the turning radius of the fish and related parameters. The significance of the motion library is that, for practical applications, control parameters of the robotic fish can be conveniently chosen so that desired speed and turning radius can be obtained. Based on the motion libraries, control strategy is designed and applied on the robotic fishes. The simulation results show that the control strategy can effectively handle different tasks. In the following, by using real-time feedback of camera, another experiment is conducted, where the robotic fish can successfully track a "U" shape trajectory. At last, some discussions are given about trajectory tracking. It is found that exact trajectory tracking can not be realized, since the robotic fish system does not have a simple mapping between the joint space and task space. A feasible way to achieve trajectory tracking is that the original trajectory can be decomposed into a few primitive trajectories, and feedback can be used to rectify possible tracking error.

Chapter 6

Locomotion Learning Using Central Pattern Generator Approach

6.1 Introduction

In previous chapters, we mainly focus on control law design of the robotic fish, different locomotion generation based on the mathematical model, motion library design and motion planning. Note that in all the previous contents, the locomotion of the fish is generated by given sinusoidal waves as joint angle references. However, in natural world, real fishes may not follow the same way. Therefore, from a biomimetic point of view, it is quite important to investigate how real fish swim, and whether the swimming pattern can be applied to the robotic fish, i.e., locomotion learning by the robotic fish from real fish. Only by learning the swimming pattern of a real fish, can we call the robotic "fish" a fish. Otherwise, it has no big difference with traditional submarine-like AUV.

The most important findings and fact in fish swimming is that: When a fish is swimming, there exists a body wave traveling along the fish body, and the direction of the body wave is opposite to the direction of its movement. Based on this finding, there are two major approaches developed in locomotion control of robotic fishes in most existing paper. The first approach is based on simple sinusoidal functions [4] [5] [54] [24] [45] [55] [56]. More specifically, sinusoidal waves are assigned to different joints of the robotic fish. The amplitudes, the phase differences, and the oscillation frequencies of the sinusoidal waves can be tuned according to specific need. The most advantage of this approach is that it is quite simple and easy to implement. However, it may not handle complicated environment or unpredictable affects. Moreover, from a biology point of view, the sinusoidal wave may hardly approximate the actual waveforms generated by real fishes. Considering these points, an alternative approach, focusing more on bio-inspired signals, is presented by many researchers, and this approach uses central pattern generator (CPG). CPG are neural circuits capable of producing coordinated patterns of high-dimensional rhythmic output signals while receiving only simple, low-dimensional input signals [33]. Biology experiments have shown clear evidence that in real fishes, the rhythms are generated centrally without requiring sensory information, and CPG are distributed networks made of multiple coupled oscillatory centers [57] [58] [59]. The advantages of CPG include that: They can exhibit limit cycle behavior, i.e., produce stable rhythmic patterns; It is convenient to use them for distributed implementation; Modulation among different locomotions can be realized by tuning a few control parameters. According to [33], there are four different mathematical models of CPG: detailed biophysical model, connectionist model, oscillator model, and neuro-mechanical model. Most existing papers of robotic fishes use the oscillator model to design CPG. In [60], the authors establish a model for a system of coupled nonlinear oscillators to construct CPG and apply it to the four-link robotic fish. Coordinated gait patterns of rhythmic movements for swimming can be obtained by modulating simple control parameters in the CPG model. In [61], the authors present a learning method to acquire fish-liking swimming with a CPG-based locomotion controller. The proposed method converts related CPG parameters into dynamical systems that evolve as part of the CPG network dynamics. The formulation of a dynamic model is presented in [21], for a free-swimming multi-joint robotic fish with a pair of wing-like pectoral fins. Furthermore, using CPG as the swimming data generator, the overall dynamic propulsive characteristics of the swimming robot are estimated. In [62], the construction and motion control of a biologically inspired, multi-mode biomimetic robotic fish is presented. The CPG are modeled as nonlinear oscillators for joints, and inter-joint coordination is achieved by altering the connection weights between joints. In [63], a bionic neural network, which consists of one high level controller and one chain of CPG, is presented for fish-robot locomotion. Each CPG contains a Zhang oscillator which shows properties similar to sine-cosine model. By using CPG, the generated signals become more robust due to the limit cycle property. Also, transition signals among different locomotions are smooth. However, in the above mentioned papers involving CPG approach, they only use coupled oscillators to build CPG, which is quite limited. Specifically, only sinusoidal waves can be generated by such coupled oscillators. In this sense, no big difference is made between sinusoidal approach and CPG approach. [64] addressed the problem of adapting the locomotor patterns to the properties of the environment, for a snake robot, and aimed at identifying fast swimming and crawling gaits for a variety of environments. The approach used a locomotion controller based on the biological concept of CPG together with a gradient-free optimization method, Powells method. In [65], the design and control of a biologically-inspired biomimetic robotic fish capable of three-dimensional locomotion was proposed. A model for a system of coupled non-linear oscillators was established to construct CPGs. The CPGs were modeled as non-linear oscillators for joints and inter-joint coordination was achieved by altering the connection weights between joints. Coordinated gait patterns of rhythmic movements for swimming could be produced by modulating simple control parameters in the CPG model. The CPG-based method showed elegant and smooth transitions between swimming gaits, and enhanced ability to cope with transient perturbations because of non-linear characteristic.

In this chapter, we present a brand new form of CPG model, which consists of coupled Andronov-Hopf oscillators, an artificial neural network (ANN), and an outer amplitude modulator. By using this model, we successfully applied swimming data of a real fish to our Anguilliform robotic fish, which type of fish is quite maneuverable and has an unique backward locomotion pattern compared with other types of fishes, and the robotic fish is able to swim forward and backward as predicted. Compared with other CPG works, the major superiority of our work is threefold: (i) Unlike previous works that use only coupled oscillators therefore can only generate fixed-pattern waveforms, we add artificial neural network and an outer amplitude modulator to the CPG structure, which makes it possible to generate different kinds of waveforms. Specifically, the CPGs in our work can generate swimming pattern of a real fish, while to the best of our knowledge, other works do not possess such capability; (ii) Three-dimensional topology is used in structure design of the coupled oscillators, and faster contraction rate can be achieved compared with those use traditional one-dimensional or two-dimensional topologies. Also, the three-dimensional topology is more robust under perturbations; (iii) By using different parameters, both forward and backward locomotion patterns can be realized within one CPG structure.

The chapter is organized as follows. In Section 2, the CPG model is given. It has three components: coupled Andronov-Hopf oscillators, an artificial neural network (ANN), and an outer amplitude modulator. The limit cycle character of the coupled oscillators, the advantage of three-dimensional topology, the properties of temporal scalability and spatial scalability and phase shift of the CPG, are discussed. In Section 3, we extract swimming data from a real Anguilliform fish. First, the properties of the CPG are verified by using the swimming data. Then, with the help of CPG, we obtain new data that on the one hand it reserves the pattern that the real fish swims, and on the other hand its values are suitable to be used on the robotic fish. At last, the effectiveness of the CPG generated data is verified by experiments. In Section 4, a conclusion is presented.

6.2 Central Pattern Generator

As shown in Fig. 6.1, the structure of the central pattern generator (CPG) contains three major components. The first component is coupled Andronov-Hopf oscillators, which are consisted of several single Andronov-Hopf oscillators. The second component is an artificial neural network (ANN). The third component is an output amplitude modulator.

6.2.1 Single Andronov-Hopf oscillator

As a basic element of coupled oscillators, single Andronov-Hopf oscillator will be introduced first in our work.

Andronov-Hopf oscillator originates from bifurcation theory. A bifurcation is a change of qualitative behavior of a dynamical system, and Andronov-Hopf bifurcation is among the most important bifurcations observed in neuron dynamics. It describes the onset (or disappearance) of periodic activity, which is ubiquitous in the neurons [66]. One of the characters of Andronov-Hopf oscillators is that, there exists a limit cycle under

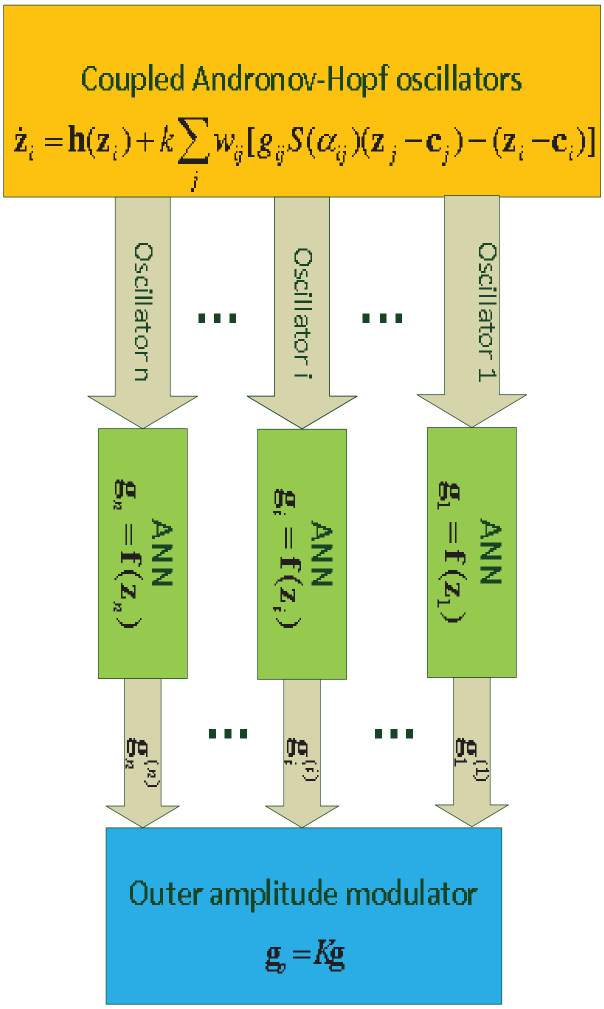

Coupled Andronov-Hopf oscillators
$\dot{\mathbf{z}}_i = \mathbf{h}(\mathbf{z}_i) + k\sum_j w_{ij}[g_{ij}S(\alpha_{ij})(\mathbf{z}_j - \mathbf{c}_j) - (\mathbf{z}_i - \mathbf{c}_i)]$
Oscillator n
...
Oscillator i
...
Oscillator 1
ANN
$\mathbf{g}_n = \mathbf{f}(\mathbf{z}_n)$
ANN
$\mathbf{g}_i = \mathbf{f}(\mathbf{z}_i)$
ANN
$\mathbf{g}_1 = \mathbf{f}(\mathbf{z}_1)$
$\mathbf{g}_n^{(n)}$
...
$\mathbf{g}_i^{(i)}$
...
$\mathbf{g}_1^{(1)}$
Outer amplitude modulator
$\mathbf{g}_o = K\mathbf{g}$

Figure 6.1: Structure of the CPG.

certain condition. The single Andronov-Hopf oscillator is able to produce sinusoidal form oscillation independently.

The dynamics of a single Andronov-Hopf oscillator is often described by differential equations, and a two-dimensional Andronov-Hopf oscillator takes the following form

$$\dot{\mathbf{z}} = \mathbf{h}(\mathbf{z}) = \begin{bmatrix} \dot{m} \\ \dot{n} \end{bmatrix} = \begin{bmatrix} -\omega(n-c_2) - \beta(\frac{(m-c_1)^2+(n-c_2)^2}{a^2} - 1)(m-c_1) \\ \omega(m-c_1) - \beta(\frac{(m-c_1)^2+(n-c_2)^2}{a^2} - 1)(n-c_2) \end{bmatrix} \tag{6.1}$$

where $\mathbf{z} = [m, n]^T$ is the state vector of single Andronov-Hopf oscillator, $\mathbf{c} = [c_1, c_2]^T$ is a constant vector representing the oscillation center, $a > 0$ represents the amplitude of the oscillator, β represents the attraction rate of the oscillator, and $\omega > 0$ is the oscillation frequency. The parameters $\mathbf{c}$, a, β and ω can be regulated according to our need.

The sign of the parameter β is critical to the existence of the limit cycle of single Andronov-Hopf oscillator. If $\beta > 0$, the phase plot (trajectory of $m-n$) of the Andronov-Hopf oscillator will form a limit cycle in the end. Now, we give a theorem and proof of it.

Theorem 6.1. *If $\beta > 0$, and the initial condition of (m, n) is not exactly (c_1, c_2), then the $m-n$ trajectory of the Andronov-Hopf oscillator will converge to a limit cycle defined by $(m-c_1)^2 + (n-c_2)^2 = a^2$.*

Proof. For convenience of expression and derivation, we give the following notation

$$\begin{aligned} m' &= m - c_1 \\ n' &= n - c_2 \end{aligned}$$

Then, the dynamics of a single Andronov-Hopf oscillator in (6.1) becomes

$$\dot{\mathbf{z}} = \mathbf{h}(\mathbf{z}) = \begin{bmatrix} \dot{m'} \\ \dot{n'} \end{bmatrix} = \begin{bmatrix} -\omega n' - \beta(\frac{m'^2+n'^2}{a^2} - 1)m' \\ \omega m' - \beta(\frac{m'^2+n'^2}{a^2} - 1)n' \end{bmatrix} \tag{6.2}$$

Define the Lyapunov function

$$V(\mathbf{z}) = \frac{1}{2}(m'^2 + n'^2 - a^2)^2 \tag{6.3}$$

Differentiating V and considering the dynamics of $\mathbf{z}$ in (6.2) into $\dot{V}$, we have

$$\begin{aligned} \dot{V}(\mathbf{z}) &= (m'^2 + n'^2 - a^2)(2m'\dot{m'} + 2n'\dot{n'}) \\ &= -\frac{2\beta}{a^2}(m'^2 + n'^2)(m'^2 + n'^2 - a^2) \end{aligned}$$

We see that the set of points in $\mathbb{R}^2$ that satisfy $\dot{V}(\mathbf{z}) = 0$ is $\{0\} \cup \{(m', n')|m'^2 + n'^2 = a^2\}$. Let $\mathbb{M}$ be the largest invariant set in $\mathbb{R}^2$. We find that $\mathbb{M} = \{0\} \cup \{(m', n')|m'^2 + n'^2 = a^2\}$, that is, $\mathbb{M}$ contains the origin and a limit cycle.

Since $V(\mathbf{z}) \to \infty$ as $||\mathbf{z}|| \to \infty$, and $\dot{V}(\mathbf{z}) \leq 0$ over the whole state space, by applying global invariant set theorem [47], we conclude that all solutions globally asymptotically converge to $\mathbb{M}$ as $t \to \infty$.

The origin $(0, 0)$ is actually unstable, and now we give the reason. Consider the region $\Omega_{\frac{1}{2}a^4}$, defined by $V(\mathbf{z}) < \frac{1}{2}a^4$. Note that the origin $(0, 0)$ does not belong to $\Omega_{\frac{1}{2}a^4}$, while the limit cycle $m'^2 + n'^2 = a^2$ is within the region. Thus, within the region $\Omega_{\frac{1}{2}a^4}$, the largest invariant set is only the limit cycle. Since the region $\Omega_{\frac{1}{2}a^4}$ is bounded, and $\dot{V(\mathbf{z})} \leq 0$ for all $\mathbf{z}$ in $\Omega_{\frac{1}{2}a^4}$, by applying local invariant set theorem [47], we conclude that every solution originating in $\Omega_{\frac{1}{2}a^4}$ tends to reach the limit cycle as $t \to \infty$. Therefore, any point near the origin actually converges to the limit cycle, which implies that the equilibrium point at the origin is unstable.

Considering the transformation between (m, n) and (m', n'), the limit cycle is a circle with its center at $\mathbf{c}$ and radius a. It completes the proof. □

Once the trajectory reaches the limit cycle, it will stay on it in all future time. Substituting $(m - c_1)^2 + (n - c_2)^2 = a^2$ into (6.1), we get

$$\begin{bmatrix} \dot{m} \\ \dot{n} \end{bmatrix} = \begin{bmatrix} -\omega(n - c_2) \\ \omega(m - c_1) \end{bmatrix} \tag{6.4}$$

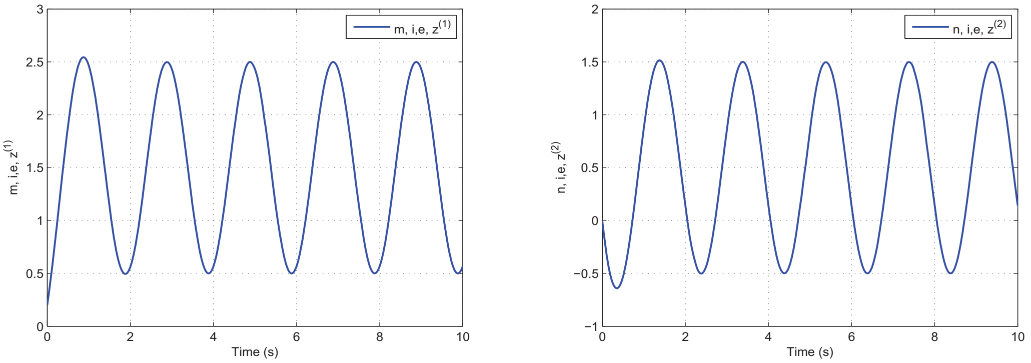
and we easily solve that $m(t) = c_1 + a\cos(\omega t + \psi)$, $n(t) = c_2 + a\sin(\omega t + \psi)$, where ψ depends on the initial condition.

From the above proof and the expression of m and n, we see that different functions that each parameter has. $\mathbf{c}$ is the center of the limit cycle. a is the radius of the limit cycle. From the expression of $\dot{V}$, we see that $\dot{V}$ is directly related with β. The bigger β is, the more quickly that V changes. Thus, the contraction rate of the Andronov-Hopf oscillator depends on β. The oscillation frequency of Andronov-Hopf oscillator is ω, and the period is $2\pi/\omega$.

Now we give some numerical examples. We select parameters $\omega = \pi$, $\beta = 1$, $a = 1$, $\mathbf{c} = [1.5 \;\; 0.5]^T$, and the initial condition is $\mathbf{z}|_{t=0} = [0.2 \;\; 0]^T$. Fig. 6.2 shows the time trajectories of the two coordinates of the vector $\mathbf{z}$, m and n, respectively. We see that the period of oscillation is exactly $2\pi/\omega$, i.e., 2 seconds.

By choosing different initial conditions, we obtain different trajectories as shown in Fig. 6.3. All trajectories tend to approach the limit cycle $(m - 1.5)^2 + (n - 0.5)^2 = 1$, no matter the starting point is inside or outside the circle (except the point $(1.5, 0.5)$).

One nice property of limit cycle is disturbance rejection. The property promises that, under the condition of existence of disturbance, the system states can still recover to the



(a) m trajectory.

(b) n trajectory.

Figure 6.2: Trajectories of single Andronov-Hopf oscillator.

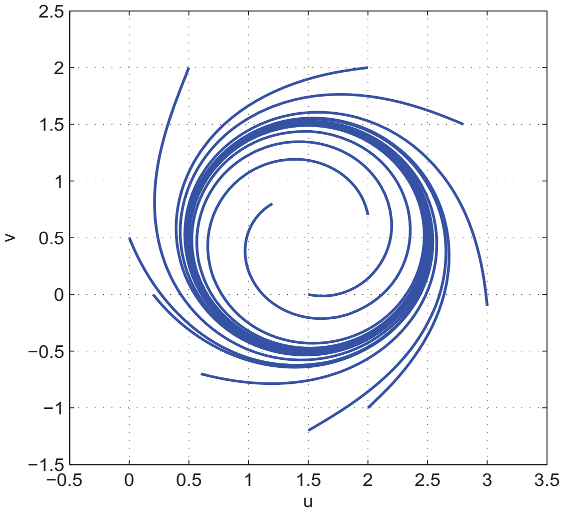


Figure 6.3: Phase plot of the limit cycle with different initial conditions.

original trajectories. To verify this property, we add pulse type disturbances to both of the states after the oscillator get into the limit cycle, at time $t = 10$ sec. Fig. 6.4 shows the phase plot, where the blue line represents the trajectory when $t < 10$, the red line represents the trajectory when $t > 10$, the blue dot and the red dot represent the points at time $t = 10$ instantly before and after the disturbance is added.

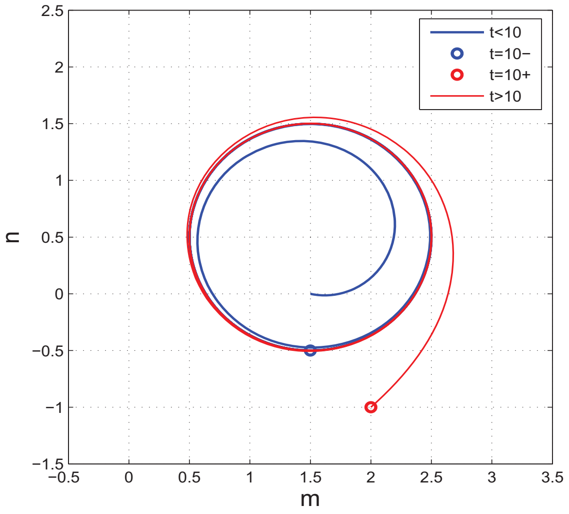


Figure 6.4: Phase plot of the limit cycle under disturbance.

6.2.2 Coupled Andronov-Hopf oscillators

The first component of CPG is coupled oscillators, whose function is to provide excitation inputs to the ANN. In this part, we will investigate the design of coupled oscillators.

Since a single oscillator corresponds to only one reference signal which is for only one joint angle, it is not enough if we want to generate reference signals for multiple joints. Additionally, phase difference can not be well expressed in a single oscillator. Thus, if we need to generate arbitrary number of reference signals, as well as to obtain desired phase differences, it is necessary to use a series of such oscillators and couple them together.

There are three key issues in designing the coupled oscillator network. First, what mathematical form that the coupled oscillators take, which means how oscillators are related and how the information is transferred. Second, topology or structure of the CPG network. Even using the same number of oscillators, we can apply different topologies on them, which result in different consequences. Actually, the performance of the network is directly related with its topology. Third, the parameters, which directly influence the dynamics of the coupled oscillators [67]. In the following contents, we will investigate these issues.

Mathematical formulation and topology of the coupled oscillators

Inspired by [68], we let the coupled oscillators take the following form:

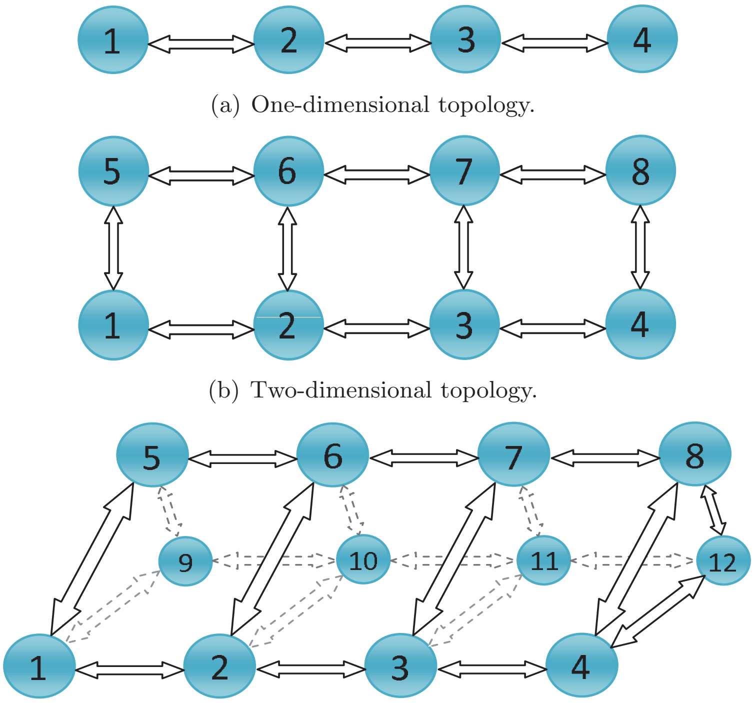
$$\dot{\mathbf{z}}_i = \mathbf{h}(\mathbf{z}_i) + k \sum_j w_{ij}[g_{ij} S(\alpha_{ij})(\mathbf{z}_j - \mathbf{c}_j) - (\mathbf{z}_i - \mathbf{c}_i)] \tag{6.5}$$

where i and j are index numbers of oscillators, oscillator j is the oscillator that has direct connections with oscillator i. The dynamics of $\mathbf{z}_i$ is composed of two terms. The first term $\mathbf{h}(\mathbf{z}_i)$, describing the effect from $\mathbf{z}_i$ itself, is the same as the one in single Andronov-Hopf oscillator. The second term describes relations with other connected oscillators. k is constant coupling strength, w_{ij} is the weight of connection between two oscillators, $g_{ij} = a_i/a_j$ is the amplitude ratio between two oscillators, α_{ij} is the desired phase difference between two oscillators, $\mathbf{c}_i$ is the oscillation center, and $S(\alpha_{ij})$ is rotation transformation, i.e.,

$$S(\alpha_{ij}) = \begin{bmatrix} \cos\alpha_{ij} & -\sin\alpha_{ij} \\ \sin\alpha_{ij} & \cos\alpha_{ij} \end{bmatrix}$$

From (6.5), we see that if all the oscillators reach their desired amplitudes and phase, the second term vanishes. Then the dynamics of $\mathbf{z}_i$ merely depends on the first term, which means all the oscillators behave the same like the single oscillator. In the end, the states of all oscillator move in their own limit cycle.

Fig. 6.5 shows three types of topologies of CPG network. All the three topologies follow the connecting rule: each oscillator connects only to the nearest oscillator(s). The rule makes the structure of the network clear and explicit. The one-dimensional topology and two-dimensional topology have appeared in previous works [61] [64]. The three-dimensional topology in this paper is designed by ourselves, and to our best of knowledge, it has not been presented in other works.



(a) One-dimensional topology.

(b) Two-dimensional topology.

(c) Three-dimensional topology.

Figure 6.5: Different topologies of CPG network.

Performance comparison of CPGs with different topologies

In this part, we compare the performance of the three topologies as shown in Fig. 6.5, and demonstrate the superiority of the three-dimensional topology in Fig. 6.5(c).

Since our robotic fish consists of four links with four orientation angles, we need to generate the same number of reference angles by using the CPG network. In one-dimensional case, there are totally four oscillators. Obviously, we only need to allocate each reference angle to each oscillator. While in two-dimensional case and three-dimensional case, the number of the oscillators are more than that of the reference angles, how to assign the reference angles remains a problem. We handle it in the following way. For the two-dimensional (three-dimensional) case, we assign one reference angle to the two (three) oscillators in the same column. That is, the two (three) oscillators are used to generate the same limit cycle and are assigned the same parameters (not necessarily the initial conditions). For example, in two-dimensional topology as shown in Fig. 6.5(b), the 1st reference angle are assigned to both oscillator 1 and oscillator 5, and in three-dimensional topology as shown in Fig. 6.5(c), the 2nd reference angle are assigned to oscillator 2 and oscillator 6 and oscillator 10. There exist coupling weights but no phase difference among the oscillators in the same column. In this way, we can choose any oscillator in the same column to generate the reference signal.

Note that there are two elements m and n in the state vector $\mathbf{z}$ of each oscillator, and the status of m and n are actually equivalent. Since only one reference signal is needed from each oscillator, we can choose either of them. Without loss of generality, we choose the first element, m, as the reference signal.

Actually, the desired amplitudes, oscillation frequency, and phase differences are tracked quite well in all the three topologies, given enough time. So the tracking performance that we compare here is mainly in terms of the settling time, or the time of re-entering the limit cycle, under the same perturbation.

The performances of CPGs with different topologies are evaluated by the settling time after adding noises/perturbations. Here we illustrate in detail how and when we add perturbations to the coupled oscillators. The total simulation time is 20 seconds. In 0–10 sec, there are no noises in the coupled oscillators; In 10–11 sec, we add the same white noise with the range of [0,1] to the three topologies of the coupled oscillators; In 11-20 sec, there are no noises. All the parameters are the same for coupled oscillators of the three topologies, the only differences among them are their topologies. The parameters are as follows: The desired amplitudes are $a_1 = 1$, $a_2 = 1.2$, $a_3 = 1.5$, $a_4 = 2$, and the desired phase differences are $\alpha_{12} = 0.20$, $\alpha_{23} = 0.25$, $\alpha_{34} = 0.40$. Other parameters are assigned as $\omega = 1, \beta = 1, k = 10$, the oscillation center of each oscillator $\mathbf{c}$ is $(0, 0)$, the coupling weights $w_{ij} = 1$ for connected oscillators and $w_{ij} = 0$ for unconnected oscillators.

We summarize the results of the settling time after adding white noise in Table 6.1. From the table, we see that the three-dimensional topology has the shortest settling time, or the time of re-entering the limit cycle. Thus, the performance of the three-dimensional topology is the best among the three.

Table 6.1: Settling time comparison of coupled oscillators of different topologies.

Topologies	Settling time
One-dimensional	1.90 sec
Two-dimensional	1.52 sec
Three-dimensional	1.21 sec

Remark 6.1. For the three-dimensional topology, we assign the same reference angle to the oscillators within the same column. Using this redundancy allocation of reference signals, the inter-connections of oscillators in the same column is strengthened as well as the connections between neighboring columns. Compared with the other two topologies,

there are more connections of both kinds in the three-dimensional case. Thus, the limit cycles are reached more quickly. From another point of view, we can conclude that three-dimensional topology is the most robust under existence of perturbations among the three topologies.

Transitions when parameters change

When the robotic fish is swimming in water, transitions among different locomotion patterns are necessary. It is important for the transition signals to be smooth. Otherwise, jerky motion occurs, which can quickly wear and break mechanical parts of the motor. Thus, abrupt changes of signals should be avoided.

Now, we check the smoothness of CPG signals during transitions between different locomotion patterns. For different time intervals, we choose different parameters, corresponding to different locomotion patterns. The parameters ω , a_i , α_{ij} and $\mathbf{c}_i$, are shown in Table 6.2. In the first time interval 0–10 sec, all the four parameters are nominal values. In the second time interval 10–20 sec, the oscillation frequency ω becomes half of the nominal value, while other parameters remain the same. In the third time interval 20–30 sec, the oscillation amplitudes a_i change, while other parameters remain the nominal values. In the fourth time interval 30–40 sec, the desired phase differences α_{ij} are reversed, while other parameters remain the nominal values. In the fifth time interval 40–50 sec, the oscillation center $\mathbf{c}_i$ is changed, while other parameters remain the nominal values.

Fig. 6.6 shows that, under the change of different parameters, the transition trajectories of the CPG output. From the figure, we see that in all the five time intervals, the desired oscillation frequency, oscillation amplitudes, phase differences and oscillation centers as shown in Table 6.2, can be reached. Compared with the signals generated by

Table 6.2: CPG parameters in different time intervals.

Time t	ω	a_i	α_{ij}	$\mathbf{c}_i$
$0 \sim 10$	π	[1,1.2,1.5,2]	[0.2,0.25,0.4]	[0,0,0,0]
$10 \sim 20$	0.5π	[1,1.2,1.5,2]	[0.2,0.25,0.4]	[0,0,0,0]
$20 \sim 30$	π	[0.5,1,1.2,1.5]	[0.2,0.25,0.4]	[0,0,0,0]
$30 \sim 40$	π	[1,1.2,1.5,2]	[-0.2,-0.25,-0.4]	[0,0,0,0]
$40 \sim 50$	π	[1,1.2,1.5,2]	[0.2,0.25,0.4]	[0.3,0.4,0.6,1]

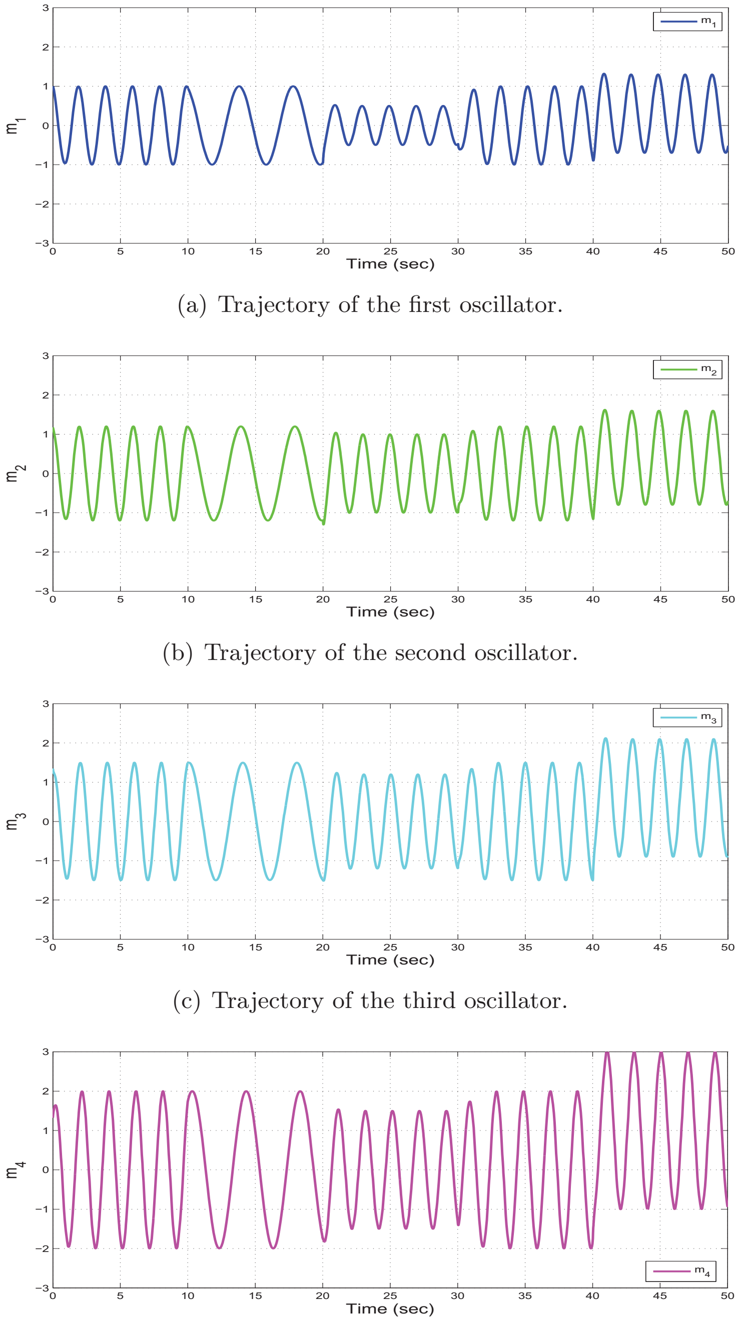
sinusoidal functions, as shown in Fig. 6.7, we see that the signals generated by CPG transit smoothly, while the signals in Fig. 6.7 change more abruptly when parameters vary.

The underlying reason, that the coupled oscillators transit smoothly under change of parameters, is due to the dynamics that the oscillators possess. So when parameters change, the transient process helps the oscillators move smoothly from one state to another, thus the states of the oscillators change gradually instead of abruptly.

From previous result in Fig. 6.6, we have the following remarks:

Remark 6.2. Arbitrary oscillation frequency, oscillation amplitude, phase differences and oscillation centers, can be obtained through tuning corresponding parameters.

Remark 6.3. When parameters change, the curves transit smoothly. Actually the smoothness is related to the parameter k in (6.5), which represents the coupling strength between different oscillators. We find that the bigger k is, the less smooth the curves are. The reason is quite straight forward. From (6.5), we see that when parameters change, the coupled oscillators need to transit from one steady state to another steady state. When transition starts, the coupling terms become non-zero. The larger k is, the larger the derivative of $\mathbf{z}$ becomes, which means the states of the oscillators will change in a more abrupt way. So when we choose the parameter k, we have to choose appropriate k



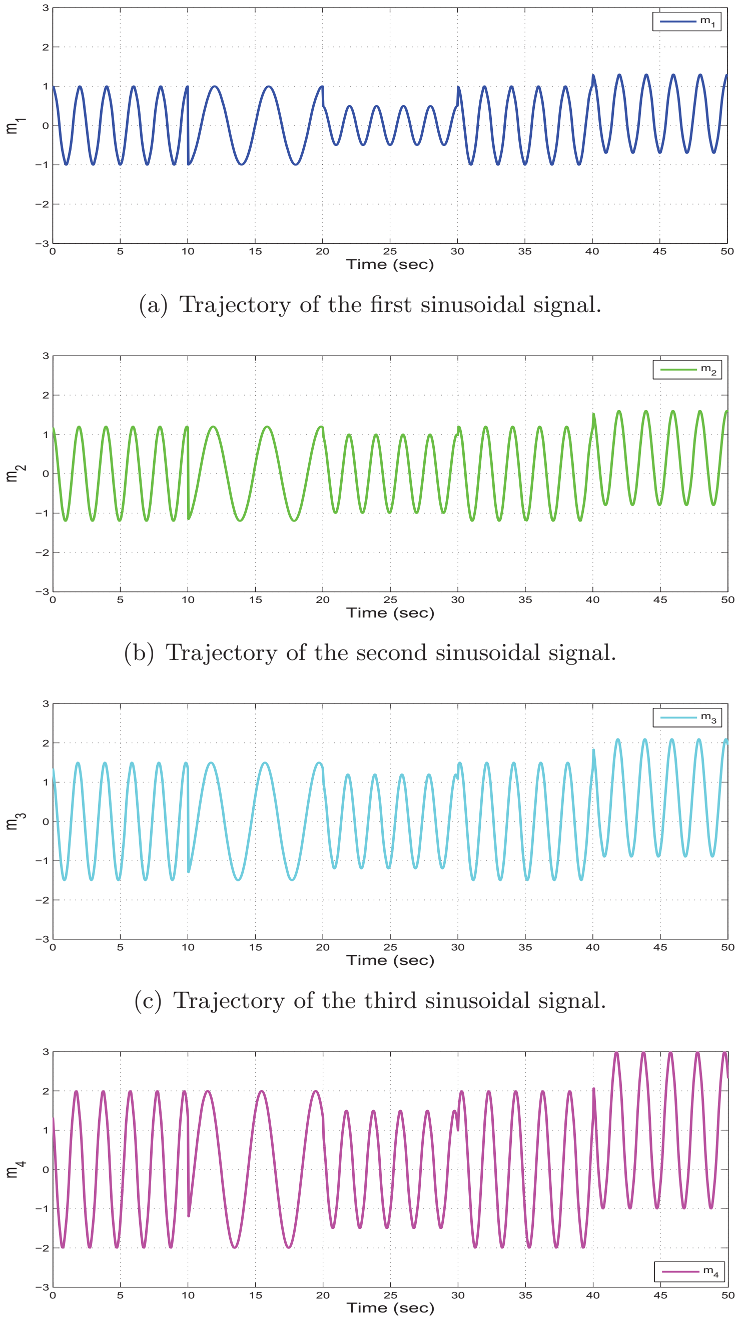
(a) Trajectory of the first oscillator.

(b) Trajectory of the second oscillator.

(c) Trajectory of the third oscillator.

(d) Trajectory of the fourth oscillator.

Figure 6.6: Transition trajectories of the CPG oscillators under change of the parameters.



(a) Trajectory of the first sinusoidal signal.

(b) Trajectory of the second sinusoidal signal.

(c) Trajectory of the third sinusoidal signal.

(d) Trajectory of the fourth sinusoidal signal.

Figure 6.7: Transition trajectories of the sinusoidal signals under change of the parameters.

to obtain smooth transitions.

6.2.3 Artificial neural network

The second component of the CPG is the artificial neural network (ANN). After getting trained by target values and receiving the excitation signals from the coupled oscillators, the ANN will output our desired waveform patterns.

The expression of the artificial neural network is

$$\mathbf{g}_i = \mathbf{f}(\mathbf{z}_i) \tag{6.6}$$

where $i = 1, \cdots, n$, $\mathbf{z}_i$ is the state vector of i-th Andronov-Hopf oscillator, also serves as the input of the i-th ANN, $\mathbf{g}_i$ is the output of the ANN, and $\mathbf{f}$ represents a nonlinear mapping. The periodic signal $\mathbf{z}_i$, obtained from the coupled oscillators, can provide sustained signals to excite the CPG.

Remark 6.4. In this work, we need the ANN to generate outputs corresponding to different locomotions of the robotic fish, thus the ANN will be trained by different groups of training input data and training output data.

Before we train the ANN, we need to assign different training inputs of the ANN for different locomotions. Here we use $\mathbf{z}_f$ and $\mathbf{z}_b$ to represent the training inputs of forward locomotion and backward locomotion, respectively, and their mathematical expressions are

$$\mathbf{z}_f = \begin{bmatrix} \cos(\omega t) \\ \sin(\omega t) \end{bmatrix} \qquad \mathbf{z}_b = \begin{bmatrix} \cos(\omega t) + 2 \\ \sin(\omega t) + 2 \end{bmatrix}$$

where ω is the oscillation frequency of the periodic motion, which can be selected according to our need. Such selection of $\mathbf{z}_f$ and $\mathbf{z}_b$ can ensure that there are no same training input at any time instant. Thus, forward locomotion and backward locomotion are differentiated by different training inputs.

The training output data is just the swimming data of real fish, which will be introduced later.

It should be noted that though CPG can handle both periodic motion learning and discrete motion learning [69], while in this paper, we mainly focus on the periodic motion case, because the locomotion of real fish is actually periodic. By replicating a series of identical motions, the fish is able to either move forward, or move backward.

Remark 6.5. Note that the dimension of the target values is n, which means the output dimension of a single ANN is also n. As shown in Fig. 6.1, the second component of the CPG contains n identical ANNs, and we extract the i-th element from the i-th ANN, i.e., extract $\mathbf{g}_i^{(i)}$ from $\mathbf{g}_i$. We use the extracted elements to define the new vector $\mathbf{g} \triangleq [\mathbf{g}_1^{(1)}, \cdots, \mathbf{g}_i^{(i)}, \cdots, \mathbf{g}_n^{(n)}]^T$. The reason we do this is that, in the training process of the ANN, the phase differences among the outputs of the ANN are actually fixed. However, in practical case, the phase differences need to be tunable. An alternative way to handle this is that, we replicate a number of n identical ANNs, and use the coupled oscillators' function to generate desired phase difference. As a result, the phase difference of different ANNs outputs can be coordinated.

6.2.4 Outer amplitude modulator

Since the ANN is a nonlinear mapping, we can not resize the amplitude of the output of the ANN by resizing the the amplitude of corresponding input of the ANN. Thus, it is necessary to add a layer after the ANNs so that we can obtain the desired amplitudes of the outputs, and that is the third component of the CPG – outer amplitude modulator.

The expression of the outer amplitude modulator is

$$\mathbf{g}_o = K\mathbf{g} \tag{6.7}$$

where $K = \text{diag}(k_1, \cdots, k_n)$ is a diagonal matrix, and k_i $(i = 1, \cdots, n)$ is a set of positive constants. $\mathbf{g}$ is the output of the ANN, and $\mathbf{g}_o$ is the output of the outer amplitude modulator. Generally, K serves as a spacial scaling matrix, through which we can tune the amplitude of $\mathbf{g}_o$.

6.2.5 Properties of the CPG

Many movements are similar in the sense that through appropriate temporal scaling, spacial scaling and phase shift, they can be transformed one another. These properties are quite useful when we need to generate a new pattern from the existing ones. In this part, we investigate these properties of the CPG, and without generality and for simplicity, we use forward locomotion as an example.

First, we investigate the temporal scalability of CPG.

Property 6.1. To stretch or compress motion patterns along the time axis by a scale of α, it is adequate to change the original temporal scaling parameter ω to $\alpha\omega$ in (6.1).

Proof. The original CPG output response at steady state is

$$\mathbf{g}_o(t) = K\mathbf{g}(t) = K\mathbf{f}(\mathbf{z}) = K\mathbf{f}(\begin{bmatrix} \cos(\omega t) \\ \sin(\omega t) \end{bmatrix})$$

The CPG output response with the time scaling is

$$\mathbf{g}'_o(t) = K\mathbf{f}(\begin{bmatrix} \cos(\alpha\omega t) \\ \sin(\alpha\omega t) \end{bmatrix})$$

Thus, we have $\mathbf{g}'_o(t) = \mathbf{g}_o(\alpha t)$. This shows that the original motion pattern $\mathbf{g}_o(t)$has been scaled by a temporal factor α. □

Remark 6.6. When $\alpha > 1$, the motion pattern is compressed along the time axis. While when $\alpha < 1$, the motion pattern is stretched along the time axis. In both cases, the spacial patterns generated are identical, but the motion speeds are different.

Next, we derive the spacial scalability property of CPG.

Property 6.2. To stretch or compress motion patterns along the spatial axis by a scale of γ (where γ =diag$\{\gamma_1, \cdots, \gamma_n\}$), it is adequate to change the original spatial scaling parameter K to γK in (6.7).

Proof. Trivial thus omitted. □

The temporal scalability and spatial scalability can be applied at the same time.

Property 6.3. To stretch or compress motion patterns along the temporal axis by a scale of α and along the spatial axis by a scale of γ, it is adequate to change the parameters ω to $\alpha\omega$, K to γK.

Proof. From Property 2, we obtain $\mathbf{g}_o(t) \to \gamma \mathbf{g}_o(t)$ when we change the parameter K to γK. Property 2 is valid for any value of ω. Thus, from Property 1, we can further derive that $\gamma \mathbf{g}_o(t) \to \gamma \mathbf{g}_o(\alpha t)$. Therefore, through appropriate parameter changes, scaled transformation of motion pattern along both temporal axis and spatial axis can be achieved. □

Property 6.4. To make motion pattern I lead motion pattern II by a time interval Δ, it is adequate to set a phase lead $\omega\Delta$ in (6.5) (i.e., tune the parameter α_{ij}) on oscillator I (compared with oscillator II) which corresponds to motion pattern I.

Proof. Denote motion pattern I as $\mathbf{g}_1(t)$, motion pattern II as $\mathbf{g}_2(t)$ (note that here the motion pattern is the output of each individual ANN, which corresponds to the individual oscillator). Motion pattern I with time advance is

$$\mathbf{g}_1(t) = \mathbf{f}(\mathbf{z}_1) = \mathbf{f}(\begin{bmatrix} \cos(\omega t + \omega\Delta) \\ \sin(\omega t + \omega\Delta) \end{bmatrix}) = \mathbf{f}(\begin{bmatrix} \cos(\omega(t + \Delta)) \\ \sin(\omega(t + \Delta)) \end{bmatrix})$$

and motion pattern II is

$$\mathbf{g}_2(t) = \mathbf{f}(\mathbf{z}_2) = \mathbf{f}(\begin{bmatrix} \cos(\omega t) \\ \sin(\omega t) \end{bmatrix})$$

Compared motion pattern I with motion pattern II, we have $\mathbf{g}_1(t) = \mathbf{g}_2(t+\Delta)$, which means motion I leads motion II by a time interval Δ. The phase lead $\omega\Delta$ can be tuned by the parameter α_{ij} in (6.5). □

Remark 6.7. Note that when a motion pattern transits from one state to a new state, we mainly consider three cases: the change in angular frequency, the change in phase difference, and the change in amplitudes. In the first two cases, the parameters that concern angular frequency and phase difference can be tuned in the coupled oscillators, thus the transient process is smooth due to the underlying dynamics of coupled oscillators. While in the third case, the amplitude parameter is tuned by the outer amplitude modulator K or γK according to Property 3. Here we must clarify that, the parameter K can be added in the very beginning at $t = 0$, which means the original motion is spatially scaled by K. In this case, K is added at the start point, no state transitions occur, thus K can be as simple as constants. While when it is concerned with amplitudes change in the middle of the process (not in the beginning), we have to consider smooth transitions of states, and the parameter γ of outer amplitude modulator is in charge of it. In the following part, we give the design of γ to meet the requirement of smooth transition.

When the amplitudes of motion patterns change from one state to another state, we let K remain the same, and in order to avoid abrupt change, we design a smooth transient process for γ in the following way. Denote that t_c is the time when the amplitude parameters change, and $\gamma_i(t)$ $(i = 1, \cdots, 4)$ is the ith element of γ, the initial value $\gamma_{i0} = \gamma_i|_{t=t_c} = 1$, $\gamma_{i\infty}$ is the final value of γ_i. Then we give the equation of $\gamma_i(t)$ as

$$\gamma_i(t) = \gamma_{i\infty} + (\gamma_{i0} - \gamma_{i\infty})\mathbf{e}^{-\mu(t-t_c)} = \gamma_{i\infty} + (1 - \gamma_{i\infty})\mathbf{e}^{-\mu(t-t_c)} \qquad (t \geq t_c) \qquad (6.8)$$

where $\mathbf{e}$ is the natural constant, and μ is the decay rate. Through the above process, the change of $\gamma_i(t)$ is made smooth.

Remark 6.8. The significance of the scalability and phase shift properties of CPG lies in that, if a new motion and an existing motion have temporal or spatial similarities, we can change corresponding parameters and apply them to the already trained ANN to generate the new motion. It is time efficient by doing so, because training an additional ANN consumes much time. For instance, if motion I has a larger amplitude but the identical duration compared with motion II, with the existing ANN of motion II, we can just use desired spatial scalability parameter to generate motion I.

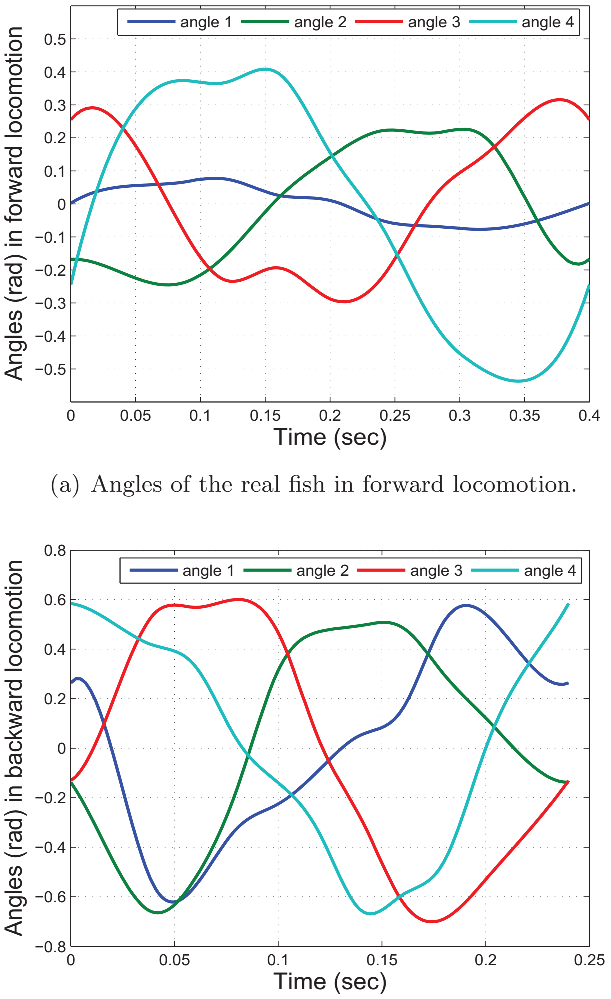
6.3 Experiments of Locomotion Learning Using Swimming Pattern of a Real Anguilliform Fish

6.3.1 Real fish swimming pattern

In [1], both forward and backward swimming locomotions of an Anguilliform fish, such as oscillation frequency, swimming speed, amplitudes and angles of the fish at different parts of the body, are investigated. From this paper, we can extract useful information about forward and backward locomotion patterns. Thus, swimming pattern of a real fish can be extracted and applied to the robotic fish. Also, applicable parameters can be generated for the robotic fish by using the properties of CPG.

As show in Fig. 6.8, the forward swimming and the backward swimming locomotions of a real Anguilliform fish are extracted from [1]. Fig. 6.8(a) shows a complete cycle of the four body points in forward locomotion, and Fig. 6.8(b) shows a complete cycle of the four body points in backward locomotion. Note that the positions of the four points on the fish are selected in such a way that they are located in similar positions along the fish body with the mid-points of the four links of the robotic fish (will be introduced later). From the figures, we see that the motion period of the forward swimming is 0.40 seconds, and the motion period of the backward swimming is 0.24 seconds.

Note that the curves that the real fish performs in Fig. 6.8 are different from sinusoidal waves, while sinusoidal waves are often used in most papers to supply for servo motors of robotic fish. Compare Fig. 6.8(a) and Fig. 6.8(b), we can find differences as well as similarities between forward locomotion and backward locomotion. In forward swimming, we see that the amplitude gradually increases from the fish's head to its tail. While in backward swimming, the amplitudes of different parts of the fish body are almost the same, and these amplitudes are all quite large. Another difference is that, in forward locomotion, the former part of the fish has a phase lead than the latter part, while in backward locomotion, the former part of the fish has a phase lag than the latter part. The similarities of forward and backward locomotions include that: First, the angle curve may contain more than one local maximum in one single period; Second, the waveforms are unsymmetrical and irregular. These two characters are quite different from sinusoidal waves.



(a) Angles of the real fish in forward locomotion.

(b) Angles of the real fish in backward locomotion.

Figure 6.8: Angle trajectories of a real Anguilliform fish in forward and backward locomotions [1].

6.3.2 Verification of CPG properties by using real fish swimming pattern

Here we adopt the forward locomotion pattern as an example to illustrate the properties of CPG. First, we use the experimental pattern of the real fish to train the ANN, then we apply desired parameters to the coupled oscillators. For purpose of clarity and simplicity, we let the dimension of the output be two: angle 2 and angle 3. In all the three sub-figures of Fig. 6.9, from time $t = 0 \sim 2$ sec, we use the original swimming pattern, where the oscillation frequency is 5π, and angle 2 has a phase lead of 0.61π compared with angle 3.

Fig. 6.9 (a) shows the property of temporal scalability. From time $t = 2 \sim 5$ sec, the angular frequency is changed to $\omega = 2\pi$, while other parameters remain the same. From the figure, we see that the oscillation frequency of the curves become 0.4 times of previous value after ω is changed, and the transition is smooth.

Fig. 6.9 (b) shows the property of spacial scalability. From time $t = 2 \sim 4$ sec, the parameters are changed to $\gamma_{2\infty} = 3$, $\gamma_{3\infty} = 2$, the decay rate $\mu = 10$, while other parameters remain the same. From the figure we see that, instead of an overall scalar change on both curves, the two curves change individually. Specifically, the amplitude of angle 2 becomes three times of previous value, while the amplitude of angle 3 becomes two times of previous value, and the transition is smooth.

Fig. 6.9 (c) shows the property of phase shift. From time $t = 2 \sim 4$ sec, we let angle 2 change from original 0.61π phase lead to 0.39π phase lag, compared with angle 3, while other parameters remain the same. The total change of phase difference is $\pi(= 0.61\pi - (-0.39\pi))$, which corresponds to the parameter $\Delta = 0.5$ in Property 4. From the figure we see that, the phase difference between angle 2 and angle 3 exactly changes from 0.61π phase lead to 0.39π phase lag, and the transition is smooth.

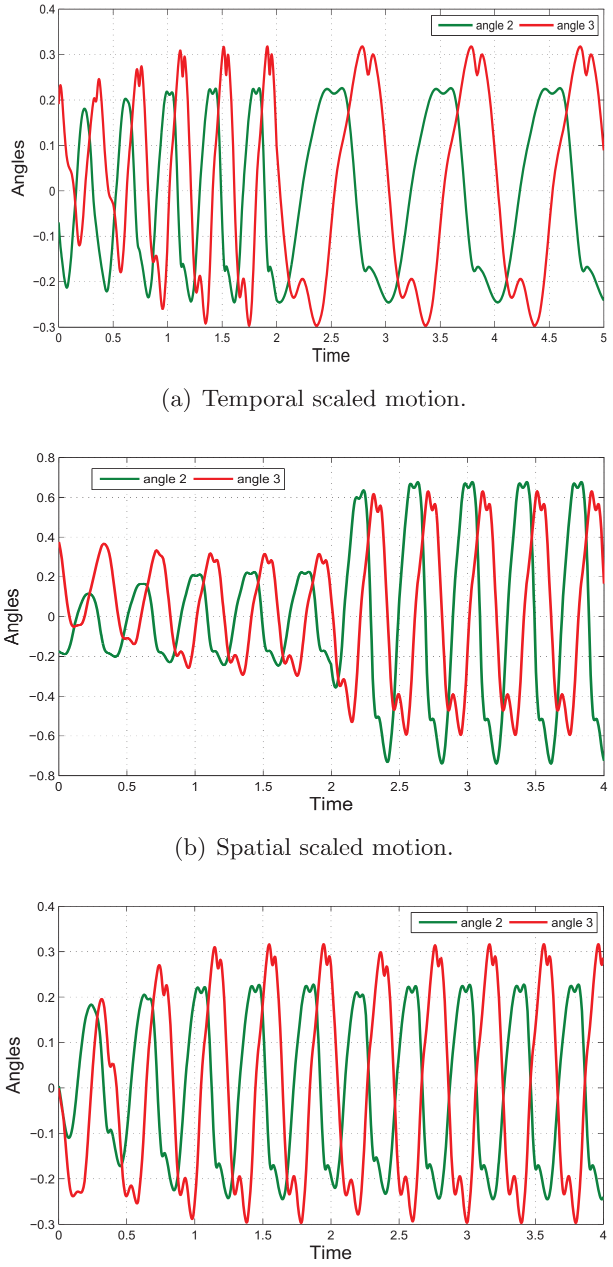
From the three sub-figures of Fig. 6.9, we see that all the transitions are smooth. After training the ANN and applied appropriate parameters to the coupled oscillators, we obtain temporal scaled motion, spatial scaled motion and phase shifted motion, respectively.

6.3.3 New swimming pattern generated by CPG

Since the motion frequency of the real fish is about $3 \sim 5$ Hz, which is higher than the value at which that the servo motor performs best. Furthermore, in the forward locomotion of the real fish [1], the amplitude difference between the anterior part and the posterior part of the fish is quite large. If we apply such angles to the robotic fish, it may cause unstable motion. Additionally, in order to produce the traveling body wave, we have to consider to add appropriate phase differences among different links. Thus, we need to modulate the amplitudes, the motion periods, and the phase differences to make the curves more applicable to the robotic fish.

Now, we give the detailed procedure that how we generate new swimming pattern using the CPG. First, we use the real fish locomotion pattern to train the ANN, thus the real fish locomotion patterns are memorized. Then, we set the CPG parameters as follows. For both forward locomotion and backward locomotion, the oscillation frequency is set to be 1 Hz, which means the period is 1 sec. For forward locomotion, we set phase difference as 0.4π, 0.4π, 0.6π (rad), and set the peak amplitude as 0.4, 0.4, 0.45, 0.5 (rad), the curves are shown in Fig. 6.10(a); For backward locomotion, we set phase difference as -0.4π, -0.4π, -0.6π (rad), the peak amplitude 0.4, 0.4, 0.45, 0.5 (rad), the curves are shown in Fig. 6.10(b).

It should be noted that, we can use the CPG to generate as many new curves as

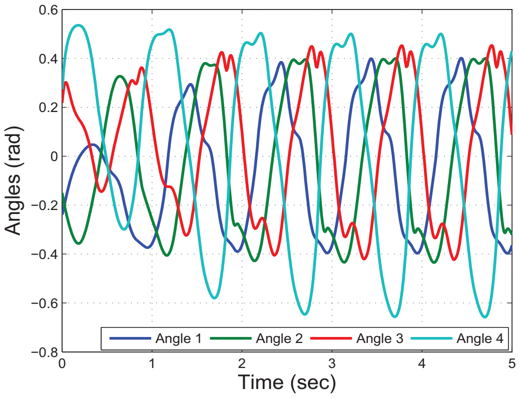


(a) Temporal scaled motion.

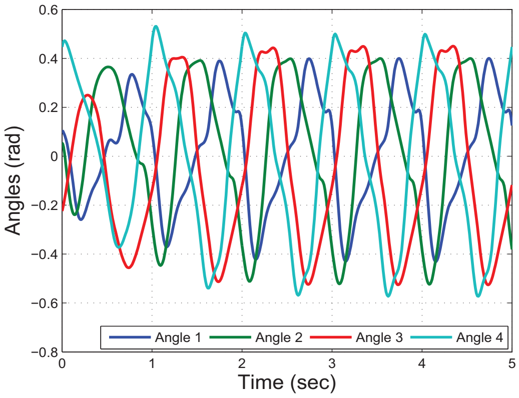
(b) Spatial scaled motion.

(c) Phase shifted motion.

Figure 6.9: Transitions from the original motion to transformed motions. (a) Temporal scaled motion with parameter $\alpha = 0.4$. (b) Spatial scaled motion with parameter $\gamma =$diag$\{3, 2\}$. (c) Phase shifted motion with parameter $\Delta = 0.5$.



(a) Angles generated by CPG in forward swimming.



(b) Angles generated by CPG in backward swimming.

Figure 6.10: Forward swimming and backward swimming locomotions generated by CPG.

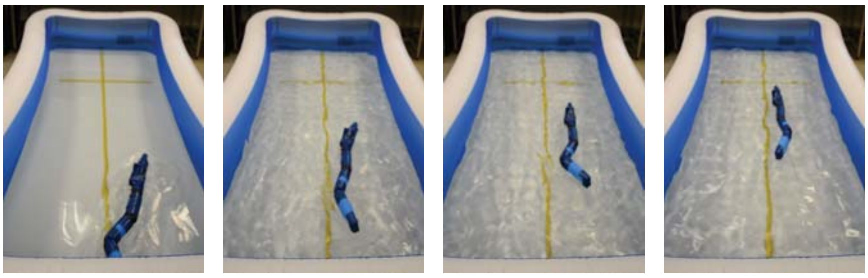
we want. Specifically, the number could be infinite by tuning those parameters. Among these new curves, many can work normally, and the pattern in Fig. 6.10 is just one of the applicable patterns to the robotic fish. We use it as an example to illustrate the effectiveness of the CPG. Details will be provided in later contents.

6.3.4 Experimental results

In this part, we present the experimental result of the robotic fish by using new swimming patterns generated from the biological data. The experimental setup is the same as that described in previous chapters.

By applying the new pattern (as shown in Fig. 6.10) generated by CPG, we give the experimental results of both forward and backward locomotions of the robotic fish, which are shown in Fig. 6.11-6.13.

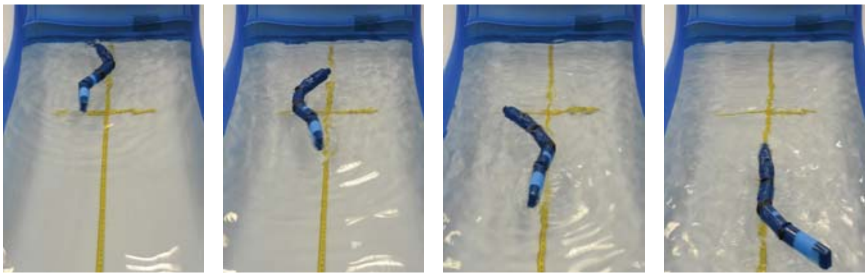
The time for both forward and backward locomotions are 20 seconds. Fig. 6.11 and Fig. 6.12 show the snapshots of forward locomotion and backward locomotion of the robotic fish at different time instants, respectively. From Fig. 6.11, we see that, the robotic fish started from a near-to-camera site, swam forward, and gradually kept moving away from the camera. While from Fig. 6.12, we see that, the robotic fish started from a far-from-camera site, swam backward, and gradually kept approaching the camera.



(a) t=0 sec. (b) t=6.7 sec. (c) t=13.3 sec. (d) t=20 sec.

Figure 6.11: Snapshots of the forward locomotion.

Note that in some of the pictures in Fig. 6.11 and Fig. 6.12, the robotic fish swings



(a) t=0 sec. (b) t=6.7 sec. (c) t=13.3 sec. (d) t=20 sec.

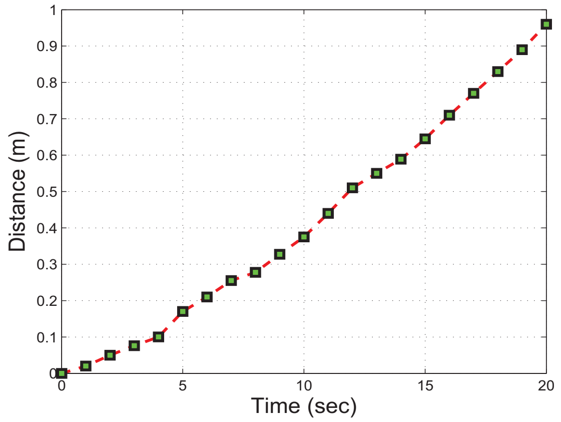
Figure 6.12: Snapshots of the backward locomotion.

"significantly". That is because the robotic fish imitates Anguilliform fish. Different from other types of fishes, one unique character of Anguilliform fish is that the whole body participates in large amplitude undulation when it is swimming [7]. Thus, the significant swing in the pictures results from the large amplitude undulation along the fish body.

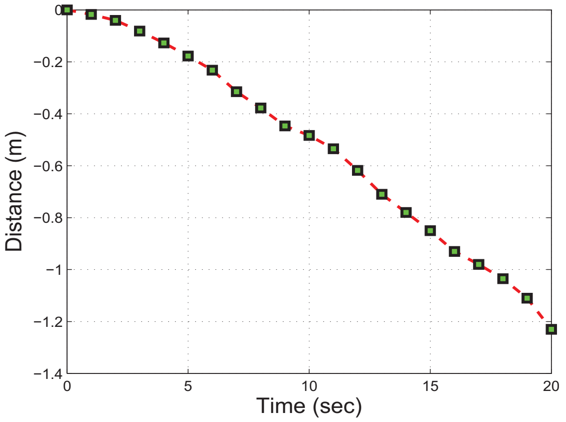
From the results shown in Fig. 6.11 and Fig. 6.12, we see that the CPG generated new swimming pattern can be successfully applied to the robotic fish, and the fish is able to swim forward and backward normally.

Fig. 6.13 shows the distance that the fish has traveled within the preset time, in both forward locomotion and backward locomotion. Since the fish starts from still, it has to accelerate itself to gain a steady speed. Thus, we can see that in the starting phase, the robotic fish swims slowly, and the distance it traveled is comparatively short. After the starting phase, the fish reaches a higher steady speed, and it can travel longer distance in the same period of time. We see that the robotic fish is able to move forward and backward, as expected.

From the two sub-figures of Fig. 6.13, we see that the robotic fish has moved different distances in the same 20 seconds. Specifically, it moves a little further in backward case. The reasons for the discrepancy are twofold. First, the joint angles in Fig. 6.10, which



(a) Distance trajectory of the forward locomotion.



(b) Distance trajectory of the backward locomotion.

Figure 6.13: Distance trajectories of forward locomotion and backward locomotion.

are applied on the robotic fish, are different. The two sets of angles are different not in a way that one can be transformed into another by using CPG properties, but in a way that they are extracted independently from two individual locomotions. Thus, the angles are essentially different in their waveforms. Second, intuitively, since the mechanical structures between the front part and the rear part of the robotic fish are not symmetrical, the movements of forward locomotion and backward locomotion can not be the same.

6.4 Conclusion

This chapter mainly focuses on the locomotion learning for an Anguilliform robotic fish. By using the central pattern generator (CPG) approach, the swimming pattern of a real Anguilliform fish is successfully learned and applied to the robotic fish.

In the beginning, we introduce the structure of the CPG. It is consisted of three parts: the coupled Andronov-Hopf oscillators, the artificial neural network (ANN), and the outer amplitude modulator. Then, the mathematical formulation and detailed discussion is provided for these three parts. For single Andronov-Hopf oscillator, which is the basic element of coupled Andronov-Hopf oscillators, we proofed that the oscillator can converge to a limit cycle. This property means that the steady state of the oscillator is irrelevant with its initial conditions. Also, we discussed the significance of some key parameters, and we find that the oscillation center, the oscillation frequency, and the radius of the limit cycle, and the contraction rate, are all tunable through specific parameters. Moreover, the property of disturbance rejection is verified by a simple simulation. For coupled Andronov-Hopf oscillators, we give the mathematical formulation and the topology. In this part, we illustrate that how each basic oscillator is connected with each other, and why desired phase difference can be produced among different oscillators. Further, we use a three-dimensional topology for our coupled oscillators, analyze its advantage, and demonstrate its better performance and robustness compared with the other two topologies. Also, it can be found that when parameters change, smoother transition can be achieved by coupled oscillators than by common sinusoidal waves. For the ANN, we assign different training inputs for it, corresponding to different locomotion patterns. After the ANN is trained, we can get the desired locomotion patterns by using some specific inputs that we previously assigned. By using the outer amplitude modulator, we can resize the outputs of the ANN in a smooth way, thus obtain the desired amplitudes that we need. After all the three components of the CPG are detailed, we introduce properties of the CPG and give proofs of them. From these properties, we know that the motion pattern generated by CPG can be compressed or stretched along the time axis and the spatial axis, and the phase differences between different outputs are tunable. Next, we extract the locomotion patterns from a real Anguilliform fish, and apply it to the robotic fish. The properties of the CPG are first verified by some numerical examples. Then new pattern is generated, which on the one hand conserves the swimming pattern of a real fish, and on the other hand is more suitable for the robotic fish. The effectiveness of the CPG approach is validated by experiments, leading a result that the robotic fish can successfully perform both forward and backward locomotions.

Chapter 7

Conclusions

7.1 Summary of Results

From a biomimetic perspective, this thesis presents mathematical model, control law design, different locomotion generation, motion library building, locomotion learning based on CPG approach, for an Anguilliform robotic fish.

In the beginning, a links-and-joints based model of Anguilliform fish is established, and hydrodynamic forces are simplified to describe the interaction between the fish and water. Through Lagrangian formulation, the mathematical model of the robotic fish is obtained. This dynamic model reveals the relation between torques added on the fish and movement of the fish. Also, the model is critical for simulating motion of the fish and developing appropriate control methods.

Given the motion dynamics of the fish, torques are developed by using computed torque control method first. Aiming at practical circumstance where parameter uncertainties exist, sliding mode control is proposed to handle the actual system. Numerical results show that the effectiveness of SMC to resist parameter uncertainties, and better tracking performance is obtained compared with that of computed torque control. Considering the chattering phenomenon that exists in the sliding mode control law, a saturation function is used to smoothen the control signals, and its performance is basically the same.

Then, a robotic fish prototype is presented which imitates the shape of an Anguilliform fish. Detailed mechanical design of the robotic fish is given, including the dimensions, the shapes, and the mass distribution of all the links. Based on the previously derived mathematical model, the relations between reference joint angles and three most useful locomotion patterns of the Anguilliform fish – forward locomotion, backward locomotion, and turning locomotion – are explored. It is found that when the former joint has a phase lead compared with the latter joint, the fish moves forward; when the former joint has a phase lag, the fish moves backward; when there exist deflections on the reference angles, the fish makes a turn. The three basic locomotion patterns serve as cornerstones for more complicated motion. The three locomotions are all verified by simulations and experiments, where the results are consistent with each other.

Simulation is also conducted on an eight-link robotic fish. Given reference joint angles which are similar to those given to the four-link fish, the eight-link robotic fish can move normally as well. The result indicates that the previously developed mathematical model and control approach can be successfully applied to robotic fishes with different number of links. It is also found that, the body wave on the eight-link fish is much smoother than that of the four-link fish, which directly results in the higher speed of the eight-link fish. For both of the two fishes, motion libraries are built which contain the relations among the speed, the turning radius and related parameters. The significance of the motion library is that, for practical applications, control parameters of the robotic fish can be conveniently chosen so that desired speed and turning radius can be obtained. Based on the motion libraries, control strategy is designed and applied to the robotic fishes. The simulation results show that the control strategy can effectively handle different tasks. By using real-time feedback of camera, an experiment is conducted, where the robotic fish can track a “U” shape trajectory. Some discussions are given for trajectory tracking of the robotic fish. A conclusion is drawn that exact trajectory tracking can not be realized, since the robotic fish system does not have a simple mapping between the joint space and task space. A feasible way to achieve trajectory tracking is that the original trajectory can be decomposed into a few simple primitive trajectories, and feedback can be used to rectify possible deviations.

By using central pattern generator (CPG) approach, the swimming pattern of a real Anguilliform fish is successfully learned and applied to the robotic fish. The CPG consists of three parts: the coupled Andronov-Hopf oscillators, the artificial neural network (ANN), and the outer amplitude modulator. The coupled oscillators possesses limit cycle property, which means that steady state is irrelevant with initial conditions. Also, the significance of some key parameters is discussed, and it is found that the oscillation center, the oscillation frequency, and the radius of the limit cycle, and the contraction rate, are all tunable through specific parameters. Moreover, the coupled oscillators also possesses property of disturbance rejection. It is demonstrated that a three-dimensional topology of coupled oscillators has better performance and robustness compared with those of the other two topologies. Also, it is found that when parameters change, smoother transition can be achieved by coupled oscillators than by common sinusoidal waves. For the ANN, different training inputs are assigned to it, corresponding to different locomotion patterns. After the ANN gets trained, desired locomotion patterns can be obtained by using specific inputs. By using the outer amplitude modulator, the desired amplitudes are obtained, and the outputs of the ANN can be resized in a smooth way. After all the three components of the CPG are detailed, properties of the CPG are introduced and proofs of them are given. From these properties, it is known that the motion pattern generated by CPG can be compressed or stretched along the time axis and the spatial axis, and the phase differences between different outputs are tunable. Next, locomotion patterns are extracted from the swimming data of a real Anguilliform fish, and applied to the robotic fish. The properties of the CPG are first verified by some numerical examples. Then new swimming pattern is generated, which on the one hand conserves the swimming pattern of a real fish, and on the other hand is more suitable for the robotic fish. The effectiveness of the CPG approach is validated by experiments, leading a result that the robotic fish can successfully perform both forward and backward locomotions which are similar to a real fish.

7.2 Suggestions for Future Work

Past research activities have laid a foundation for the future work. Based on the prior research, the following questions deserve further consideration and investigation.

1. The mathematical model developed in this thesis is a planar (2D) model. For future work, 3D model can be explored. Thus, the robotic fish can not only swim on surface of the water, but also dive into the water.

2. Diving system needs to be implemented and corresponding hardware needs to be designed and installed on the robotic fish. New control laws for depth control needs to be investigated, so that the robotic fish is able to submerge and rise in the water.

3. The tasks given in this thesis are all for single robotic fish. For future work, multiple fishes cooperation and coordination need to be explored. It can be imagined that multiple fishes can achieve much more complicated tasks compared to those conducted

by a single fish. Furthermore, control strategies on the issue of multi-agent needs to be developed for the multiple-fish system.

Bibliography

[1] K. D'Aout and P. Aerts. A kinematic comparison of forward and backward swimming in the eel Anguilla anguilla. *Journal of Experimental Biology*, 202(11):1511–1521, June 1999.

[2] R. M. Murray, S. S. Sastry, and Zexiang Li. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., Boca Raton, FL, USA, 1994.

[3] Bill Gates. A Robot in Every Home. *Scientific American Magazine*, January 2007.

[4] Junzhi Yu, Long Wang, and Min Tan. Geometric optimization of relative link lengths for biomimetic robotic fish. *IEEE Transactions on Robotics*, 23(2):382–386, Apr 2007.

[5] Chao Zhou, Min Tan, Zhiqiang Cao, Shuo Wang, D. Creighton, Nong Gu, and S. Nahavandi. Kinematic modeling of a bio-inspired robotic fish. In *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008.*, pages 695 –699, May 2008.

[6] Kexu Zou, Chen Wang, Guangming Xie, Tianguang Chu, Long Wang, and Yingmin Jia. Cooperative control for trajectory tracking of robotic fish. In *American Control Conference, 2009. ACC '09.*, pages 5504 –5509, June 2009.

[7] 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, April 1999.

[8] Yonghua Zhang, Jianhui He, and K. H. Low. Parametric Study of an Underwater Finned Propulsor Inspired by Bluespotted Ray. *Journal of Bionic Engineering*, 9(2):166–176, Jun. 2012.

[9] Yonghua Zhang, Jianhui He, and Guoqing Zhang. Measurement on Morphology and Kinematics of Crucian Vertebral Joints. *Journal of Bionic Engineering*, 8(1):10–17, Mar 2011.

[10] Phi Luan Nguyen, Van Phu Do, and Byung Ryong Lee. Dynamic Modeling and Experiment of a Fish Robot with a Flexible Tail Fin. *Journal of Bionic Engineering*, 10(1):39–45, Jan 2013.

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

[12] M. J. Lighthill. Large-amplitude elongated-body theory of fish locomotion. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 179(1055):125–138, 1971.

[13] J. Z. Yu, M. Tan, S. Wang, and E. Chen. Development of a biomimetic robotic fish and its control algorithm. *IEEE Transactions on Systems Man and Cybernetics Part B: Cybernetics*, 34(4):1798–1810, Aug. 2004.

[14] K. A. Morgansen, V. Duidam, R. J. Mason, J. W. Burdick, and R. M. Murray. Nonlinear control methods for planar carangiform robot fish locomotion. In *IEEE International Conference on Robotics and Automation, 2001. Proceedings 2001 ICRA.*, volume 1, pages 427 – 434, 2001.

[15] K. A. Morgansen, P. A. Vela, and J. W. Burdick. Trajectory stabilization for a planar carangiform robot fish. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 1, pages 756 – 762, 2002.

[16] K. A. Morgansen, B. I. Triplett, and D. J. Klein. Geometric methods for modeling and control of free-swimming fin-actuated underwater vehicles. *Robotics, IEEE Transactions on*, 23(6):1184 –1199, Dec. 2007.

[17] F. Boyer, M. Porez, and W. Khalil. Macro-continuous computed torque algorithm for a three-dimensional eel-like robot. *IEEE Transactions on Robotics*, 22(4):763 –775, Aug. 2006.

[18] Michael Sfakiotakis and Dimitris P. Tsakiris. Biomimetic centering for undulatory robots. *International Journal of Robotics Research*, 26(11–12):1267–1282, Nov. 2007.

[19] Junzhi Yu, Long Wang, Jinyan Shao, and Min Tan. Control and coordination of multiple biomimetic robotic fish. *Control Systems Technology, IEEE Transactions on*, 15(1):176–183, 2007.

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

[21] Junzhi Yu, Ming Wang, Zongshuai Su, Min Tan, and Jianwei Zhang. Dynamic modeling and its application for a cpg-coupled robotic fish. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 159–164, May 2011.

[22] K. A. McIsaac and J. P. Ostrowski. Open-loop verification of motion planning for an underwater eel-like robot. In *7th International Symposium on Experimental Robotics.*, pages 271–280, Dec. 11-13, 2000.

[23] K.A. McIsaac and J.P. Ostrowski. Experimental verification of open-loop control for an underwater eel-like robot. *International Journal of Robotics Research*, 21(10-11):849–859, Oct.-Nov. 2002.

[24] Junzhi Yu, Lizhong Liu, Long Wang, Min Tan, and De Xu. Turning control of a multilink biomimetic robotic fish. *Robotics, IEEE Transactions on*, 24(1):201 –206, Feb. 2008.

[25] K. H. Low, Chunlin Zhou, and Yu Zhong. Gait Planning for Steady Swimming Control of Biomimetic Fish Robots. *Advanced Robotics*, 23(7-8):805–829, 2009.

[26] K. A. McIsaac and J. P. Ostrowski. Motion planning for anguilliform locomotion. *IEEE Transactions on Robotics and Automation*, 19(4):637 – 652, Aug. 2003.

[27] D. Zhang, Long Wang, Junzhi Yu, and Guangming Xie. Robotic fish motion planning under inherent kinematic constraints. In *American Control Conference*, pages 4135–4140, June 2006.

[28] Jinyan Shao, Long Wang, and Junzhi Yu. Collision-free motion planning for a biomimetic robotic fish based on numerical flow field. In *American Control Conference*, pages 2736–2741, June 2006.

[29] Yongnan Jia, Guangming Xie, and Long Wang. Path planning for robot fish in water-polo game: Tangent circle method. In *Intelligent Control and Automation (WCICA), 2011 9th World Congress on*, pages 730 –735, June 2011.

[30] Y. Hu, L. Wang, J. Liang, and T. Wang. Cooperative box-pushing with multiple autonomous robotic fish in underwater environment. *IET Control Theory and Applications*, 5(17):2015–2022, Nov. 2011.

[31] Yonghui Hu, Wei Zhao, and Long Wang. Vision-based target tracking and collision avoidance for two autonomous robotic fish. *Industrial Electronics, IEEE Transactions on*, 56(5):1401 –1410, May 2009.

[32] Qian Yang, Mei Yu, Shu Liu, and Zhong ming Chai. Path planning of robotic fish based on genetic algorithm and modified dynamic programming. In *Advanced Mechatronic Systems (ICAMechS), 2011 International Conference on*, pages 419–424, 2011.

[33] Auke Jan Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks*, 21(4):642–653, May 2008.

[34] Joseph Ayers, Cricket Wilbur, and Chris Olcott. Lamprey robots. In *In Proceedings of the International Symposium on Aqua Biomechanisms*, 2000.

[35] C. Rossi, W. Coral, J. Colorado, and A. Barrientos. A motor-less and gear-less bio-mimetic robotic fish design. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3646 –3651, May 2011.

[36] Zheng Chen, S. 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, June 2010.

[37] M. Anton, Zheng Chen, M. Kruusmaa, and Xiaobo Tan. Analytical and computational modeling of robotic fish propelled by soft actuation material-based active joints. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009.*, pages 2126 –2131, 2009.

[38] M. Aureli, V. Kopman, and M. Porfiri. Free-locomotion of underwater vehicles actuated by ionic polymer metal composites. *IEEE/ASME Transactions on Mechatronics*, 15(4):603 –614, Aug. 2010.

[39] M. Borgen, G. Washington, and G. Kinzel. Introducing the carangithopter: A small piezoelectrically actuated swimming vehicle. In *Adaptive Structures Material Systems Symp., ASME Int. Congress Exposition*, 2000.

[40] G. Barbera, Lijuan Pi, and Xinyan Deng. Attitude control for a pectoral fin actuated bio-inspired robotic fish. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 526 –531, May 2011.

[41] Scott D. Kelly and Richard M. Murray. Modelling efficient pisciform swimming for control. *International Journal of Robust and Nonlinear Control*, 10:217–241, 2000.

[42] O. Ekeberg. A combined neuronal and mechanical model of fish swimming. *Biological Cybernetics*, 69(5-6):363–374, Oct. 1993.

[43] J.E. Colgate and K.M. Lynch. Mechanics and control of swimming: A review. *IEEE Journal of Oceanic Engineering*, 29(3):660 – 673, July 2004.

[44] John J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[45] Koichi Hirata, Tadanori Takimoto, and Kenkichi Tamura. Study on turning performance of a fish robot. In *Inproceedings 1st International Symposium on Aqua Bio-Mechemics*, pages 287–292, Aug. 2000.

[46] Yi-Ling Yang, P.C.P. Chao, and Cheng-Kuo Sung. Landing posture control for a generalized twin-body system using methods of inputcoutput linearization and computed torque. *IEEE/ASME Transactions on Mechatronics*, (3):326–336.

[47] J.J.E. Slotine and W. Li. *Applied nonlinear control*. Prentice Hall, 1991.

[48] W.J. Cao and J.X. Xu. Nonlinear integral-type sliding surface for both matched and unmatched uncertain systems. *IEEE Transaction on Automatic Control*, 49(8):1355–1360, Aug. 2004.

[49] A.C. Smith, F. Mobasser, and K. Hashtrudi-Zaad. Neural-network-based contact force observers for haptic applications. *Robotics, IEEE Transactions on*, 22(6):1163 –1175, Dec. 2006.

[50] S. M. LaValle. Motion planning. *Robotics Automation Magazine, IEEE*, 18(1):79–89, 2011.

[51] Saroj Saimek and Perry Y. Li. Motion planning and control of a swimming machine. *International Journal of Robotic Research*, 23(1):27–53, 2004.

[52] M. Porez, V. Lebastard, A.J. Ijspeert, and F. Boyer. Multi-physics model of an electric fish-like robot: Numerical aspects and application to obstacle avoidance. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1901–1906, 2011.

[53] D. Zhang, L. Wang, and J. Yu. Coordinated control of two biomimetic robotic fish in pushing-object task. *Control Theory and Applications, IET*, 1(5):1200–1207, 2007.

[54] Junzhi Yu, Lizhong Liu, and Long Wang. Dynamic modeling and experimental validation of biomimetic robotic fish. In *American Control Conference, 2006*, pages 4129–4134, June 2006.

[55] M.S. Triantafyllou, A.H. Techet, and F.S. Hover. Review of experimental work in biomimetic foils. *Oceanic Engineering, IEEE Journal of*, 29(3):585 – 594, July 2004.

[56] K.A. Harper, M.D. Berkemeier, and S. Grace. Modeling the dynamics of spring-driven oscillating-foil propulsion. *Oceanic Engineering, IEEE Journal of*, 23(3):285 –296, July 1998.

[57] I. Delvolve, P. Branchereau, R. Dubuc, and J.M. Cabelguen. Fictive rhythmic motor patterns induced by NMDA in an in vitro brain stem-spinal cord preparation from an adult urodele. *Journal of Neurophysiology*, 82(2):1074–1077, Aug. 1999.

[58] J.G. Cheng, R.B. Stein, K. Jovanovic, K. Yoshida, D.J. Bennett, and Y.C. Han. Identification, localization, and modulation of neural networks for walking in the mudpuppy (Necturus maculatus) spinal cord. *Journal of Neuroscience*, 18(11):4295–4304, June 1998.

[59] P. S. G. Stein, S. Grillner, A. Selverston, and D. G. Stuart. *Neurons, networks and motor behavior*. MIT Press, 1997.

[60] W. Zhao, Y. Hu, L. Zhang, and L. Wang. Design and CPG-based control of biomimetic robotic fish. *IET Control Theory and Applications*, 3(3):281–293, Mar. 2009.

[61] Yonghui Hu, Weicheng Tian, Jianhong Liang, and Tianmiao Wang. Learning fish-like swimming with a cpg-based locomotion controller. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1863 –1868, Sep. 2011.

[62] Wei Zhao, Junzhi Yu, Yimin Fang, and Long Wang. Development of multi-mode biomimetic robotic fish based on central pattern generator. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3891–3896, Oct. 2006.

[63] Daibing Zhang, Dewen Hu, Lincheng Shen, and Haibin Xie. A bionic neural network for fish-robot locomotion. *Journal of Bionic Engineering*, 3(4):187 – 194, 2006.

[64] A. Crespi and A.J. Ijspeert. Online optimization of swimming and crawling in an amphibious snake robot. *Robotics, IEEE Transactions on*, 24(1):75–87, Feb. 2008.

[65] W. Zhao, Y. Hu, L. Zhang, and L. Wang. Design and cpg-based control of biomimetic robotic fish. *Control Theory Applications, IET*, 3(3):281 –293, Mar. 2009.

[66] F.C. Hoppensteadt and E.M. Izhikevich. *Weakly Connected Neural Networks*. Number 126 in Applied Mathematical Sciences. Springer, 1997.

[67] Junzhi Yu, Ming Wang, Weibing Wang, Min Tan, and Jianwei Zhang. Design and control of a fish-inspired multimodal swimming robot. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3664 –3669, May 2011.

[68] Keehong Seo, Soon-Jo Chung, and Jean-Jacques E. Slotine. CPG-based control of a turtle-like underwater vehicle. *Autonomous Robots*, 28(3):247–269, Apr. 2010.

[69] Jian-Xin Xu and Wei Wang. A general internal model approach for motion learning. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 38(2):477–487, Apr. 2008.

Appendix: Author's Publications

Journal Papers

[1] Xue-Lei Niu, Jian-Xin Xu, Qin-Yuan Ren, Qing-Guo Wang. Real-time path planning for an Anguilliform robotic fish using visual feedback. Preparing.

[2] Xue-Lei Niu, Jian-Xin Xu, Qin-Yuan Ren, Qing-Guo Wang. Locomotion learning for an Anguilliform robotic fish using central pattern generator approach. *IEEE Trans. on Industrial Electronics*. Revised.

[3] Xue-Lei Niu, Jian-Xin Xu, Qin-Yuan Ren, Qing-Guo Wang. Locomotion generation and motion library design for an Anguilliform robotic fish. *Journal of Bionic Engineering*. Accepted.

[4] Jian-Xin Xu, Xue-Lei Niu, Qin-Yuan Ren. Modeling and control design of an Anguilliform robotic fish. *International Journal of Modeling, Simulation, and Scientific Computing*, 3(4), 2012.

Conference Papers

[5] Qinyuan Ren, Jianxin Xu, Wenchao Gao, Xuelei Niu. Generation of robotic fish locomotion through biomimetic learning. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 815–821, 2012.

[6] Qinyuan Ren, Jianxin Xu, Xuelei Niu. A GIM-based approach for biomimetic robot motion learning. In *WASA '12 Proceedings of the Workshop at 5th ACM SIGGRAPH Asia*, 97–103, 2012.

[7] Jian-Xin Xu, Xue-Lei Niu, Qin-Yuan Ren, Qing-Guo Wang. Collision-free motion planning for an Anguilliform robotic fish. In *Industrial Electronics (ISIE), 2012 IEEE International Symposium on*, 1268–1273, 2012.

[8] Jian-Xin Xu, Qinyuan Ren, Wenchao Gao, Xue-Lei Niu. Mimicry of fish swimming patterns in a robotic fish. In *Industrial Electronics (ISIE), 2012 IEEE International Symposium on*, 1274–1279, 2012.

[9] Jian-Xin Xu, K. Abidi, Xue-Lei Niu, De-Qing Huang. Sampled-data iterative learning control for a piezoelectric motor. In *Industrial Electronics (ISIE), 2012 IEEE International Symposium on*, 899–904, 2012.

[10] Jian-Xin Xu, Xue-Lei Niu, Zhao-Qin Guo. Sliding mode control design for a Carangiform robotic fish. In *Variable Structure Systems (VSS), 2012 12th International Workshop on*, 308–313, 2012.

[11] Jian-Xin Xu, Xue-Lei Niu, Zhao-Qin Guo. Gait generation and sliding mode control design for anguilliform biomimetic robotic fish. In *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, 3947–3952, 2011.

[12] Jian-Xin Xu, Xue-Lei Niu. Analytical control design for a biomimetic robotic fish. In *Industrial Electronics (ISIE), 2011 IEEE International Symposium on*, 964–869, 2011.