NANYANG TECHNOLOGICAL UNIVERSITY

MODELING AND CONTROL OF SWIMMING GAITS FOR FISH-LIKE ROBOTS USING COUPLED NONLINEAR OSCILLATORS

ZHOU CHUNLIN

SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING

2012

MODELING AND CONTROL OF SWIMMING GAITS FOR FISH-LIKE ROBOTS USING COUPLED NONLINEAR OSCILLATORS ZHOU CHUNLIN 2012

MODELING AND CONTROL OF SWIMMING GAITS FOR FISH-LIKE ROBOTS USING COUPLED NONLINEAR OSCILLATORS

ZHOU CHUNLIN

ZHOU CHUNLIN

School of Mechanical and Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2012

Acknowledgements

First and foremost, I would like to thank Prof. Low Kin Huat, my supervisor, for giving me the opportunity to work with him and for his guidance and insightful comments on my research. The completion of the thesis is achieved through many rounds of discussions with him in lab, in his office or at his home. Sometimes Prof. Low was so devoted and the discussion even lasted until midnight. My most sincere gratitude is given to him for his great help in my four year study at NTU.

I would like to share my achievements with my friends, Mr. Chong Chee Wee, Mr. Zhong Yu, who ever worked together with me in my project. Some results in the thesis are outcomes of exchange of ideas with them. A lot of experiments were conducted with their helps. Mr. Chong finished most of the work on the design and testing of fish robot NAF-II. My friends, Mr. Lim Hup Boon, Ms. Huang Yunyun, Ms. Wang Ping, Mr. Luu Trieu Phat, Miss Pang Wee Ching, Mr. Huang Junyi and all technicians in Robotics Research Center are acknowledged for their supports and listening ears whenever I meet difficulties. Final-year-project undergraduate students who have worked in our robotic fish team are also acknowledged for their contributions.

Gratitude is given to Dr. Michael Lau for his thorough comments on my first year report, Dr. Gerald Seet for providing the research facilities at Robotics Research Center, Dr. Srigrarom Sutthiphong and Mr. Vincent Chai for sharing their knowledge and assistance in water tunnel experiments.

Finally, I want to mention that without the support of my families, I could never move to the current stage. All my achievements also belong to them.

The financial support by research fund NTU-MINDEF-JPP/06/01 and AcRF-RG23/06 is acknowledged.

Zhou, Chunlin

NTU, January 2012

Table of Contents

Acknowledgements .... **iii**

Table of Contents .... **iv**

List of Figures .... **vii**

List of Tables .... **xiii**

List of Symbols .... **xiv**

Abstract .... **xviii**

Chapter 1 Introduction **1**

1.1 Background 1
1.2 Motivation 3
1.3 Objectives, Methodologies and Scope 6
1.4 Outline of the Thesis 7

Chapter 2 Review of Related Works **9**

2.1 Biological Inspirations to the Design of Propulsor 9
2.1.1 Classification of Fish Swimming 9
2.1.2 Fish Swimming Modes 11
2.1.3 Generation of Thrust 14
2.2 Mechanisms of Biomimetic Propulsors and Robotic Fishes 15
2.2.1 Median and/or Paired Fin (MPF) Type Fish Robots 16
2.2.2 Body and/or Caudal Fin (BCF) Type Fish Robots 20
2.3 Fish Robots Developed by Robotic Fish Team of Nanyang Technological University 23
2.3.1 Design of Fish Robots with Parallel Mechanism 25
2.3.2 Designs of Fish Robots with Serial Open Chain Mechanism 34
2.4 Modeling and Control of Swimming Gaits for Fish Robots 38
2.4.1 Kinematics Modeling of Swimming Gait Based on Sine Generators 38
2.4.2 Models of Central Pattern Generators (CPGs) 40
2.4.3 Modeling of CPGs by Using Coupled Nonlinear Oscillators (CNLOs) 43
2.4.4 Control of Swimming Gaits by Using CPGs 44
2.5 Summary 47

Chapter 3 Off-line Generation of Swimming Gaits Using Sine Generators **49**

3.1 Modeling of Swimming Gaits 49
3.1.1 Modeling of Harmonic Swimming Gait 49
3.1.2 Modeling of Non-harmonic Swimming Gait 51
3.2 Analysis of Swimming Performance Based on Lighthill's Elongated Body Theory 55
3.2.1 Lighthill's Elongated Body Theory 55
3.2.2 Application of EBT in Multi-DOF Fish Robots 57
3.3 Experiments of Gait Control on Fish Robots 58
3.3.1 Theoretical Prediction of Swimming Speed 58
3.3.2 Experiments on NKF-II 60
3.4 Summary 63

Chapter 4 On-line Generation of Swimming Gaits Using Nonlinear Oscillators 65

4.1 Gait Generator Based on a Nonlinear Oscillator 66

4.1.1 Generation of Harmonic Swimming Gait Pattern 68
4.1.2 Generation of Non-harmonic Swimming Gait Pattern 70

4.2 Closed-loop Swimming Control: A Case Study of Station-holding 73

4.2.1 Station-holding in Adverse Unsteady Flow 73
4.2.2 Control Laws for Station-holding Control 77

4.3 Simulation of Station-holding Control 80

4.4 Experiment of Station-holding Control 83

4.4.1 Experiment Setup 83
4.4.2 Experimental Results 85

4.5 Summary 89

Chapter 5 Coordination of Swimming Gaits Using Coupled Nonlinear Oscillators 90

5.1 Coupling of Multiple Oscillators: A Model of Central Pattern Generators (CPGs) 91

5.1.1 Coupled Oscillators with Stable Phase Differences 91
5.1.2 Formulations of Coupling Schemes 93
5.1.3 Configurations of Oscillators 97

5.2 Application of the CPG Model in on a Fish Robot with Single Fin Propulsor 100

5.2.1 Configuration of Oscillators for NKF-II 100
5.2.2 Experiments 102

5.3 Application of the CPG Model on a Fish Robot with Paired Fins 105

5.3.1 Configuration of Oscillators for RoMan-II 105
5.3.2 Control of Turning by Tuning the Phase Difference 107
5.3.3 Experiments 109

5.4 Summary 113

Chapter 6 Robust Tracking of Swimming Gaits 115

6.1 PID Controller for Swimming Gait Tracking 116

6.1.1 Motion Control Model of a Biomimetic Propulsor 116
6.1.2 Limitation of PID Controller 120

6.2 Design of Robust Tracking Controller 121

6.2.1 Modeling of the Disturbance 122
6.2.2 Design of a Robust Gait Tracking Controller 123

6.3 Experiments of Gait Tracking Control 129

6.3.1 Control Hardware 129
6.3.2 Experimental Results 131

6.4 Summary 135

Chapter 7 On-line Optimization of Swimming Locomotion 136

7.1 Gait Generation Based on CPGs and Genetic Algorithm (GA) 137

7.1.1 Problem Definition 138
7.1.2 Swimming Gait Generation by Using CPGs 139
7.1.3 Application of GA in Swimming Control 141

7.2 Application of GA in Fast Swimming Mode 143

7.2.1 Choosing of Fitness Function 143
7.2.2 Representation of Population 143
7.2.3 Selection, Crossover and Mutation 144

7.3 Application of GA in Efficient Swimming Mode 144

7.3.1 Measurement of Power in Locomotion of Fish Robot 145
7.3.2 Evaluation of Swimming Efficiency: Economic Perspective 146
7.4 Experimental Results 148
7.4.1 Experiment Setup 148
7.4.2 Performance Testing Based on Offline Approach 150
7.4.3 On-line Searching of Optimal Swimming Speed 152
7.4.4 Optimization of Swimming Energy Efficiency 157
7.5 Optimization of Locomotion for Different Propulsors 161
7.6 Summary 165

Chapter 8 Conclusions and Future Works 168

8.1 Thesis Contributions 168
8.2 Discussions 169
8.3 Future Works 171

Appendix A Design of a PD Tracking Controller for NAF-II 174

Appendix B Modeling of Gaits for Fish Robots with Serial Open Chain Mechanism with Sine Generators 178

Appendix C Specifications of Force Sensors 183

Appendix D Source Codes of GA 186

List of Publications 198

References 200

List of Figures

Fig. 2.1 Terminology used to identify fins in different swimming modes [8]. 10

Fig. 2.2 Relationship between swimming propulsion and swimming functions [32]. 10

Fig. 2.3 Swimming modes identified within BCF propulsion and their example species (adopted from [32]). 11

Fig. 2.4 Gradual changing of BCF swimming movements from (a) anguilliform, through (b) subcarangiform and (c) carangiform to (d) thunniform mode [8]..11

Fig. 2.5 Swimming modes identified within MPF propulsion and their example species (adopted from [32]). 12

Fig. 2.6 Klausewitz's illustration of manta ray locomotion: (a)–(e) the down stroke flapping motion and (f)–(i) upstroke flapping motion (adopted from [38]). 13

Fig. 2.7 The von Kármán vortex street (left) and the wake of a swimming fish that has reverse rotational direction of vortices [8]. 14

Fig. 2.8 Parallel links used to fit the body/fin of MPF swimmers. 17

Fig. 2.9 Fundamental structures of undulatory fins found in amguilliform, gymnotiform and rajiform swimmers respectively [12]. 17

Fig. 2.10 An experimental undulating fin device using parallel bellows actuators [53]. 18

Fig. 2.11 Undulatory propulsor in MPF fish robots. Pictures of prototypes (a)-(h) are adopted from [26, 54-59] respectively. 19

Fig. 2.12 Serial chain of linkages used to fit the body/fin of BCF swimmers. 20

Fig. 2.13 Small-Scale Automatic Mechanical Fish [66]. 21

Fig. 2.14 RoboTuna (left) and its improved version RoboPike (right) developed in MIT in 1994 and 1995 respectively [67-70]. 21

Fig. 2.15 Robotic fish developed in Essex University: photo of the swimming robot (left) and its internal structure (right) [71-73]. 22

Fig. 2.16 Examples of BCF type fish robots. Those pictures of prototypes (a)-(h) are adopted from [12, 74-78] respectively. 22

Fig. 2.17 Prototypes of fish robots developed in Nanyang Technological University in recent years. . 24

Fig. 2.18 Black Ghost Knifefish. 26

Fig. 2.19 Prototype of an Eight-DOF Knifefish Robot: (a) isometric view and (b) front view. 27

Fig. 2.20 The fundamental element of the propulsor: (a) CAD model of the propulsor in an undulatory waveform; (b) a pair of sliders; and (c) the schematics of the two-DOF five-bar slider mechanism. 28

Fig. 2.21 (a) A manta ray and (b) terminologies used to identify its bones, muscles, and fins (adapted from the work by Prof. R. Aidan Martin at the Zoology Department in the University of British Columbia). 30

Fig. 2.22 Scaled diagram of a manta ray. Dimensions of the two fins and body are normalized to the wing span [86]. 30

Fig. 2.23 (a) CAD model of a manta ray robot (the figure shows the internal frame and fin membrane is not covered on fin rays); and (b) fin ray and fin membrane configuration of one side. 31

Fig. 2.24 Overview of the manta ray robot RoMan-II: (a) dimension of key features and (b) isometric view. 32

Fig. 2.25 A simulation of the first-mode vibration of the middle fin ray by using ANSYS Workbench. 34

Fig. 2.26 Frequency response of fin ray in water. The vertical axis denotes the ratio between the amplitude of distal end of fin (A_d) and the excitation frequency of motor (A_e). Horizontal axis is the excitation frequency in Hz. 34

Fig. 2.27 CAD model of a two-DOF BCF type fish robot [81]. 35

Fig. 2.28 Swimming speed test of NAF-II under different values of tail beat frequency (f) and amplitude (A). The maximum speed reaches 0.33 m/s at f = 1.2 Hz and A = 12^o. 37

Fig. 2.29 Type of Tail Fins Tested: Area A: 7711 mm^2, Aspect Ratio: 8.71; Area B: 9906 mm^2, Aspect Ratio: 6.82; Area C: 14096 mm^2, Aspect Ratio: 4.80; Area D: 21082 mm^2, Aspect Ratio: 3.21. 37

Fig. 2.30 Illustration of fish body curve and envelop during swimming (top view). In robotic fish control, the body curve is usually defined with a 2-D travelling wave equation called body motion function [99]. 39

Fig. 2.31 Waveform commonly used by undulatory swimming machines with parallel mechanism designs (a) waveform with linear envelop; (b) waveform with parabola envelops; and (c) waveform with elliptic envelop [12]. 39

Fig. 2.32 Segmental neuronal network that coordinates locomotion: (a) a schematic representation of the brainstem, segmental and sensory components of the neural system that generate burst activity; and (b) a schematic representation of the activity patterns of the different nerve cells and their time relations (adopted from [121]). 42

Fig. 2.33 An application of artificial CPG in controlling of swimming and walking of a salamander robot: (a) the robot prototype and (b) configuration of the CPG model (adopted from [11]). 45

Fig. 3.1 Modeling of waveform on a long undulatory fin: (a) the profile of intersection line between fin surface and the x-y plane is normally modeled as (b) a harmonic wave or (c) non-harmonic wave. 50

Fig. 3.2 Parallel linkage mechanisms used in the fish robots: (a) the parallel connection of fin rays and (b) the structure of single fin ray (rigid link). 50

Fig. 3.3 Non-harmonic waveform generated by Gymnotiform swimmers in forward/backward swimming. 52

Fig. 3.4 The fin rays with different inclined angles. The left unit with β=90° is named the upright ray, while the right unit with β< 90° is called the inclined ray [109]. 52

Fig. 3.5 A curve of non-harmonic oscillation for the i^{th} fin ray. 53

Fig. 3.6 A slender fish model. 55

Fig. 3.7 Swimming speed predicted by using Eq.(3.21) for NKF-II (C = 0.181 m^2). 60

Fig. 3.8 (a) Front view illustration of experiment configuration. The propulsor is in the rest position. Three laser sensors (A, B, C) are used to detect positions. (b) The propulsor prototype hanging in the lab water tank. 61

Fig. 3.9 Comparison of theoretical speed (dashed curves) and experimental speed (solid curves): (a) theoretical calculation is in good agreement with the experimental results in the frequency of 1.0-1.6 Hz and amplitude of 0.03-0.06 BL. The speed and amplitude are normalized in body length. 62

Fig. 4.1 Outputs of the sine generator where amplitude A is changed from 1 to 2 (upper) and frequency f is changed from 1 to 2 (lower). All changes of parameters occur at time t*=2.5s. At other time (t* is random), discontinuity may also appear. 66

Fig. 4.2 An example of unsmooth motions recorded in experiment. The tail fin of NAF-II was driven by a DC motor and controlled with a sine function. The angular position is shown and normalized to 1. The change of frequency is changed from 1Hz to 2Hz at t*. 67

Fig. 4.3 Output of Hopf oscillator. The initial states are set to (0.5, 0.5). The amplitude and frequency are normalized to 1. 69

Fig. 4.4 The phase portrait of the oscillator. The figure illustrates that the system starts evolving from eight different initial states (point 1-8) and they finally converge to the limit cycle. The radius and frequency are all normalized to 1. 69

Fig. 4.5 An example of fin ray motion controlled by Hopf oscillator. State u of the oscillator is used as the control signal, i.e. $\theta(t)=u(t)$. The initial state is the same as point 1 in Fig. 4.4. The radius is changed from 1 to 2 (upper) or frequency is changed from 1 Hz to 2 Hz (lower) at t^*. 70

Fig. 4.6 Non-harmonic gait pattern with adaptive frequencies. 71

Fig. 4.7 (a) Piecewise function and (b) uni-polar sigmoid function. 72

Fig. 4.8 The swimming gait patterns modeled by Hopf oscillator ($\theta=u$). In this example, frequency f=1Hz and τ =40. The amplitude is normalized to 1. 72

Fig. 4.9 A water tunnel apparatus. The observation of fish sample [81] and the robotic fish experiment are conducted in the Test Section. 74

Fig. 4.10 (a) Snapshot of fish swimming in the water tunnel. It swims against water flow at the second stage with about 4 Hz tail beat frequency. These pictures show the case that fish swims in the second stage where the high frequency tail oscillation is performed. (b) The position variation of the sample fish in the adverse flow. The flow rate is 0.2m/s (approximately 1 BL/s). 75

Fig. 4.11 Illustration of a robotic fish swimming against flow. d is the actual displacement. d_r denotes the reference position of the fish. V is the flow velocity. 77

Fig. 4.12 Control structure of station-holding control on multi-DOF fish robots. 79

Fig. 4.13 Overview of the control scheme. The Hopf oscillator generates the periodic tail motion pattern. A feedback control scheme is applied to regulate the frequency and amplitude. The amplitude is also couppled with frequency by a proportional coefficient k_6. Flow rate is a feedforward quantity to the controller. 80

Fig. 4.14 Simulated results of station-holding control. The displacement and the absolute velocity are shown. The parameters in (9) are M=6.8kg, ρ=1000kg/m^3, C_d=0.63 and S_c=0.02 m^2. 81

Fig. 4.15 Evolving of the tail beat frequency，amplitude and the swimming gait control signal. The paramters in (3)-(5) are chosen as k=1, k_1=2.0, k_2=0.8, k_3=25.0, k_4=0.5, k_5=28.0 and k_6=0.02. 81

Fig. 4.16 Simulated swimming gaits for a multi-DOF fish robot. Control parameters are set as: n_λ=1, k=8.0, ε=1.0, k_1=8.0, k_2=4.0, k_3=6.0, k_4=200, k_5=240, and k_6=1.2. 83

Fig. 4.17 Configuration of experiment in water tunnel. 84

Fig. 4.18 Control hardware based on NI-9073 CompactRIO™ real-time controller. 84

Fig. 4.19 Snapshots of the adaptive swimming from t_2 to t_3. The locomotion experienced three stages: I) both controlling parameters increase as the robot is flushed away in t=25-30s; II) swims back in t=30-42s and III) get stable when t>42s. 86

Fig. 4.20 (a) Variation of flow speed in water tunnel; and (b) the response of the station-holding control. The coefficients in (1)-(6) are k=8.0, λ=1.0, k_1=16, k_2=1.5, k_3=12, k_4=28, k_5=105, k_6=1.6, θ_m=0.785rad/s and α_m=2.36 rad/s^2. Note that the flow rate may not be changed sharply as described in the figure. The figure only shows the flow control command. 87

Fig. 4.21 The evolving of the tail beat frequency and amplitude. The time line definition follows Fig. 4.20. 88

Fig. 5.1 Output of two uncoupled (a) and coupled oscillators (b). 92

Fig. 5.2 Single –directional coupling between two oscillators. 93

Fig. 5.3 Illustration of perturbation in direction of phase angle φ. 94

Fig. 5.4 Mutual coupling between two oscillators. 96

Fig. 5.5 Couplings among three oscillators. 96

Fig. 5.6 Outputs of two oscillators with one-directional coupling and different coupling strength: (a) ε=5 and (b) ε=0.8. Other parameters are: k=10, φ_d=-1rad, r=1, and f=1Hz. 97

Fig. 5.7 Evolving of gait transition and phase difference: (a) gait transition under one-directional coupling; (b) gait transition under mutual coupling; and (c) evolving of phase difference. 97

Fig. 5.8 Fundamental topological structures of n multiple coupled oscillators: (a) chain type; (b) radial type; (c) ring type; and (d) fully connected type (number of oscillators is restricted to four). 99

Fig. 5.9 A chain structure CPG model with chain structure and single-directional coupling for NKF-II. 100

Fig. 5.10 Configuration of multiple oscillators with radial structure and single-directional coupling. 101

Fig. 5.11 Control architecture based on CPGs. 102

Fig. 5.12 The LabVIEW program used to control the swimming motion. The user defined sub VI, osc.vi, solves equations of CPG model by fourth-order Runge-Kutta method. 103

Fig. 5.13 The experimental swimming gait control signals generated by the CNLOs (CPG model): (a)-(c) on-line tuning of control parameters and (d) the evolving of gaits for eight driving cranks. The amplitudes are normalized to 1. Parameters in Eq. (5.23)-(5.27) are: n=8, ε=1 and k=10. 104

Fig. 5.14 Swimming speed of NKF-II under control of gait signals given in Fig. 5.13. Amplitude is 0.05 BL (30mm) Negative speed indicates (t > 22s) the prototype performs backward swimming. 105

Fig. 5.15 CPG based motion control structure for RoMan-II. CPG1-3 control the left fin motion and CPG4-6 control the right. For this robot, n=3 and n_λ=0.5. 106

Fig. 5.16 Illustration of thrust forces of a manta ray in steering: (a) forward swimming; (b) backward swimming; (c) turning left; and (d) pivot turning to the left. $F_{T,L}$ and $F_{T,R}$ are thrust forces generated by left side fin and right side fin, respectively. The flapping frequency and amplitude for both sides are assumed identical. 108

Fig. 5.17 Phase difference for the left side fin (a) and right side fin (b) under the control of q. 108

Fig. 5.18 Control hardware hierarchy for RoMan-II. 110

Fig. 5.19 Illustrated experimental results (a) control input f; (b) control input q; and (c) the experimental swimming gait control signals generated by the CPG model. The six signals, $\theta_1 - \theta_6$, are obtained by solving (5.28)-(5.34). The amplitudes are normalized to 1. Other parameters are: τ'=4, τ=20, n=6, α=0.33, ε=2.5 and k=10. 111

Fig. 5.20 Snapshots of RoMan-II in swimming locomotion in one second (f = 1 Hz). The robot performs down stoke flapping during t = 0 s to 0.375 s and upstroke flapping during t = 0.375 s to 0.875 s. Red curves indicate the profile of the left fin edge. 112

Fig. 6.1 (a) Tail fin mechanism of NAF-II and (b) the schematic of tail linkages and the water reactive force (top view of NAF-II). 117

Fig. 6.2 Water tunnel and force measurement configuration [81]. 118

Fig. 6.3 Block diagram of tail motion control with PID controller. 119

Fig. 6.4 Closed-loop gait tracking control. 120

Fig. 6.5 Tracking performances for different gait inputs: tracking response by changing input amplitude (case 2) and influence from change of frequency (case 3). 121

Fig. 6.6 Gait control signal and the water reactive force acting on the caudal fin. 122

Fig. 6.7 Block diagram of the closed-loop system with a robust controller. 126

Fig. 6.8 Response to gait control input with f=1Hz and A=20 º. 128

Fig. 6.9 Response to gait control input with f=2.5 Hz and A=12 º. 129

Fig. 6.10 Control hardware of NAF-II: (a) a master-slave control hierarchy and (b) master controller circuit board. 130

Fig. 6.11 Experiment results of PD controller with theoretical parameters (output 1) and manually adjusted parameters (output 2). 132

Fig. 6.12 Experiment results of PD control (K_p = 47.5, K_d =0.73, f=1.75Hz, and A=12°). 132

Fig. 6.13 Experiment results of the robust tracking: (a) A=12°, f=2.5 Hz and (b) A=12°, f =1.75 Hz. The controller parameters are k_1=0.0081, k_2=5.035, k_3=-81.92, k_4=150.6, and f_s=256 Hz. 134

Fig. 7.1 An on-line swimming control scheme. The on-line motion planning is achived by GA. The robust gait tracking is applied to perform high performance low level actuaotor control, which has been detailed in Chapter 6 and [30]. 137

Fig. 7.2 Illustration of fish body curve and envelop during forward swimming (top view). The dark solid curve represents the midline (or vertebrate) of an undulatory swimmer. The dimensions are normalized to the body length. 138

Fig. 7.3 GA-based closed loop control and configuration of oscillators (CPGs). 140

Fig. 7.4 Flow chart of Genetic Algorithm. 142

Fig. 7.5 Torque measurement at each crank. 147

Fig. 7.6 Flow chart of the optimization process in experiments. 149

Fig. 7.7 Off-line testing results of swimming performance: (a) swimming speed vs. frequency; and (b) energy efficiency vs. frequency. The experiments are conducted in several groups with different amplitudes. Selection of parameters is subjected to the power limitation of motors. 151

Fig. 7.8 Body curve associated to the optimal speed obtained by off-line testing. 152

Fig. 7.9 Tracking the performance of GA in optimization of speed. Red curves represent the evolving of the best individual in 50 generations. Blue curves reflect the averaged performance of all individuals in 50 generations. 154

Fig. 7.10 Body curve associated to the optimal speed: (a) examples of profiles at different time in one undulation cycle (T = 0.439 s); and (b) the curves captured in the experiment at the same time instances. All dimensions are normalized for the propulsor length. The blue lines illustrate the fish body profiles at different time instances. 156

Fig. 7.11 Tracking the performance of GA in optimization of efficiency. The best individual in each generation is plotted in red square. The averaged efficiency index of 10 individuals in each generation is plotted in blue circle. 159

Fig. 7.12 Body curve associated to the optimal efficiency. All dimensions are normalized for the propulsor length. The blue lines illustrate the fish body profiles at different time instances. 160

Fig. 7.13 Experimental results of the speed optimization for (a) the 6-DOF propulsor and (b) the 4-DOF propulsor. The left figures show the performance of GA in the optimization and the right figures show the optimal fin curvatures associated to the best speed at different time instances. 162

Fig. 7.14 Experimental results of the energy efficiency optimization for (a) the 6-DOF propulsor and (b) the 4-DOF propulsor. The left figures show the performance of GA in the optimization and the right figures show the optimal fin curvatures associated to the best speed at different time instances. 163

Fig. A.1 Using graphics to calculate the parameter of controller. 175

Fig. A.2 Closed-loop motion control of the fish tail with a PD controller. 176

Fig. A.3 Step response of gait tracking with a PD controller. 176

Fig. A.4 Block diagram of closed-loop system with fore filter. 177

Fig. A.5 Step response with PD controller and fore filter. 177

Fig. B.1 Fish body curve fitting through rigid linkages. 179

Fig. B.2 A body curve fitting example by rigid body linkages. Three links are employed. All dimensions are normalized to the body length. 181

Fig. B.3 Joint angles in one cycle. 181

Fig. C.1 Geometries of a crank and the bonding of strain gauges. 183

Fig. C.2 Connection of strain gauges (R1-R4) in form of Wheatstone bridge. A precision constant voltage source is applied to drive the circuit. 183

Fig. C.3 Output voltage of force sensors under different balance weights. Circle dots denote data recorded in calibration. Lines are linear fittings of experimental data. y denotes the voltage output and x is the mass of balance weights. 184

List of Tables

Table 2.1 A list of fish robots developed by the Robotic Fish Team of Nanyang Technological University. 25

Table 2.2 Physical specifications of NKF-II. 27

Table 2.3 Physical specifications of RoMan-II. 33

Table 2.4 Physical specifications of NAF-II [95]. 35

Table 3.1 Data of speed collected in experiments 63

Table 5.1 The relationship between the swimming direction and *q*. 108

Table 5.2 Turing radius test of RoMan-II 113

Table 6.1 Parameters of motion control model at nominal supply voltage. 119

Table 6.2 Expressions of coefficients in Eq. (6.39). 133

Table 7.1 Individuals of the first generation in optimization of speed. 153

Table 7.2 Individuals of the 24^{th} generation in optimization of speed. 155

Table 7.3 Initial individuals in optimization of efficiency. 158

Table 7.4 Individuals of the 55^{th} generation in optimization of energy efficiency. 158

Table 7.5 Comparison of the speed optimization for different propulsors. 164

Table 7.6 Comparison of energy efficiency optimization for different propulsors. 165

Table C.1 Specification of strain gauge. 185

List of Symbols

Chapter 2:

x, y, z	Axis of Cartesian coordinates
i, j	Numerators: i, j =1, 2, 3,…
t	Time (sec)
$y = f(x, t)$	Fish body motion function
$f_e(x)$	Envelop function of fish body curve
f, $f(t)$, f_i, $f_i(t)$	Oscillation frequency of gait generators, nonlinear oscillators, CPGs, fish body segments, or driving cranks (Hz)
λ	Length of the wave generated by fish or fish robots (m)

Chapter 3:

A, $A(t)$, A_i, $A_i(t)$	Oscillation amplitude of gait generators, nonlinear oscillators, CPGs, fish body segments, or driving cranks (deg)
φ_d, $\varphi_{d,i}$	Phase difference among gait generators, nonlinear oscillators, or CPGs (deg)
θ, $\theta(t)$, θ_i, $\theta_i(t)$, $\theta_i(x,t)$	Gait control signal or gait control function of fish robots (deg)
T	Period of oscillation motion (sec)
t_m	Peak time (sec)
B_i	Coefficients Cubic Spline
b_j	Coefficients of Fourier Series
$v(x,t)$	Lateral velocity of fin/body segments
F_y	Lateral force
U	Forward steady swimming speed (m/s)
M	Added mass of water slice or mass of a fish robot/propulsor (kg)
H	Height of a propulsor (m)
L	Total length of fish body/propulsor
F_T	Thrust force (N)
F_d	Drag force (N)
$P(t)$	Instantaneous power (W)

ρ	Density of water (kg/m^3)
C	A constant associated to the drag of a swimming fish robot (m^2)
k	Wave number
ω	Round frequency ($\omega=2\pi f$)

Chapter 4:

u, u_i, v, v_i	States of Hopf nonlinear oscillator
k	Positive constant that modulates the speed of convergence of Hopf oscillator
$S(t)$	Frequency switching function
τ	Time constant tuning the speed of frequency switching (s)
d	Displacement of fish robot (m)
d_r	Reference displacement (target position) of fish robot (m)
V	Adverse flow speed (m/s)
Ω_m	Maximum allowable angular velocity of actuator (rad/s)
α_m	Maximum allowable angular acceleration of actuator (rad/s^2)
k_1 - k_6	Coefficients of station-holding controller
S_c	Cross-sectional area of fish robot (m^2)
C_d	Drag coefficient

Chapter 5:

$F(\mathbf{X}_i)$	Nonlinear systems defined by Hopf oscillator
$\mathbf{X}_i=(u_i, v_i)^T$	State vector of oscillators or CPGs
φ, φ_i	Phase angle of gait generators, nonlinear oscillators, or CPGs
r_i	Magnitude of the state vector in polar coordinates
$\mathbf{P}_i=(p_{u,i}, p_{v,i})^T$	Perturbation vector
$(p_{r,i}, p_{\varphi,i})^T$	Perturbation vector in polar coordinates
$\mathbf{e}_{\varphi,i}$	Unit tangential vector to limit cycle
ε	Coupling strength

m	Number of waves generated by fish or fish robots
n	Number of oscillators, linkages, body segments, driving cranks, CPGs
q	Steering controlling parameter
τ'	Time constant tuning the speed of direction switching

Chapter 6:

F_R	Reactive force from water (N)
V_s	Sweeping speed of fish tail (m/s)
S	Area of the caudal fin (m^2)
b	Linear damping coefficient (N s/rad)
R	Distance between the joint 1 of NAF-II and the caudal fin (m)
G	Gear ratio
T_R, $T_{R,i}$	Torque caused by water reactive force on the propulsor
T_w	Equivalent resistance torque on the motor shaft (N m)
T_m	Motor output torque (N m)
K_m	Torque constant of motor (N m/A)
I	Electric current through the motor armature (A)
J	Equivalent moment of inertia of the tail mechanism (kg m^2)
V_{in}	Supply voltage (volt)
K_e	Back electromotive force coefficient (V s/rad)
L_m	Inductance of motor (H)
R_m	Resistance of motor armature (Ω)
A_R	Magnitude of the water reactive force (N)
φ_R	Phase difference between the water reactive force and the gait control signal (deg)
θ_d	Reference gait signals (deg)
$\mathbf{\Theta}$	State vector of the tracking control system
$\mathbf{A}$	State matrix of the tracking control system
$\mathbf{B}$	Input matrix

B'	Disturbance matrix
y	System output
C	Output matrix
e	Tracking error
s	Laplace operator
z	z operator
$u(s)$, $u(kT)$	Feedback control signals
f_s	Sampling frequency or sample rate (Hz)

Chapter 7:

l_i	Length of driving crank or fin ray (m)
FIT	Fitness function
P_u	Effective power or useful power (W)
P_t	Total power consumption
P_l	Power loss in motion transmission from the actuators to the propulsor
P_p	Power used to drive the propulsor itself
P_w	Power wasted in the wake
η	Power efficiency
I_E	Ratio between the swimming speed and averaged power (speed per unit power, m/W s)

Note: all undefined symbols in a chapter follow the definitions given in previous chapters unless otherwise specified.

Abstract

Fish have perfect body mechanisms and swimming modes for underwater locomotion, and provide ideal models for the propulsor design of fish robots. This thesis investigates modeling and control of swimming gaits for multiple degree of freedom (DOF) fish-like robots. An experimental-based approach is conducted, i.e. online generation and tuning of swimming gaits for fish robots according to its locomotion performance.

The online modeling aims to achieve fish-like swimming locomotion on robots, and optimal swimming gaits associated to fast swimming speed and high energy efficiency of locomotion. The investigation consists of two parts. Firstly, the online gait generators are formulated by using artificial central pattern generators (CPGs). Secondly, the gait is tuned by using a closed-loop swimming control system incorporated with the CPG model. In order to obtain diversified gait patterns and gait transitions, different structure and coupling ways of CPGs are discussed so that feedback control laws can be imposed on the CPGs to shape the swimming pattern. The parameters of the oscillators are tuned as fish robots swims, without causing jerks or losing coordination of motions. The implementation of the system includes the combination of CPGs and closed-loop swimming control, which is initially applied in a special case of adaptive swimming: the station-holding control in adverse unsteady flow. Subsequently, the system is utilized, together with genetic algorithm (GA), to produce fast or energy-efficient swimming gaits for multi-DOF fish robots.

Three types of fish robots have been deployed as experimental platforms for the research work here. Experiments are conducted to assess the CPG-based gait generation and the closed-loop swimming control system, including sine function based gait generators, robust gait generation and station-holding control of fish robots, multi-DOF swimming gait coordination and gait transition by using artificial CPGs, robust swimming gait tracking control, and online optimization of swimming performance by using GA and CPGs.

Chapter 1 Introduction

This thesis presents the research work on modeling and control of swimming gaits for multiple degree-of-freedom (DOF) fish-like swimming robots. Focus is the on-line swimming gait generation by using coupled nonlinear oscillators (which are also regarded as model of artificial central pattern generators) and on-line optimization of swimming performance by combining Genetic Algorithm (GA) and CPG model. The research background, motivation, and outline of the thesis will be discussed in this chapter.

1.1 Background

The advent of new technologies in biology, materials and robotics brings the possibility to the realization of biomimetic propulsion and fish-like swimming robots [1-3]. Such a robot is a special type of Biologically-inspired Underwater Vehicle (BIUV) that obtains thrust by imitating behaviors of aquatic animals [4]. BIUVs are driven by biomimetic fins, flippers, or bodies rather than conventional screw propellers. Similar as the conventional Autonomous Underwater Vehicles (AUVs), the BIUV systems have a wide-range of applications in many different areas: marine sourcing, seabed charting, military surveillance, environmental assessments, sea exploring, uncovering of mines and conducting of scientific research, etc. [3, 5]. Besides, the BIUVs have their unique properties superior over the conventional AUVs, especially in propulsion efficiency aspect. Among locomotion patterns of underwater animals, the swimming of fish attracts extensive interests of research [2].

Over millions of years of evolution and natural selection, fishes have developed perfect body mechanisms and swimming modes for underwater locomotion. It has been reported that the swimming efficiencies for most of the fish species is higher than 80% [6]. Some thunniform fishes can even swim with efficiency more than 90%, while the average efficiency of the current screw propellers are located between 40% to 50% [6]. Additionally, fishes can make turning with a turning radius less than 10% to 30% of their body length and maintain the high speed at the same time. This remarkable talent is far beyond the capability of any current ships whose turning radius is usually much bigger than its hull length and the turning speed must be less than the half of the normal cruising speed [6].

Compared to the screw propellers, the motion of fins or bodies of fish can provide more flexible maneuverability, which can be used to achieve fine posture adjustment for underwater robots [7]. These abilities inspire innovative designs which improve the ways that man-made systems operate in and interact with the aquatic environment [8]. The underwater ecology is also an important concern in the study of the BIUVs especially when marine life has been deteriorating due to the extensive use of propellers which produce large bandwidth noises in the wake. The swimming patterns of fish generate noise-free thrust. This consideration also pushes engineers to innovate by constructing vehicles without rotary propellers [9].

Fish provide ideal models to the biomimetic propulsor design of swimming machines [5, 6, 10]. Over the past two decades, the interests on robotic fishes have experienced an increasing trend. In the robotics community, the research on the fish robots has a straightforward objective: to transform the conceptual design of the biomimetic fish into novel underwater vehicles that can serve humans. The accomplishment of this objective requires studies in many aspects: the mechanical design of the fish robots, materials of biomimetic propulsors, actuation and actuator designs for underwater environments, the sensors and electronic systems for underwater measurements, control of swimming for highly efficient locomotion, intelligent control strategy for autonomous manipulations, etc [2, 3, 11, 12].

In this thesis, the kinematics modeling and control of swimming gaits for multi-DOF fish-like swimming robots will be discussed. Modeling and control of fish-like robots include several different levels. High level control concerns about the problem of navigation, obstacle avoidance, path following, swarm, etc; the middle level addresses the issue of swimming motion planning and swimming gait generation by using kinematics models; the low level control of swimming implements planned swimming gaits on actual fish prototypes which ensures that the mechanical propulsor can precisely track those control signals. The thesis presents the research work covering the last two categories: the kinematics modeling of swimming gaits, on-line generation and tracking control of gaits for multi-DOF robotic fish systems. In the thesis, the multi-DOF refers to those coordinated and involved in thrust generation. Each DOF corresponds to an actuator running with a continuously oscillatory pattern (some special cases such as pectoral fin in BCF type robot which sometimes runs in a non-continuous mode like a rudder to control the posture of the robot [13, 14] will not be the issue of this research).

The purpose of the modeling in this study is not only to achieve fish-like swimming motion on robots, but also to find optimal swimming gaits that can enhance the locomotion performance in terms of swimming speed and efficiency. Here the term *gait* is borrowed from the legged locomotion or robotic arms to define the control input for the actuator that drives an individual robotic body element involved in thrust generation. In essence, the motion planning and gait generation for a multi-DOF robotic fish is not much different from the classical planning problem in robotic arms dealing with the issue of trajectory following versus gait generation. However, the former has its own specialties: 1) the motion should be coordinated in order to produce propulsion for the robot in water environment; and 2) such motion pattern should yield an energy-efficient locomotion since increasing the efficiency of underwater vehicles is a major goal in development of robotic fishes.

1.2 Motivation

From a robotics perspective, enhancement of the energy efficiency of locomotion is still the kernel issue in the study on robotic fishes even if current research interests extend to diversified topics such as application of smart materials, new actuation methodology and micro swimming robots, etc [12, 15]. Finding optimal swimming patterns for a fish robot is a complex problem due to the unknown hydrodynamics and physical properties of actual robots. Biomimetic methods are traditionally adopted to obtain fish-like motions on swimming machines. Observations on real fish's swimming motion provide reference systems, the body motion functions which are used to generate swimming gaits for multi-DOF swimming robots. Then the research will focus on finding a suitable body motion function that can produce optimal locomotion. The disadvantage is that the machine swimming like a real fish in shapes does not guarantee it achieves the same high efficiency [5]. Another solution is to conduct large amounts of experiments and find empirical results to refine the body motion functions. Stochastic searching techniques such as Genetic Algorithm are applied in optimization of control parameters [16-18].

Direct theoretical analysis of hydrodynamics, computation fluid dynamics (CFD) simulation or indirect analyzing of flow by measuring the vortices shed in the wake of swimming fish have been studied for years to explain the secrets of highly efficient fish swimming. The results help to supervise the locomotion control of fish robots [2, 8]. In-depth understanding of swimming hydrodynamics is always preferred because it provides guideline to find the optimal system parameters. With the advancement in computing technology, simulation of swimming becomes available through Computational Fluid Dynamics (CFD) tools, which helps to find the relationship between swimming performance and its motion patterns [16]. Recently, the theories about wake structure and vorticity control and measurement technologies based on Particle Image Velocimetry (PIV) provide more direct way of analyzing the water flow around fish body, which effectively guides the design of swimming control systems [17-21].

In the present research, the intention is not to develop deterministic theories that can provide a full explanation of swimming dynamics of fish robots, but solve the problem by using experiment-based approaches, i.e. through on-line processing of performance data of a fish robot to find optimal swimming patterns. For a given fish robot, digging its ultimate potentials in the locomotion is of more value for real-world applications. The intention of on-line swimming optimization of swimming locomotion is stem from this consideration. The term *on-line* means that the swimming motion is not predefined but autonomously generated by the control system as the robot runs. It also requires that real-time feedback of the locomotion performance can be included in the swimming gait generation.

Current robotic fish systems suffer from a common problem: a closed-loop control scheme for swimming machines is not well established yet due to the absence of real-time feedback of water flow information around the fish body [5]. Without this closed-loop control, the swimming gait can hardly be adaptive to the flow. Coordination of swimming motions for multi-DOF fish robot is also an issue to be addressed. Coupled nonlinear oscillators (CNLOs) will be used to perform the on-line swimming gait generation and coordination of motions. The CNLOs are also regarded a model of artificial Central Pattern Generators (CPGs). Biological CPGs are entities found in the spinal cord of vertebrates or in some invertebrates [15, 22, 23]. They produce rhythmic motion patterns and coordinate the moving elements of bodies, limbs, fins, or wings for locomotion of animals [24-26]. The CPG model can be applied in control and coordination of locomotion for multi-DOF robots [25, 27, 28].

The CPG based approach is adopted for four reasons: (i) CPGs produce stable rhythmic patterns which are ideal for controlling the oscillation or undulation of swimming robots; (ii) the gait patterns produced by CPGs are robust to reject the external disturbance; (iii) the mechanism of interlinked CPG neurons provides a solution to coordinate motion of multiple-joint robots and (iv) the most attractive property for the present research, CPGs are ideally suited to integrate feedback signals into differential equations, which provide opportunity to obtain mutual entrainment between the CPG and the robot to achieve adaptation of swimming gait to water flow [15]. With the sensory feedback from robot-water interaction, locomotion with optimal energy consumption can be expected [29]. A closed-loop locomotion control system will be developed for robotic fishes [15] and Genetic Algorithm (GA) will be applied to refine the parameters of CPG based control system. The proposed CPG model for locomotion is expected to generate gait patterns adaptive to the surrounding water flow [29].

By virtue of the above-mentioned works, the proposed research will contribute to the performance enhancement of swimming robots in terms of swimming speed, energy efficiency, and maneuverability. In long term, the methodology developed in this research is expected to be applied on other types of robots to obtain natural and energy-efficient locomotion.

1.3 Objectives, Methodologies and Scope

This research aims to obtain a model which can on-line generate swimming gaits for multi-DOF robotic fish systems by using CNLOs. Together with Genetic Algorithm optimizing the system parameters, a closed-loop swimming motion control system with feedbacks from robot-water interactions will be also developed to produce optimal swimming patterns in terms of swimming speed or energy efficiency. In order to achieve the specified objectives, the following works have been accomplished:

1. Several robotic fishes for specified task requirements have been developed to achieve fish-like locomotion. Offline motion planning based on sine generators has been conducted and tested on different types of fish robots.
2. The applications of nonlinear oscillator in generation of robust swimming gaits were investigated. A special case of adaptive swimming, the station-holding performance of real fish is implemented on a fish robot. The feedback control laws that tune the parameters of nonlinear oscillators were derived.
3. Multiple coupled nonlinear oscillators were adopted to build artificial CPGs and model the kinematics of multi-DOF swimming gaits. The structure, equations and coupling schemes of the CPG model were detailed. The applications of the CPG model in different fish robots are discussed and tested in experiments.
4. A robust tracking controller was applied in tracking the gait signals produced by CPGs.
5. Genetic Algorithm was adopted as an optimization tool to on-line tune the parameters of CPGs in the closed-loop control system. The application of GA in fish robots was studied. An efficiency measurement method based on feedback of water reactive force exerted on fish robots was developed to evaluate the swimming efficiency. The fast swimming mode and energy-efficient swimming mode of a multi-DOF fish robot were found in experiments.

1.4 Outline of the Thesis

There are eight chapters in the thesis. Following the general introduction to the present research in this chapter, the literatures in related areas are reviewed in Chapter 2. The contents of literature review include three major parts: theories regarding the kinematics and dynamics of fish swimming, designs of biomimetic propulsors and fishes, modeling and control of swimming motion by using nonlinear oscillators or CPGs. The designs of fish robot prototypes used in the present research are also introduced in this chapter.

Modeling of swimming gaits for multi-DOF fish robots based on sine functions is discussed in Chapter 3. The highlight is to derive analytical gait control functions according to the topological structures of biomimetic propulsors. Also, an analytical method for predicting the swimming performance of fish robots is discussed in the chapter. Experiments were conducted on a fish robot to validate the theoretical analysis.

Gait generation by using a nonlinear oscillator will be discussed in Chapter 4. The property of the oscillator is analyzed. As an application of the robust swimming gait generation, the station-holding performance of real fish is implemented on a fish robot. Feedback control laws are developed to tune the parameters of the gait generator. Experiments are provided to show the effectiveness of the control method.

In Chapter 5, the coupling schemes for multiple nonlinear oscillators are developed, based on which an artificial CPG model used to coordinate swimming motions will be formulated. The couplings schemes and the topological structures of multiple oscillators are the focuses. Applications of the CPG model are illustrated on an eight-DOF knifefish prototype with a single fin propulsor and a six-DOF manta ray robot with two sided flapping fin propulsors respectively.

Robust swimming gait control includes two aspects: robust gait generation and gait tracking. The latter is discussed in Chapter 6. Usually PID controllers are adopted as this motor-level controller [5, 15, 30]. However, it was found in experiments that precise tracking of the gait signals in water environment is hardly obtained due to the un-modeled hydrodynamics. A robust tracing system is developed to solve this problem.

Genetic Algorithm in optimization of swimming locomotion together with CPGs in generation and coordination of swimming gaits for multi-DOF fish robots is explored in Chapter 7. It is aimed to find optimal body curves which are associated to optimal swimming performance. An efficiency measurement method based on feedback of water reactive force exerted on the fish robot is developed to evaluate the swimming performance. The feedback is used to establish a closed-loop swimming control system. The eight-DOF biomimetic propulsor is used to test the optimization method. Two objectives are achieved: the optimal swimming speed and optimal energy efficiency.

The last chapter, Chapter 8, is a summary of the whole work discussed in the thesis. A plan of the future search is also proposed in this chapter. Four appendixes are provided as supplementary materials of the thesis. The contents include: design process of a PD controller for gait tracking control in Appendix A, gait modeling for fish robots with serial open chain mechanisms in Appendix B, technical details of force sensors in fish robots in Appendix C, and source codes of GA Appendix D.

Chapter 2 Review of Related Works

The research on robotic fishes is related to many different areas like biology, robotics, fluid mechanics, mechatronics, material, control science, etc. In this chapter, works that directly contribute to the present thesis are outlined. Those works are briefly introduced in three parts. In Section 2.1, the inspirations from fish anatomy and morphology to the bionic fish design will be reviewed. The current designs of fish robots will be discussed in Section 2.2. The designs of fish robot prototypes for the present research will be introduced in Section 2.3. The work on modeling and control of swimming gaits for multi-DOF fish robots, especially the coupled nonlinear oscillator and CPG based approaches will be outlined in Section 2.4.

2.1 Biological Inspirations to the Design of Propulsor

Different communities in science and engineering are involved in the study of fish-like BIUVs, i.e. robotic fishes. Among them, the works from fields of biology, fluid mechanics, hydrodynamics and robotics make major contributions to this area. Terminologies and nomenclatures in biology routinely transplanted and/or totally accepted by engineers in design of robotic fishes will be introduced in this section. And the inspirations from fish anatomy and swimming modes to the design of bionic propulsors will be discussed.

2.1.1 Classification of Fish Swimming

Biological fins and body structure of fishes are good models for engineers in designing the bionic propulsor. Fig. 2.1 illustrates the terminology used to identify morphological features of fish, as it is most commonly found in literature and used throughout this text. The fin dimensions normal and parallel to the water flow are called span and chord, respectively [8]. Breder [31] presents a set of nomenclatures classifying fish into several basic types according to the body structure and swimming pattern. The classification of fish is based on two main factors: (1) the extent to which propelling process is based on undulatory motion versus oscillatory motion and (2) the body structures or fin segments that are most active in generating propulsion. Fish swimming is divided into body and/or caudal fin (BCF) locomotion and median and/or paired fin (MPF) locomotion. This classification consists of two levels: the first level characterized by BCF or MPF describes what may be called the swimming *type* and the second level describes the swimming *mode* for each type.

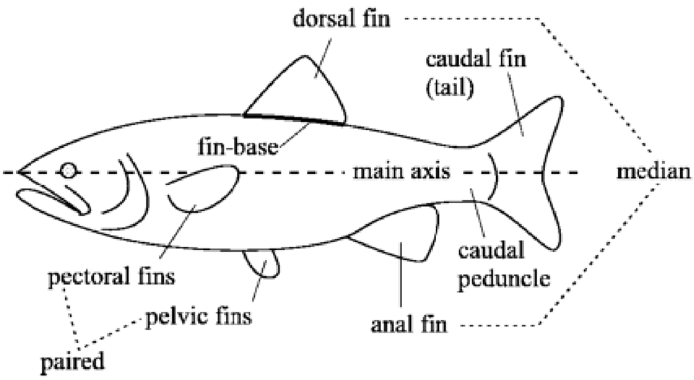


Fig. 2.1 Terminology used to identify fins in different swimming modes [8].

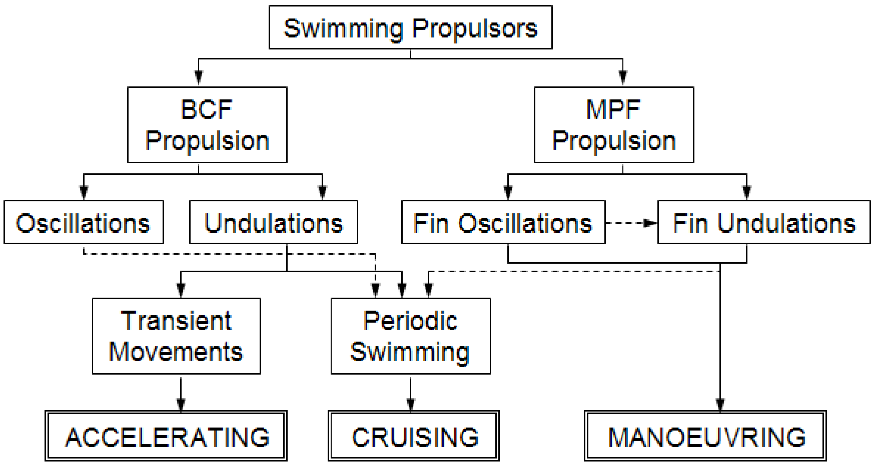


Fig. 2.2 Relationship between swimming propulsion and swimming functions [32].

Based on this classification, Lindsey [32] and Webb [33] summarized the internal links among the types classified by Breder. Further, Webb [34] presented a more detailed and comprehensive classification which categorizes the fish swimming into BCF periodical propulsion, BCF transient propulsion, MPF propulsion and MPF casual swimming or no swimming. Fig. 2.2 shows the relationship between swimming propulsion and swimming functions, which is given by Lindsey [32]. Although the classification is questioned for being oversimplified in biology area, the nomenclatures are widely accepted by researchers in engineering areas [8]. For biologically inspired underwater vehicle, Berder's classification is sufficient to provide information for the design of the bionic propulsor.

2.1.2 Fish Swimming Modes

(a) Swimming Modes with BCF Propulsion

Most of fish families obtain propulsion through BCF swimming. This swimming type is characterized by five modes: anguilliform, subcarangiform, carangiform, thunniform and ostraciiform. Fig. 2.3 depicts the five swimming modes identified within BCF propulsion, based on the degree of body undulation.

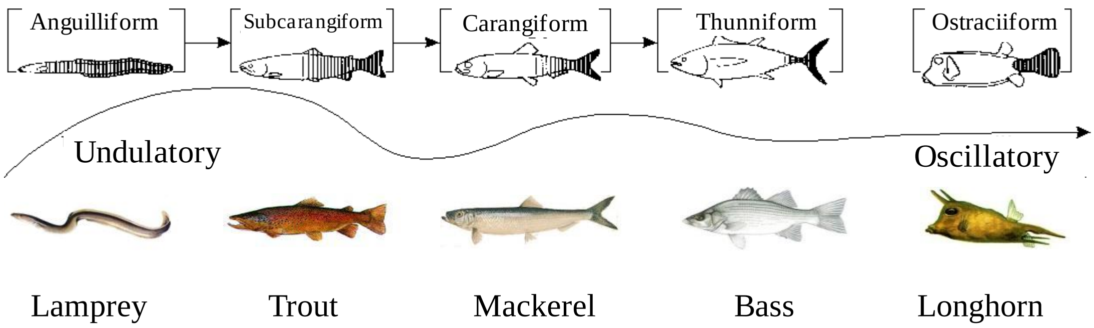


Fig. 2.3 Swimming modes identified within BCF propulsion and their example species (adopted from [32]).

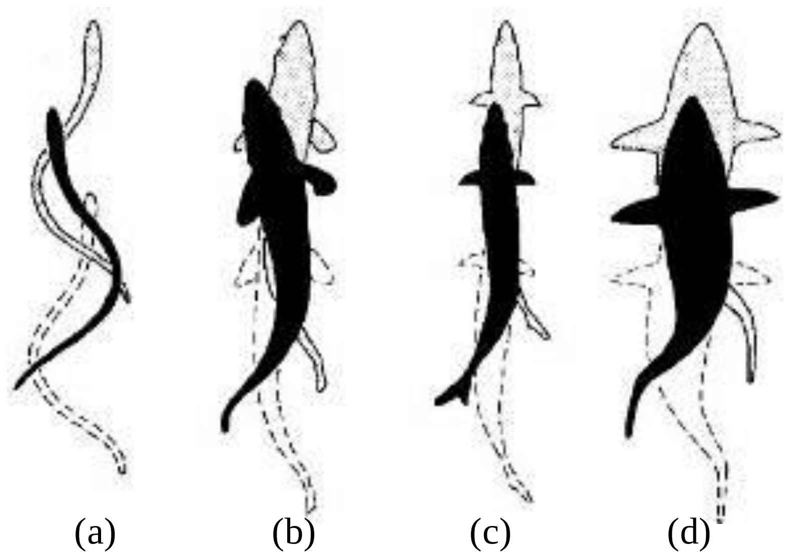


Fig. 2.4 Gradual changing of BCF swimming movements from (a) anguilliform, through (b) subcarangiform and (c) carangiform to (d) thunniform mode [8].

The swimming modes involved in BCF swimming may be more than one type, either at the same time or at different speeds [8]. Fig. 2.4 illustrates the gradual changes of swimming modes from purely undulating (anguilliform) to purely oscillation (ostraciiform). These varying swimming modes come along with distinct body structure whose active portion involved in propulsion generation is different [2, 8].

(a) Swimming Modes with MPF Propulsion

Fig. 2.5 shows the specific swimming modes identified within MPF propulsion and the typical species of each kind, based on the (extended) classification scheme proposed by Breder. Undulatory swimming modes are also used by many fish as the major way of propulsion in steady swimming [18].

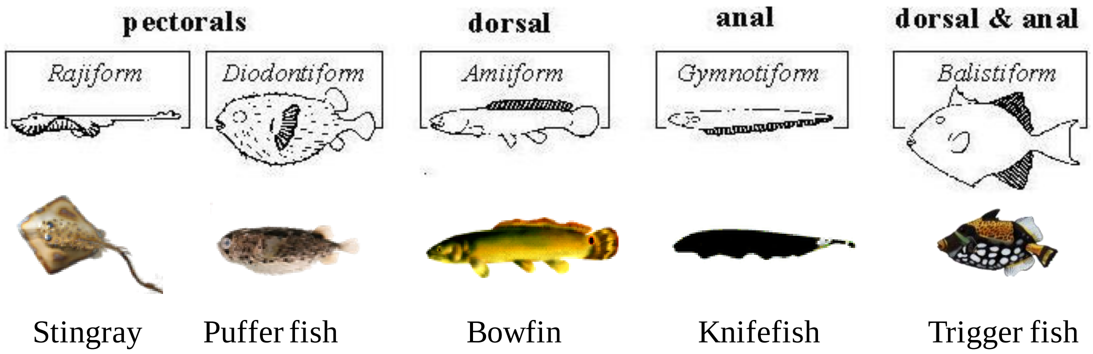


Fig. 2.5 Swimming modes identified within MPF propulsion and their example species (adopted from [32]).

Among those types of fish, a significant number of MPF swimmers obtain thrust through the purely undulatory fin motions. An estimated 15% of the fish families use non-BCF modes as their routine propulsive means [8]. And a significant number of fish typically relying on BCF modes for propulsion employ MPF modes for maneuvering and stabilization. Median or paired fins are normally used as the propulsor. Such fins consist of a set of muscles and bones that can produce backward traveling wave toward the opposite direction of heading during steady forward locomotion. They are also used for highly flexible maneuvering and stabilization in transient movements. A remarkable feature of this type is that MPF swimmers using long fin as propulsor can generate waves of opposite direction and hence swim backward without turning of the whole body. Like BCF propulsion, they can also generate adequate thrust to be used for locomotion, but at generally low speeds compared with that of the BCF type [8].

(c) Swimming Mode of Manta Ray

Manta ray (manta birostris) use median paired fins to swim. They are the largest species of ray's family, classified under class chondrichthyes, along with skates and sharks [35]. Manta ray could be recognized from its diamond-shaped flat body. Brower used several sequentially posed pictures of manta ray to produce the quantified dimensions of manta ray body [36]. The motion exhibited by manta rays is not undulating as stingrays or other types of rays. Similar to cownose, eagle and bat ray, manta ray is considered using flapping pectoral fin to swim [37]. It exhibits rajiform motion during its swimming across the ocean. A full sequence of manta ray locomotion in both downwards and upwards flapping is illustrated in Fig. 2.6.

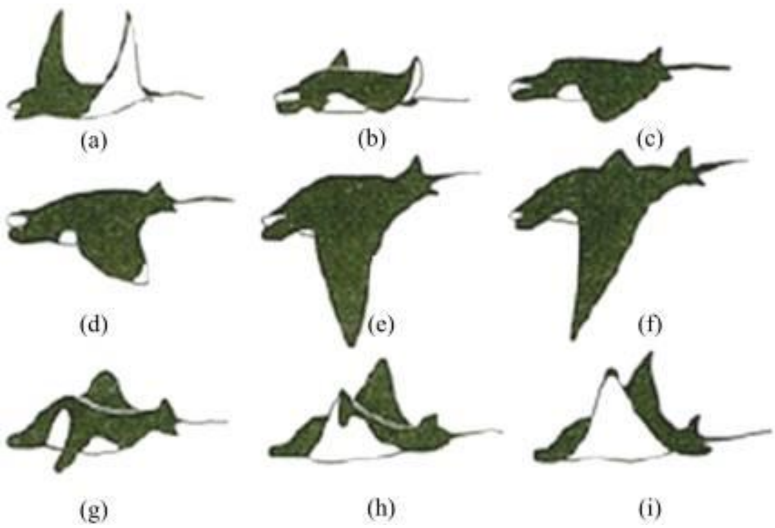


Fig. 2.6 Klausewitz's illustration of manta ray locomotion: (a)–(e) the down stroke flapping motion and (f)–(i) upstroke flapping motion (adopted from [38]).

Klausewitz [38] divided the structure of manta ray's pectoral fin into basal row, the first third of the fin adjacent to the body, and distal row, the rest of the fin up to the tip. The locomotion of manta ray can be qualified as illustrated in Fig. 3: (a) Distal part is at its extreme upwards orientation, basal part relatively stays horizontal; (b)–(d) distal part is pulled down gradually, basal part is kept approximately horizontal; (e) Completion of downward flap, distal part nears its extreme downward orientation, basal part bends slightly downward; (f) The distal part still nears its extreme downward orientation, basal part immediately starts to move upward, pulling the distal after it; (g)–(h) Distal part is pulled up gradually, basal part is kept approximately horizontal; (i) Completion of upward flap, distal part approaches its extreme upward orientation, basal part bends slightly upward. This analysis is further investigated by the motion study conducted by Brower[36]. In his work, he obtained the front profile of a swimming manta ray. The profiles show that the upstroke only takes about half time of down stroke [36].

2.1.3 Generation of Thrust

It is commonly accepted that the propulsive force of swimming is highly related to the mechanism of momentum transferring between fish and water interaction [10, 39, 40]. One of the effective ways to study this mechanism is to study the vortex caused by fish swimming. Vortices in the wake have a reversed rotational direction compared to the von Kármán vortex street, which is observed in the wake of stationary objects such as cylinders or airfoils [8, 41] (Fig. 2.7).

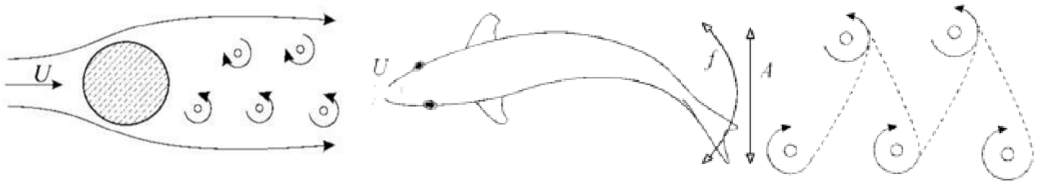


Fig. 2.7 The von Kármán vortex street (left) and the wake of a swimming fish that has reverse rotational direction of vortices [8].

Although the mechanism of the vortices generation is still unclear, the observed phenomenon, named reverse von Kármán street, appears to be tightly associated with thrust generation [8, 42]. It has been demonstrated, that fish uses two different modes of vorticity control in straight-line swimming to optimize performance by utilizing body-generated vortices [43]. The constructive mode employs a vortex reinforcement scheme, whereby the oncoming body-generated vortices are repositioned and then paired with tail-generated same-sign vortices, resulting in a strong reverse Kármán street, and hence increased thrust force. The destructive mode, in contrast, employs a destructive interference scheme, in which the body-generated vortices are repositioned and then paired with tail-generated opposite-sign vortices, resulting in a weakened reverse Kármán street, thus extracting energy from the oncoming body-shed vortices and increasing swimming efficiency [43].

The vortexes in the wake of fish swimming can be measured by instruments. The well-developed Computational Fluid Dynamics (CFD) software provides powerful tool to analyze and predict these vortices. The deforming body of fish interacting with the surrounding fluid is described by the incompressible Navier-Stokes equations. CFD software is used to solve the Navier-Stokes equations completely. The principle of CFD is based on finite difference method and finite volume method. Both attempt to change differential equation to discrete equations [44-46]. It helps to simulate the vortex generated in the swimming from which information of flow filed is obtained and hence helps to understand the dynamics of fish-water interaction [16, 18-20, 47-52]. Vortex can also be generated in a pulsatile way as exhibited by squid [7]. The thrust generation by those biorobotic devices is believed to be useful to improve the low-speed maneuvering and station-holding capabilities of traditional propeller driven underwater vehicles [7].

As computational tools, the CFD and DPIV are not suitable for the design of control scheme for the control of bionic propulsor, especially for the design of a real-time controller. However, the results from these technologies provide quantitative information about fish-flow interaction and therefore help in the modeling of aquatic locomotion.

2.2 Mechanisms of Biomimetic Propulsors and Robotic Fishes

From mechanical viewpoints, it is feasible to model vertebras of slender fish body or bones of its flexible fins as short rigid links. Subsequently, the body or fin of a robotic fish can be schematically represented as mechanical linkage systems connected by means of revolute joints. A robotic fish contains limited number of such links, which renders it an approximation of real fish. The more the links of a robotic fish has, the more approximation can be achieved. As a result, more degrees of freedom incorporated in the design of robotic fish would be preferred for flexible moving performance. There are two main topological structures in the mechanical design: the parallel linkage mechanism and planar serial open chain mechanism.

A good review of biorobotic AUVs can be found in [4]. In this thesis, the multi-DOF refers to those coordinated and involved in thrust generation. Each DOF corresponds to an actuator running with a continuously oscillatory pattern. Some other patterns such as using of pectoral fins in BCF type robot which sometimes runs in a non-continuous mode like a rudder to control the posture of the robot [14] is not the scope of the thesis. The fish robots typically using oscillation or undulation of body fins as major way of propulsion will be reviewed.

2.2.1 Median and/or Paired Fin (MPF) Type Fish Robots

The development of biomimetics, bio-fluid mechanics, intelligent control, mechatronics and new materials make it realizable for a machine to obtain thrust in water by learning the movement of real fish. From the last decade of the 20th century, many of such fish-like machines have been built. Breder's classification of fish swimming is commonly accepted in engineering community [8]. According to its classification, the robotic fish are also developed to mimic the BCF and MPF swimming types.

Bionic MPF propulsion is a hot topic in robotic fishes owing to its flexible locomotion capability. Undulatory swimming motions are often observed in MPF fish. Fish generating thrust through undulating fins shows relatively high flexibility and maneuverability. Its muscle and bone structure are important reasons accounting for this superior performance. As a result, undulating species are also good models for the propulsor design. To achieve the same maneuverability as real fish, undulatory propulsor routinely adopts the parallel mechanism and flexible membrane material that quite resembles actual fin of real fish that has significant number of degrees of freedom. Fig. 2.8 shows the basic topology of such parallel mechanism design inspired by the MPF undulatory swimming modes.

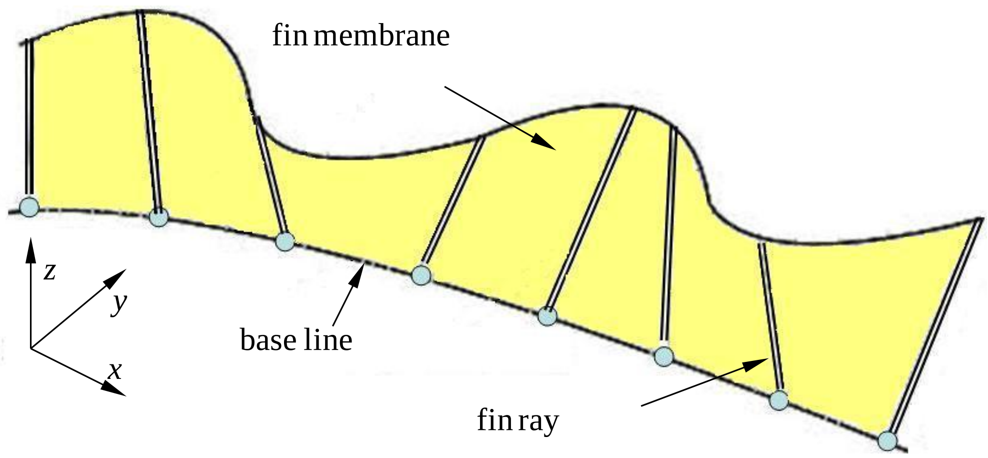


Fig. 2.8 Parallel links used to fit the body/fin of MPF swimmers.

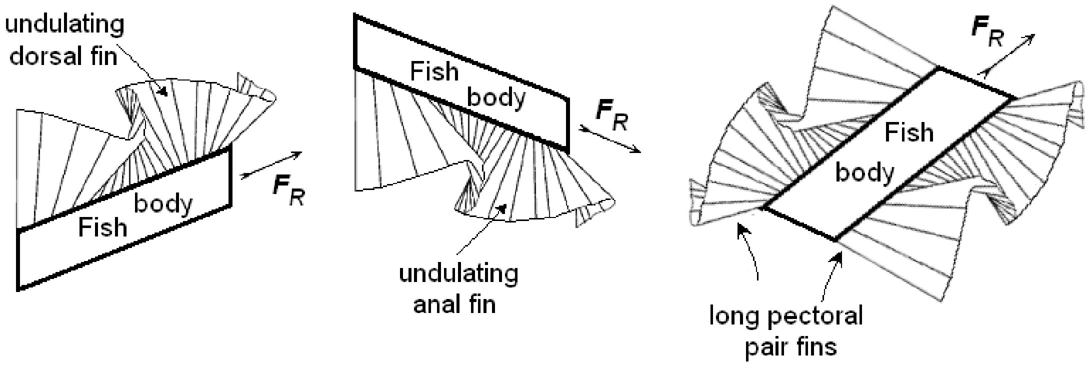


Fig. 2.9 Fundamental structures of undulatory fins found in amguilliform, gymnotiform and rajiform swimmers respectively [12].

The mechanical fin rays replace the bones of real fish in undulatory machines to provide actuation of biomimetic fins. Generally these fins are made of flexible and elastic material to enhance its compliance in swimming. Fin rays oscillate alternately with specific amplitude and frequency, which generate a traveling wave along the long fin. Nearly all the current fish prototypes of undulatory MPF fish adopt this scheme as the design model. Configuration of long fins on fish robots follows the cases in real fishes. Single or paired long undulatory fins are commonly adopted in robotic fishes. Fig. 2.9 shows three fundamental structures in robotic fishes with long fin propulsor.

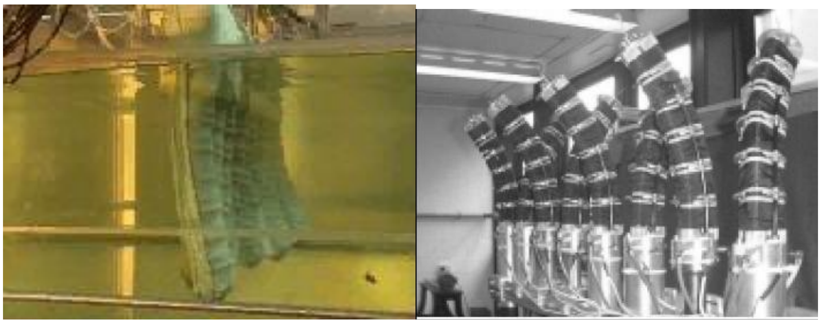


Fig. 2.10 An experimental undulating fin device using parallel bellows actuators [53].

Pure undulatory swimming mode exists in both BCF and MPF swimmers. For undulatory BCF swimmer, both serial chain mechanism and the parallel mechanism are used. For the design of the latter, the number of fin rays appearing at the parallel mechanisms can be expanded by incorporating more actuators without increase the payload for actuators and the motion of those fin rays can be controlled independently, which totally differs from the serial chain mechanism employed by BCF swimming machines. As a result, the parallel mechanism is adopted more commonly in the MPF undulatory swimming machines.

Fig. 2.10 shows an early design of undulating fin device built in Heriot-Watt University in 2001. This mechanical fin consists of elastic fin membrane driven by eight parallel bellows actuator (PBA). However, due to the bulky size and large inertia of the actuator, the swimming performance of this bionic fin is limited.

In the design of undulatory propulsors, the mechanical fin rays replace the bones of real fish. Generally, these fins are made of flexible materials to enhance its compliance in swimming. Fin rays oscillate alternately with specific amplitude and frequency, which generates a traveling wave to obtain thrust. Nearly all the current prototypes of undulatory MPF fish with undulatory propulsors adopt this scheme as the design model [54, 56, 57, 60].

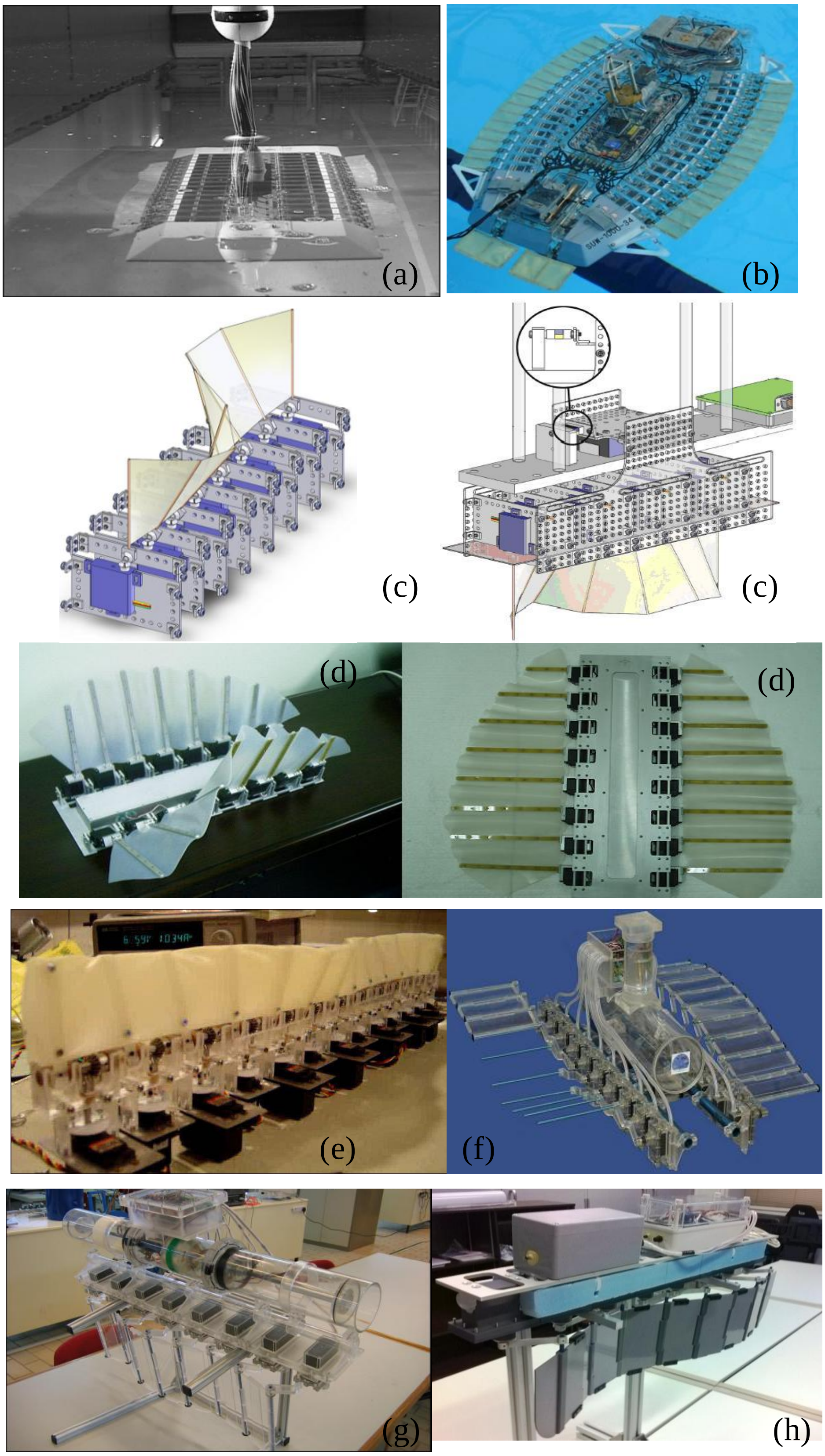


Fig. 2.11 Undulatory propulsor in MPF fish robots. Pictures of prototypes (a)-(h) are adopted from [26, 54-59] respectively.

Similar design makes use of other types of actuators, which can effectively increase the velocity of swimming. One of the representative achievements in designs of MPF swimming machines in recent years is the robotic fish developed by Osaka University. Three generation of prototypes are built since 2002 [59]. Fig. 2.11 shows the prototypes of first and third generation. With the multi-DOF mechanical design and high performance servo system, the robotic squid can perform versatile maneuverability in swimming [55]. Fig. 2.11f-h show the robotic stingray [60] and robotic Knifefish (NKF-I and NKF-II) [57, 58] developed in Nanyang Technological University. Modular designs are adopted in the prototypes. Other examples are shown by Fig. 2.11c-e [26, 55, 56]. More fish robots using MPF swimming type in recent literatures can be found in [61-65].

2.2.2 Body and/or Caudal Fin (BCF) Type Fish Robots

Most of the BCF swimming machines are designed by using planar serial chain mechanisms. Fig. 2.12 shows an example of design that simulates the flexible fish body or fins with 5 rigid links that fit the segment of fish body or fin. This structure shares the same idea with the design of robotic arm. The difference lies in that usually one end a robotic arm is constrained to a base and an end-effector is mounted on the other end, while both ends of the chain of a robotic fish are free in aquatic locomotion. One end of the chain is designed as the fish head and the other end, the so-called end-effector, which is commonly seen in robotic arm acts as the tail or caudal fin.

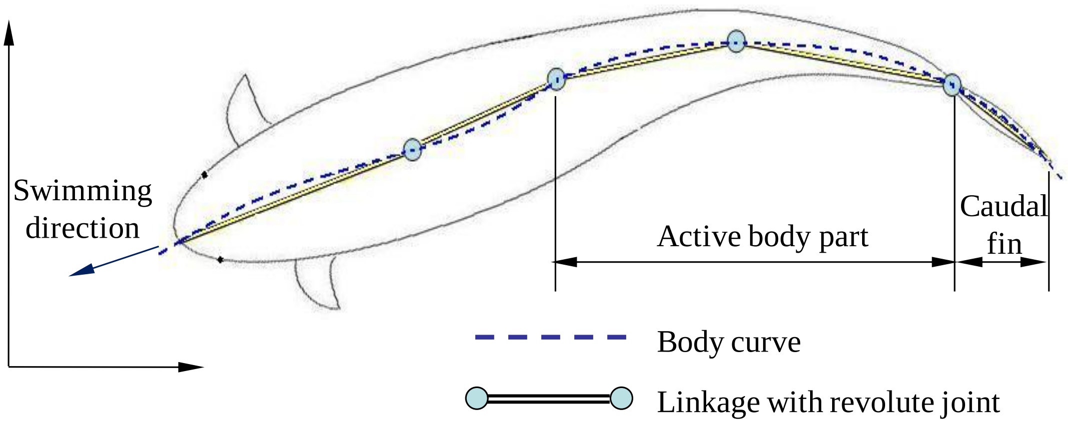


Fig. 2.12 Serial chain of linkages used to fit the body/fin of BCF swimmers.

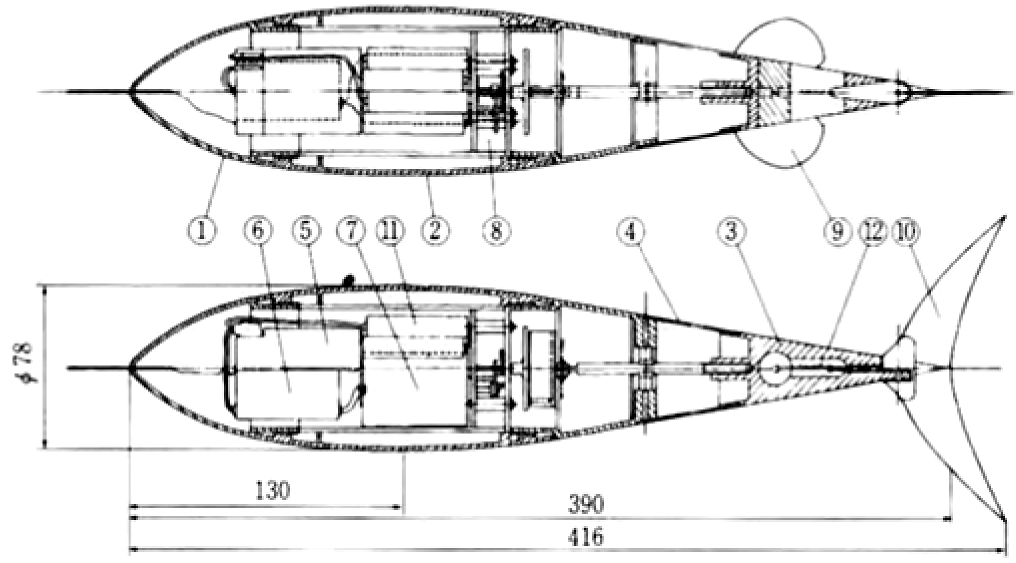


Fig. 2.13 Small-Scale Automatic Mechanical Fish [66].

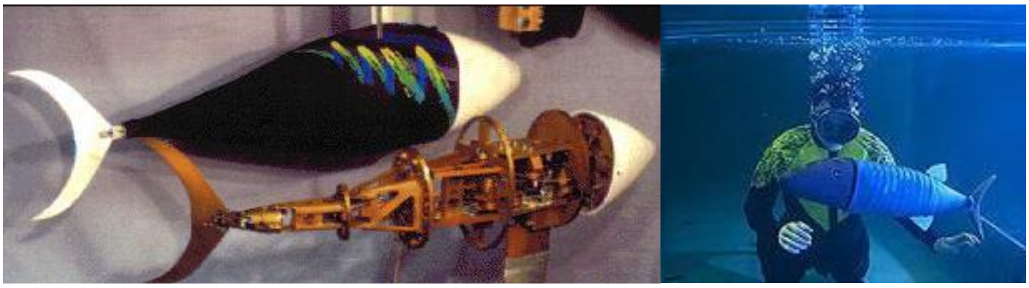


Fig. 2.14 RoboTuna (left) and its improved version RoboPike (right) developed in MIT in 1994 and 1995 respectively [67-70].

In the control of robotic fish locomotion, the rigid links fit the actual body curve of real fish. If the body curve of the real fish is interpreted by its center line of its spine, the serial chain design is actually a kind of curve fitting by using the straight line segments. The fitting error depends on the number of links and the extent that the body bends. The ideal design comprises lots of links to fit the curve. However, this brings about the difficulty in control and mechanical component design. What's more, the design with higher degrees of freedom may not be energy-efficient because the actuators pay more energy to drive the bulky body. As a result, a lot of the current designs of robotic fish employ two to five links and they provide good performance in simulating the swimming of real fish in terms of locomotion when equipped with well selected body shape and material.

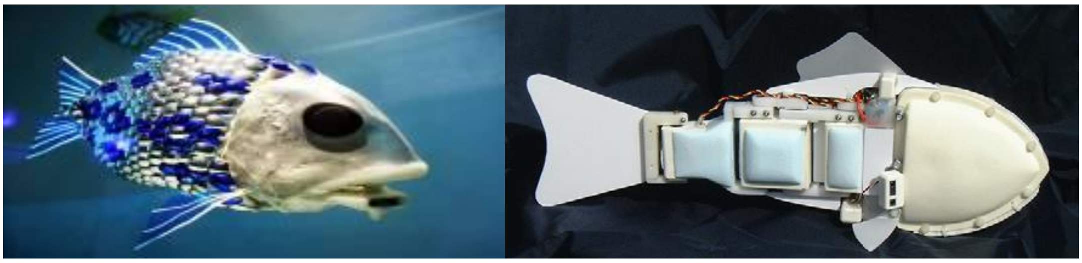


Fig. 2.15 Robotic fish developed in Essex University: photo of the swimming robot (left) and its internal structure (right) [71-73].

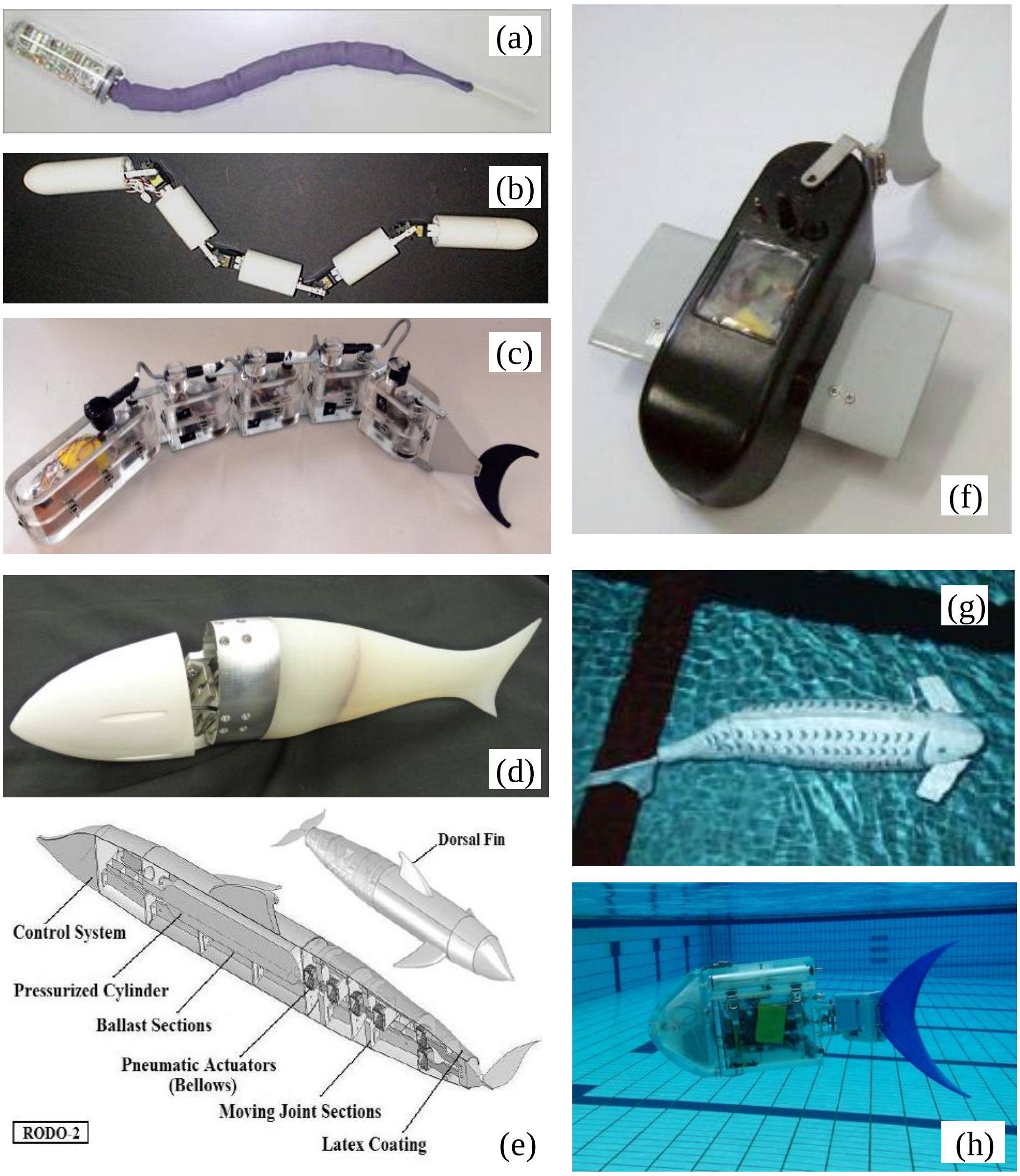


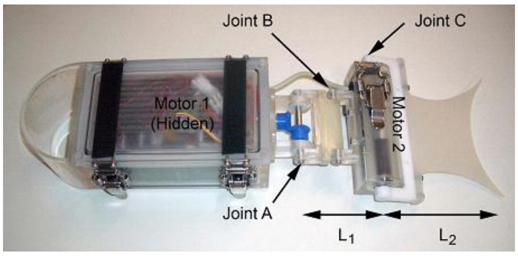
Fig. 2.16 Examples of BCF type fish robots. Those pictures of prototypes (a)-(h) are adopted from [12, 74-78] respectively.

Early realization of robotic fish can be traced to the year of 1978 when the Small-scaled Automatic Mechanical Fish was developed in University of the Ryukyus in Japan [66] (see Fig. 2.13). There were two hinges: one connecting the body and the tail, and another connecting the tail and the caudal fin. Both hinges were free to rotate around the joints. The Scotch-yoke mechanism was employed to perform oscillatory motion [66]. Afterwards, the birth of RoboTuna (see Fig. 2.14) in MIT in 1994 triggered the extensive research interests in biomimetic swimming robots all over the world. RoboTuna has a 1.2m long body made of 2843 mechanical components. It simulates the swimming and body structure of a real tuna. The swimming speed reaches 2 m/s through the oscillation of the body and caudal fin. The improve version, the RoboPike was built in 1995 (see Fig. 2.14). It was used to study the fast maneuverability and the Vorticity Control Theory [67-70].

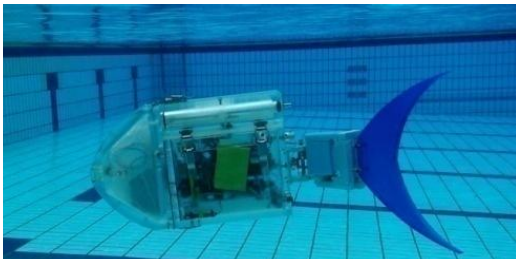
Fig. 2.15 shows another famous fish robot which demonstrates the state-of-art technology in the prototype design. It was developed by the team of Dr. Hu in Essex University in 2006 and was thought as the one most closest to a real fish till now [71]. It mimics a typical carangiform swimmer with flexible and smooth swimming motions in water. Many other BCF type fish robots were developed in the study of bionic propulsion in recent years (see Fig. 2.16). The propulsor designs mostly adopted serial open chain mechanism, i.e. using linkages to fitting the body curves of fishes. The swimming modes cover from pure undulation in anguiliform to pure oscillation in ostraciiform. Some recent designs of fish prototype can be found in [12, 74-79].

2.3 Fish Robots Developed by Robotic Fish Team of Nanyang Technological University

Several fish robots have been developed by the Robotic Fish Team of Nanyang Technological University, including two BCF type fish robots inspired by Asian arowana fish (NAF-I & II), two MPF type fish robots inspired by the Black Ghost Knifefish, and two MPF type robots inspired by manta ray. Every fish robot has two generations of prototype. The second generation is improved from the previous one and has been used as experimental platforms to validate the proposed gait control methodologies for the present work. These prototypes are shown in Fig. 2.17. The captions (a)-(f) in the figure explains the name of each robot. A summary of the fish robots are given in Table 2.1.



(a) **N**anyang **A**rowana **F**ish – I (NAF-I) [80]



(b) **N**anyang **A**rowana **F**ish – II (NAF-II) [81]

(c) **N**anyang **K**nife**f**ish - I (NKF-I) [60]

(d) **N**anyang **K**nife**f**ish - II (NKF-II) [82]

(e) **Ro**botic **Man**ta - I (RoMan-I) [83]

(f) **Ro**botic **Man**ta - II (RoMan-II) [84]

Fig. 2.17 Prototypes of fish robots developed in Nanyang Technological University in recent years.

Table 2.1 A list of fish robots developed by the Robotic Fish Team of Nanyang Technological University.

No.	Name of Robot	Year	Type	Swimming Mode	Dimension, L x H x W (mm)	Mass (Kg)	DOF	Max. Speed
1	**N**anyang **A**rowana **F**ish - I (NAF-I)	2006	BCF	Sub-carangiform	450 x 100 x 120	4.9	2 active	0.04 m/s 0.08 BL/s
2	**N**anyang **A**rowana **F**ish - II (NAF-II)	2008			661 x 260 x100	6.8	1 active 1 passive	0.33 m/s 0.5 BL/s
3	**N**anyang **K**nife**f**ish - I (NKF-I)	2005	MPF	Gymnotiform	850 x 200 x 15	10	8 active	0.09 m/s 0.11 BL/s
4	**N**anyang **K**nife**f**ish - II (NKF-II)	2009			600 x 150 x 8	5.6	8 active	0.35 m/s 0.58 BL/s
5	**Ro**botic **Man**ta - I (RoMan-I)	2009		Rajiform	550 x 180 x 900	6.1	8 active 6 passive	0.25 m/s 0.45 BL/s
6	**Ro**botic **Man**ta - II (RoMan-II)	2010			500 x 105 x 960	7.3	6 active	0.4 m/s 0.8 BL/s

2.3.1 Design of Fish Robots with Parallel Mechanism

Among the fish robots listed in Fig. 2.17 and Table 2.1, the NAF-II, NKF-II, and RoMan-II are improvements of previous designs and have been adopted in the present work. The prototype designs of these robots are presented in this section.

(a) Prototype Design of an Eight-DOF Knifefish Robot (NKF-II)

Although the propulsor designs for different fish robots may not be the same, their fundamental mechanisms share the same topological structure: either parallel mechanism or serial open chain mechanism. The types of fish robots are divided by this categorization. An eight-DOF fish robot inspired by Black Ghost Knifefish (see Fig. 2.18) has been developed by our team. This kind of fish uses a long anal fin beneath the body to generate thrust. When it swims, the fin produces traveling wave propagating to the opposite direction of swimming. The rigid body holds straight unless it performs turning. The wave direction can be switched and by this way, the fish is able to swim backward without turning its body. The fish robot is designed to simulate such flexible motions.

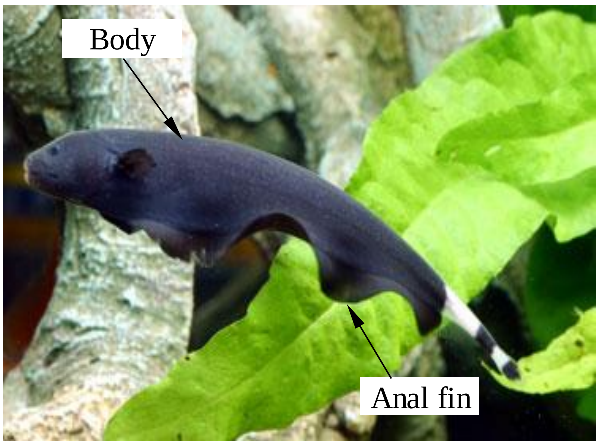


Fig. 2.18 Black Ghost Knifefish.

Parallel fin mechanism is applied in the propulsor design. Usually fin rays are covered and interlinked by membranes made of flexible materials such as silicon rubber. As shown by examples in Fig. 2.11(a)-(e), the biomimetic fin rays are parallel-arranged on a fixed baseline and covered by a flexible membrane. The fin rays are connected on a free baseline through cranks. The use of parallel mechanism may benefit from easy implementation of motion control. However, it suffers from the limitation of work space. When fin rays oscillate, the membrane will expand and retract in certain range. The oscillation amplitude of a fin ray is limited by the range. Therefore, in designs of current MPF fish robots, the fin ray is normally short in order to oscillate within the workspace and reduce the water reaction [26, 54-59].

In the design of Knifefish robot (NKF-I & II), a different design is proposed. The fin rays are connected on a free baseline through cranks. Each crank is directly driven by a motor and baseline moves as all cranks oscillate. With this mechanism, the fin membrane can be rigid and pure undulation can be obtained by the propulsor. Additionally, the fin ray can be longer than those in other designs, which helps to increase the area of fins. Fig. 2.19 shows the overall view of the prototype. The prototype is made up of a controller box, a buoyancy tank, eight waterproofed motor housings and eight pairs of fin plates which construct the long propulsor. Physical specifications of this propulsor prototype are listed in Table 2.2.

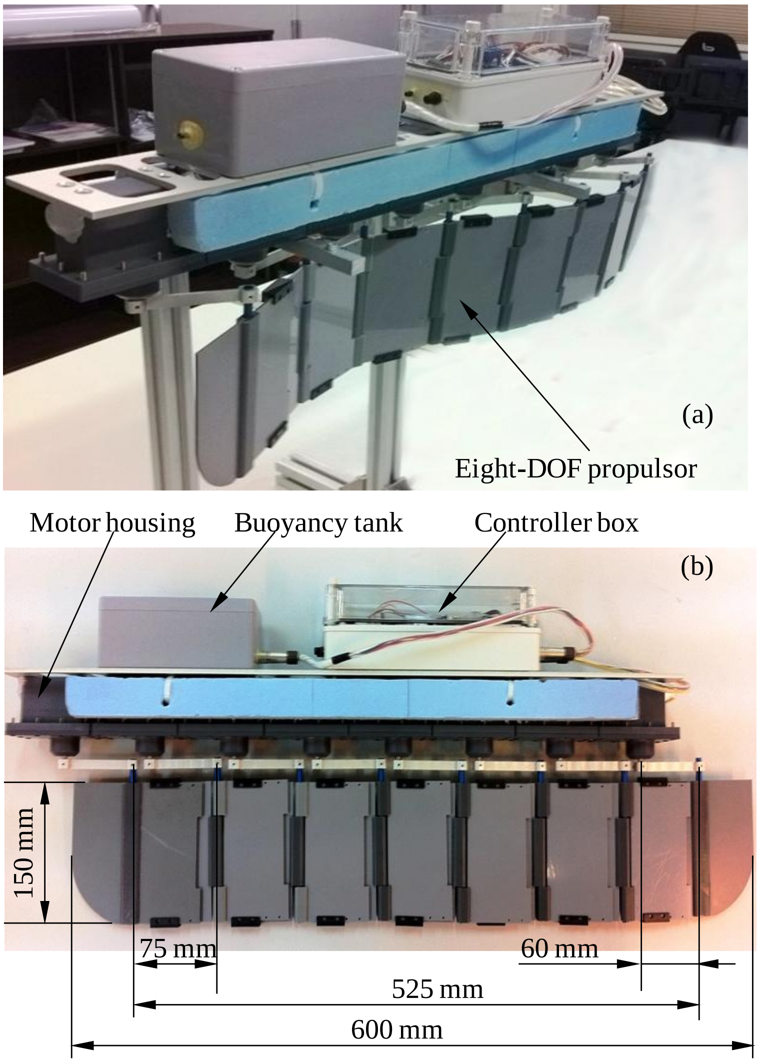


Fig. 2.19 Prototype of an Eight-DOF Knifefish Robot: (a) isometric view and (b) front view.

Table 2.2 Physical specifications of NKF-II.

Mass	Approximate 5.62 kg
Propulsor Dimension	600mm (L) x 150mm (H) x 8mm (W)
Actuator	Eight brushless DC servo motors
Controller	PIC18F-2431 and PIC18F4520 for free-run robot CompactRIO™ embedded system for lab testing
Power source	7.2VDC, 4500mAH Ni-MH battery
Swimming speed	0.35 m/s, 0.58 BL/s (max)
Operating time	Approximately 1 hours (fully charged)
Communication	FM radio frequency or wired connection to computer

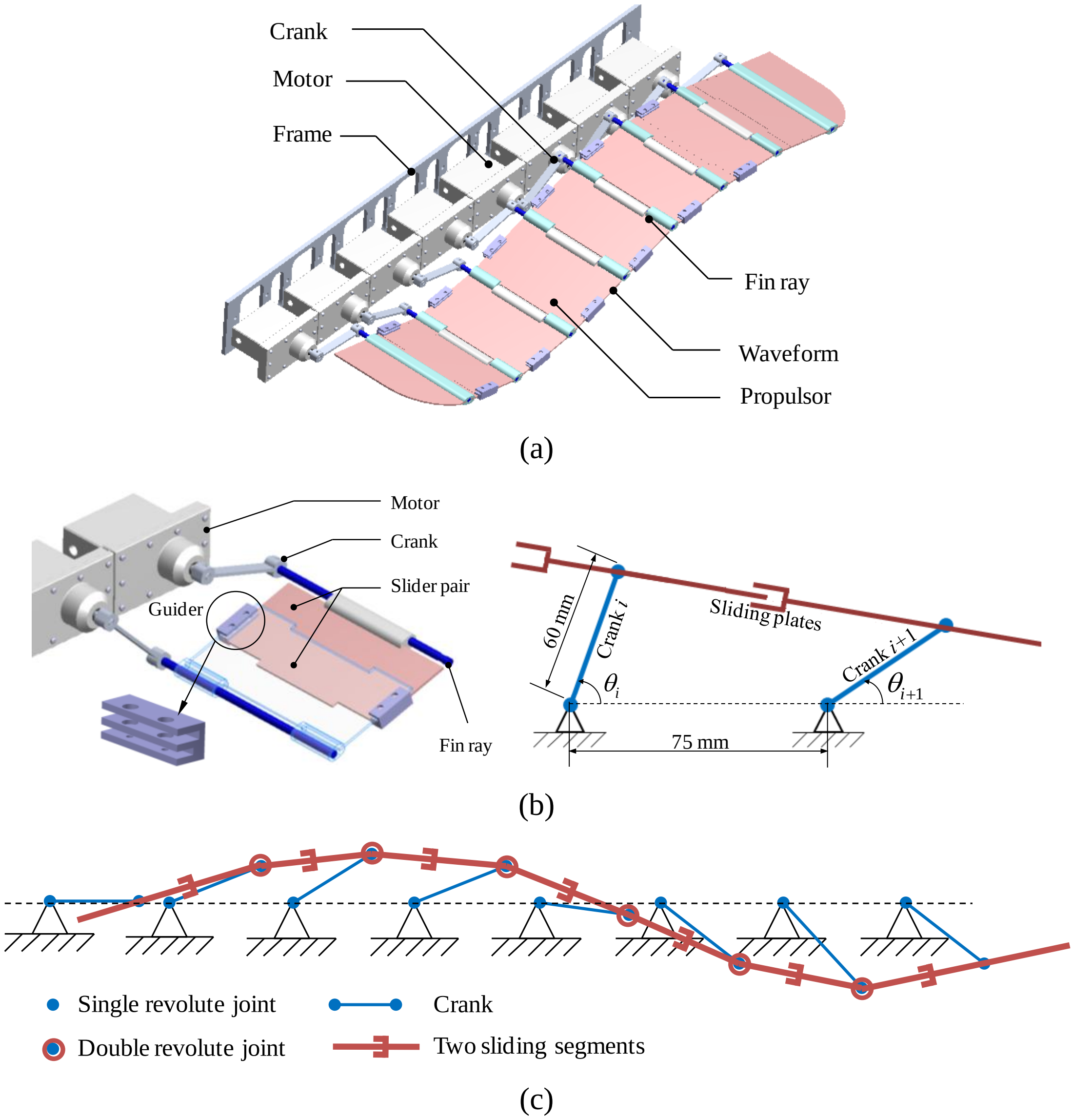


Fig. 2.20 The fundamental element of the propulsor: (a) CAD model of the propulsor in an undulatory waveform; (b) a pair of sliders; and (c) the schematics of the two-DOF five-bar slider mechanism.

The long propulsor is made of interconnected thin sliding plate modules. Each module comprises two relatively sliding plates. The fundamental mechanism is a two-DOF five bar linkage, as shown in Fig. 2.20b-c. The prototype contains eight such slider pairs and it can have more if necessary. According to Fig. 2.20c, the mobility (DOF) of the propulsor can be obtained through the following calculation:

Number of cranks:	8
Number of sliding segments:	14

Number of single revolute joints: 10

Number of double revolute joints: 6

Number of sliding joints: 7

Mobility M = 3(8+14) - 2(10+2X6+7) = 8

This propulsor can be regarded as a two-dimensional waving plate. An arbitrary traveling wave can be generated through alternate oscillations of cranks (see Fig. 2.20a). The parallel mechanism of the propulsor makes it possible to control the oscillation of the each crank independently. This mechanism design also suffers from the problem of workspace because large-amplitude oscillation will cause disconnect of fin sliders. For KNF-II, the maximum oscillation angel of cranks is 50^o.

(b) Prototype Design of a Six-DOF Manta Ray Robot (RoMan-II)

Many MPF type fish adopts two sided fins to produce thrust. The motion patterns of the later are more versatile. A fish robot (RoMan-II) inspired by manta ray has been developed. As shown in Fig. 2.21, two sided flapping fins are employed as propulsors. A manta ray exhibits rajiform motion (the thrust mainly generated by the movements of two flapping pectoral fins) during its swimming in the ocean [7]. Several reasons motivate the development of the fish robot.

Firstly, the flapping motion is thought of an effective way of producing thrust [7]. It is hoped that a robotic manta ray could benefit from such efficient propulsive mechanism, if it can simulate the motions of a real manta ray. Secondly, unlike some of other fish that obtain thrust by oscillating their bodies and caudal fins, a manta ray has a big and flat body that can keep it stable during swimming in the course of flapping, undulating, and/or gliding. Such a unique feature facilitates the robotic manta ray a promising underwater vehicle for marine purposes as instruments, equipment, or payloads can be accommodated inside the body. Thirdly, like other fishes using Median/Paired Fin (MPF) locomotion mode [35, 85], the manta ray exhibits high performance maneuverability in water, which is an ideal feature for a fish robot to possess.

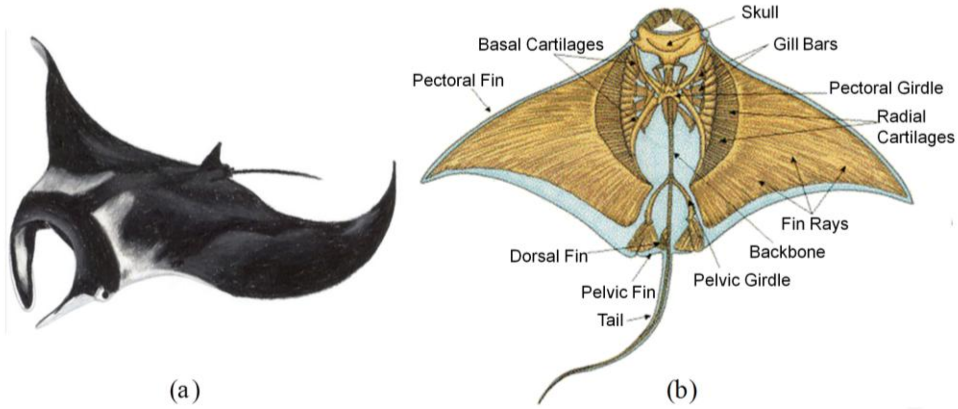


Fig. 2.21 (a) A manta ray and (b) terminologies used to identify its bones, muscles, and fins (adapted from the work by Prof. R. Aidan Martin at the Zoology Department in the University of British Columbia).

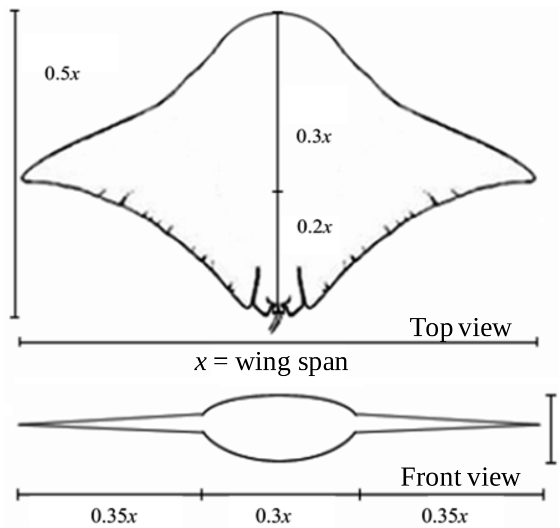


Fig. 2.22 Scaled diagram of a manta ray. Dimensions of the two fins and body are normalized to the wing span [86].

Manta ray could be recognized from its diamond-shaped flat body. The shape and size of its two wing-like fins and the streamlined body play important roles in the swimming. The scaled diagram of a manta ray is shown in Fig. 2.22. The structure of a manta ray robot simulates that of the real fish, which is expected to share the similar high performance for the robot. Prototype of the robotic manta, RoMan-II is shown in [86]. The proportions of all body parts approximately follow those of the real manta. Note that due to the mechanical limitation, the fin width of RoMan-II is slightly smaller than 0.35 wing span. Such configuration ensures enough strength of the large soft fins.

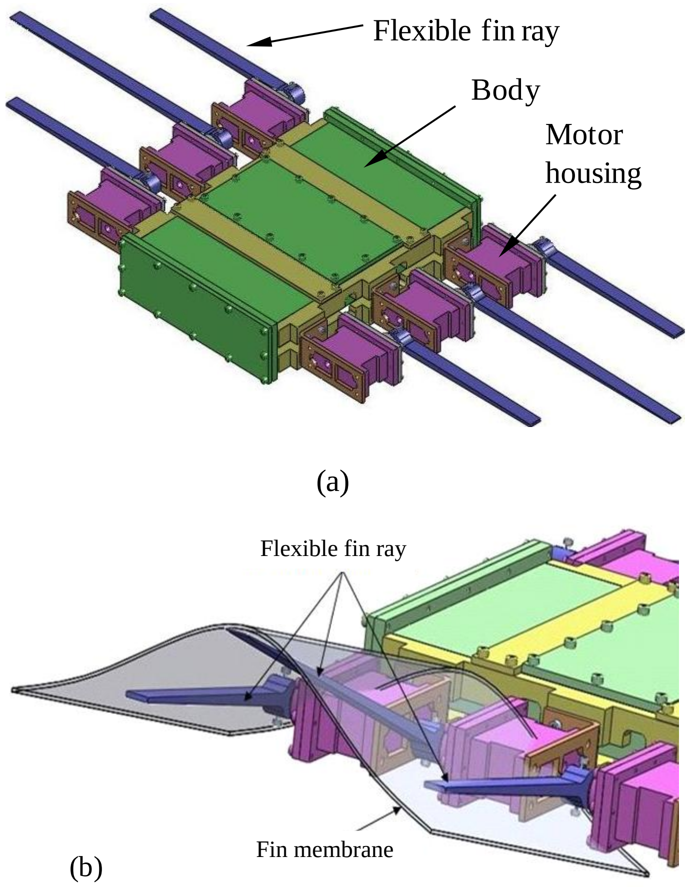


Fig. 2.23 (a) CAD model of a manta ray robot (the figure shows the internal frame and fin membrane is not covered on fin rays); and (b) fin ray and fin membrane configuration of one side.

The manta ray robot discussed in the present work is an improvement of RoMan-I [87]. The fish prototype is shown in Fig. 2.23. There are totally six DOF involved in flapping motion and three controlled fin rays are arranged on each side. The bionic propulsor (the flapping fin) is designed with three fin rays at each side, as shown in Fig. 2.23. Fin rays are parallel-connected and actuated independently by brushless servo motors. Fin rays are made of elastic material (Polypropylene) in order to add compliance of motion. This design is simpler than those using smart materials [88, 89] or multi-joint linkage mechanisms [1, 2, 12]. Fin rays are designed as uniform strength cantilever beams in order to reduce the mass without causing structure failure in oscillations. Fin membrane is made of silicon rubber and attached to the three fin rays.

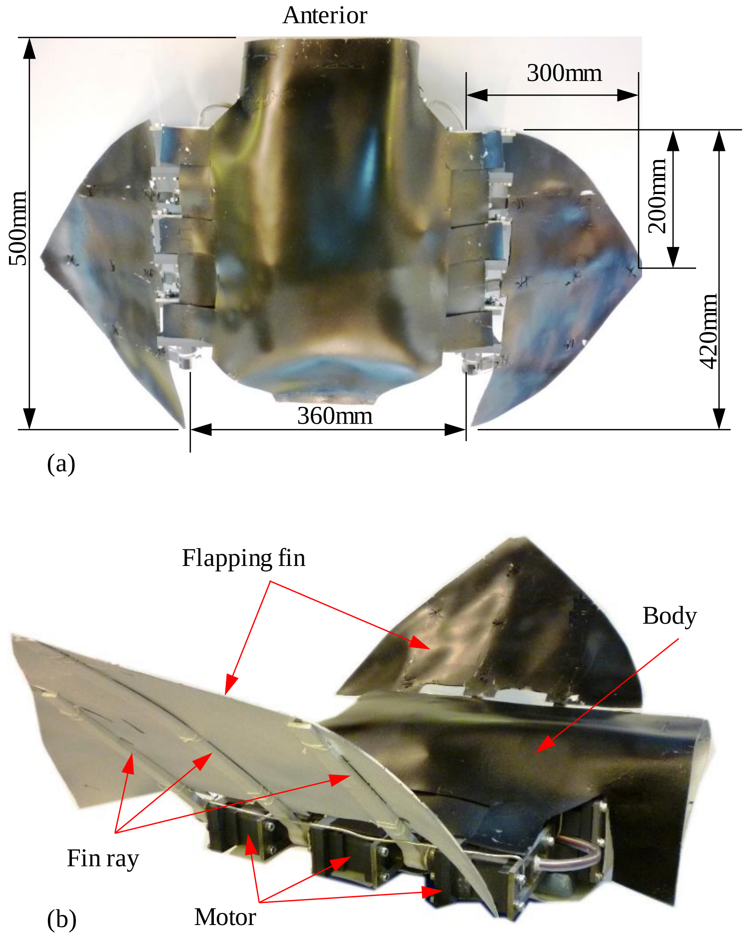


Fig. 2.24 Overview of the manta ray robot RoMan-II: (a) dimension of key features and (b) isometric view.

In the prototype design, the elastic fin ray is made of Polypropylene (Young's modulus is 1350 MPa and density is 946 kg/m^3). A fin membrane in silicon rubber sheet is taken as the cover of the fin rays. With such an arrangement, fin motion control becomes less complex and consumption of energy could also be reduced. On the other hand, fin rays and actuators are mounted to the fish body where the control electronics and buoyancy modules are accommodated. The huge-and-flat body enables the fish robot has better payload capacity. The payload could be any devices that can be accommodated into the body of the robot, such as the extra battery, communication devices and various sensors for autonomous locomotion control. The physical specifications of the manta ray robot are listed in Table 2.3. The mass of the robot without payload is 4.8 Kg. With extra 2.5 Kg, the total mass reaches 7.3 Kg and in this case the robot obtains neutral buoyancy in water. With one full charge of battery, the robot can operate 6 hours in testing.

Table 2.3 Physical specifications of RoMan-II.

Mass	Approximate 7.3 kg (include 2.5 Kg payload)
Propulsor Dimension	500mm (L) x 105mm (H) x 960mm (W)
Actuator	Six brushless DC servo motors
Controller	PIC18F-2431 and PIC18F4520
Power source	7.2VDC, two sets of 4500mAH Ni-MH battery
Swimming speed	0.4 m/s, 0.8 BL/s (max)
Operating time	Approximately 6 hours for swimming (fully charged)
	Approximately 9 hours for gliding (fully charged)
Communication	FM radio frequency or wired connection to computer

The use of elastic fin rays can improve the propulsion efficiency [77, 90]. When flapping in water, fin rays oscillate alternately to drive the fin membrane so as to generate thrust. Larger thrust can be obtained by increasing of frequency and/or amplitude of flapping fins. However, due to the power limitation of actuator, the effect of this method is also limited. As the fin rays can be regarded as vibrating cantilever beams (see Fig. 2.23b), another way to augment the thrust is to drive a fin ray at its resonance frequency so as to increase the flapping amplitude.

When a fin ray oscillates in water, the alternating reactive force from water is equivalent to a nonlinear damping effect. Besides, the fin rays together with soft fin membranes (silicon rubber) cause the movement of surrounding water, which leads to the added-mass effect [90-92]. It has been reported that the added-mass will reduce the natural frequency of a cantilever beam immersed in water [91, 92]. As a result, the resonance frequency of fin ray oscillation will be lower than the first order natural frequency of the beam in the air. Ansys Workbench^{TM} software has been used to find the first order natural frequency of fin rays (see Fig. 2.25). For example, the middle ray has a natural frequency of 11.3 Hz in air. The efficient oscillation frequency was found to be 1.3 Hz in water testing (see Fig. 2.26).

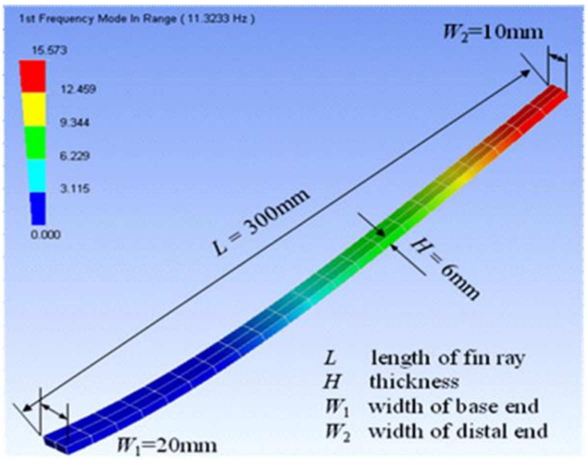


Fig. 2.25 A simulation of the first-mode vibration of the middle fin ray by using ANSYS Workbench.

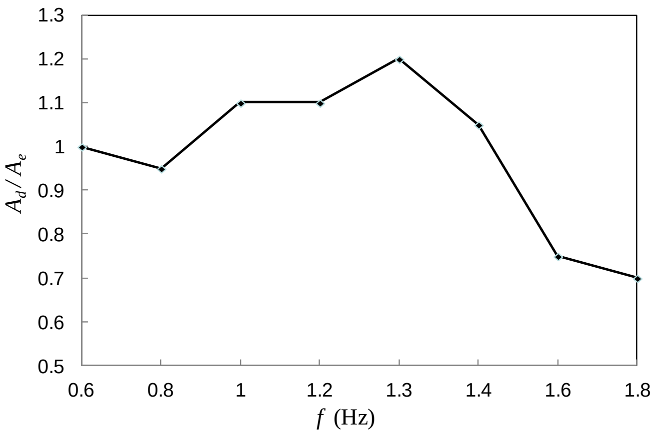


Fig. 2.26 Frequency response of fin ray in water. The vertical axis denotes the ratio between the amplitude of distal end of fin (A_d) and the excitation frequency of motor (A_e). Horizontal axis is the excitation frequency in Hz.

2.3.2 Designs of Fish Robots with Serial Open Chain Mechanism

A two-DOF BCF type fish robot (one controlled DOF and one passive DOF) is developed by our team. This robot is an improvement of the robot presented in [93]. Original design of the fish robot (NAF-I, [93]) is found not so efficient in thrust generation, which is not a suitable platform to perform further research regarding the on-line motion planning. As a result, the mechanisms are simplified and it comes to the NAF-II. This fish robot has been detailed in [30, 81, 94]. In this section, mechanisms of the robot are reviewed. The contents of this section are modified from [30] and [81].

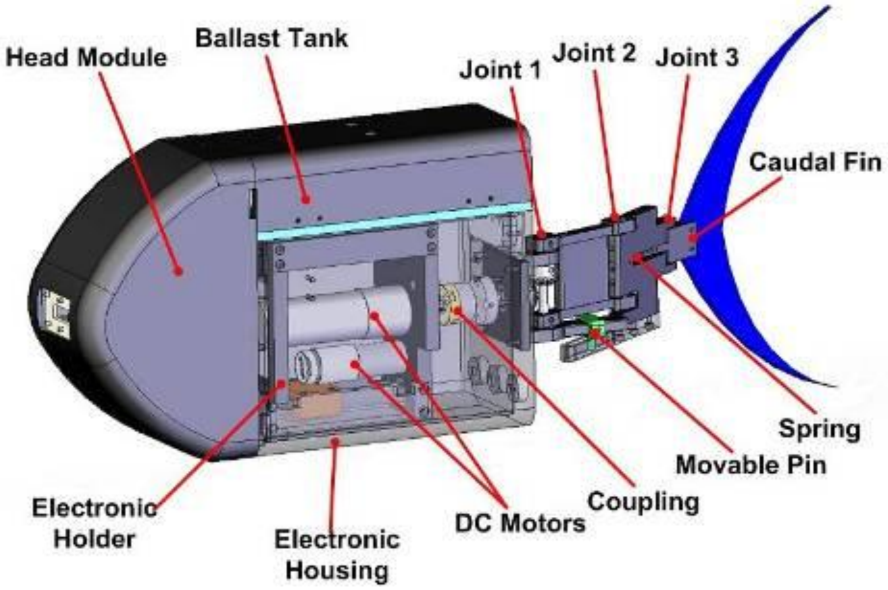


Fig. 2.27 CAD model of a two-DOF BCF type fish robot [81].

Table 2.4 Physical specifications of NAF-II [95].

Mass	Approximate 6.8 kg
Dimension	661mm (L) x 260mm (H) x 100mm (W)
Actuator	2 units 6W Maxon^{TM} DC motor
Microcontroller	PIC18F-2431, PIC18F4520
Power source	15VDC, 4000mAH Ni-H battery
Swimming speed	0.33 m/s, 0.5 BL/s (max)
Operating time	Approximately 4 hours (fully charged)
Communication	433 MHz RF serial communication

Fig. 2.27 shows the CAD model of a fully assembled fish-like underwater vehicle after the embodiment design process. The NAF-II comprises of four main individual modules: tail fin module, electronics housing module, ballast tank module and head module (including sensors and power unit). As these parts are designed in a modular manner, each module can be easily replaced should there be a change in the design or additional functions are required to be incorporated to the robotic fish.

NAF-II uses a caudal fin to provide the propulsion, which is commonly employed by carangiform swimmers, such as the arowana or the herring. The caudal fin mechanism is driven by a single DC motor with a set of miter gears to convert the horizontal axis of rotation into vertical axis. The adjustable movable pin and prismatic joint will determine the resultant angle at Joint 2. The angle at Joint 2 can be adjusted by sliding the movable pin along the bar. By using this design, a parametric study on the effect of Joint 2 angles of the tail mechanism during the experimental phase can be conducted. The last joint of the tail fin mechanism is a spring-loaded passive joint, which introduces a certain degree of flexibility to the tail fin mechanism. Studies have shown that a properly selected chord-wise flexibility can have significant effect on the performance of oscillating foils [90, 95]. Joint 3 was designed in such a way that the spring constant can be changed, in order to study the effect of the different degree of flexibility of the caudal fin had on the swimming performance of the robotic fish.

A counter weight driven by a DC motor through a lead screw is mounted on the electronic holder. The function of the counter weight is to provide titling feature to NAF-II. Thus, even without the present of ballast tank, the robot can still dive into the water by adjusting the position of the counter weight. A ballast tank is incorporated onto the robotic fish to allow the fish to navigate in the vertical direction. This ballast tank utilizes a single piston and is driven by a DC motor though a pulley and belt system. Physical specifications of the fish robot are listed in Table 2.4.

The fish head module was designed to house various sensors, which are sensitive to noise and the battery pack. Moreover, the fish head has a smooth surface to viscous drag. The round shape in the front also helps to reduce the form drag during swimming. The electronics housing module shields the electronic components from the harsh underwater environment and acts as a base body where enhancement modules can be mounted. This is also the module where the microprocessor, motor controllers, radio frequency receiver and navigation sensor for the robotic fish are located.

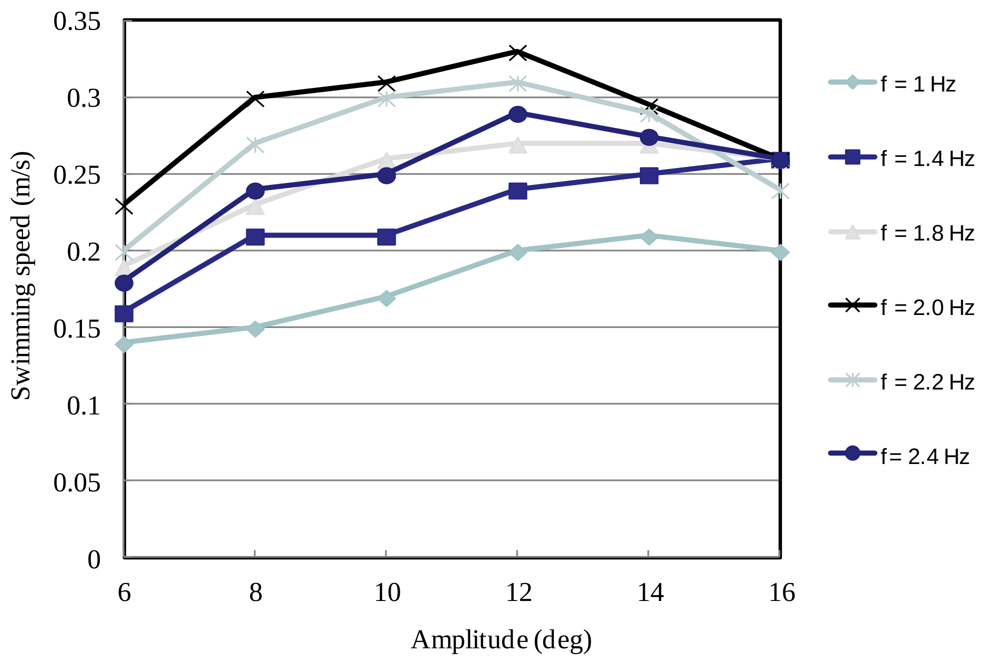
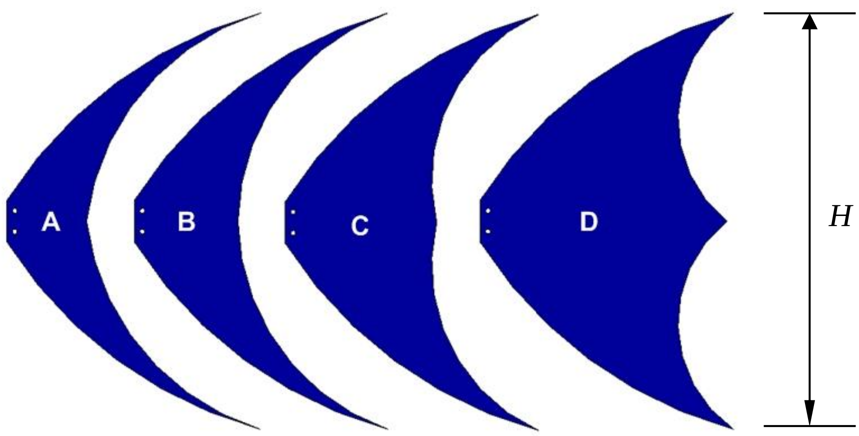


Fig. 2.28 Swimming speed test of NAF-II under different values of tail beat frequency (f) and amplitude (A). The maximum speed reaches 0.33 m/s at f = 1.2 Hz and $A = 12^o$.



$$\text{Aspect Ratio} = \frac{H^2}{\text{Fin Area}}$$

Fig. 2.29 Type of Tail Fins Tested: Area A: 7711 mm^2, Aspect Ratio: 8.71; Area B: 9906 mm^2, Aspect Ratio: 6.82; Area C: 14096 mm^2, Aspect Ratio: 4.80; Area D: 21082 mm^2, Aspect Ratio: 3.21.

In the design of NAF-II, the following system parameters should be considered as they have significant influence on the swimming performance: the drag of the body, the shape of the tail fin, and the stiffness of the spring. The suitable values of parameters can be found from large amount of empirical tests with the use of statistical methods [95]. Rigid connection of the tail fin and three different spring connections with constants 0.98, 2.45 and 4.61 N/m^2 were tested. It is found that using of a spring will improve the propulsion performance. The 0.98 N/m^2 spring produces largest thrust [95]. The selection of caudal fin is based on the aspect ratio (see Fig. 2.29). Basically, a fin with a higher aspect ratio is more efficient than that with a lower aspect ratio because the fin would induce less drag per unit of lift or thrust produced [8]. The fin A in Fig. 2.29 is selected for NAF-II [95]. With these configurations, the swimming performance was tested and plotted in Fig. 2.28.

2.4 Modeling and Control of Swimming Gaits for Fish Robots

Traditionally, the rhythmic pattern of fish swimming is modeled by sine or cosine function in offline motion planning. Predefined body motion functions are used to find the amplitude and phase of those sinusoidal gait generators. The coordination of multiple gaits depends on the phases of gait generators. More recently, CNLOs are more and more often applied in swimming control, which makes the on-line generation of swimming gaits possible. Gait generators are used in swimming locomotion control with open-loop or closed-loop schemes. Some work on kinematics modeling and control of swimming gaits for multi-DOF fish robots will be reviewed in this section.

2.4.1 Kinematics Modeling of Swimming Gait Based on Sine Generators

Locomotion of fishes relies on rhythmic activity to move forward through undulations and/or oscillations of fins and/or body [30]. Badly chosen gaits will agitate the robot without guarantee that all system elements (vertebrae) will be involved in the thrust generation effectively [97]. A vast of applications of swimming gait control is based on sine or cosine functions decomposed from fish body motion functions. Other methods are also documented in current literatures such as the finite-state machines, heuristic control laws [98] and Lagrangian function [78].

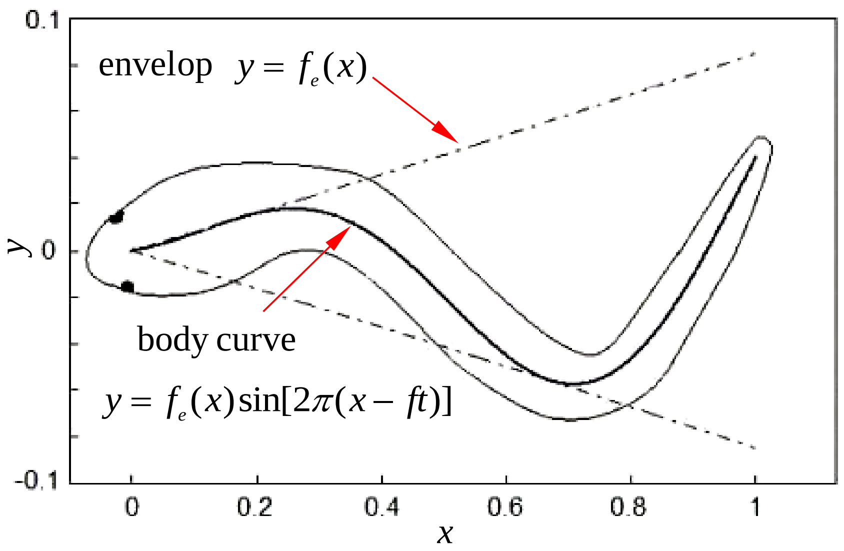


Fig. 2.30 Illustration of fish body curve and envelop during swimming (top view). In robotic fish control, the body curve is usually defined with a 2-D travelling wave equation called body motion function [99].

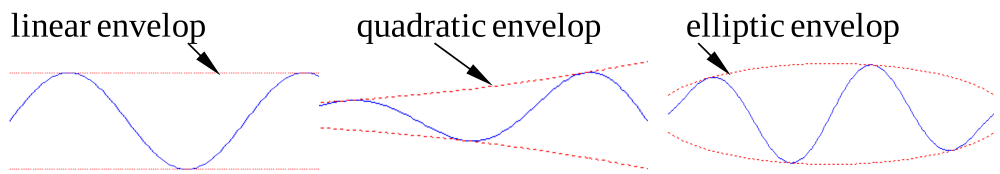


Fig. 2.31 Waveform commonly used by undulatory swimming machines with parallel mechanism designs (a) waveform with linear envelop; (b) waveform with parabola envelops; and (c) waveform with elliptic envelop [12].

In order to maintain the coordination of multiple gaits, those gait generators are derived from predefined motion reference system. As one kind of guidance system, the body motion functions defined in biological literatures is applied to generate proper swimming gait of swimming machines. By using the body motion function (also called motion library in some literatures such as [100]), the separate gait function (control signals for each actuator) for each actuator will be derived from this function. The resulting motion of the robotic fish is obtained by composition of the elementary motions for each actuator. One commonly used fish motion function is suggested by Lighthill [101] as

$$y = f(x,t) = f_e(x)\sin(2\pi ft + \frac{2\pi}{\lambda}x) \tag{2.1}$$

where $f(x, t)$ is a 2-D wave function that describes the position of fish body along its central line of spinal cord at a specific time instance t, $f_e(x)$ is the envelop function, which could be in linear, quadratic or elliptic forms, f is the oscillation frequency of body segment and determines the speed of the traveling wave, and λ is the wave length. The body curve is illustrated by Fig. 2.30. The commonly used envelop is shown in Fig. 2.31. The figure shows the body wave adopted in the straight-line motion. Other types of motion such as turning can be controlled by adding modulation terms to Eq. (2.1) [71, 102, 103] or by a framework derived from the hydrodynamics analysis of fish swimming [104-106].

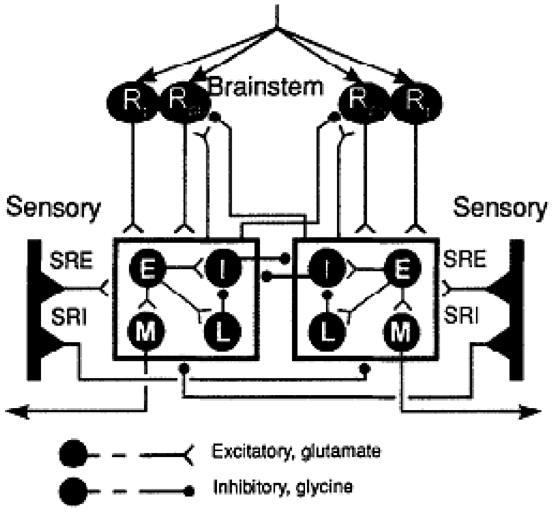
Other guidance systems are derived from the Lighthill's. The body motion function can also be extended to control different swimming motion such as turning by superposition of modulating terms [97]. Similar method by using modulating terms can also be used for anguilliform locomotion in [74], carangiform locomotion in [71] and many other robotic fishes [57, 107, 108]. It is natural to control oscillation of fish body segments by using sine or cosine functions to fit the curve defined by body motion functions. The gait control functions are completely determined by the body motion functions and they can hardly be tuned when the fish robot swims. Otherwise, those gaits will lose coordination.

2.4.2 Models of Central Pattern Generators (CPGs)

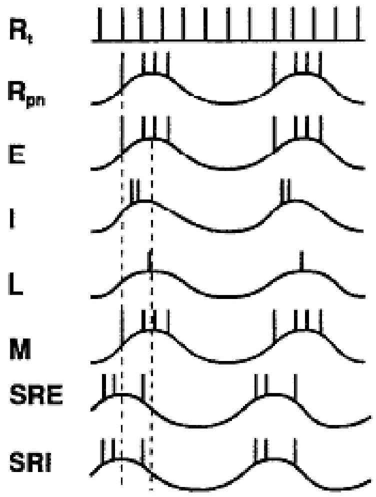
It is believed that the fish swimming performance is related to the shape of the body curve. The envelope parameters in Eq. (2.1) can be obtained by observation of behaviors of real fish captured by video cameras [12, 101, 109]. To enhance the performance of swimming, some literatures also suggest the application of the computational fluid dynamics (CFD) analysis techniques to find the optimal values of these parameters either in energy-efficient mode or fast-swimming mode [16, 108]. In experiment-based research, global searching techniques such as Genetic Algorithm is employed to study envelop parameters [110-112]. Those methods are mostly on an offline basis. Once the body motion function is changed, the gait control functions have to be re-derived in order to maintain coordination of motions. It is desired to have a method that when parameters of each gait function are changed, the gaits are still coordinated. The artificial Central Pattern Generators (CPGs) provide such facilities.

CPGs are neural networks capable of producing coordinated patterns of rhythmic activity without any rhythmic inputs from sensory feedback or from higher control centers [11]. Biological CPG is a kind of neural circuit found in both invertebrate and vertebrate animals, including humans [113] that can produce rhythmic patterns of neural activity. The rhythmic actions of animals (running, chewing, swimming, flying, etc.) produced by biological CPGs can be simulated by artificial neural networks in the locomotion of robots. One special property in CPG is that the generation of rhythmic motion patterns does not need rhythmic inputs, i.e. CPGs are self-sustained systems. Reviews of CPGs in their neurobiological aspects can be found in [15, 114, 115].

Early investigation of CPGs can be found in [116] where the rhythmic animal's nerve behavior under simple stimuli were reported in the year of 1912. Later, CPGs were found generally existing in many kinds of animals and many different ways of rhythmic activities. Representative researches were made on primitive fishes like lamprey [117-121] and even mammals such as the cat [122]. In the experiments on lamprey, when the spinal cord of the animal was isolated for the fish body and was activated by simple electrical or chemical stimulations, it still produced locomotion patterns (the so-called fictive locomotion [23]) which were quite similar to that produced by the intact spinal cord [15, 22, 23, 25]. Grillner [25, 121] studied swimming in the lamprey and presented a neural network model of CPGs that controls motion in the animal. A lamprey consists of approximately 100 segments in its body. Its locomotion is obtained by propagating waves of excitation down the spinal cord with a constant phase delay between equidistant segments so that the resulting muscle contractions lead to a traveling wave [25, 78]. The spinal cord of the lamprey can live for several days after removal from the body. In experiments of fictive swimming [25], the motor pattern produced in the isolated lamprey spinal cord show that if excitation is introduced at the "head" of the spinal cord, the motor impulses generated by the spinal cord match the traveling wave used for forward locomotion (this is fictive swimming). Excitation introduced near the "tail" of the spinal cord lead to motor impulses that would lead to a wave traveling toward the head of the animal (backward fictive swimming), leading to backward propulsion [25]. This feature is modeled with neural networks and applied on the swimming robotic fishes (see Fig. 2.32). Those results show that the CPGs in spinal cord are sufficient generate coordinated locomotion pattern without the extra commands from the brain.



(a)



(b)

Fig. 2.32 Segmental neuronal network that coordinates locomotion: (a) a schematic representation of the brainstem, segmental and sensory components of the neural system that generate burst activity; and (b) a schematic representation of the activity patterns of the different nerve cells and their time relations (adopted from [121]).

In the experiment on a cat [122], when the de-cerebrated cat was put on a running treadmill, the cat could still walk on the treadmill. It could even change its gait patterns by changing the stimulation level to the spinal cord. This result reveals that the spinal cord itself can control the complicated locomotion and the brain only provides the stimuli with simple signals (electrical or chemical signals) [122]. And it provides an excellent example of distributed control.

A lot of work has been done toward explain or simulate behavior of biological CPGs by using math tools. Different levels of the artificial CPGs are established [15]. The biophysical model is constructed based on the Hodgkin–Huxley type of neuron models. Such modeling method is somewhat like the modeling of artificial neural networks: explain how ion pumps and ion channels influence membrane potentials and the generation of action potential [15, 22]. The second way is to use leaky-integrator neurons or integrate-and-fire neurons to establish a connectionist model. The research on these models focuses on the generation of oscillation and the coupling of neurons by using the well-developed structure of oscillator networks [123]. Coupled nonlinear oscillator model is another important way of simulate CPGs. Although some researchers insist that CPGs should only be modeled by the first way, the CNLO model has been well accepted as artificial CPGs. Technically, as long as the behavior of a dynamical system can successfully simulate the rhythmic motion pattern, that system can be regarded as a CPG model. The research of this kind of model focuses on design of coupling schemes in order to achieved synchronization of large population of oscillators [11, 15, 124, 125].

2.4.3 Modeling of CPGs by Using Coupled Nonlinear Oscillators (CNLOs)

An excellent summarization and formulation of CPGs based on CNLOs are documented in [126]. More generally, fundamentals of the synchronization of large population oscillators have been well documented in text books such as [127] and [128]. In the CNLO model, the major issue is to find the connections among oscillations in order to maintain stable phase difference, i.e. phase locking. As a result, development of coupling schemes will be the focus [126]. Besides, the theoretical work like formulation of an oscillator, stability analysis, and bifurcations are the issues in application of this model.

Many different CNLO systems can be applied to establish artificial CPGs. Matsuoka oscillator is one of the most commonly used oscillators in locomotion control of robots. This is a switched linear system (or piecewise linear system [126]) which consists of a set of ordinary differential equations (ODEs) with piecewise state variables [26, 129-131]. In [11], amplitude-controlled oscillator model was presented to control the locomotion of a robotic salamander. In the nonlinear ODEs of this oscillator, the phase difference, intrinsic frequency and the amplitude are explicitly defined with tunable parameters, which provide a simple way to on-line adjust the locomotion gaits by tuning the corresponding parameters [11, 28, 132, 133]. Hopf oscillator is also a commonly used tool to model CPG. This is another kind of amplitude-controlled oscillator. As it produces harmonic patterns, it is suitable to generate swimming gaits for fish robots [134-137]. Van Der Pol oscillator is another well-know tool in modeling of CPGs partly because CPGs based on large population of this oscillator can benefit from the fast phase locking [126, 138].

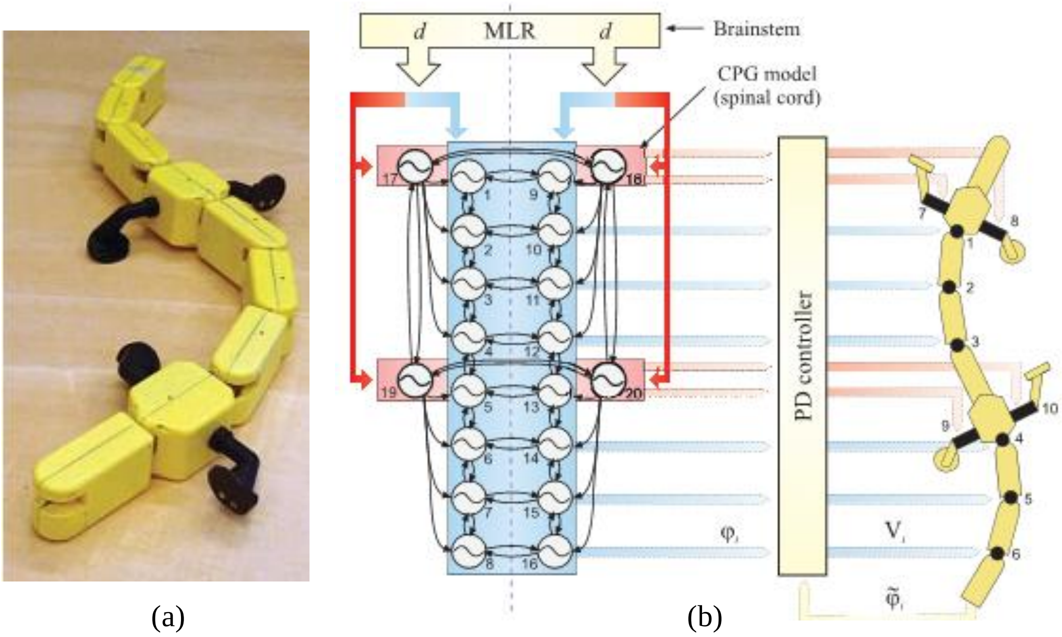
As the CNLO model is defined by ODEs, methods for analyzing ODEs can also be applied to the CPG model. Therefore, the CNLO based CPG model is more flexible in applications of locomotion control.

2.4.4 Control of Swimming Gaits by Using CPGs

A very excellent review on the applications of CPGs in locomotion control of robots can be found in the work of Ijspeert [15]. In his paper, Ijspeert identified several advantages of CPG in control of robots: (i) rhythmic pattern generated by CPGs is robust due to the limit cycle; (ii) CPGs are suitable for the distributed implementation; (iii) on-line modulation of CPG parameters are possible; (iv) sensory feedback can be coupled into the CPG model so as to shaping the CPG output; and (v) learning or optimization algorithms can be applied together with CPGs. This is a complete summarization of the benefits by using CPGs in robots. Actually, all these advantages will be utilized in the present thesis.

As CPGs are capable of producing high-dimensional coordinated oscillatory patterns, they are suitable to control the swimming gaits of fish robots. Fig. 2.33 shows an excellent example of CPG in locomotion control of robot. The robot can walk on land with the coordinated motions of four limbs and the six-DOF body performing a standing wave. In water, a traveling wave is produced by the body and swimming locomotion can be obtained. CPG model is used produce the gait control functions that coordinate motions and control the gait transition between walking and swimming. This is a typical application of CPG in locomotion control of robots. In the year of 2008 when [28] is published, the author argued that the CPG in swimming robot control was still rare.

However, in the past two year, many works attempt to use CPGs in control of swimming robots. In anguilliform swimming, a travelling undulation of the elongated body from head to tail causes the forward swimming. To simulate this kind of motion, CPG models are developed and implemented using finite-state machines in [139] and systems of coupled oscillators in [11].



(a) (b)

Fig. 2.33 An application of artificial CPG in controlling of swimming and walking of a salamander robot: (a) the robot prototype and (b) configuration of the CPG model (adopted from [11]).

Similar to the aquatic anguilliform swimming gait, the gait of a terrestrial snake robot can also be driven by CPG models, which is illustrated in [27] and [140]. On a fish prototype [26] inspired by Gymnarchus niloticus and modeled as a multi-joint dynamic system with parallel connections, a CPG governing the locomotion of fish is implemented. In another work [28], a CPG model is implemented on a fish robot with three actuated fins capable of swimming in water and crawling on firm ground. More recent applications of CPGs in swimming robot control are documented in [130, 133, 136, 141].

The control system of a biomimetic underwater vehicle generates actuator signals to drive the actual velocity and attitude of the vehicle to the value commanded by the guidance system [5]. There are many topics in the control of swimming machines. A good review of mechanisms and control of swimming robots can be found in [100]. In the work, the considerations for the control of biomimetic swimming machines are outlined [100]. This thesis talks mainly about deriving of swimming gait control functions and gait tracking control. For the former, current literatures focus mostly on the motion planning problems by using external cameras to provide the environment feedback [142]. The key issue is to devise a proper gait generator that can coordinate swimming gaits for multiple degree-of-freedom (DOF) and hence achieve desired performance such as obstacle avoidance and path tracking [76, 78, 143].

An on-line swimming gait generation without machine vision feedback is discussed in [28]. By using an artificial Central Pattern Generator (CPG), the fish robot in [28] can perform smooth gait transition from crawling to swimming according to the feedback provided by a water detection sensor. In another example of closed-loop swimming control [144], adaptation to irregular boundaries of locomotion area is implemented on an undulatory robot. In [135], the station-holding performance in one-dimensional unsteady flow is implemented on a multi-DOF fish robot. A nonlinear oscillator is used as gait generator and feedback control laws are designed to tune the output of the oscillator [135]. In this thesis, GA will be used to optimize the locomotion performance by tuning parameters of CPGs directly. Another way is illustrated in [145], where the frequency adaptive oscillator is used to manipulate the vorticities. The advantages of on-line motion planning and closed-loop swimming control in fish robots are well illustrated by these examples.

Once the gait signal (gait control function) is at hand, the tracking of the signal turns to be a classical motion control problem [30]. The low-level controller (motor controller) will be used to tracking the given swimming gaits. Because of the nonlinear hydrodynamics of water environment, such low level controller is necessary to regulate the system to track the desired the swimming gait. PID controllers are the most commonly used tools [30, 146, 147]. Similar work can also be found in [11, 23, 137, 148]. Other control methods are also implemented [146].

Morgansen *et al.* build a nonlinear model for the control of carangiform locomotion [149]. This model is based on quasi-steady fluid flow. By using this model, the authors propose gaits for forward and turning trajectories and analyze system response under such control strategies. Yang *et al.* studied the nonlinear tracking problem [150]. A time-varying, smooth Lyapunov-based feedback control law for tracking the nonholonomic fish robot is presented, which gives asymptotical convergence to a desired trajectory generated by a reference fish robot.

Ohgishi *et al.* uses a numerical and experimental method to study a double jointed fish robot utilizing self-excitation control [151]. The fish robot is composed of a streamlined body and a rectangular caudal fin. The body length is 280 mm and it has a DC motor to actuate the first joint and a potentiometer to detect the angle of the second joint. The signal from the potentiometer is fed back into the DC motor, so that the system can be self-excited. In order to obtain a stable oscillation and resultant stable propulsion, a torque limiter circuit is employed. Ariyanto *et al.* use the method of identification to figure out the control model of robotic fish [152]. By using data collected from inertial sensors, a suitable mathematical model explaining fish turning dynamics is obtained through system identification method. The identified model matches well with the experimental results and thus can be used for the design of controllers [152].

2.5 Summary

The chapter reviews the work related the current research in three major aspects: biological and biomimetic aspects of fish and their inspirations to the fish robots, the fish robot designs, and control of fish robots based on artificial CPGs or CNLOs. The research on fish robots actually covers more comprehensive aspects. Detailed analysis of those works and the inspirations to the present thesis will be specified in the following chapters.

A part of literatures directly relevant to the present thesis is reviewed in the chapter. A complete review on fish robots can hardly be achieved. Kinematics of real fish's swimming has long been a research focus in fish biology. Plenty of the results in this area (the body motion function, for example) can benefit the control fish robots. However, those biological works are not detailed in this chapter because the thesis will finally focus on the on-line generation of motion in a purely engineering perspective. The mechanical designs of fish robots are not confined in those with multiple rigid linkages, as listed in Section 2.2. Other types of mechanical designs such as those made up of smart materials and cable-driven elastic linkages are not reviewed. The mechanisms of majorities in current literatures, which are helpful for our own designs, are summarized. Finally, the control aspect of the fish robots covers varieties of knowledge in engineering. Even the artificial CPG itself can be a big topic. Those mostly related are included in this chapter.

Chapter 3 Off-line Generation of Swimming Gaits Using Sine Generators

This chapter is dedicated to develop an off-line approach to modeling the swimming gaits of multi-DOF fish robots by using sine function based generators. For fish robots, since linkages move with oscillatory patterns, it is natural to use the sine or cosine functions to model the swimming gaits. This is a kinematics aspect of fish swimming. The swimming mechanics will also be analyzed by using Lighthill's Elongated Body Theory. The chapter starts with the discussion on modeling swimming gaits by sine generators in Section 3.1. In Section 3.2, the research focuses on deriving the equations that can predict the swimming performance of a fish robot. Experiments on a robot will be presented in Section 3.4 to validate the theoretical analysis.

3.1 Modeling of Swimming Gaits

Generation of an undulatory waveform by fins can be regarded as a process that all fin rays of fish oscillate alternately in a certain sequence and drive the membrane of the fin to stretch or retract, which forms an undulatory wave along the fin. As a result, the gait planning for generation of undulatory motion becomes the problem of finding the sequence and oscillation functions of biomimetic fin rays. Application of sine functions in swimming gait generation will be discussed in this section.

3.1.1 Modeling of Harmonic Swimming Gait

The aim of swimming gait modeling is to find out the appropriate time-dependant control functions. For the parallel mechanism, the linkages are controlled to produce traveling waves. Because the linked fin rays are controlled independently within its workspace, the function describes the absolute angle of those links. The traveling wave is regarded as harmonic wave in most cases or non-harmonic wave for some other species (see Fig. 3.1).

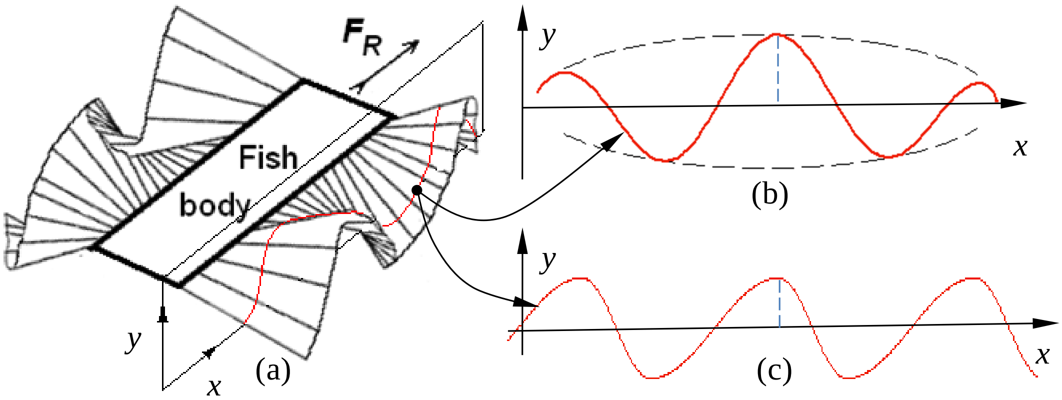


Fig. 3.1 Modeling of waveform on a long undulatory fin: (a) the profile of intersection line between fin surface and the x-y plane is normally modeled as (b) a harmonic wave or (c) non-harmonic wave.

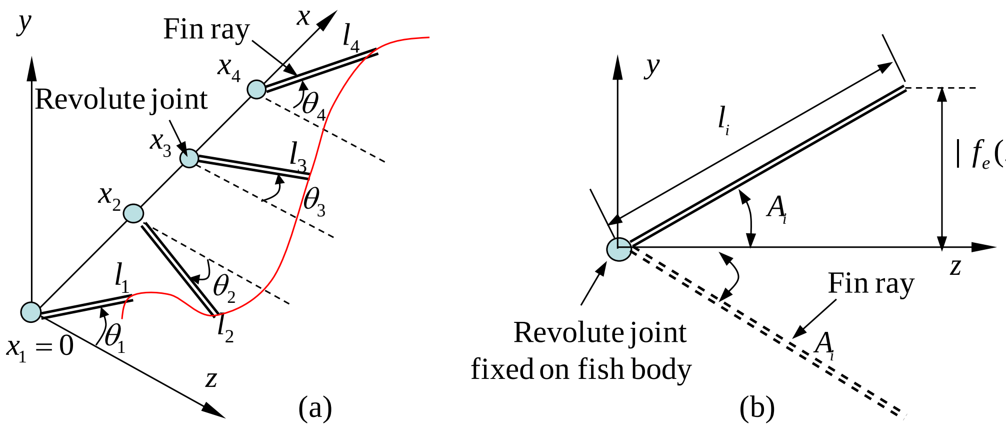


Fig. 3.2 Parallel linkage mechanisms used in the fish robots: (a) the parallel connection of fin rays and (b) the structure of single fin ray (rigid link).

Fig. 3.2 illustrates the parallel connection of four such links and the basic structure of a single fin ray (rigid link). To generate a travelling wave, the basic gait function is still in a sinusoidal form. The successful simulation of undulatory swimming depends on the determination of oscillation amplitude of fin rays and the phase lag between adjacent links. Here the position of first link joint is set as the origin, the longitude of fin as the *x* axis. For simple harmonic waveform the amplitude of oscillation A_i can be derived from the following equation:

$$A_i = \sin^{-1}\frac{|f_e(x_i)|}{l_i} \tag{3.1}$$

where $f_e(x_i)$ is the envelop equation (see Fig. 2.30), A_i is the oscillation amplitude, l_i is the length of fin ray and x_i is the longitudinal position of the i^{th} joint, as depicted in Fig. 3.2. The phase difference between two adjacent links depends on the wave number, the wavelength, the distance between two adjacent links and the number of links used to simulate the undulatory wave. Their relationship is given by following equation:

$$\varphi_{d,i} = \varphi_{d,1} - \frac{2\pi}{\lambda} x_i \quad i = 1,2,\ldots,n \tag{3.2}$$

where $\varphi_{d,i}$ is the phase difference between adjacent fin rays. Gait functions for the generation of simple harmonic undulatory wave by parallel connected fin rays can then be obtained. Their motions are governed by a set of time-dependant sinusoidal function given by:

$$\theta_i(t) = A_i \sin(2\pi f t + \varphi_{d,i}) \tag{3.3}$$

The bi-directional swimming locomotion can be obtained by changing the sign of the phase difference, $\varphi_{d,i} = -\varphi_{d,i}$.

3.1.2 Modeling of Non-harmonic Swimming Gait

In Fig. 3.1 and many of current literatures [26, 54-59], an assumption is made that the fin rays are rigid ones and they do not bend during swimming. This is a simplification of fin rays of real fish, which are usually flexible ones. Bending of flexible fin rays exists due to the reactive force when they oscillate in water. With such an assumption, two kinds of motion will be ignored. Firstly, if fin rays in Fig. 3.1 or Fig. 3.2 are flexible, undulatory motion will appear along the *z*-axis. This undulation can help to reduce the reaction force on the fin membrane, as manta ray's case mentioned in Section 2.3.1 [77, 90].

Another motion loss is the no-harmonic waveform (See Fig. 3.3). It has been reported that Gymnotiform swimmers produces non-harmonic wave pattern during forward swimming [109]. In [109], it is proved that such wave is caused by the effect of the natural compliance of flexible fin rays to the oncoming flow. As a result, by using inclined fin rays (see Fig. 3.4b), non-harmonic waveform can be produced.

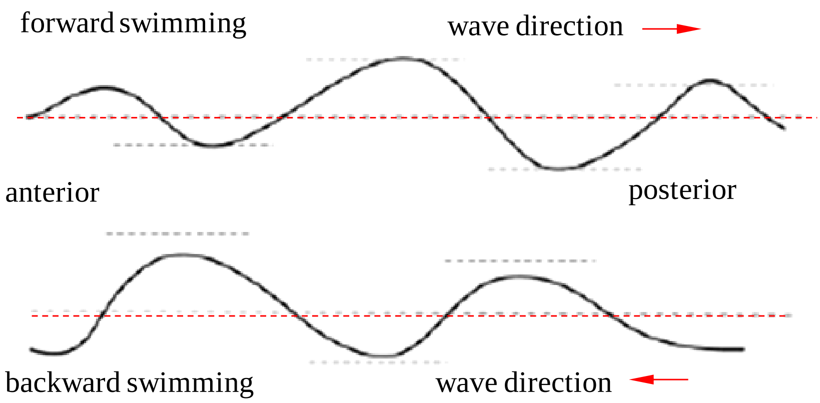


Fig. 3.3 Non-harmonic waveform generated by Gymnotiform swimmers in forward/backward swimming.

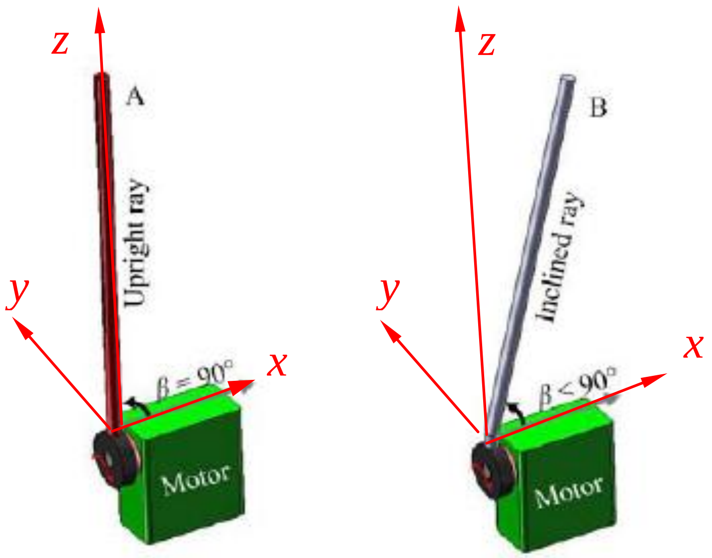


Fig. 3.4 The fin rays with different inclined angles. The left unit with β=90 °is named the upright ray, while the right unit with β< 90 °is called the inclined ray [109].

Non-harmonic gaits are also observed in Manta Rays. Klausewitz studied the flapping of manta ray and found that the upstroke only takes about half time of the down-stroke in one flapping cycle [36]. This indicates that harmonic functions (sine function, for example) may not be an accurate way to simulate the locomotion pattern of a manta ray. The forward thrust of manta ray can be produced on both upstroke and down-stroke flapping. However, the down-stroke motion will contribute major part of thrust [36]. Therefore, it is worth applying non-harmonic gait patterns in locomotion control of fish robots may in order to improve the propulsion performance.

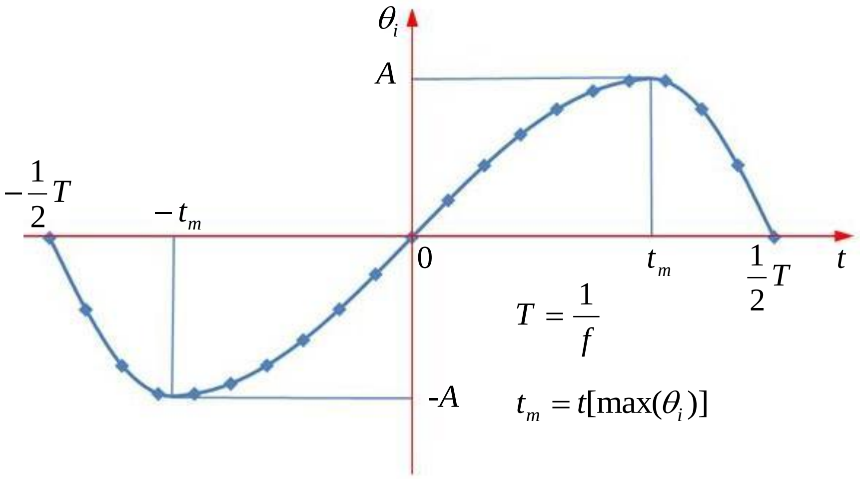


Fig. 3.5 A curve of non-harmonic oscillation for the i^{th} fin ray.

The gait function of the single fin ray developed in [109] is still a harmonic function. Because of the inclination of fin rays, the wave generated by the undulatory fin is asymmetrical [109]. Instead of using this hardware method, gait functions for general designs with straight (upright) fin ray will be obtained in the following derivation. By changing the wave shape parameters, fish prototype can simulate waves with different shapes.

It can be assumed from Fig. 3.3, that this kind of wave is formed by oscillations of particles with more than one frequency component in a continuum of media. Therefore, the aim of motion planning is to find the oscillation functions with those frequency components for every biomimetic fin rays. In order to perform parametric study in future work, the oscillation is modeled by cubic spline curve fitting with a parameter α that defines the shape of the curve. Fig. 3.5 shows a curve of non-harmonic oscillation for a biomimetic fin ray. The non-dimensional parameter α is defined as $t_m/0.5T$. Then

$$t_m = t[\max(\theta_i)] = \frac{1}{2}\alpha T \tag{3.4}$$

According to Eq. (3.4) and Fig. 3.5, if α=0.5, the curve becomes symmetrical, if 0.5<α<1, the oscillation of fin rays will cause a forward swimming waveform as depicted in Fig. 3.7(a) and a backward swimming waveform is obtained as 0<α<0.5. Next cubic spline is applied to fit this curve in time domain [0, T/2]. The general form of the fitting function is

$$\theta_i(t) = B_3 t^3 + B_2 t^2 + B_1 t + B_0 \tag{3.5}$$

From Fig. 3.5, conditions (by assuming unit amplitude, A=1) are obtained as

$$\theta_i(t_m) = 1, \quad \theta_i(0) = \theta_i(\frac{1}{2}T) = 0 \text{ and } \dot{\theta}_i(t_m) = 0 \tag{3.6}$$

and subsequently

$$\begin{pmatrix} B_3 \\ B_2 \\ B_1 \\ B_0 \end{pmatrix} = \begin{pmatrix} -\dfrac{2t_m + T/2}{t_m{}^2(t_m - T/2)^2} \\ -1/t_m{}^2 - 2t_m B_3 \\ 2/t_m + t_m^2 B_3 \\ 0 \end{pmatrix} \tag{3.7}$$

Now the gait function $\theta_i(t)$ within [0, T/2] can be obtained. Because of the periodicity of gaits, the gait function should be odd expanded in the region of [-T/2, T/2]. After the odd expanding, the oscillation equation can be given by:

$$\theta_i(x,t) = \begin{cases} B_3 t^3 + B_2 t^2 + B_1 t & t \in (0, T/2] \\ B_3 t^3 - B_2 t^2 + B_1 t & t \in (-T/2, 0] \end{cases} \tag{3.8}$$

It should be noted that the gait function given by Eq. (3.8) is still coupled with the position of fin rays implicitly. To control the separate actuators, it is necessary to find control functions dependent only on time. Since the wave is non-harmonic which contains more than one frequency component, it is desired to express this wave as

$$y = f_e(x) \sum_{j=1}^{\infty} b_j \sin\left[j(2\pi f t + \varphi_{d,i}) \right] \tag{3.9}$$

where $\varphi_j(x)$ is the phase lag of different fin rays, b_j is the coefficient of components with different frequencies in Fourier series of $\theta_i(t)$ and $\varphi_{d,i}$ is determined by Eq. (3.2). Therefore, the gait for separate fin rays should be

$$\theta_i(t) = f(x_i, t) = f_e(x_i) \sum_{j=1}^{\infty} b_j \sin(2j\pi f t + j\varphi_{d,i}) \tag{3.10}$$

Backward swimming gait can be obtained by changing the sign of φ_i. In the actual application, it is not necessary to include infinite number of frequency components. Out experiments show that smooth swimming can be achieved by using only three components.

3.2 Analysis of Swimming Performance Based on Lighthill's Elongated Body Theory

This section discusses the application of Lighthill's Elongated Body Theory (EBT) [101] in analysis of swimming performance of fish robots. The theory makes use of a set of partial differential equations to predict the thrust and efficiency of swimming under rhythmic and symmetric body motions [101, 153]. EBT is a descriptive tool, other than a predictive tool for fish swimming. It can well explain the relation between the swimming motion pattern and the swimming performance. It cannot be used to predict the performance. The section shows that by applying certain assumptions, it can also be extended to predict the swimming performance of fish robots [153].

3.2.1 Lighthill's Elongated Body Theory

In 1971, Lighthill developed the Large-amplitude Elongated Body Theory (EBT) [154]. In the theory, the fish is modeled as an elastic thin plate. Lighthill's theory is applicable to long slender fish without prominent body-fins and a gradual taper in dimensions to the caudal fin. In order to simplify the problem, flow is usually assumed inviscid, incompressible and irrotational [78, 155, 156]. Based on this theory, kinematics model of the fish locomotion can also be obtained. In the work by Tan *et al*. [105], it is used to establish a turning motion control framework for BCF type locomotion. Other applications can be found in more recent studies [18, 99, 105, 153, 157]. For a swimming robot, it is important to explore the swimming performance regarding the swimming energy consumption, velocity and thrust. Lighthill's EBT theory is sufficient to give these quantities.

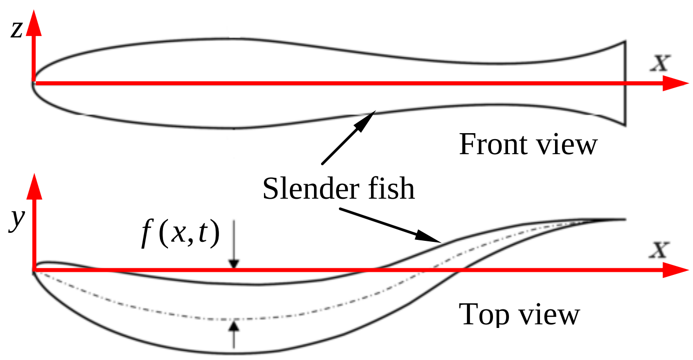


Fig. 3.6 A slender fish model.

Lighthill [101] was the first to apply the methods of slender body theory (see Fig. 3.6) to fish swimming in an inviscid fluid medium. The theory gives a solution to evaluation of swimming performance at the equilibrium speed, i.e. the steady swimming. Although it intends to explain a biological phenomenon, it is of great value when used to the engineering area, especially to the biomimetic fish control. According to Lighthill's model [8], the velocity of a fish body's cross section can be seen as a moving water slice. The velocity of lateral swimming motion, $v(x, t)$, can be derived by the total differential of the body motion function:

$$v(x,t) = \frac{\partial f(x,t)}{\partial t} + U\frac{\partial f(x,t)}{\partial x} = Df \tag{3.11}$$

in which

$$D = \frac{d}{dt} = \frac{\partial}{\partial t} + U\frac{\partial}{\partial x} \tag{3.12}$$

Note that U is the forward steady swimming speed of fish, t is time, and D is a total differential operator used to simplify the derivation. The lateral force, F_y, exerted by the body on the water slice can be computed by the rate of change of the momentum:

$$F_y = D[M(x)v(x,t)] \tag{3.13}$$

where $M(x) = \rho\pi(H/8)^2$ is the apparent mass of the cross section per unit length, H is the height of the propulsor and ρ is the water density. In the following derivation, assume that M and v denote $M(x)$ and $v(x,t)$, respectively. The power used to generate lateral motion, P, can then be obtained as follows:

$$P(t) = \int_0^L \frac{\partial f(x,t)}{\partial t} D(Mv)dx = \frac{\partial}{\partial t}\int_0^L (Mv\frac{\partial f}{\partial t} - \frac{1}{2}Mv^2)dx + UMv\frac{\partial f}{\partial t}\bigg|_{x=L} \tag{3.14}$$

in which L is the total length of fish body. In another hand, by applying the energy balance principle, the energy produced by the lateral movements of fish can also be thought of the sum of two following parts, as given by

$$P(t) = F_T U + \int_0^L D(\frac{1}{2}Mv^2)dx \tag{3.15}$$

where F_T (= F_x) is the thrust force and F_TU the propulsive power, the second term (the integration) on the right hand side is kinetic energy acquired by water slice and fish body. By comparing the two expressions of $P(t)$ from Eq. (3.14) and (3.15), the thrust force F_T can be expressed as

$$F_T(t) = \left[M(v\frac{\partial f}{\partial t} - \frac{1}{2}v^2) \right]_{x=L} - \frac{\partial}{\partial t}\int_0^L Mv\frac{\partial f}{\partial t}dx \tag{3.16}$$

3.2.2 Application of EBT in Multi-DOF Fish Robots

Note that the EBT is a descriptive tool, other than a predictive tool for fish swimming. When the swimming motion pattern (the BMF) is given and the swimming speed is known, the thrust and power consumption can be computed by using Eq. (3.16). That is to say, it can well explain the relation between the swimming motion pattern and the swimming performance. It cannot be used to analyze the performance. For fish robots, it is needed to predict the swimming speed and thrust under given BMF, which is a different problem. Therefore, the EBT should be modified in order to become a predictive tool. This can be done by applying the following assumption.

At steady swimming, the swimming speed reaches equilibrium at a constant value *U*. It is natural to assume that, under steady swimming, i.e. at the equilibrium forward velocity *U*, the averaged thrust over one period of fish undulating or oscillation is balanced by the averaged drag force of water. This equilibrium state can be reflected from the balanced force on the robotic fish as:

$$\overline{F_T} \approx \overline{F_d} = \frac{1}{2}\rho C U^2 \tag{3.17}$$

where C is a coefficient related to characteristic cross-sectional area of fish and its body drag. The expression of swimming speed U can then be predicted by equating Eqs. (3.16) and (3.17):

$$U = \sqrt{\frac{\overline{(\partial f / \partial t)^2}}{\overline{(\partial f / \partial x)^2} + 64C / (\pi H^2)}} \tag{3.18}$$

By substituting Eq. (3.18) into Eq. (3.17), the thrust force can be predicted as

$$\overline{F_T} \approx \frac{\rho C \overline{(\partial f / \partial t)^2}}{2\left[\overline{(\partial f / \partial x)^2} + 64C / \pi H^2 \right]} \tag{3.19}$$

By using above mentioned derivations, the swimming speed and thrust can be predicted. Equations (3.18) and (3.19) build a bridge between the swimming motion patterns and the consequent swimming performance.

3.3 Experiments of Gait Control on Fish Robots

The EBT is applied to analyze the swimming performance of the fish robot. This theoretical tool is validated by experiments on NKF-II. The fish robot is controlled by the gait functions discussed in Section 3.1.1. The theoretical and experimental results are presented in this section.

3.3.1 Theoretical Prediction of Swimming Speed

When a swimming motion pattern (described by the BMF) is given, Eq. (3.18) can be used to evaluate the swimming performance. It is necessary to know the value of C in order to compute the swimming speed. One way is to measure the drag by pulling a fish body at a given speed in still water [108] An alternative way to experimental determination is to perform a CFD simulation on the fish robot [16]. In our research, we adopted the first method with the configuration shown in Fig. 3.8 since a robot prototype is available to facilitate the experiment. The propulsor prototype was pulled to move in water by a string at certain speed. A force sensor was used to measure the tension on the string. At steady swimming, it can be assumed that the constant moving is the result of balance between the drag and the tension. By using Eq. (3.17), the value of C can be obtained and C = 0.181 m^2 for NKF=II.

Different body motion functions will result in distinct swimming speed. In order validate the equations derived in this chapter, the following BMF is adopted to test the gait generation and performance prediction:

$$f(x,t) = A\sin(kx - \omega t) \tag{3.20}$$

where $\omega=2\pi f$ and k is a wave number. The partial derivatives of Eq. (3.20) used in Eq. (3.18) and (3.19) should be found for the given motion of the fish prototype. The average value over one cycle of rhythmic swimming should also be derived. These time-averaged values are simply obtained by integrating each function of the body motion and dividing by the period.

Partial derivatives with respect to time t

$$\left.\frac{\partial f}{\partial t}\right|_{x=L} = -\omega A\cos(kL-\omega t)$$

$$\left.(\frac{\partial f}{\partial t})^2\right|_{x=L} = \omega^2 A^2 \cos^2(kL-\omega t)$$

$$\left.\overline{(\frac{\partial f}{\partial t})^2}\right|_{x=L} = \frac{\omega}{2\pi}\int_0^{\frac{\omega}{2\pi}} \omega^2 A^2 \cos^2(kL-\omega t)dt = \frac{1}{2}\omega^2 A^2$$

Partial derivatives with respect to longitudinal position x

$$\left.\frac{\partial f}{\partial x}\right|_{x=L} = kA\cos(kL-\omega t)$$

$$\left.(\frac{\partial f}{\partial x})^2\right|_{x=L} = k^2 A^2 \cos^2(kL-\omega t)$$

$$\left.\overline{(\frac{\partial f}{\partial x})^2}\right|_{x=L} = \frac{\omega}{2\pi}\int_0^{\frac{\omega}{2\pi}} k^2 A^2 \cos^2(kL-\omega t)dt = \frac{1}{2}k^2 A^2$$

and

$$\left.\overline{\frac{\partial f}{\partial x}\cdot\frac{\partial f}{\partial x}}\right|_{x=L} = \frac{\omega}{2\pi}\int_0^{\frac{\omega}{2\pi}} -\omega k A^2 \cos^2(kL-\omega t)dt = -\frac{1}{2}\omega k A^2$$

By substituting the partial derivatives into Eq. (3.18), the steady swimming speed is predicted as:

$$U = \frac{2\pi f}{\sqrt{k^2 + 96C/(\pi A^2 H^2)}} \tag{3.21}$$

The geometries of fish prototype are L=0.65m and H=0.15m. By substituting the geometry and various frequency-amplitude combinations into Eq. (3.21), the quantities related to the steady swimming speed can be predicted. Theoretical plots of steady swimming speed as functions of amplitude for various frequencies are given in Fig. 3.7. The overall trend is that swimming motion with the higher frequency or amplitude theoretically generates the higher speed.

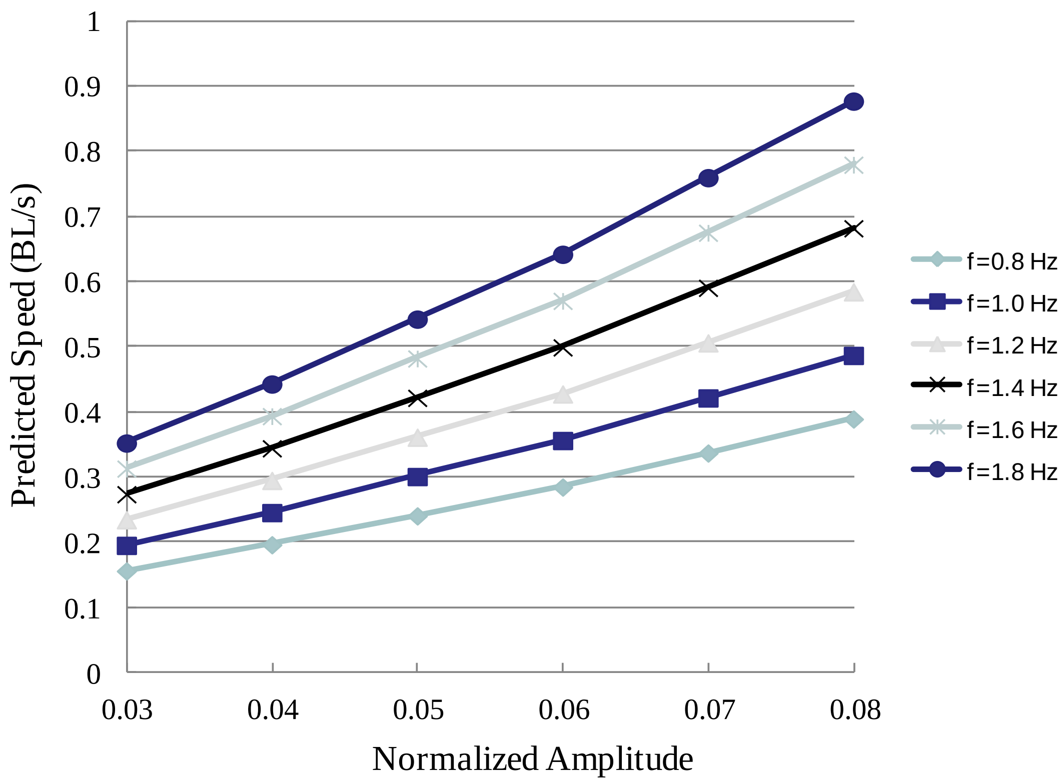
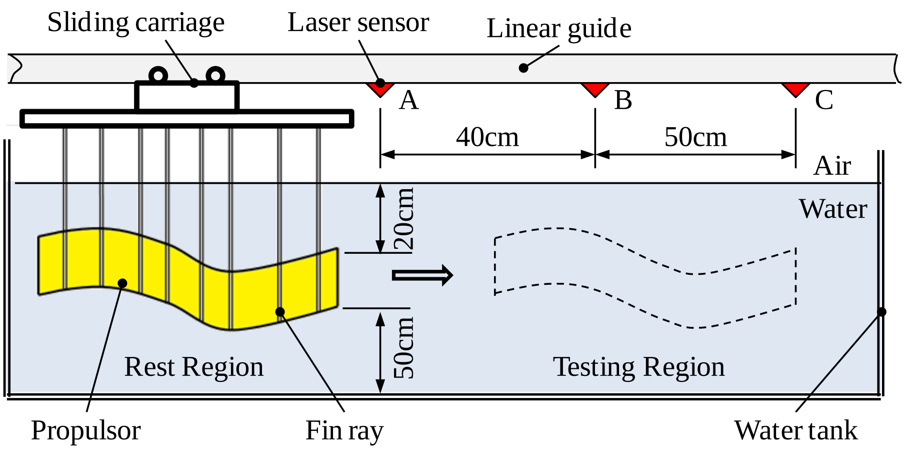


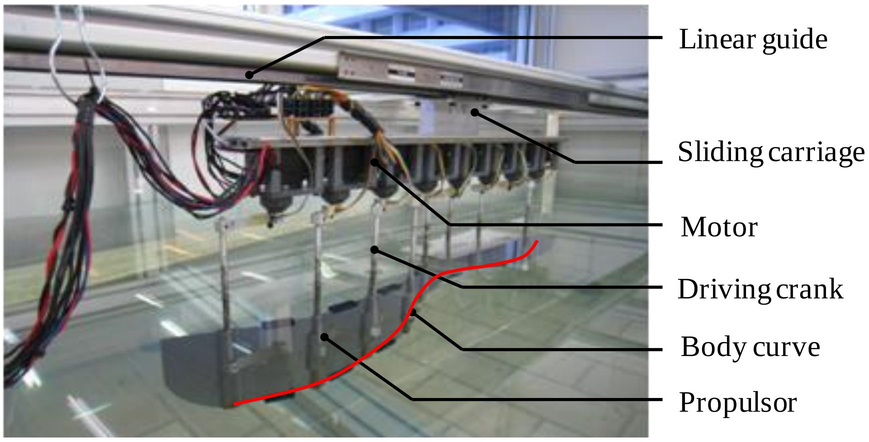
Fig. 3.7 Swimming speed predicted by using Eq.(3.21) for NKF-II (C = 0.181 m^2).

3.3.2 Experiments on NKF-II

The NKF-II is designed for free-run mode in open water area. Here, *free-run* refers to that the fish robot swims in open water area (for example, a large pool) without any external constrains like a harness or rope to constrain its motion. For ease of the measurement of swimming performance, a duplicate of the biomimetic propulsor in the fish robot is made and tested in a lab water tank, as shown in Fig. 3.8. In the experiments, the prototype performs straight-line swimming on the linear guide. Three laser sensors (A, B, and C) are used to detect the edge of the head and control the process. The inactive parts of the prototype (including actuators, wires, sensors, and structural frames) are exposed in air while the active part of the prototype (the propuslor) is submerged in water to propel the entire prototype. The bulky inactive part is intended to be off the water because the large body drag can be avoided in this way. Given that the friction force from the linear guide is insignificant, this configuration allows us to concentrate only on the optimal swimming patterns without considering the effects casued by the body drag and frictions.



(a)



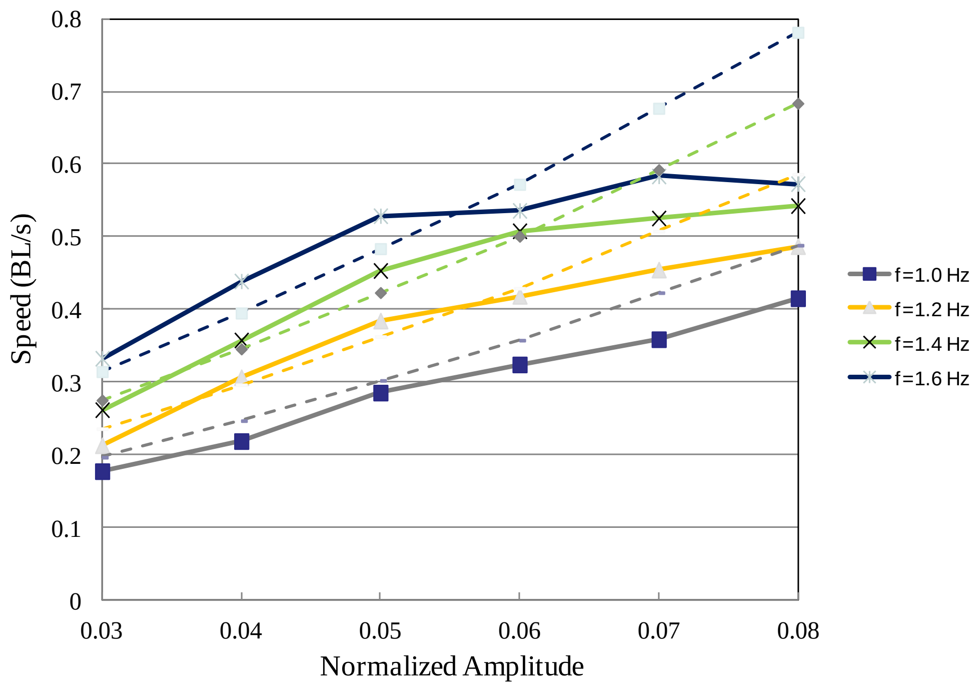
(b)

Fig. 3.8 (a) Front view illustration of experiment configuration. The propulsor is in the rest position. Three laser sensors (A, B, C) are used to detect positions. (b) The propulsor prototype hanging in the lab water tank.

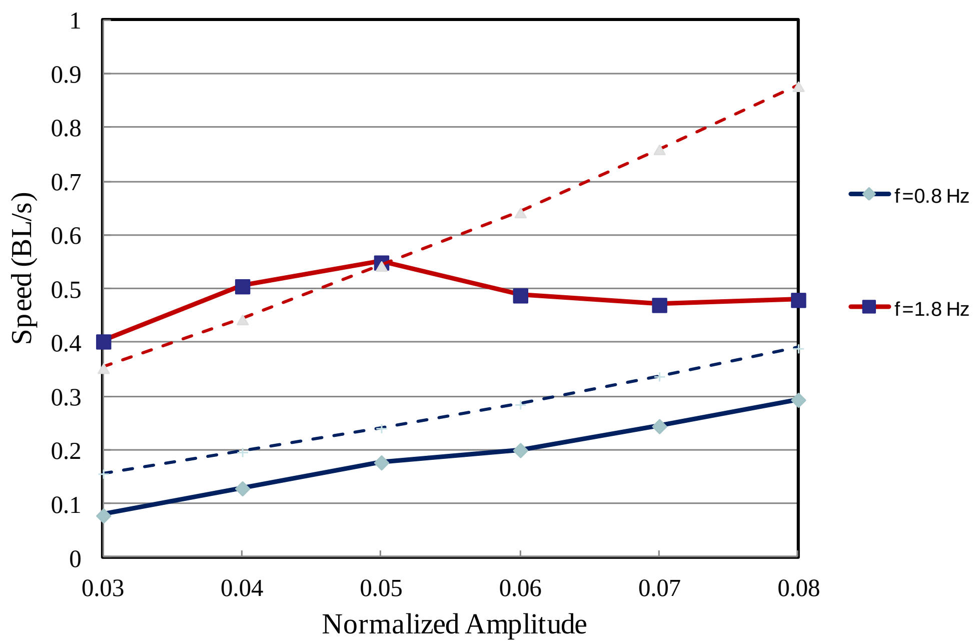
By using this simple harmonic wave function, the eight swimming gaits can be derived from Eq. (3.1)-(3.3) and given by

$$\theta_i(t) = \sin^{-1}(\frac{A}{0.06})\sin(2\pi f t - \frac{\pi}{4}) \quad i=1,2,\ldots,8 \tag{3.22}$$

Empirical tests have been conducted to evaluate the swimming speed of the biomimetic propulsor. Eq. (3.22) is adopted to control the eight motors. Different frequency and amplitude were applied. The experiment data are listed in Table 3.1. Comparison between the swimming speed and controlling parameters is shown in Fig. 3.9.



(a)



(b)

Fig. 3.9 Comparison of theoretical speed (dashed curves) and experimental speed (solid curves): (a) theoretical calculation is in good agreement with the experimental results in the frequency of 1.0-1.6 Hz and amplitude of 0.03-0.06 BL. The speed and amplitude are normalized in body length.

Table 3.1 Data of speed collected in experiments

Amplitude	Speed at different frequencies, (BL/s)					
(per BL)	0.8 Hz	1 Hz	1.2 Hz	1.4 Hz	1.6 Hz	1.8 Hz
0.03	0.08	0.17	0.21	0.26	0.33	0.40
0.04	0.13	0.22	0.31	0.36	0.44	0.51
0.05	0.18	0.29	0.38	0.45	0.53	0.55
0.06	0.20	0.32	0.42	0.51	0.54	0.49
0.07	0.25	0.36	0.45	0.53	0.55	0.47
0.08	0.29	0.42	0.49	0.54	0.56	0.48

A general trend can be observed from experiments that the swimming speed increases as the increasing of both amplitude and frequency, which is in accordance with the theoretical prediction. Both groups of data are plotted in Fig. 3.9 for a comparison. It can be seen from Fig. 3.9a that in 0.8 Hz < f < 1.8 Hz and A < 0.06 BL, the theoretical speeds is in good agreement with the experimental one. Fig. 3.9b shows two cases that there are significant discrepancies between the theoretical speeds and the experimental speeds. The speed decreases with the increasing of amplitude at f=1.8 Hz. This speed drop is caused by the power limitation of electrical motors because power drain exceeds the capability of servo motors. Additionally, when large amplitude is applied, much energy is wasted in pushing water aside, other than propel the fish robot forward. This reason also accounts for the dropping of the speed at large amplitude. At f=0.8 Hz, the experimental device swims at relatively low speed, the friction on the sliding carriage (see Fig. 3.8) may dominate and it reduces the swimming speed.

3.4 Summary

This chapter discussed sine function based swimming gait planning and an analytical explanation of swimming mechanics of fish robots. The solution to gait planning problem for fish robot with parallel mechanism is illustrated and tested in experiments. The EBT is applied to analyze the swimming performance of a fish robot and validated by experiments on NKF-II. This chapter shows an off-line control method for fish robots, which is a conventional way to obtain swimming locomotion. It does not solve the problem that in what manner the swimming performance of a fish robot will be better. This question will be answered in the following chapters by using on-line gait generation method.

Only basic fish locomotion, the straight-line forward/backward swimming, is addressed in this chapter. Other types of locomotion such as turning, fast start, and C-sharp turning are not discussed. In [105] and [104], a turning control framework has been developed. More locomotion types have been well studied in [71, 106, 158]. These works will inspire the future studies on more versatile locomotion.

Chapter 4 On-line Generation of Swimming Gaits Using Nonlinear Oscillators

Sine functions are commonly used models to generate oscillatory swimming gait of fish robot and have been successfully implemented in many applications [12, 54, 56, 109]. However, there are some limitations with these generators. Firstly, it is not suitable for on-line gait generation and gait tuning. The amplitude, frequency and the phase of a sine generator is defined based on the pre-selected fin motions. They can hardly be on-line tuned for the need of adaptation to the environment because this may cause the unsmooth motions. Secondly, the modeling method is not generic. When used for different fin motions or different fin prototypes it has to be tailored to accommodate various applications. To avoid these disadvantages, a nonlinear oscillator is adopted as the swimming gait generator.

The nonlinear oscillator provides a solution to on-line adjustment of the swimming gait according to the environment feedback, which will make the adaptive swimming locomotion possible for fish robots. Combined with feedback control laws, the parameters of the generator can be dynamically tuned, from which the adaptive swimming can be obtained. As an example of adaptive swimming, station-holding performance exhibited by real fish is studied and implemented on a biomimetic fish robot. A control method that enables the swimming of robotic fish systems adaptive to variational flow will be developed. The application of a nonlinear oscillator in swimming gait generation is discussed in Section 4.1. As a case of adaptive swimming, station-holding performance of real fish in adverse unsteady flow is studied and implemented on a fish robot, which is discussed in Section 4.2. Simulations are performed to test the feedback control laws in Section 4.3. Experimental results will be presented in Section 4.4.

4.1 Gait Generator Based on a Nonlinear Oscillator

High performance aquatic locomotion needs on-line motion planning to achieve the swimming adaptation to the environment surrounding fish robots. This property can hardly be achieved with sinusoidal function based gait generator because the on-line changing of parameters (phase lag, frequency or amplitude of a sine generator) of these generators may cause discontinuity or jerk of motions. A mathematic model that generates on-line steady harmonic gaits is necessary. The gaits are also required to be adjusted continuously and smoothly according to environment feedback. A nonlinear oscillator will be applied to model the rhythmic swimming gait.

The normal sine generator is not robust because when there is an abrupt change in amplitude or frequency, the motion may not be smooth, even not continuous. Fig. 4.1 shows the outputs of a sine generator in the case that amplitude or frequency is increased at the time instance *t**. It can be seen that the motion is not smooth in both cases. The decrease of those parameters causes the similar results. The smooth transition of swimming gaits is important for fish robots because it ensures that there is no jerk or discontinuity in locomotion. The actuators therefore get free from the potential damage caused by the jerk.

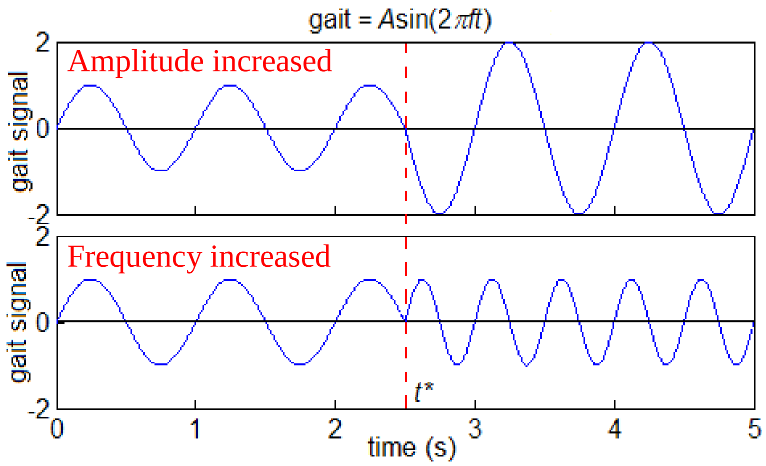


Fig. 4.1 Outputs of the sine generator where amplitude A is changed from 1 to 2 (upper) and frequency f is changed from 1 to 2 (lower). All changes of parameters occur at time t^*=2.5s. At other time (t^* is random), discontinuity may also appear.

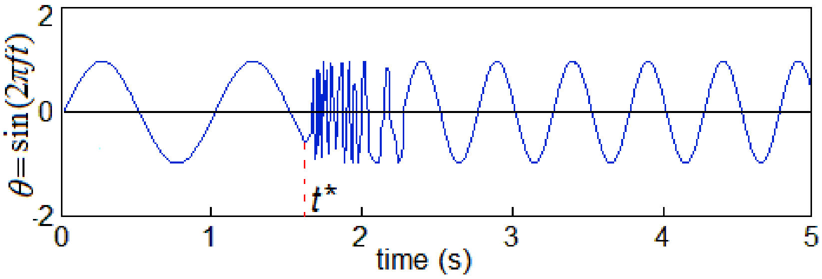


Fig. 4.2 An example of unsmooth motions recorded in experiment. The tail fin of NAF-II was driven by a DC motor and controlled with a sine function. The angular position is shown and normalized to 1. The change of frequency is changed from 1Hz to 2Hz at t^*.

Fig. 4.1 only shows theoretical case that the online changing of parameters will cause unsmooth of motions. In actual experiments on the tail oscillation of NAF-II (see Fig. 2.27), the situation is even worse. A sine signals (frequency f = 1 Hz) was applied to the fish tail and the tail beat is recorded by using the rotary encoder. At a random time t^*, the frequency was manually changed to 2 Hz. This changing causes the irregular shaking of the fish tail. The shaking appeared because in this case, there is a large discrepancy between the actual position of the motor and the command signals. This situation would not stop until the motor could track the control command after several seconds. This phenomenon was recorded and plotted in Fig. 4.2

In order to solve the above-mentioned problems, a nonlinear oscillator is applied to model the swimming gait. Here the Hopf oscillator is selected, which is defined by:

$$\dot{u} = k(A^2 - u^2 - v^2)u - 2\pi f v \tag{4.1}$$

$$\dot{v} = k(A^2 - u^2 - v^2)v + 2\pi f u \tag{4.2}$$

where u and v are two state variables of the oscillator which are all functions of time t ($t \geq 0$), A is a positive number that determines the amplitude of the steady state oscillation, f is the oscillation frequency and k is a positive constant which regulates the speed of convergence.

4.1.1 Generation of Harmonic Swimming Gait Pattern

Many different oscillators can be used for modeling the motion pattern of oscillatory motion of caudal fin [22, 28]. Hopf oscillator is chosed because of its harmonic steady state output, which is suitable for modeling the swimming gaits along a straight path. The output of the oscillator is illustrated in Fig. 4.3. The steady state output of the Hopf oscillator can be given by

$$u \approx A\cos(2\pi ft + \varphi_0) \tag{4.3}$$

$$v \approx A\sin(2\pi ft + \varphi_0) \tag{4.4}$$

where φ_0 is determined by the initial states of the oscillator.

Another attraction of this oscillator is its limit cycle, which is illustrated in Fig. 4.4. Starting from eight different points (four inside the limit cycle and four outside the limit cycle), the states always asymptotically converge to a limit cycle with radius A as time gets large. Given that (0, 0) is a stable equilibrium point (a point attractor) of the Hopf oscillator, the initial value should avoid this point. They can be theoretically set as any values near the limit cycle.

Robustness of the oscillator is also a significant property. Under perturbations of two cases: the system starts evolving from a random initial condition and parameters of the oscillators are changed, the system can also continuously converge to a new stable state. The output of the oscillator is always kept smooth owing to the integration in solving the nonlinear differential equations [15]. As shown in Fig. 4.3, initial state of the oscillator is set to (0.5, 0.5) at t=0s. The oscillation naturally evolves into a stable harmonic pattern with A=1 and f=1 Hz. This feature makes it possible the posture of a robot will not necessarily return to a prescribed status when the system restarts in locomotion control. Instead, the limbs of robots can start to move from any initial positions in the workspace.

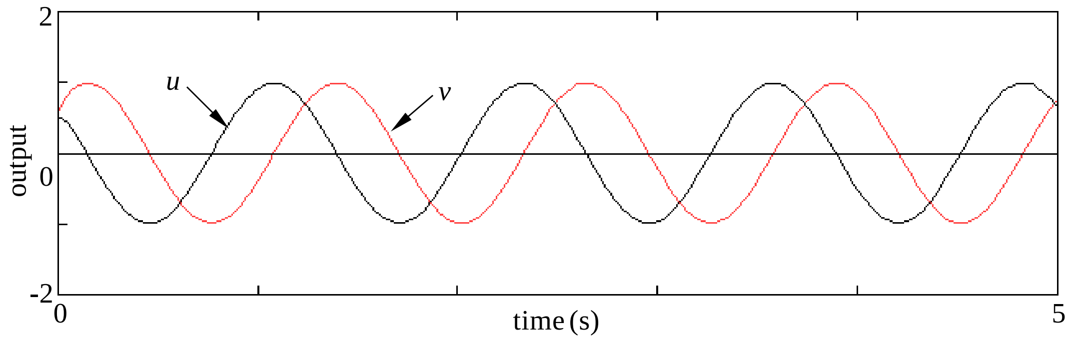


Fig. 4.3 Output of Hopf oscillator. The initial states are set to (0.5, 0.5). The amplitude and frequency are normalized to 1.

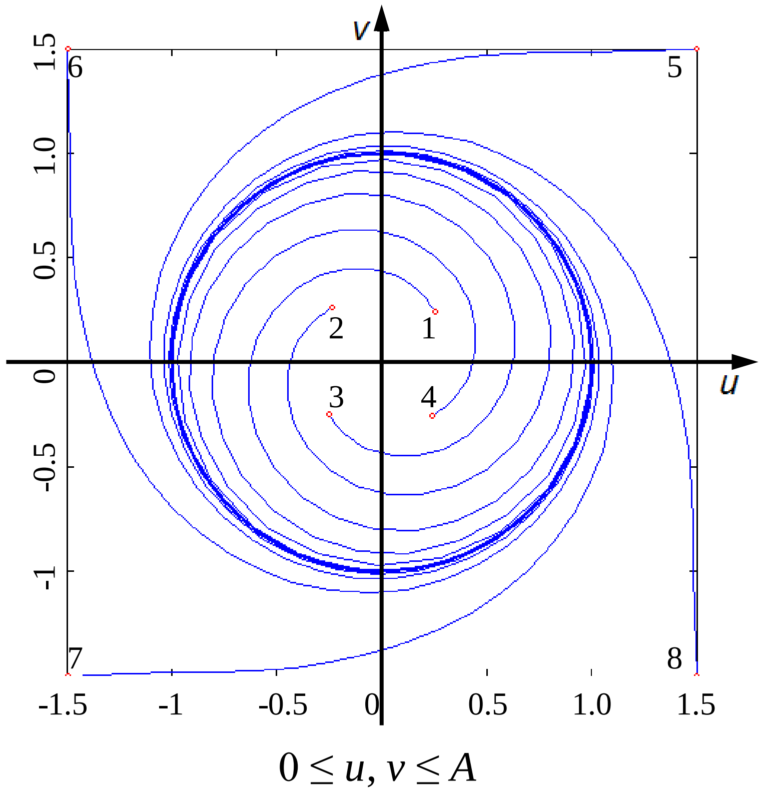


Fig. 4.4 The phase portrait of the oscillator. The figure illustrates that the system starts evolving from eight different initial states (point 1-8) and they finally converge to the limit cycle. The radius and frequency are all normalized to 1.

In Fig. 4.5, when the radius is changed from 1 to 2 (upper) or frequency is changed from 1 Hz to 2 Hz (lower), it can be seen that the oscillations naturally become stronger or faster with smooth transition respectively, which is different from the cases in Fig. 4.1. When used in control of swimming, this asymptotic stability provides possibility to generate gradually changing gait signals, which makes the adaptive swimming gaits generation feasible. The robustness of the oscillator allows a continuous and smooth transition of gait. And the asymptotic stability ensures that the transition always converges to a stable one.

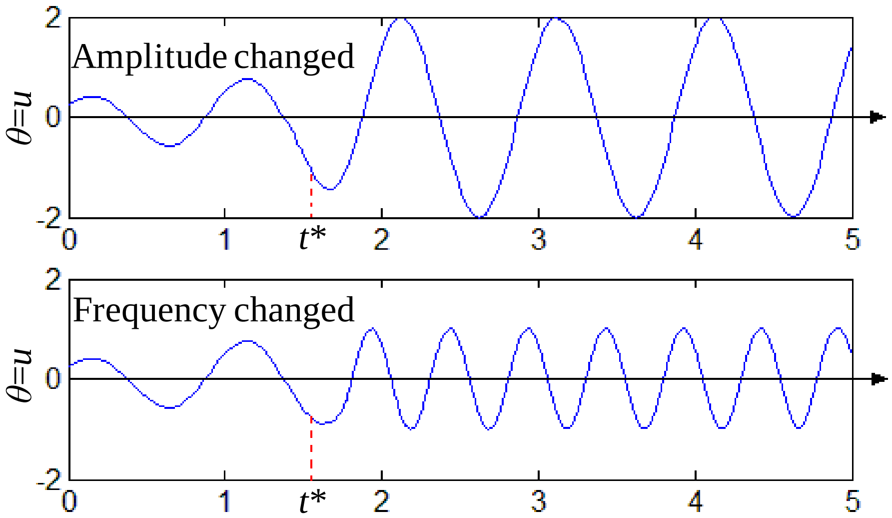


Fig. 4.5 An example of fin ray motion controlled by Hopf oscillator. State u of the oscillator is used as the control signal, i.e. $\theta(t)=u(t)$. The initial state is the same as point 1 in Fig. 4.4. The radius is changed from 1 to 2 (upper) or frequency is changed from 1 Hz to 2 Hz (lower) at t^*.

4.1.2 Generation of Non-harmonic Swimming Gait Pattern

Steady states output of the Hopf oscillator have harmonic patterns (see Fig. 4.3). Therefore it is suitable to model the swimming gait depicted in Fig. 3.5. Non-harmonic gait can also be achieved with Hopf oscillator by applying adaptive frequencies in one cycle of fin ray oscillation. It can be assumed that the asymmetry of the wave shape depicted in Fig. 4.6 is caused by the different frequencies involved in one cycle. The switching of frequencies depends on the shape of the non-harmonic wave. It can be assumed that higher frequency component f_1 accounts for the fast rising phase (upstroke) and lower frequency component f_2 explains the slow declining phase (down stroke), respectively.

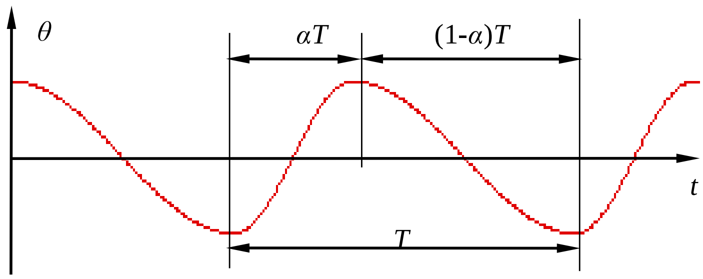


Fig. 4.6 Non-harmonic gait pattern with adaptive frequencies.

According to the symbol definition in Fig. 4.6, we have

$$f_1 = \frac{1}{2\alpha T} \tag{4.5}$$

$$f_2 = \frac{1}{2(1-\alpha)T} \tag{4.6}$$

where α is a shape ratio that determines how much time the rising phase takes in one cycle and $0 \le \alpha < 1$

A function used to switch the frequency between f_1 and f_2 is necessary. Instead of normally used piecewise function which can provide a non-continuous switching, the uni-polar sigmoid function is employed to achieve the smooth switching, which is given by

$$S(t) = \frac{1}{1+e^{-t}} \tag{4.7}$$

During one cycle of the oscillation, two different frequencies are exhibited by the non-harmonic gait. It is necessary to adopt a function to switch the two frequencies. A normal way is to use piecewise function by defining the region where the two frequencies should be applied, as shown in Fig. 4.7a. However, one intension adopting CPG-based control is that we hope that the online modulation of swimming gaits can be controlled by a simple and continuous signal, instead of manual setting of different control values. Comparison of the piecewise function and the sigmoid function is illustrated in Fig. 4.7. This property can be achieved by the sigmoid function. It can switch the two frequencies continuously and smoothly. In the following chapters, this idea will also be used to perform other types control signals in CPG-based control model in order to achieve easy tuning of motions.

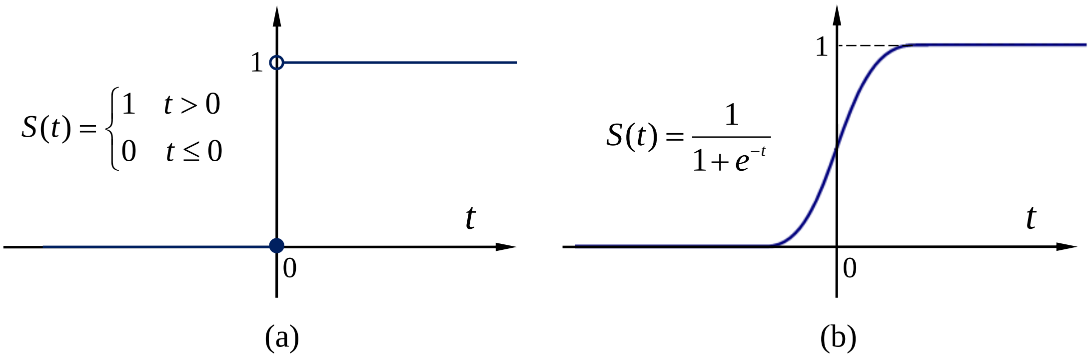


Fig. 4.7 (a) Piecewise function and (b) uni-polar sigmoid function.

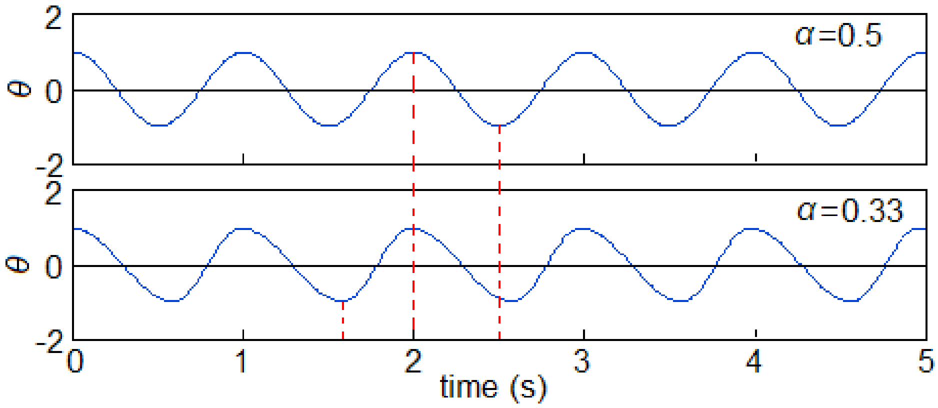


Fig. 4.8 The swimming gait patterns modeled by Hopf oscillator (θ=u). In this example, frequency f=1Hz and τ =40. The amplitude is normalized to 1.

If the state u of the Hopf oscillator (Eqs. (4.1) and (4.2)) is selected as the output to control the swimming gait, state v can be used to determine when the frequency will be switched between f_1 and f_2. The adaptive frequency f can then be given by

$$f(t) = f_1 + (f_2 - f_1)S(\tau v) \tag{4.8}$$

where τ is a positive time constant that tunes the speed of switching and v is a state variable of Hopf oscillator. In this case, the state variable u will be used as the output to control the swimming gait. The switching of frequency occurs when $|u|$ reaches its maximum value and v is zero (see Fig. 4.3,). There is 90^{o} phase difference between u and v, which is satisfy the switching condition. Therefore, the variable v can be used to trigger the switching. Substitute Eqs.(4.5)-(4.7) into (4.8) and obtain

$$f = \frac{1}{2\alpha T} + \frac{(2\alpha - 1)}{2\alpha(1-\alpha)(1+e^{-\tau v})T} \tag{4.9}$$

Fig. 4.8 shows gait patterns generated by the nonlinear oscillator. With Eq. (4.9), both harmonic gait (α=0.5, i.e. the rising phase equals to the decline phase) and non-harmonic gait are obtained. Eq. (4.9) gives a solution to the problem that the shape of swimming gait shape can be adjusted by tuning of only one parameter, α. The sigmoid function ensures that the switching is continuous for α. Therefore, the gait pattern can also be on-line tuned smoothly.

4.2 Closed-loop Swimming Control: A Case Study of Station-holding

Since the controlling parameters of the nonlinear oscillator can be on-line tuned causing no jerks of motion, it is suitable to be used in closed-loop swimming control so as to achieve high performance locomotion. This section discusses an example of its application in station-holding control of fish robots. The aim of is to control swimming of robotic fish system adaptive to the adverse flow with varying flow rate and capable of holding its position.

4.2.1 Station-holding in Adverse Unsteady Flow

Station-holding in fish is a remarkable example of adaptation to unsteady flow, which has been well documented in biology studies [14, 159, 160]. Firstly, observation on the behavior of a real carp fish in such a condition in a water tunnel (see Fig. 4.9) is made. The fish sample belongs to typical BCF type species. It uses the oscillation of its caudal fin as propulsor to generate thrust [8], which is also a major way of propulsion in current fish robots [12]. The motion of fish in still water shows steady patterns with approximately constant tail beat frequency and amplitude in a relatively long time period (here amplitude and frequency are defined as *controlling parameters*). If there is a sudden increase of incoming flow rate, i.e. the fish have to swim against the water current, it can quickly perceive this change and make adjustments on the swimming pattern. Both controlling parameters are augmented with the increase of flow rate. In such a way, fish can generate larger thrust to against the higher drag and hold their positions un-shifted.

Assume that the maximum steady swimming speed of a fish is V_f. The flow rate is said to be low if the adverse flow speed is less than $0.5V_f$ (Reynolds Number Re $\leq$ $2.24\text{x}10^{-4}$) and it is said to be high if the adverse flow speed is higher than $0.5V_f$

($2.24 \times 10^{-4} < Re \leq 4.48 \times 10^{-4}$). At low flow rate, the fish sample can quickly adjust its tail beat frequency as soon as it senses the change of flow. The fish is not flushed away from its original place. The fish sample has to produce larger thrust to swim against the higher adverse flow speed. Once the flow rate is larger than V_f the fish can hardly hold its positions in the flow.

At relatively high flow rate ($2.24 \times 10^{-4} < Re \leq 4.48 \times 10^{-4}$), the adjustment of gaits exhibits three stages. At first, the fish does not respond so quickly and is washed away from its original place along the flow direction. In this stage, the fish tries to swim back to its original position with the two controlling parameters increased. Larger amplitude is observed, which is due to the need of sufficient thrust [99]. Then the fish gets adaptive to the flow environment and swims back. As the fish approaching the original place, it reduces the amplitude slightly and keeps the beat frequency higher than that at low flow rate. The decrease may be explained by the fact the fish reduces the energy cost as it becomes adaptive to the flow. After short period of regulating, the gait is gradually changed to a new steady pattern. The swimming motion goes into the third stage.

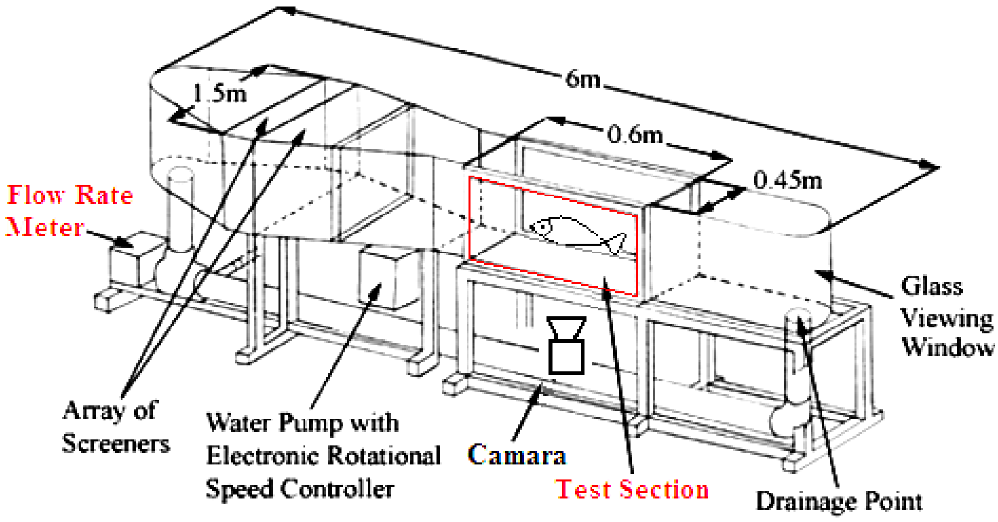
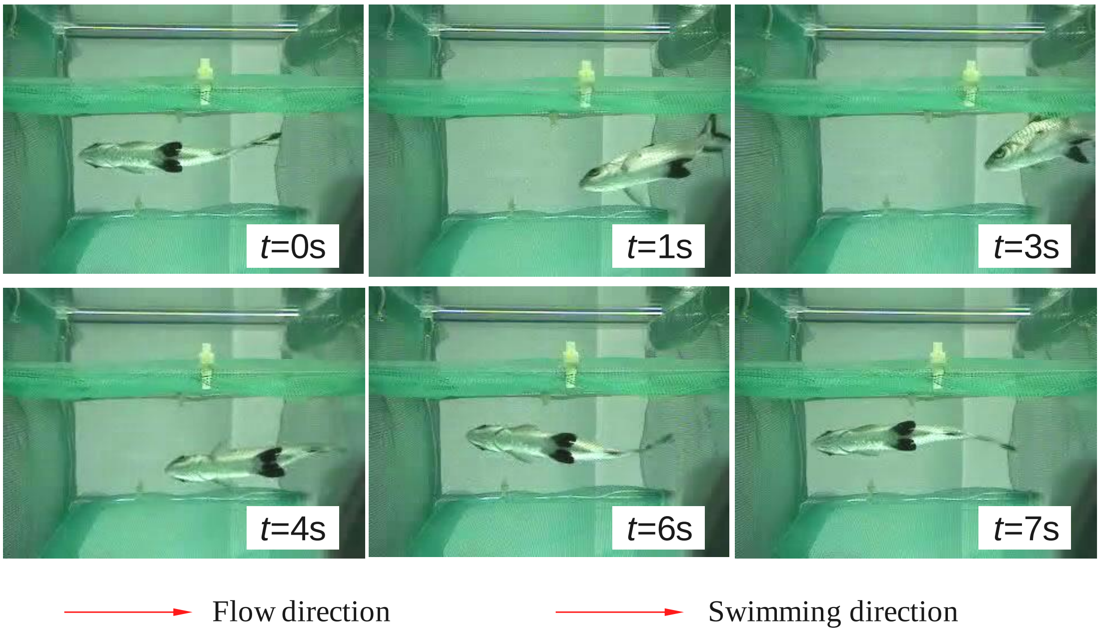
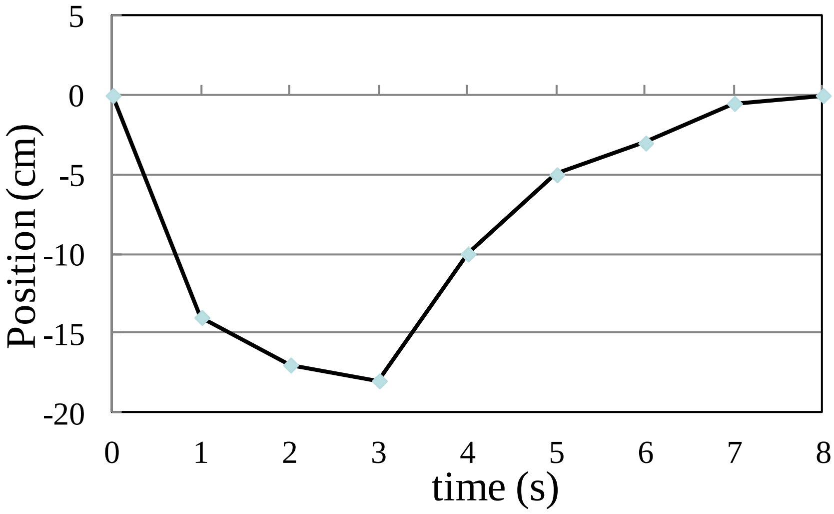


Fig. 4.9 A water tunnel apparatus. The observation of fish sample [81] and the robotic fish experiment are conducted in the Test Section.



(a)



(b)

Fig. 4.10 (a) Snapshot of fish swimming in the water tunnel. It swims against water flow at the second stage with about 4 Hz tail beat frequency. These pictures show the case that fish swims in the second stage where the high frequency tail oscillation is performed. (b) The position variation of the sample fish in the adverse flow. The flow rate is 0.2m/s (approximately 1 BL/s).

Development of a control system that can handle all kinds of flow conditions is not a scope of this chapter. The swimming motion control of robotic fish in unknown unsteady flow is a challenge. One important reason is that the control system of a fish robot lacks of effective sensing system to know the real-time flow information. A suitable control strategy can hardly be decided without this facility. As far as we know, the station-holding control discussed in this thesis is the first implementation of such capability on a fish robot, although it is conducted in a very ideal environment by adopting one dimension unsteady flow. To simplify the problem, the flow situation is restricted by following properties: 1) the flow rate is variable; and 2) the flow is one-way and one-dimensional. Two issues exist in the station-holding control. Firstly, the adaptive swimming control needs feedback of water environment, which is hardly obtained due to the lack of proper sensors [5, 161]. Secondly, the current study on the adaptive swimming control method is not sufficient yet.

Real fish possess sensing organs called lateral lines by which they perceive the pressure, velocity and the orientation [162]. Fish are capable of adapting themselves to the turbulence by changing the swimming gaits relying on the sensory feedback from the lateral line. However, there are currently no sensing devices like the lateral line available for a robotic fish. The idea of the artificial lateral line is still under proof of concept [161], which makes the closed-loop swimming control difficult. In this initial study, developing a sensing system for fish robots is not an issue in the present work. The sensory signals are obtained through normal displacement and velocity sensors in a water tunnel apparatus.

There are some studies on the station control of large autonomous underwater vehicles (AUVs) in complex water environment like rivers and oceans. The actual implementation on AUVs is based on the large-area flow information (obtained by meteorological satellite, for example) [163, 164]. In recent years, some research teams are devoted to develop artificial lateral lines to detect the micro change of flow situation [161]. The more complicated station-holding control may benefit from these new sensors in future works.

4.2.2 Control Laws for Station-holding Control

(a) Cases with Single DOF

The fish robot (NAF-II) shown in Fig. 2.27 will be used to illustrate the station-holding control. As implied by the real fish, a BCF-type robotic fish can achieve the similar adaption to unsteady flow by real-time adjusting two controlling parameters. Feedback control laws that regulate the controlling parameters can be employed to guide the tuning. It is naturally assumed that these two parameters all rise if there is an abrupt speed increase of incoming flow. In the beginning, the fish augments both parameters for larger thrust to overcome the drag. Then as the swimming becomes adaptive to the flow, the frequency or amplitude may be slightly decreased to save the energy. Additionally, the evolving of both parameters is coupled.

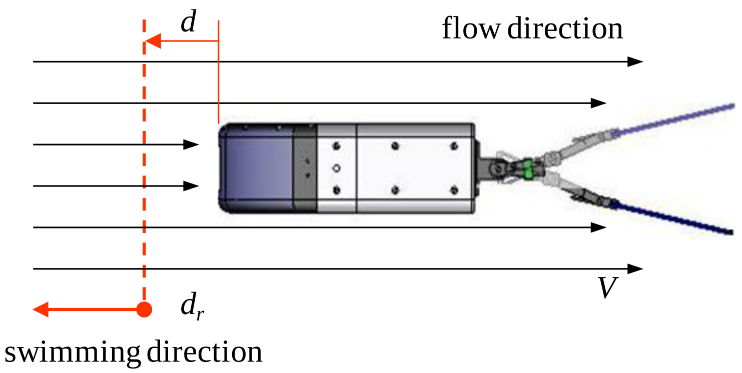


Fig. 4.11 Illustration of a robotic fish swimming against flow. *d* is the actual displacement. d_r denotes the reference position of the fish. *V* is the flow velocity.

Several assumptions are made. Firstly, it is assumed that once the fish senses the increase of incoming flow, its tail beat frequency jumps to a certain value proportional to the flow rate. Secondly, as the fish is washed away by water flow, the frequency gets larger with the distance increment with respect to its original position. Thirdly, the frequency becomes larger as the relative speed between fish locomotion and the water flow rises. This is apparent because the drag increases proportionally to square of relative speed [8]. Finally as the fish is flushed away from the origin, the controller should 'memorize' this position error and try to reduce the distance. Therefore, the controller for the frequency should also contain the information of the integration of position errors. Given that the reference input position is zero (d_r=0) in station-holding problem (in this case the position error is $e=d_r-d$), the control law for frequency can be interpreted as:

$$f = -k_1 d - k_2(V - \dot{d}) - k_3 \int_0^t d\,\delta t \tag{4.10}$$

where k_1, k_2 and k_3 are positive constants. Definitions of other symbols can be found in Fig. 4.11. Rewrite Eq. (4.10) and obtain

$$f = -(k_1 d - k_2 \dot{d} + k_3 \int_0^t d\,\delta t) - k_2 V \tag{4.11}$$

The last term of Eq. (4.11) can be seen as a feed forward of the flow information. The control law for the amplitude can also be structured in a similar manner. The value of amplitude is proportional to the adverse flow rate and the distance *d*. With the coupling of frequency, the control law can be given by the following equation.

$$A = -k_4 d - k_5 V - k_6 f \tag{4.12}$$

where k_4, and k_5 are positive coefficients; k_6 represents the coupling strength with the frequency. Note that the coupling coefficient, k_6 is necessary because it ensures that the actuator will not reach its power limitation when frequency increases. For a practical physical system, the value of *A* and *f* are bounded by the maximum allowable power. Given the steady state output of the Hopf oscillator can be given by Eq. (4.3), its first order and second order derivatives are limited by the maximum speed and acceleration of an actual actuator respectively. Therefore, following conditions for *A* and *f* (by differentiating *u*) exist:

$$Af \leq \Omega_m \tag{4.13}$$

$$Af^2 \leq \alpha_m \tag{4.14}$$

where Ω_m and α_m are angular and acceleration limits determined by the actuator.

(b) Cases with multi-DOF

The station-holding control for multi-DOF fish robots consists of two aspects: coordination of swimming gaits [125, 165] and the adaptation process where the gait patterns are modulated by feedback control laws. The former (see Fig. 4.12) can be achieved by multiple CNLOs, which will be detailed in the following chapters and the later is discussed here. The control law for one-DOF fish robot in previous sections can be extended to the multi-DOF case. According to Eqs. (4.11)-(4.12), six parameters are needed to decide one pair of controlling parameters for each oscillator, and there are totally eighty-four parameters need to be found for the prototype in Fig. 2.19. Actually, it is not necessary to develop such a complex mathematical method to determine all these parameters. There are ways to reduce the complexity of the computation by looking into the swimming patterns of real fish.

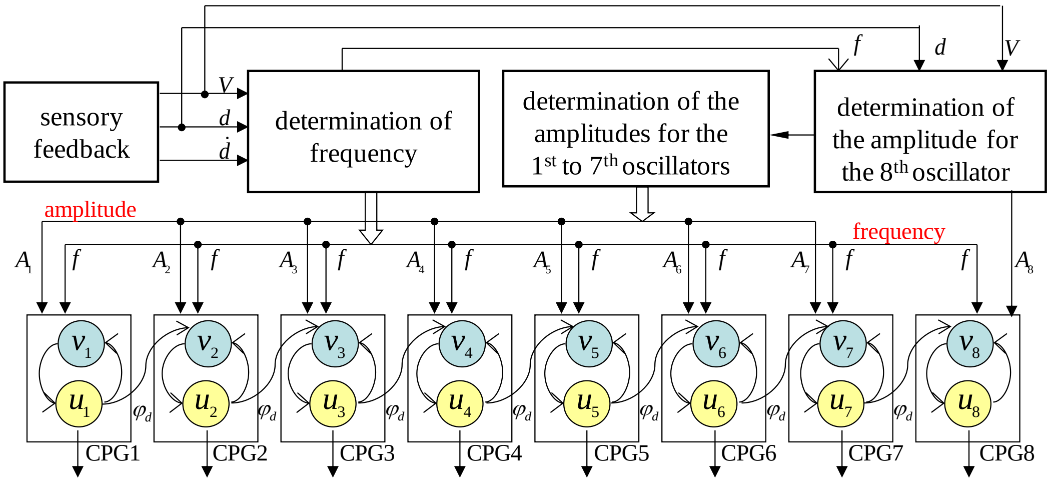


Fig. 4.12 Control structure of station-holding control on multi-DOF fish robots.

For the prototype in , this means the amplitude for the last actuator is derived from (4.12) and the rest are derived from the envelop function. The whole control structure can be described by Fig. 4.12. The determination of amplitudes and frequency can be made by the following control laws:

$$f_i = f = -(k_1 d - k_2 \dot{d} + k_3 \int_0^t d\,\delta t) - k_2 V \tag{4.15}$$

$$A_n = -k_4 d - k_5 V - k_6 f \tag{4.16}$$

$$A_i = \frac{A_n}{n} \cdot i \tag{4.17}$$

$$A_i f \le \Omega_m \tag{4.18}$$

$$A_i f^2 \le \alpha_{i,m} \tag{4.19}$$

The gait signals for the propulsor prototype can be obtained by solving equations (4.15)-(4.19). See the illustration of the fish body curve in Fig. 2.30. The amplitudes of oscillations become gradually larger from head to tail and they obey satisfy the envelop function, which is thought to be an efficient way of locomotion [16]. Here, the linear envelop is adopted to simplify the problem [16, 71]. Given that higher amplitude consumes more energy and there is power limit for an actual actuator, it is necessary to determine the maximum amplitude with the feedback control law.

4.3 Simulation of Station-holding Control

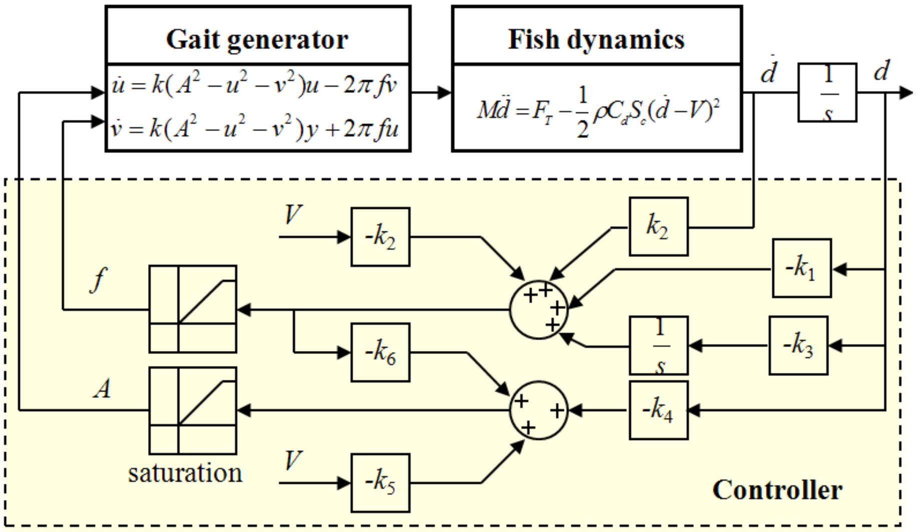


Fig. 4.13 Overview of the control scheme. The Hopf oscillator generates the periodic tail motion pattern. A feedback control scheme is applied to regulate the frequency and amplitude. The amplitude is also coupled with frequency by a proportional coefficient k_6. Flow rate is a feedforward quantity to the controller.

A simulation is performed to test station-holding control laws. Swimming gait of the fish robot is produced by the Hopf oscillator. Feedback control laws are applied to modulate the output of the oscillator. The closed-loop control scheme is illustrated in Fig. 4.13. The dynamics modeling of the robot locomotion is necessary for the simulation. Let M denote the mass property of the robotic fish, F_D be the body drag and F_T be the thrust generated by tail. The dynamics of the fish can be simply interpreted by:

$$M\ddot{d} = F_T - F_D \tag{4.20}$$

It can be assumed that the fish tail is a pure propulsor and the drag completely comes from the fish body. Then F_D is given by:

$$F_D = \frac{1}{2}\rho C_d S_c (\dot{d} - V)^2 \tag{4.21}$$

where ρ is the water density, C_d is the drag coefficient and S is the cross-sectional area of fish body.

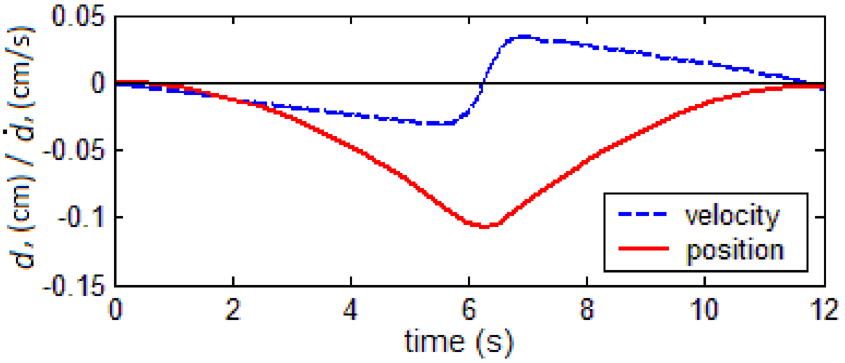


Fig. 4.14 Simulated results of station-holding control. The displacement and the absolute velocity are shown. The parameters in (9) are M=6.8kg, ρ=1000kg/m^3, C_d=0.63 and S_c=0.02 m^2.

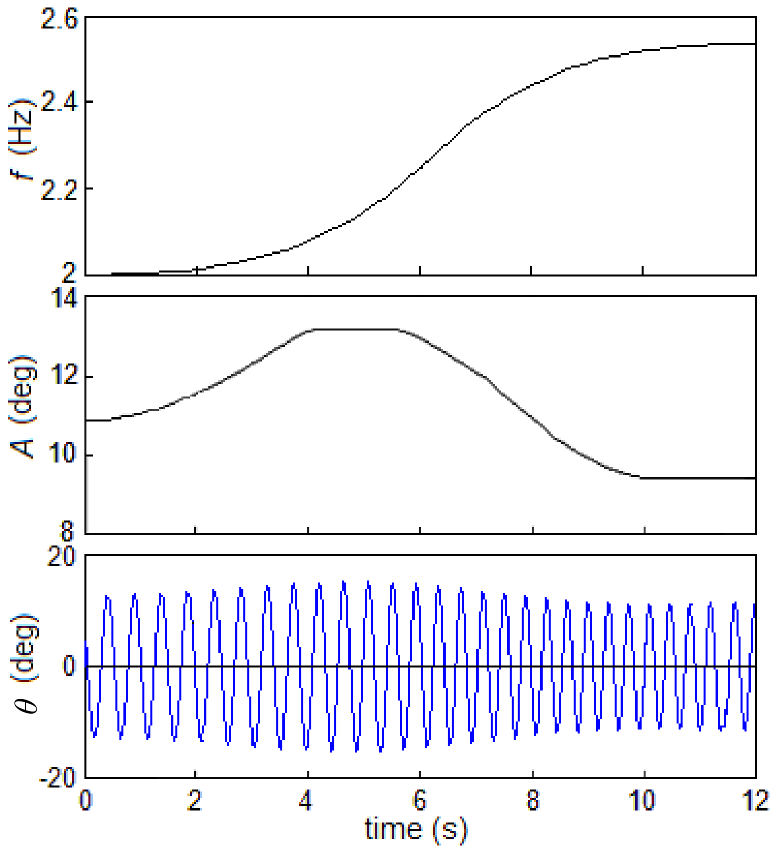


Fig. 4.15 Evolving of the tail beat frequency, amplitude and the swimming gait control signal. The paramters in (3)-(5) are chosen as k=1, k_1=2.0, k_2=0.8, k_3=25.0, k_4=0.5, k_5=28.0 and k_6=0.02.

To evaluate the thrust force, experiments were conducted to find the relation between the thrust and the controlling parameters by using statistical methods based on

Buckingham's theorem [166]. A semi-empirical model, $F_T = F_T(A, f, V)$, is developed in to model the thrust. According to the derivation detailed in [166]. The dynamics of the swimming robots can then be given by:

$$\ddot{d} = \frac{1}{M}\left[F_T(A,\ f, V) - \frac{1}{2}\rho C_d S_c (\dot{d} - V)^2 \right] \tag{4.22}$$

$$F_T = \frac{0.4032A^{0.8} f^2}{V + 0.28Af)^{0.3}}\left[\frac{3.9A^{6.5} f^{0.9}}{(V + 0.28Af)^4} + 0.65\right](2\sin\frac{A}{2} + \sin\frac{\sqrt{3}A}{2} + \sin A) \tag{4.23}$$

With Eq. (4.23), the thrust force can be predicted by knowing controlling parameters and the flow speed. It is only used in the simulation to find the some general trend of coefficients for Eqs. (4.11) and (4.12). Better coefficients will be determined in actual experiments. The problem of station-holding in the adverse water flow is actually a position control with zero input (d_r=0). Fig. 4.14 shows a result of the simulation. When the flow rate steps to a certain value and the robot swims in the adverse current, the robot begins to regulate the tail oscillation patterns. In the first stage, the robot is washed away from zero position. The absolute swimming is toward the negative direction. After several seconds setting, the robot adapts itself to the adverse current and approaches to its origin gradually. The evolving of the controlling parameters is illustrated in Fig. 4.15. The robot keeps tuning the motion of its tail continuously and smoothly until the oscillation evolves to a new stable pattern.

Due to the size limitation of our lab water tunnel, actual experiment was not conducted. Instead, a simulation is performed by the help of the thrust predictive model of this prototype obtained in experiments, as that for the one-DOF fish. In the simulation, the flow rate V jumps from 0.03m/s to 0.1m/s. Linear body curve envelop is adopted.The maximum amplitude θ_8 is derived from Eq. (4.16) and all other amplitudes are obtain from Eq. (4.17). The gaits keep varying depending on the feedback from environment and holding a stable phase difference. Those gaits generate a full cycle traveling waveform along the fish body (n_λ=1). The evolving of the eight swimming gaits is illustrated in Fig. 4.16. The station-holding performance is as similar as that of the one-DOF fish (in Fig. 4.14).

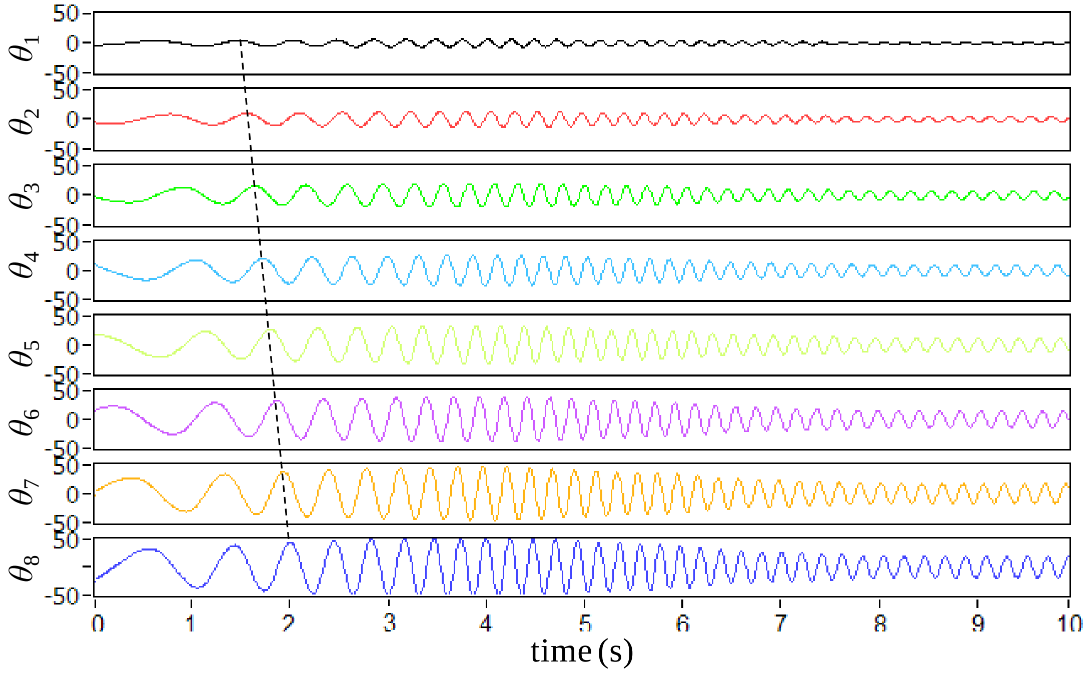


Fig. 4.16 Simulated swimming gaits for a multi-DOF fish robot. Control parameters are set as: n_λ=1, k=8.0, ε=1.0, k_1=8.0, k_2=4.0, k_3=6.0, k_4=200, k_5=240, and k_6=1.2.

4.4 Experiment of Station-holding Control

4.4.1 Experiment Setup

The experiment is conducted in the water tunnel apparatus shown in Fig. 4.9. The configuration of the experiment is shown in Fig. 4.17. The fish swims against one-dimensional adverse flow in the water tunnel. A linear conductive plastic resistance sensor is mounted on the fish robot through a rigid bar connection to measure its actual position. The absolute swimming velocity is obtained from the derivative of the position. The forth-order backward derivative algorithm is applied to achieve high precision differentiation. A pendulum pulled by a soft string is submerged in water to visually indicate the strength of flow speed by its deflection angle. The accurate information of flow is provided by a volumetric flow rate meter to the closed-loop control system.

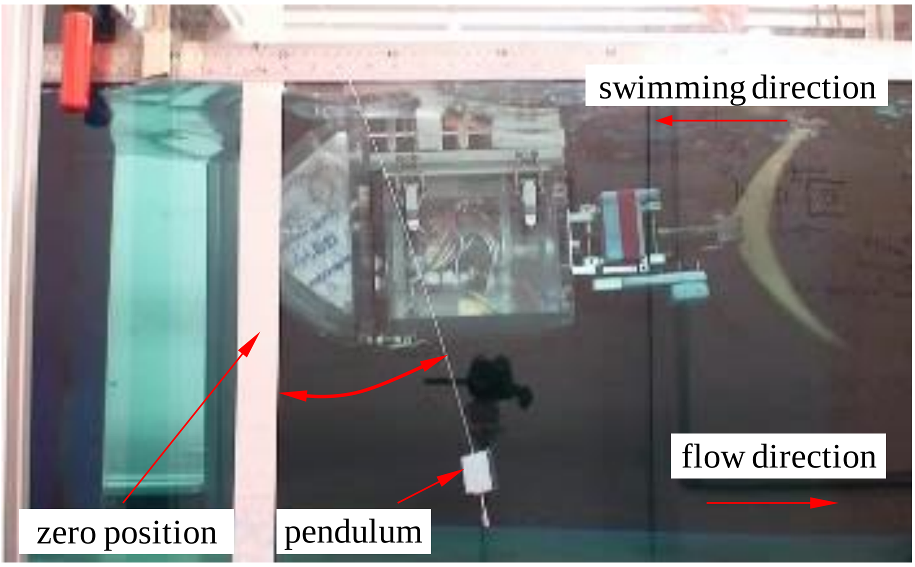


Fig. 4.17 Configuration of experiment in water tunnel.

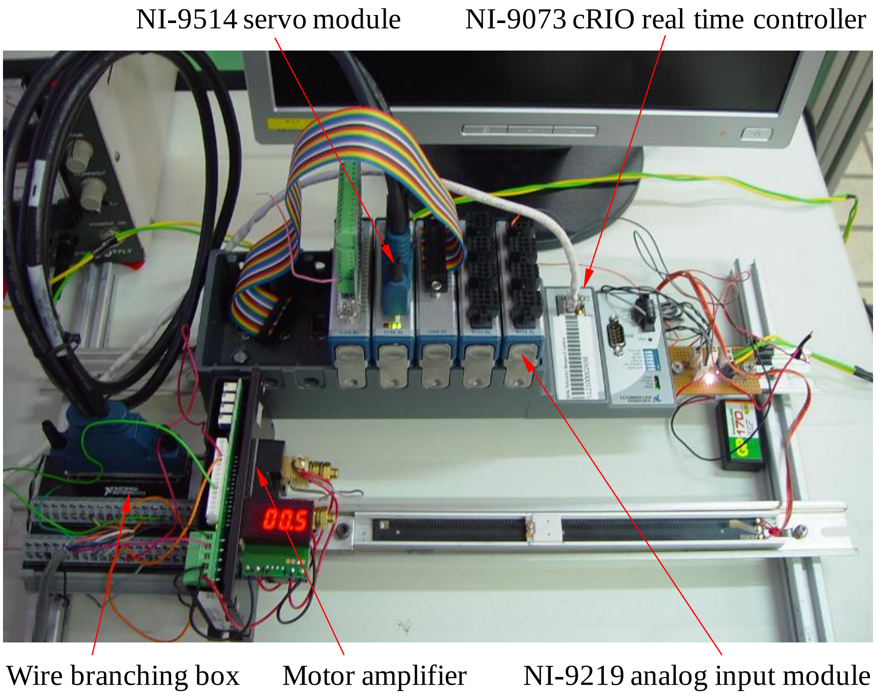


Fig. 4.18 Control hardware based on NI-9073 CompactRIO™ real-time controller.

Equations (4.1)-(4.14) are solved in Labview™ development environment and implemented with the CompactRIO™ embedded controller. The sampling period of control loop is chosen as 20ms, which is proved sufficient to maintain the real-time property in the experiment. The whole control philosophy is that the nonlinear oscillators work as an oscillatory swimming pattern generator with constant steady amplitude and variable frequency. The feedback controller tunes this frequency and modulates the output of the oscillator. Upon implementation, a robust tracking controller (will be discussed in Chapter 6) is applied as a low level controller in order that the tail DC motor can track the set points obtained from the oscillators. This is necessary because it makes sure that the tail motion exactly follows the commands derived from the oscillator [30].

4.4.2 Experimental Results

Firstly, parameters k_3 and k_5 are evaluated for feed forward terms in Eqs. (4.11)-(4.12). It is assumed that the fish robot in the following two scenarios swims with the same controlling parameters: 1) water is still and robot moves forward at constant absolute speed U with respect to the ground; and 2) water moves at constant speed U and the robot swims against flow while keeping stationary with respect to the ground. The fish robot can be tested in free run mode in still water. The relation between the swimming speed and controlling parameters can be obtained. Then values of k_3 and k_5 are found through dividing frequency/amplitude by flow velocity respectively. For our fish robot, the predictive model in [166] is applicable for finding the values of these parameters. One way to evaluate k_1-k_6 is to use the Bode diagram, as suggested in [167]. But an accurate dynamic model of the fish robot is needed to plot the Bode diagram. Although a predictive model in the simulation is available, it is not so accurate for finding the proper parameters. In our experiment, these parameters are tuned manually based on actual response of station-holding control [168].

t=25s, f=1.7Hz, A=7.9º　　　　t=30s, f=3Hz, A=11.2º

t=34s, f=2.1Hz, A=10.5º　　　　t=32s, f=2.6Hz, A=10.5º

t=36s, f=2.1Hz, A=10º　　　　t=42s, f=2.2Hz, A=10º

Fig. 4.19 Snapshots of the adaptive swimming from t_2 to t_3. The locomotion experienced three stages: I) both controlling parameters increase as the robot is flushed away in t=25-30s; II) swims back in t=30-42s and III) get stable when t>42s.

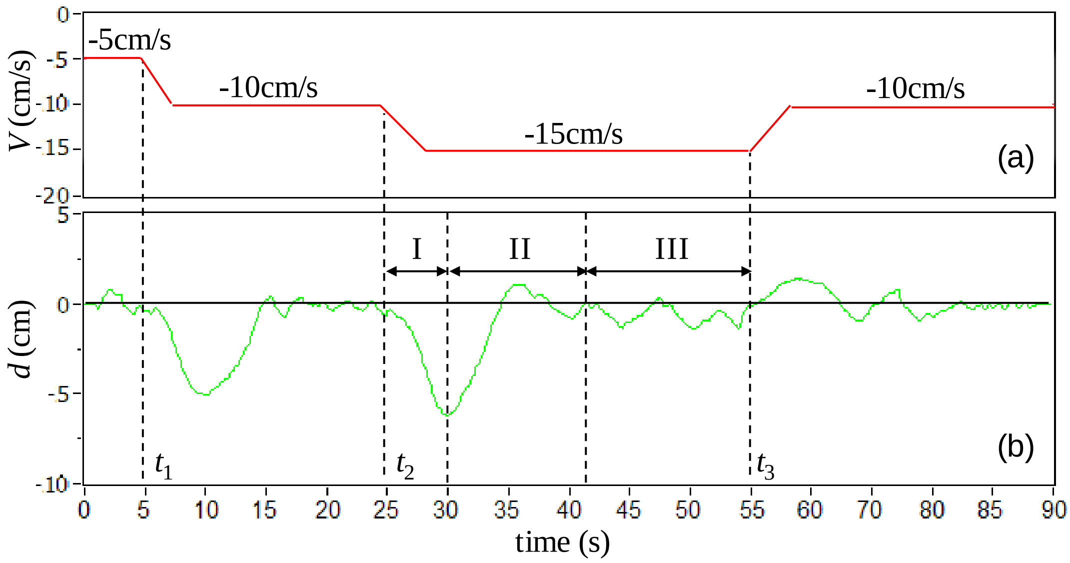


Fig. 4.20 (a) Variation of flow speed in water tunnel; and (b) the response of the station-holding control. The coefficients in (1)-(6) are k=8.0, λ=1.0, k_1=16, k_2=1.5, k_3=12, k_4=28, k_5=105, k_6=1.6, θ_m=0.785rad/s and α_m=2.36 rad/s^2. Note that the flow rate may not be changed sharply as described in the figure. The figure only shows the flow control command.

The step responses of the control system are tested, which is depicted in Fig. 4.20. The example of swimming between t_2 and t_3 is illustrated in Fig. 4.19. Firstly, a low adverse flow rate is set and the robot easily gets adaptive to the flow (the fish robot quickly adjust its swimming gaits to hold its original position). Then the flow speed is increased to 10cm/s at time t_1. At first the robot is flushed away from the original position. The controlling parameters increase correspondingly to generate higher thrust against the flow. After about ten-second setting process, the robot swims back to zero position and maintains stable. At t_2 the flow rate is increased to 15cm/s (this flow rate is equivalent to the 0.5BL/s, which is a relatively large value for the present fish robot). The setting process is similar except that the there is an obvious overshoot (t=35-38s) and the fluctuation near the zero position is relatively large (t=40-55s). At t_3 a decreasing step response is tested by reducing the flow rate. As shown from t=55s to 70s, after short period of overshooting, the robot gets stable again. The evolving of the controlling parameters is shown in Fig. 4.21.

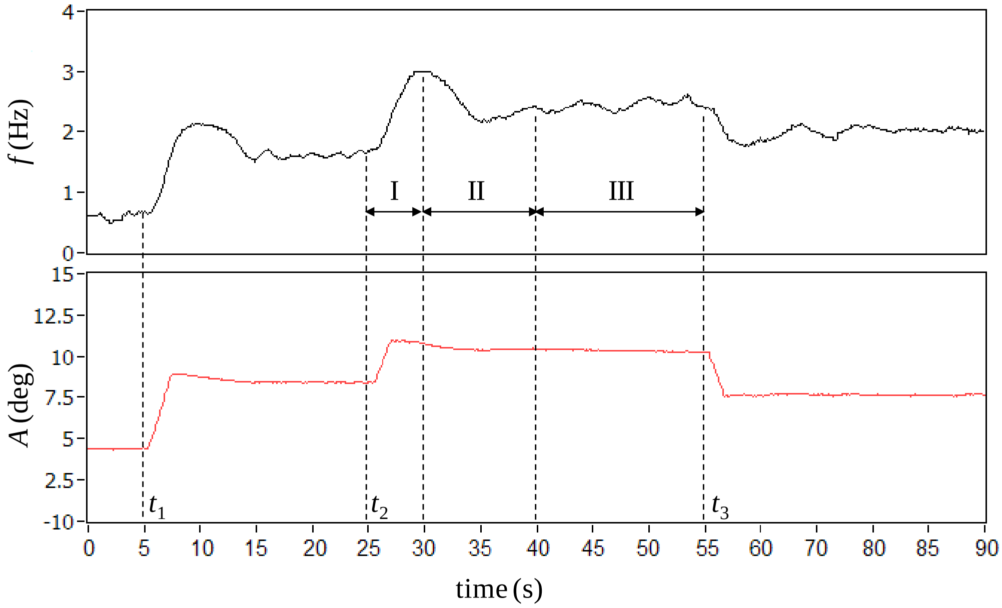


Fig. 4.21 The evolving of the tail beat frequency and amplitude. The time line definition follows Fig. 4.20.

Experimental results show that the station-holding control system is stable and the controlling parameters are convergent as time gets large. There is an obvious gap between the actual coefficients for control laws obtained from experiments and those from the simulation. This suggests that the model in (4.22) is not sufficient to describe the dynamics of the robot-water interaction. This equation provides only a static description of thrust and controlling parameters. In the transient swimming condition, the accuracy of this model may not be sufficient. However, the simulation is still useful and it can provide a qualitative understanding of the controller design.

In the simulation, the coefficients in the control laws are k=1, k_1=2.0, k_2=0.8, k_3=25.0, k_4=0.5, k_5=28.0 and k_6=0.02, which will achieve station holding performance on NAF-II as shown in Fig. 4.14. In experiments, the coefficients are initially set as same as those in the simulation. But due to the discrepancies between the NAF-II and its mathematical model in the simulation, these coefficients do not work well for the actual robot. The coefficients are tuned manually according to the experiment results. Fig. 4.19 - Fig. 4.20 show the improve results with coefficients of k_1=16, k_2=1.5, k_3=12, k_4=28, k_5=105, and k_6=1.6. These values are adopted in the experiment on NAF-II.

4.5 Summary

The modulation of parameters of Hopf oscillator will not cause the unsmooth motions, even if the system is under evolving. Therefore, it is suitable to be a robust swimming gait generator. As an application of the nonlinear oscillator, swimming adaption to one-dimensional adverse unsteady flow is studied and implemented on a fish robot. Feedback control laws that enable a fish robot to keep its station are proposed. By dynamically tuning the controlling parameters, the fish robot can adjust its swimming gaits gradually adaptive to the variation of flow. The control laws were tested on a BCF type fish prototype in water tunnel.

The station-holding control discussed in this chapter is a relatively simple case. As far as we know, this is the first implementation of station-holding on fish robots. The inspirations from this simple illustration are expected to inspire more complex adaptive swimming control. The actual situation of water environment (river, sea, for instance) may be much more complex where the flow is multi-dimensional and turbulent. The manipulation of fin/body motion is therefore a more complicated exercise, which can hardly be achieved on a fish robot. Lack of effective flow sensor is also a big obstacle to achieve the station-holding performance in such flow environment. These problems are expected to be solved in future studies.

Chapter 5 Coordination of Swimming Gaits Using Coupled Nonlinear Oscillators

Pure oscillation is seldom adopted as a major way to obtain thrust. Usually, multi-DOF undulatory motions are more or less involved in thrust generation for most fishes. Bionic propulsor in fish robots simulates natural and flexible undulatory motions by incorporating multiple oscillatory joints in mechanisms. In this chapter, the application of nonlinear oscillators in modeling of swimming gaits for bionic propulsors is explored. A single nonlinear oscillator can be used to generate the fundamental rhythmic pattern for an oscillatory joint and multiple such oscillators can be interlinked to coordinate motions of a multi-DOF propulsor. The analytical derivation of coupling scheme among oscillators will be discussed.

CNLOs are more and more often employed to model the rhythmic movements in robotics area such as legged walking, flying and swimming [22, 28, 144, 148]. They can also be regarded as models of artificial Central Pattern Generators (CPGs) used in the inter-limb coordination for multi-DOF legged robots and more recently in swimming gait generation for fish robots [26, 28, 129, 130]. Compared with the sine function based gait generator, this modeling method is expected to have the following advantages: i) the swimming gaits for multiple joints are not pre-defined but on-line generated, which makes the adaptive swimming to the environment feasible; ii) different types of gait patterns can be obtained to achieve versatile aquatic locomotion; iii) an analytical and generic solution to the problem of modeling the biomimetic undulatory fin motion can be available; iv) the swimming gait can be on-line tuned with simple control signals.

These features will be illustrated through theoretical analysis and experiments on fish robots. Firstly, the connections for multiple oscillators are established in Section 5.1, i.e. synchronization of multiple nonlinear oscillators. An artificial CPG model used for coordinating swimming motions will be formulated. The application of the CPG model is then illustrated by experiments on an eight-DOF undulatory long fin prototype, which is described in Section 5.2. In Section 5.3, the model is applied to the locomotion control of another fish robot, a six-DOF robotic manta ray, which illustrates generation of versatile locomotion by different configurations of coupling schemes.

5.1 Coupling of Multiple Oscillators: A Model of Central Pattern Generators (CPGs)

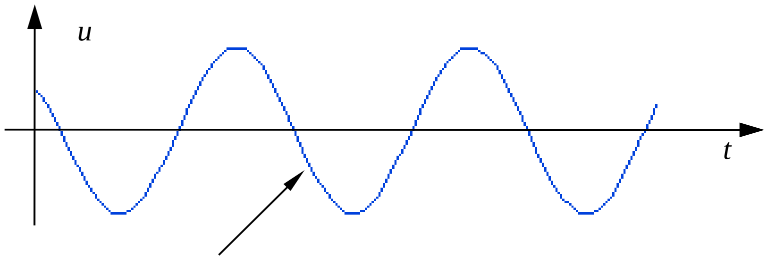
Here a method used to model the CPGs for quadruped locomotion in [169] is adopted and modified to derive the swimming gaits for bionic propulsors. It can be assumed that movements of all oscillating joints are driven by the identical nonlinear oscillators. Multiple Hopf oscillators are chosen to perform the gait pattern generation for all joints and coupling scheme will be developed to make those oscillation synchronized, i.e. achieve phase locking among oscillators.

5.1.1 Coupled Oscillators with Stable Phase Differences

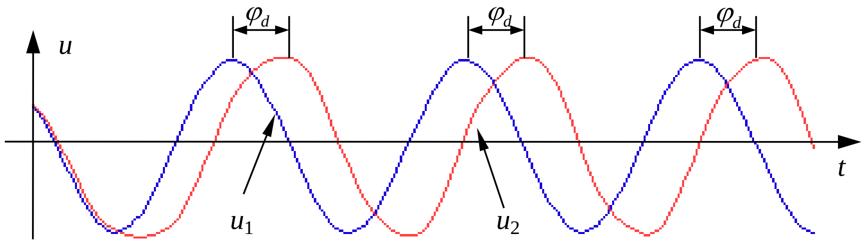
Without lose of generality, consider the canonical form of identical nonlinear dynamical systems defined in Eqs. (4.1) and (4.2) [170]:

$$\dot{\mathbf{X}}_i = F(\mathbf{X}_i) + \mathbf{P}_i = \begin{bmatrix} k(1-u_i^2 - v_i^2)u_i - 2\pi f v_i \\ k(1-u_i^2 - v_i^2)v_i + 2\pi f u_i \end{bmatrix} + \begin{bmatrix} p_{u,i} \\ p_{v,i} \end{bmatrix} \tag{5.1}$$

where $F(\mathbf{X}_\mathrm{i})$ represents a nonlinear system defined by the Hopf oscillator, $\mathbf{X}_i=(u_i, v_i)^T$ is the state vector of the i^{th} system and $\mathbf{P}_i=(p_{u,i}, p_{v,i})^T$ is a perturbation vector. The coupling of oscillators can be achieved in the way that one oscillator perturbs another and in order that a stable phase difference between the two oscillators are maintained. This phenomenon is a case of phase locking [127]. The difference between uncoupled and coupled oscillators is explained in Fig. 5.1.



(a)



(b)

Fig. 5.1 Output of two uncoupled (a) and coupled oscillators (b).

At first, the coupling term for two Hopf oscillators is derived and extended to multiple oscillators afterwards. For the ease of deriving, Eq. (5.1) can be transformed into polar coordinates interpretation by using relations $u_i=r_i\cos\varphi_i$ and $v_i=r_i\sin\varphi_i$:

$$\dot{u}_i = \dot{r}_i \cos\varphi_i - r_i\dot{\varphi}_i \sin\varphi_i + p_{u,i} = k(1-r_i^2)r_i \cos\varphi_i - \omega r_i \sin\varphi_i + p_{u,i} \tag{5.2}$$

$$\dot{v}_i = \dot{r}_i \sin\varphi_i + r_i\dot{\varphi}_i \cos\varphi_i + p_{v,i} = k(1-r_i^2)r_i \sin\varphi_i + \omega r_i \cos\varphi_i + p_{v,i} \tag{5.3}$$

where r and θ are the length and angle of the vector starting for origin to the point (x, y) in the phase plot. After equating the coefficients of $\cos\varphi_i$ and $\sin\varphi_i$, we obtain:

$$\begin{bmatrix} \dot{\varphi}_i \\ \dot{r}_i \end{bmatrix} = \begin{bmatrix} \omega \\ k(1-r_i^2)r_i \end{bmatrix} + \begin{bmatrix} p_{\varphi,i} \\ p_{r,i} \end{bmatrix} \tag{5.4}$$

where $(r_i, \varphi_i)^T$ is the system state vector in polar coordinates, ω ($\omega=2\pi f$) is the natural frequency of the unperturbed system, F_φ and F_r are dynamical systems describing the evolution of φ_i and r_i, vector $(p_{r,i}, p_{\varphi,i})^T$ corresponds to $\mathbf{P}_i$. Its two elements are components of the perturbation acting in direction of the radius and phase respectively.

5.1.2 Formulations of Coupling Schemes

(a) Two Oscillators with One-directional Coupling

To simplify the problem, consider holding of an arbitrary phase difference between two oscillators by introducing a one-directional coupling: oscillator one perturbs oscillator two and there is no reverse perturbation. In this case, $\mathbf{P}_1$ equals to zero and $\mathbf{P}_2$ is an unknown vector which needs to be derived. The coupling scheme is shown in Fig. 5.2.

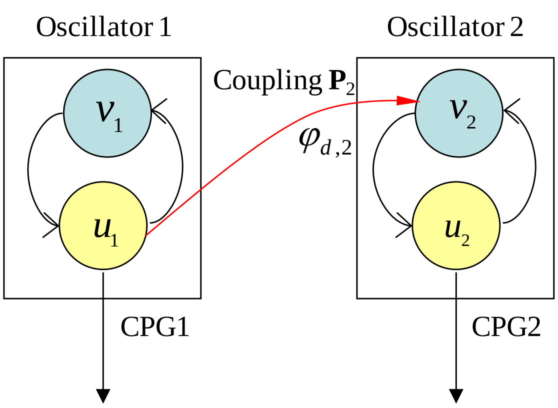


Fig. 5.2 Single –directional coupling between two oscillators.

As u_2 is taken as the output of oscillator two, it is not perturbed directly by u_1. An indirect manner is applied by using the output of the oscillator one to influence the state v_2 of the oscillators 2, as shown in Fig. 5.2. This can be done because u_2 is internally coupled with v_1. Therefore,

$$p_{u,2} = 0 \tag{5.5}$$

If there is no coupling between two oscillators and they start to evolve from the same initial state, u_1 always equals to u_2. With coupling, it can be assumed that oscillator two gets synchronized to oscillator one when the phase of oscillator one reaches $\varphi_1+\delta$. Here an intermediate angle δ is introduced to control the phase difference φ_d. The coupling term on oscillator two can then be defined by

$$\mathbf{P}_2 = \begin{bmatrix} p_{u,2} \\ p_{v,2} \end{bmatrix} = \begin{bmatrix} 0 \\ \varepsilon u_1 \end{bmatrix} = \begin{bmatrix} 0 \\ \varepsilon r_1 \cos(\varphi_1 + \delta) \end{bmatrix} \tag{5.6}$$

where ε is a positive constant that determines the coupling strength.

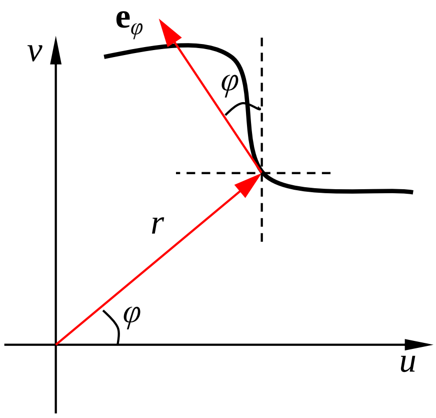


Fig. 5.3 Illustration of perturbation in direction of phase angle φ.

In a polar coordinate system, the phase difference is caused by perturbation acting in direction of phase (see Fig. 5.3), i.e. tangential to the limit cycle. This direction for oscillator two (unperturbed) is given by:

$$\mathbf{e}_{\varphi,2} = \frac{\dot{\mathbf{X}}_2}{|\dot{\mathbf{X}}_2|} = (-\sin\varphi_2,\ \cos\varphi_2)^T \tag{5.7}$$

The perturbation on the phase is obtained by

$$p_{\varphi,2} = \mathbf{P}_2^T \cdot \mathbf{e}_{\varphi,2} \tag{5.8}$$

By substituting Eqs. (5.6) and (5.7) into (5.8), $p_{\varphi,2}$ can be given in terms δ and φ_2:

$$p_{\varphi,2} = \mathbf{P}_2^T \cdot \mathbf{e}_{\varphi,2} = \varepsilon r_1 \left[\cos(\delta - \varphi_{d,2}) \cos^2 \varphi_2 - \frac{1}{2} \sin(\delta - \varphi_{d,2}) \sin(2\varphi_2) \right] \tag{5.9}$$

In a phase locking regime, two oscillators evolve with stable phase difference $\varphi_{d,2}$:

$$\varphi_{d,2} = \varphi_2 - \varphi_1 \approx \text{const.}\big|_{t\to+\infty} \tag{5.10}$$

From Eq. (5.10), we have

$$\dot{\varphi}_{d,2} = \dot{\varphi}_2 - \dot{\varphi}_1 = \omega_{d,2} + p_{\varphi,2} \tag{5.11}$$

where $\omega_{d,2}=\omega_2-\omega_1$. The two oscillators are synchronized with a constant phase difference $\varphi_{d,2}$ after a short transient phase evolving (from 0 to t_0). Therefore, in steady state we have:

$$\lim_{t\to\infty} \int_{t_0}^{t} \dot{\varphi}_{d,2} dt = 0 \tag{5.12}$$

The integration of Eq. (5.12) over time t can be done implicitly by integration over φ_2 in the steady state of the system [126]. If the system evolves into the steady state since $\varphi_2=\Phi_0$ (at $t=t_0$), then phase locking is maintained after $\varphi_2=\Phi>\Phi_0$ ($t>t_0$). Eq. (5.12) can be rewritten as:

$$\dot{\varphi}_{d,res} = \lim_{\Phi\to\infty}\int_{\Phi_0}^{\Phi} \dot{\varphi}_{d,2} d\varphi_2 = \sum_{j=0}^{\infty}\int_{\Phi_0+2j\pi}^{\Phi_0+2(j+1)\pi} \dot{\varphi}_{d,2} d\varphi_2 = 0 \tag{5.13}$$

Then substitute the (5.11) into (5.13), the following result can be obtained:

$$\dot{\varphi}_{d,res} = 2\pi\omega_{d,2} + \varepsilon r_1 \pi(\cos\varphi_{d,2}\cos\delta + \sin\varphi_{d,2}\sin\delta) = 0 \tag{5.14}$$

By solving Eq. (5.14), the intermediate angle δ can be obtained:

$$\delta \approx \varphi_{d,2} - \frac{\pi}{2} \tag{5.15}$$

Substitute (5.15) into (5.9), we obtain:

$$\begin{aligned} p_{v,2} &= \varepsilon r_1 \cos(\varphi_1 + \varphi_{d,2} - \frac{\pi}{2}) \\ &= \varepsilon(u_1\cos\varphi_{d,2} + v_1\sin\varphi_{d,2}) \end{aligned} \tag{5.16}$$

The coupling terms in condition of one-directional coupling can be given by

$$\mathbf{P}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5.17}$$

$$\mathbf{P}_2 = \begin{bmatrix} 0 \\ \varepsilon(u_1\sin\varphi_{d,2} + v_1\cos\varphi_{d,2}) \end{bmatrix} \tag{5.18}$$

(b) Two Oscillators with Mutual Coupling

The coupling between two oscillators is not necessarily one-directional [169]. The output of oscillator *i* can also be used to influence the evolving of oscillator *i*-1. In this case, oscillator *i*-1 is coupled with oscillator *i* with a contrary phase difference, as shown in Fig. 5.4. The coupling terms $\mathbf{P}_{i-1}$ and $\mathbf{P}_i$ can be obtained in a similar manner as that in the derivation for Eq. (5.6), which means $\mathbf{P}_{i-1}=[0, \varepsilon u_i]^T$ and $\mathbf{P}_i=[0, \varepsilon u_{i-1}]^T$. The coupling terms for two mutually coupled oscillators can then be given by

$$\mathbf{P}_{i-1} = \begin{bmatrix} 0 \\ \varepsilon(v_i\cos\varphi_{d,i} - u_i\sin\varphi_{d,i}) \end{bmatrix} \tag{5.19}$$

$$\mathbf{P}_i = \begin{bmatrix} 0 \\ \varepsilon(u_{i-1}\sin\varphi_{d,i} + v_{i-1}\cos\varphi_{d,i}) \end{bmatrix} \tag{5.20}$$

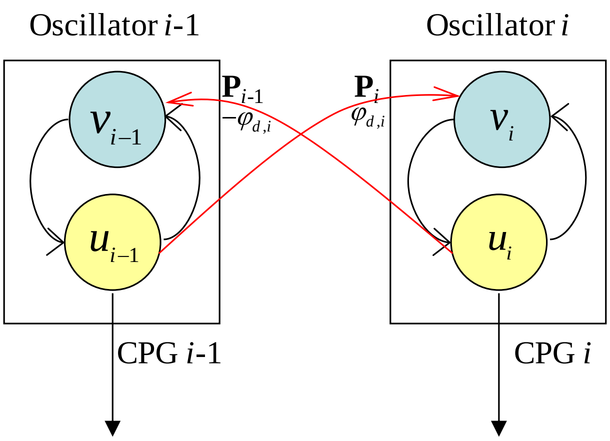


Fig. 5.4 Mutual coupling between two oscillators.

(b) Multiple Coupled Oscillators

For an actual animal or a biomimetic robot, multiple oscillatory joints may be involved in the locomotion, which requires more oscillators to generate coordinated gaits. In this case, multiple coupling may be imposed on one oscillator, for example, the oscillator *i* in Fig. 5.5. The figure shows a chain structure of three serially connected oscillators. Oscillator *i* are influenced by the outputs of the oscillator *i*-1 and oscillator *i*+1. In this case, $\mathbf{P}_{i\text{-}1}$ can also be defined by Eq. (5.19). $\mathbf{P}_i$ and $\mathbf{P}_{i+1}$ are defined by:

$$\mathbf{P}_i = \begin{bmatrix} 0 \\ \varepsilon(u_{i-1}\sin\varphi_{d,i} + v_{i-1}\cos\varphi_{d,i} - u_i\sin\varphi_{d,i+1} + v_{i+1}\cos\varphi_{d,i+1}) \end{bmatrix} \tag{5.21}$$

$$\mathbf{P}_{i+1} = \begin{bmatrix} 0 \\ \varepsilon(u_i\sin\varphi_{d,i+1} + v_i\cos\varphi_{d,i+1}) \end{bmatrix} \tag{5.22}$$

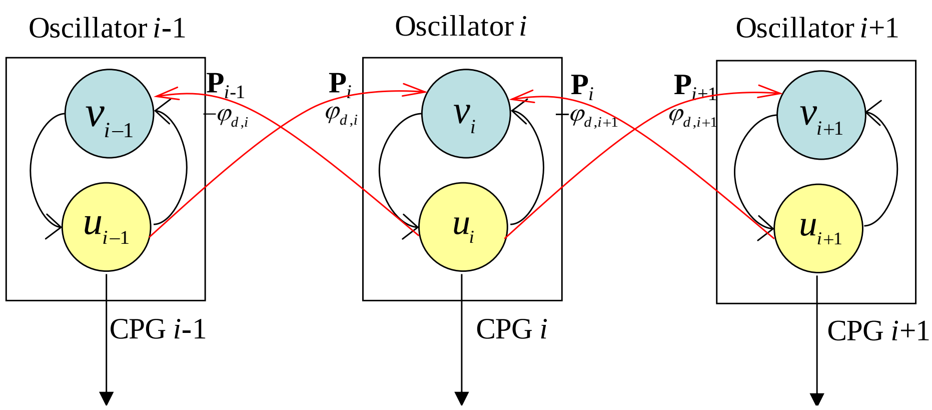


Fig. 5.5 Couplings among three oscillators.

5.1.3 Configurations of Oscillators

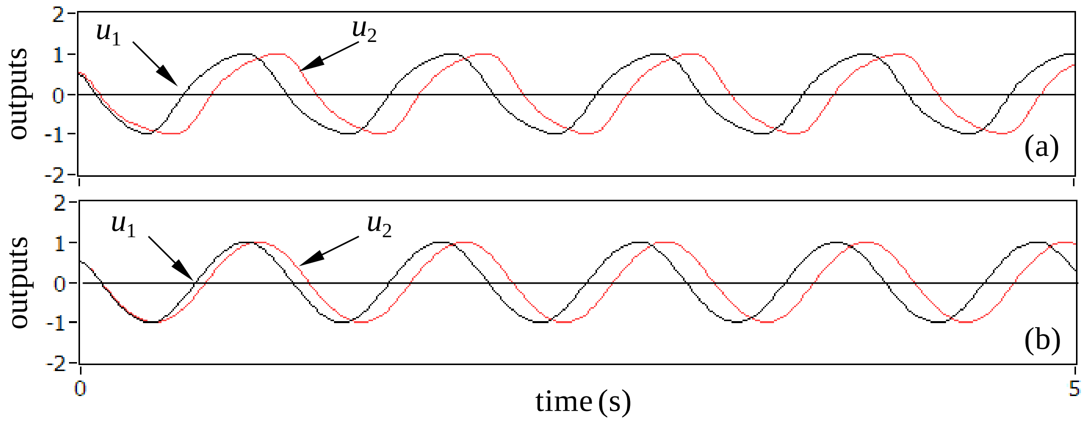


Fig. 5.6 Outputs of two oscillators with one-directional coupling and different coupling strength: (a) ε=5 and (b) ε=0.8. Other parameters are: k=10, φ_d=-1rad, r=1, and f=1Hz.

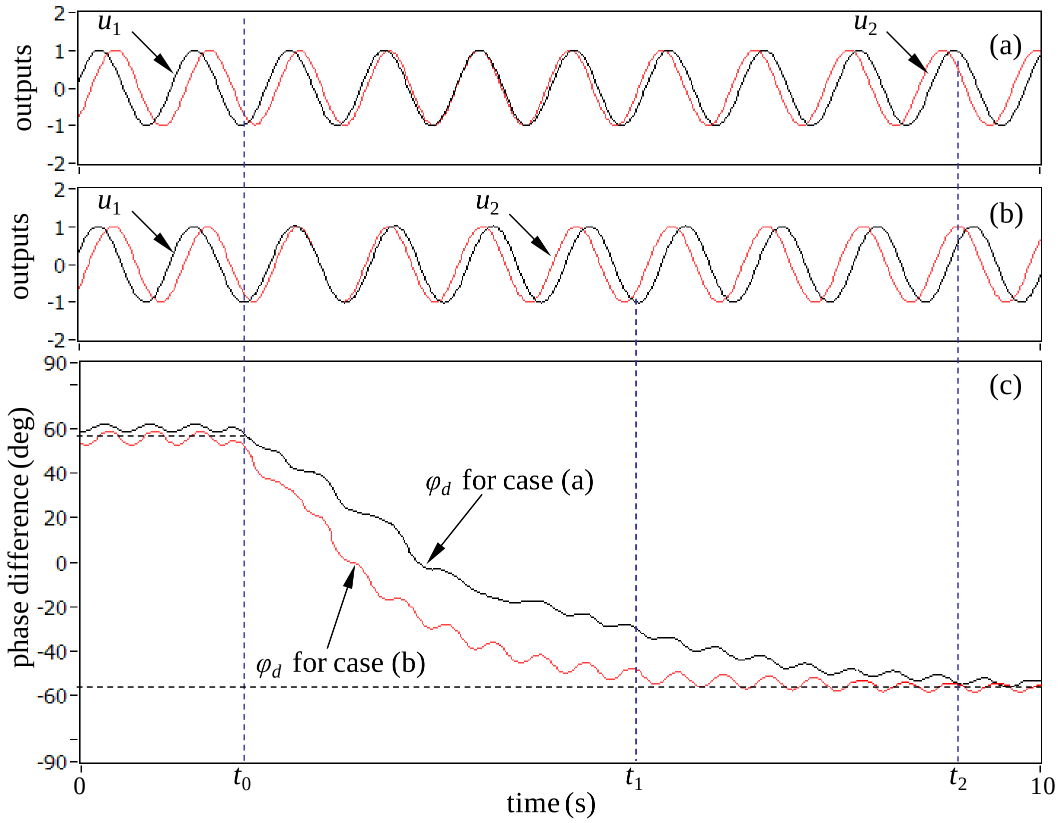


Fig. 5.7 Evolving of gait transition and phase difference: (a) gait transition under one-directional coupling; (b) gait transition under mutual coupling; and (c) evolving of phase difference.

One-directional coupling is adequate to maintain stable phase difference between oscillators. And this scheme is easy to be implemented in robot control. The shortcoming is that the gait transition will be relatively slow. In undulatory motion control, when a forward swimming becomes a backward swimming, the phase lag between two adjacent oscillators will flip into the phase lead. Such large changing of phase difference will cause the slow gait transition. This problem can partly solved by increasing the coupling strength. However, large coupling strength distort the shape of output wave (see the example shown in Fig. 5.6a). In Fig. 5.6b, it can be seen that the harmonic gait pattern is maintained, but the evolving of the phase difference becomes slower.

By applying mutually coupling scheme, the transition becomes faster. Fig. 5.7 shows the evolving of the phase difference for different coupling. For (a) and (b), initially the phase lag between u_2 and u_1 is set to 1 rad. At time t_0, the phase lag is changed to -1 rad. Parameters are: ε=0.8, k=10, r=1, and f=1Hz. It can be seen that mutually coupling results in faster transition ($t_1<t_2$). The disadvantage of mutual coupling is that implementation of this scheme requires more computing cost. In circumstances where fast gait transition is not a critical requirement, one-directional coupling is sufficient.

The coupling of two oscillators is fundamental for construction of artificial CPGs. There exist several topological structures of CPGs when more oscillators are involved. Commonly adopted structures include the chain type [26], the ring type [171], the radial type [172], the fully connected type [120] and the combinations of above [11]. Besides, given one type of topological structure, the coupling can be different, as discussed in previous sections. Fig. 5.8 shows four fundamental structures.

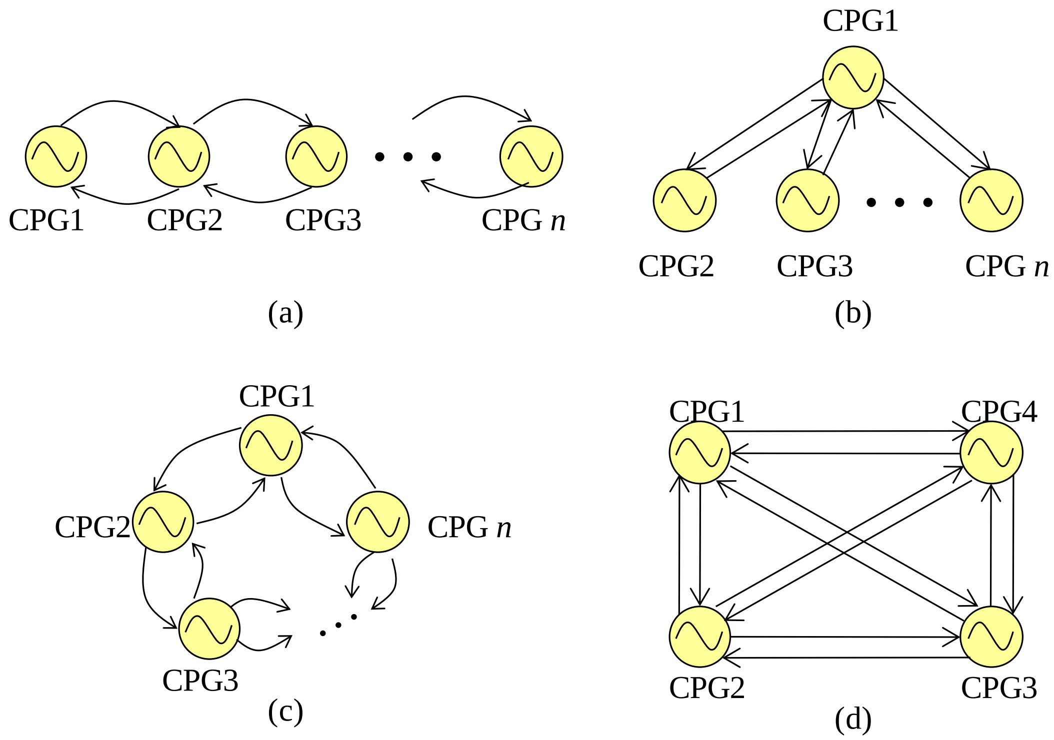


Fig. 5.8 Fundamental topological structures of *n* multiple coupled oscillators: (a) chain type; (b) radial type; (c) ring type; and (d) fully connected type (number of oscillators is restricted to four).

The topological structure of coupled oscillators simulates that of biological CPGs by connecting mutually-affected neurons together. However, the structure shown in Fig. 5.8 is only an engineering simplification of the real CPGs which are complex network of large amount of neurons. It actually has no direct counterparts in biological CPGs. Therefore, there are no standard principles that determine which structure should be adopted. The selection of the structure depends on actual situation of applications in a robot. For example, for the fish robot, NKF-II, developed in the present research, the chain structure can be adopted as the fin segments are serially connected. For a quadruped robot, motions of all its four legs must be coupled and the fully connected structure can be applied.

5.2 Application of the CPG Model in on a Fish Robot with Single Fin Propulsor

The synchronization of multiple oscillators is an extension of two coupled oscillators. Those coupled oscillators can be thought of CPGs controlling the locomotion of multi-DOF swimming robots. With different configurations of oscillators, various swimming motion patterns can be obtained. In this section, an eight-DOF biomimetic propulsor is taken as an example to illustrate the application of the CPG model.

5.2.1 Configuration of Oscillators for NKF-II

For the propulsor prototype shown in Fig. 3.8b, there are totally eight oscillatory joints involved in thrust generation. In order to generate undulatory wave on the long propulsor, eight oscillatory signals are necessary to coordinated motions. The oscillation of each joint has specific phase difference with the adjacent ones. Therefore, it is natural to apply a chain structure of oscillators to build the CPG model, which is illustrated by Fig. 5.9.

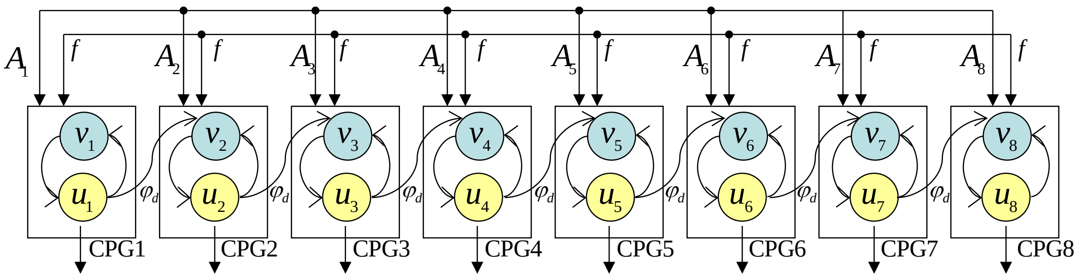


Fig. 5.9 A chain structure CPG model with chain structure and single-directional coupling for NKF-II.

To simplify the implementation, one-directional coupling is applied. The amplitudes A_i are determined by the profile of the undulatory wave. All oscillators (CPGs) share the same frequency to keep the integrity of the motion [33]. The phase differences for any adjacent joints are also identical because all joints are equally spaced for the given prototype. In our experiments, the coupling scheme given in Fig. 5.9 works well in steady state swimming. However, the gait transition between forward swimming gait and backward gait under this chain structure is slow. The reason is that each oscillator is synchronized with the previous one and the lag dynamics propagates and accumulates one by one, especially when the number of oscillators is large. An alternative solution can be adopted by employing the coupling scheme shown in Fig. 5.10. All oscillators are synchronized to the first one with different stable phase lags depending on the position of the associated joint on the prototype. It is observed in experiments that the actual transition time is significant reduced with the revised scheme.

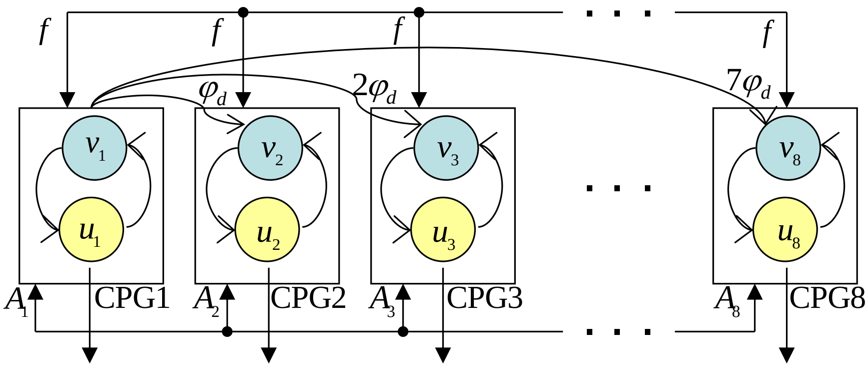


Fig. 5.10 Configuration of multiple oscillators with radial structure and single-directional coupling.

According to configuration shown in Fig. 5.10, the undulatory swimming motion can be controlled by the following artificial CPGs:

$$\theta_i(t) = A_i u_i \tag{5.23}$$

$$A_i = \sin^{-1}\frac{f_e(x_i)}{l_i} \tag{5.24}$$

$$\begin{bmatrix} \dot{u}_i \\ \dot{v}_i \end{bmatrix} = \begin{bmatrix} k(1-u_i^2 - v_i^2)u_i - 2\pi f v_i \\ k(1-u_i^2 - v_i^2)v_i + 2\pi f u_i \end{bmatrix} + \begin{bmatrix} 0 \\ u_{i-1}\sin\varphi_{d,i} + v_{i-1}\cos\varphi_{d,i} \end{bmatrix} \tag{5.25}$$

$$\varphi_{d,i} = \varphi_d = -\frac{2\pi m}{n} \tag{5.26}$$

$$\varepsilon = \begin{cases} 0 & i=1 \\ \text{positive const.} & i>1 \end{cases} \tag{5.27}$$

where i=1, 2, …, 8, n is the number of DOFs (actuators), n_λ is the number of waves on the long propulsor, A_i is the amplitude of the i^{th} oscillator, l_i is the length of the i^{th} driving crank, and $f_e(x_i)$ is the envelop equation shaping the profile of the undulatory wave. Output Eq. (5.23) generates harmonic gait control signals for oscillatory joints involved in propulsion.

5.2.2 Experiments

Experiments were conducted in the lab water tank with configurations described in Fig. 3.8. The testing prototype performs straight-line motion on the rail. The experimental control architecture is described in Fig. 5.11. The hardware configuration is as same as that in Fig. 4.18. A part of control program is shown in Fig. 5.12. The nonlinear equations of the oscillators are solved with the fourth-order Runge-Kutta method in Labview^{TM} running in the NI-9073 real-time controller. Control commands can be manually set in PC and sent to the real-time controller through Ethernet connection. Gait signals obtained from Eq. (5.23) are encoded into PWM signals to control servo motors, which will be transmitted to motor drive through a high speed I/O module NI-9401 (see Fig. 4.18).

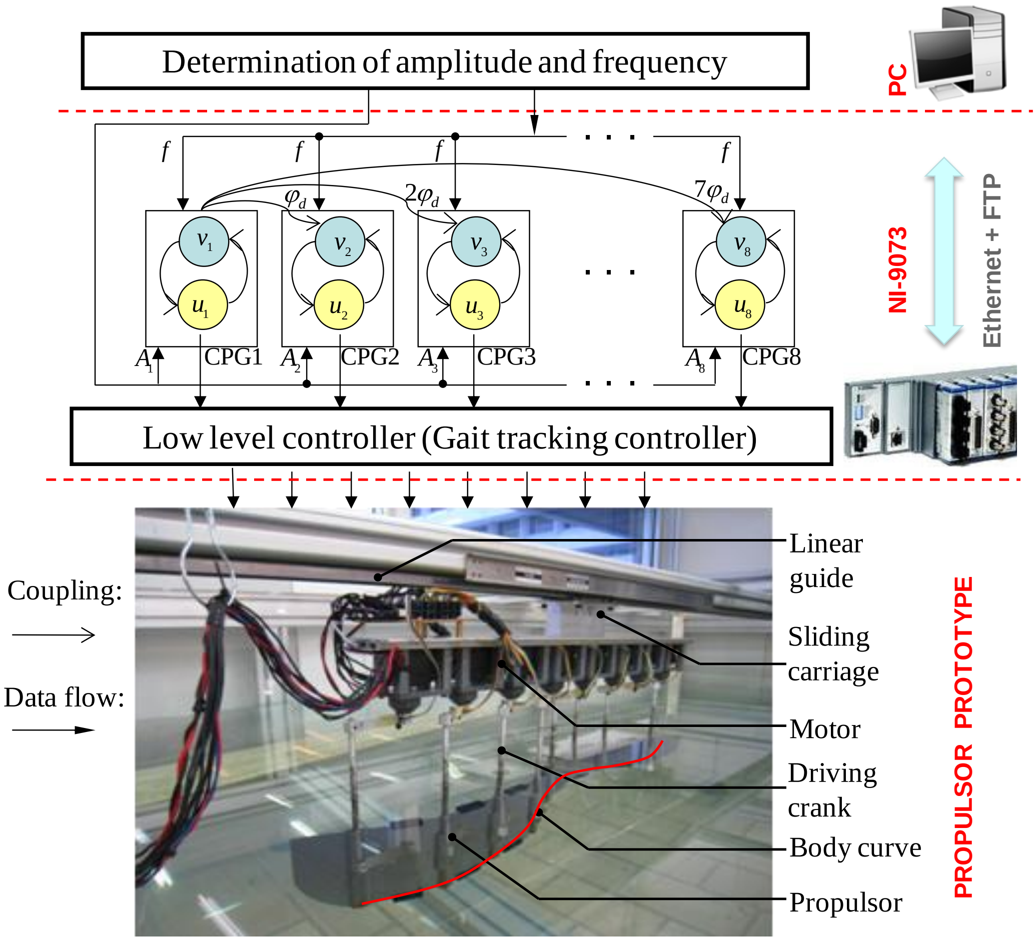


Fig. 5.11 Control architecture based on CPGs.

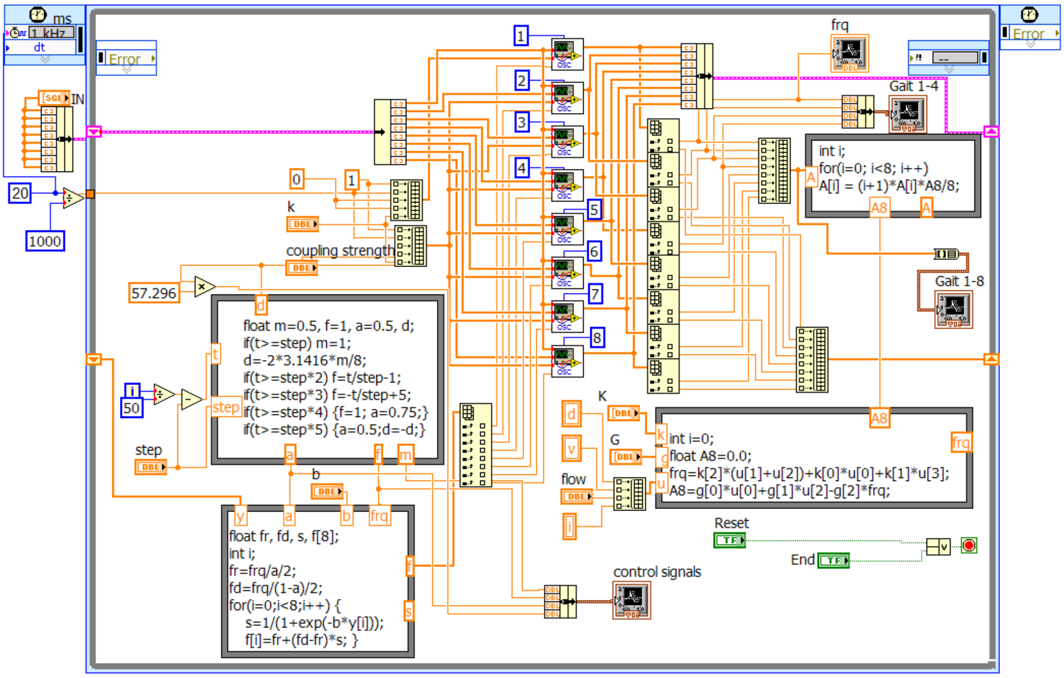


Fig. 5.12 The LabVIEW program used to control the swimming motion. The user defined sub VI, osc.vi, solves equations of CPG model by fourth-order Runge-Kutta method.

Experiments were conducted to test the swimming performance under CPG-based control. The testing includes several aspects: i) on-line change of wave numbers by tuning n_λ (see Fig. 5.13a); ii) smooth accelerating and decelerating by increasing and decreasing the oscillation frequency f (see Fig. 5.13b); and iii) the forward-backward swimming gait transition by flipping the sign of the phase difference φ_d (see Fig. 5.13c).

The evolving of the swimming gaits for eight fin rays are illustrated in Fig. 5.13d. When t<5s, harmonic (α=0.5) forward (φ_d<0) swimming gaits are applied to generate a half waveform (n_λ=0.5) on the long propulsor. From t=5s, full length waveform is performed with the wave number jumps to one and the phase lag between two adjacent oscillators increases to 45° subsequently. A seamless evolving can be seen from Fig. 11d. During t=5s~10s, the accelerating and decelerating process by changing the oscillation frequency is tested. The traveling speed of the undulatory wave is tuned in this way and swimming speed is changed subsequently. The oscillations of joints smoothly get faster, then smoothly slower. The transition of forward gaits to backward gaits are performed staring form t=20s. The three red lines in Fig. 5.13d shows the difference: in forward swimming, the motion of the i^{th} oscillator lags behind the $(i+1)^{th}$ while in backward swimming the motion of the i^{th} oscillator leads the $(i+1)^{th}$.

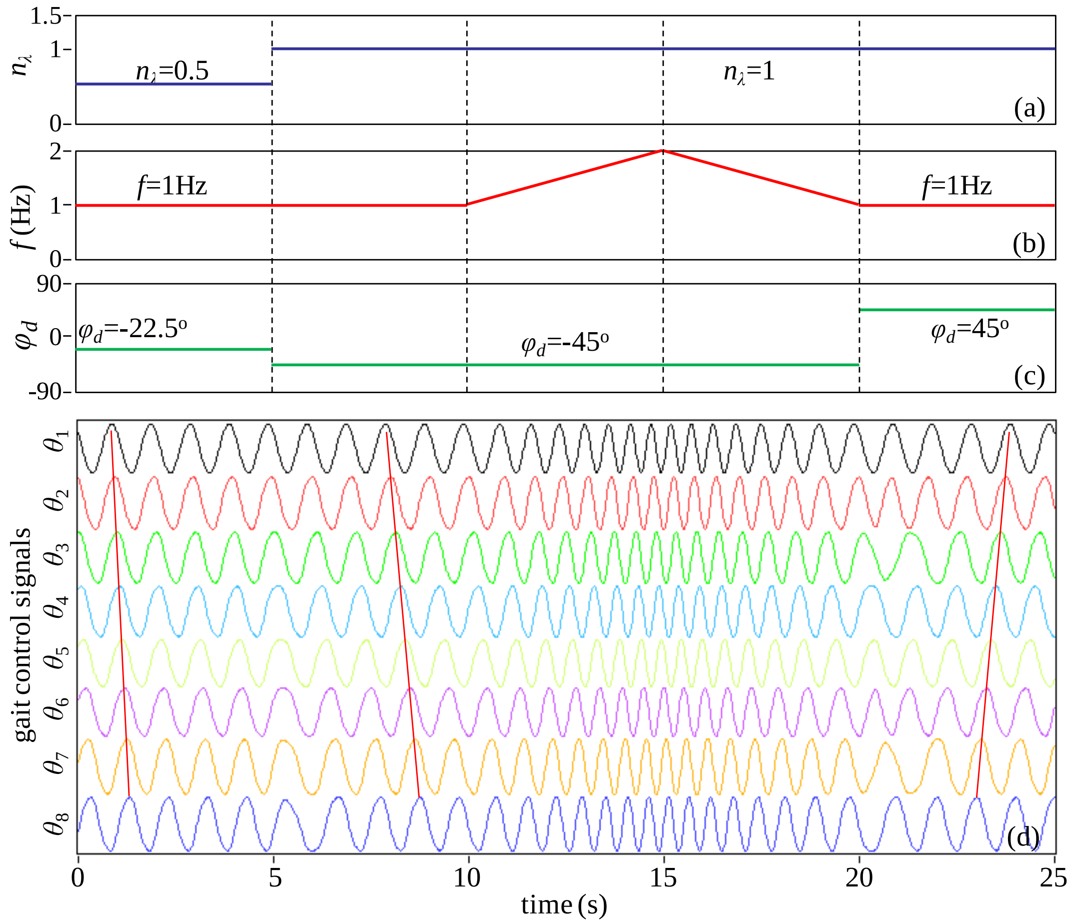


Fig. 5.13 The experimental swimming gait control signals generated by the CNLOs (CPG model): (a)-(c) on-line tuning of control parameters and (d) the evolving of gaits for eight driving cranks. The amplitudes are normalized to 1. Parameters in Eq. (5.23)-(5.27) are: n=8, ε=1 and k=10.

Under the control signals given in Fig. 5.13, swimming speeds were recorded and plotted in Fig. 5.14. Note that the control parameters illustrated in Fig. 5.13a-c are modeled with piecewise functions in experiments. It can be seen that even if those parameters are not smoothly tuned, the gaits are still continuous and smooth. This is one of the most attractive properties of the CPG based method. By applying the gait control signals, smooth and natural locomotion of the biomimetic propulsor is obtained in experiments.

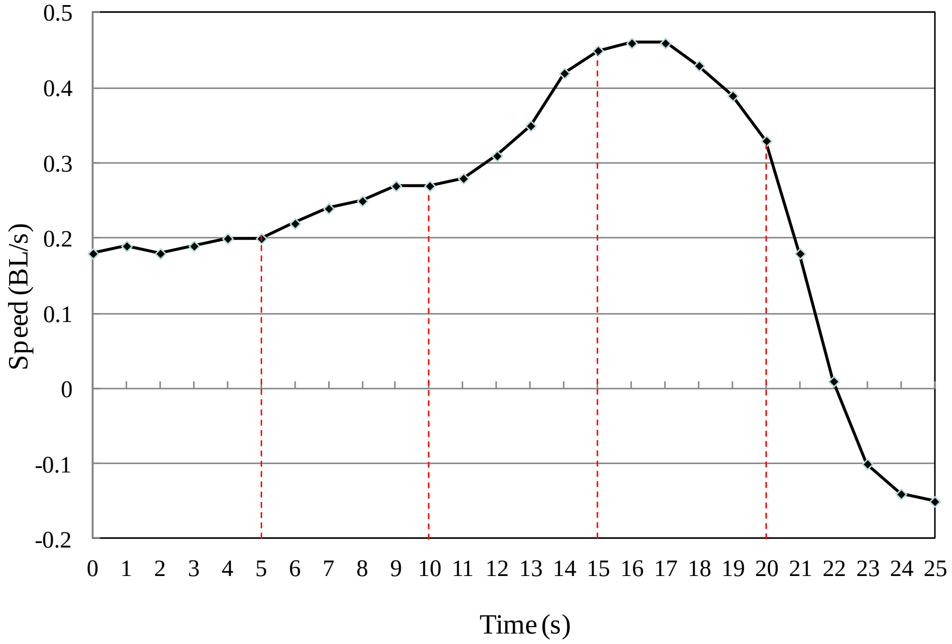


Fig. 5.14 Swimming speed of NKF-II under control of gait signals given in Fig. 5.13. Amplitude is 0.05 BL (30mm) Negative speed indicates (t > 22s) the prototype performs backward swimming.

5.3 Application of the CPG Model on a Fish Robot with Paired Fins

The manta ray robot (RoMan-II) uses two symmetrical sided fins as propulsor. Each fin consists of three driving fin rays which oscillate alternately to produce thrust. The CPG model is also suitable to control the locomotion of RoMan-II.

5.3.1 Configuration of Oscillators for RoMan-II

For RoMan-II, each fin is driven by three brushless servo motors and totally six oscillators are needed to generate the swimming gaits for six fin rays. Every three oscillators are coupled with a chain structure for one side of fin control. All oscillators are coupled with the first oscillator. The configurations of oscillators are shown in Fig. 5.15. Both sides can be controlled independently and there are no mutual couplings between the two sides, expect for the first oscillators of both sides (oscillator one and oscillator four in Fig. 5.15).

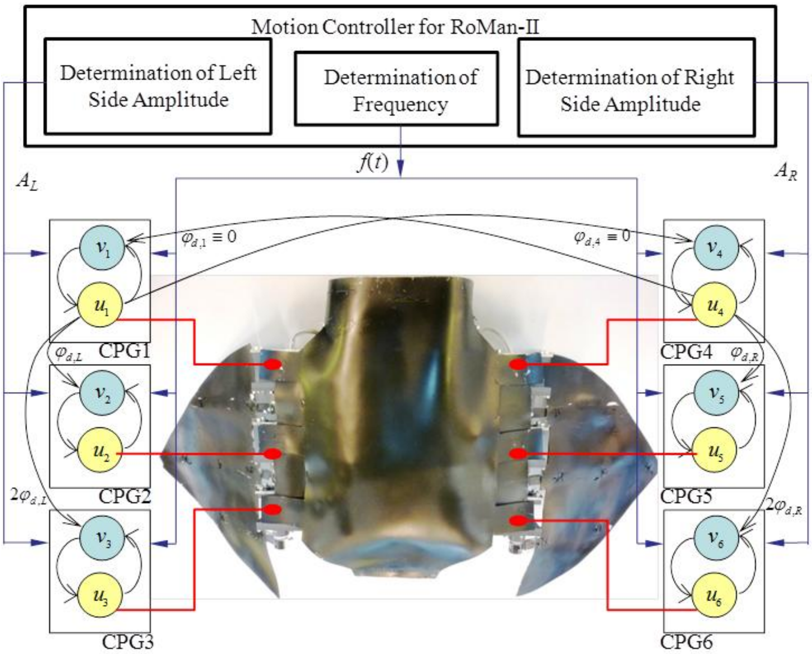


Fig. 5.15 CPG based motion control structure for RoMan-II. CPG1-3 control the left fin motion and CPG4-6 control the right. For this robot, n=3 and n_λ=0.5.

According to the analysis in Chapter 2, in one cycle oscillation of a fin ray, the upstroke only takes about half time of the down stroke [36]. This indicates that harmonic functions (sine function for example) may not be accurate models to generate gait pattern of a manta ray robot. The swimming gait for RoMan-II can be modeled with a non-harmonic pattern. The adaptive frequency defined in Eq. (4.9) will be used to shape the gait pattern with α=1/3. The locomotion control equations for left side motion of RoMan-II are given below:

$$\theta_i(t) = A_i u_i \tag{5.28}$$

$$\begin{bmatrix} \dot{u}_i \\ \dot{v}_i \end{bmatrix} = \begin{cases} F(\mathbf{X}_i) + \varepsilon \begin{bmatrix} 0 \\ v_4 \end{bmatrix}, & i = 1 \\ F(\mathbf{X}_i) + \varepsilon \begin{bmatrix} 0 \\ u_1 \sin[(i-1)\varphi_{d,L}] + v_1 \cos[(i-1)\varphi_{d,L}] \end{bmatrix}, & i = 2, 3 \end{cases} \tag{5.29}$$

$$f(t) = \frac{3}{4T}\left(1 + \frac{1}{1 + e^{-\tau v}}\right) \tag{5.30}$$

where $F(\mathbf{X}_i)$ is the i^{th} Hopf oscillator defined in Eq. (5.1), T is the nominal flapping period, and $\varphi_{d,L}$ is the phase difference between two adjacent oscillators in the left fin.

The equations for the right side are similar to that of the left side. The mutual coupling between the first two oscillators with zero phase difference is necessary in this case. When the robot swims forward along a straight path, it is necessary that the two fins perform exactly the same motion. This symmetry of motion may not be maintained without the mutual coupling. The coupling with zero phase difference will make oscillator one and four always locked in phase. All other oscillators are coupled with the first oscillators. Given that the flapping motion of the manta ray is not pure undulation, it is assumed that the fin generates half cycle waveform in forward swimming (n_λ=0.5). As a result, the phase difference can be given by:

$$\varphi_{d,L} = \varphi_{d,R} = \varphi_d = -\frac{2\pi m}{n} = -\frac{\pi}{3} \tag{5.31}$$

5.3.2 Control of Turning by Tuning the Phase Difference

CPGs in one side share the same nominal frequency to maintain the continuity of motions. The frequency and amplitude for both sides may be different when the robot performs turning motion. The unbalanced thrust forces generated by the two sides can cause the steering of locomotion. Fig. 5.16 illustrates several cases of steering. For example, if left side amplitude or frequency is smaller than that of the right side, the robot will turn left (Fig. 5.16c).

The unbalanced thrust forces of two sides may be explained by the different amplitude and/or flapping frequency. Instead of changing amplitudes or frequencies of both sides, here another way to control the turning motion by tuning the phase differences is presented. Take cases in Fig. 5.16c-d as examples. If the phase difference in the left side, $\varphi_{d,L}$, decreases from φ_d, the left side will generates weaker and weaker forward thrust and the robot will turn to left (Fig. 5.16c). If $\varphi_{d,L}$ is smaller than zero, the left side will generate backward thrust and if $\varphi_{d,L}$=- φ_d, the robot can perform pivot turning (Fig. 5.16d). This analysis is also applicable for $\varphi_{d,R}$ when the robot turns to the right. A new variable, q, is introduced to control the turning motion. By changing value of q, the swimming direction can be controlled. The relationship between the swimming direction and q is listed in Table 5.1 and described by Fig. 5.17.

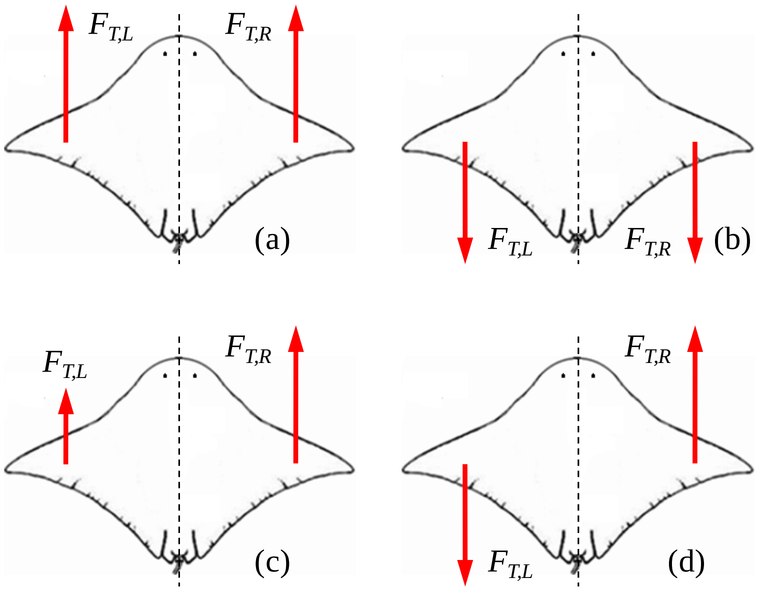


Fig. 5.16 Illustration of thrust forces of a manta ray in steering: (a) forward swimming; (b) backward swimming; (c) turning left; and (d) pivot turning to the left. $F_{T,L}$ and $F_{T,R}$ are thrust forces generated by left side fin and right side fin, respectively. The flapping frequency and amplitude for both sides are assumed identical.

Table 5.1 The relationship between the swimming direction and q.

q (φ_d<0)	q =-1	-1< q <0	q =0	0< q <1	q =1
Swimming direction	Right pivot turning	Right turning	Swimming forward	Left turning	Left pivot turning

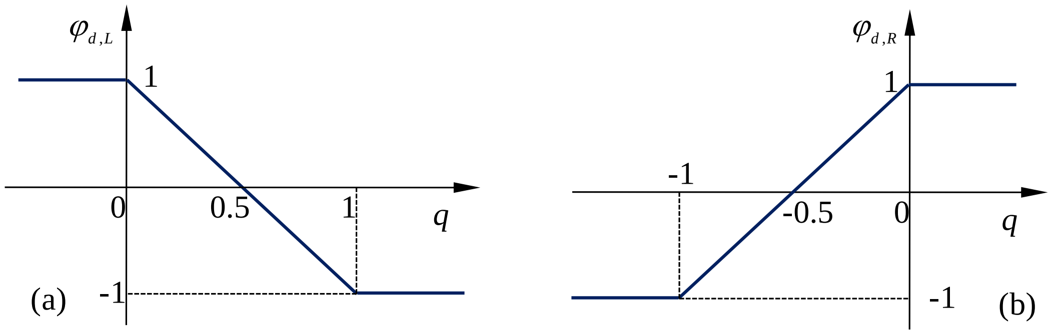


Fig. 5.17 Phase difference for the left side fin (a) and right side fin (b) under the control of q.

According to Fig. 5.15, the phase differences in the two fins can be defined by using the bi-polar sigmoid function:

$$\varphi_{d,L} = \frac{e^{-\tau'(q-0.5)} - 1}{e^{-\tau'(q-0.5)} + 1}\varphi_d \tag{5.32}$$

$$\varphi_{d,R} = \frac{1 - e^{-\tau'(q+0.5)}}{1 + e^{-\tau'(q+0.5)}}\varphi_d \tag{5.33}$$

where *τ'* is a time constant tuning the convergence speed. The sigmoid function is used because it ensures a continuous and smooth tuning of phase difference. In order to match the evolving curves shown in Fig. 5.17, the following requirements are set:

$$|\varphi_d| - |\varphi_{d,L}| \le 5\% \text{ and } |\varphi_d| - |\varphi_{d,R}| \le 5\% \tag{5.34}$$

By combining (5.32)-(5.34), it can be known that the minimum value of *g* should be 3.66. In actual applications, *τ'* can be set to 4. The significance of introducing *q* in the control is that the steering of the robot can be controlled by using this simple variable. It provides a way to incorporate sensory feedback into the CPG model to tune the swimming direction. Like tuning the frequency and amplitude in station-holding control, feedback control laws can also be fabricated to adjust *q*, which is expect to be useful in accurate direction control.

5.3.3 Experiments

Initial Experiments were conducted on RoMan-II in a swimming pool. The nonlinear differential equations in Eqs. (5.28)-(5.30) are solved by the fourth-order Runge-Kutta method with C programming [173]. The control of motors is achieved through microcontroller (μC) based hardware. Fig. 5.18 shows the hardware hierarchy. One PIC18F4520 μC is used as master controller whose function includes: determining frequency and amplitude to slaves, determining phase difference between oscillators, processing the sensory feedback signal and wireless communication with PC. Functions of six PIC18F2431 slave μC include: receiving the frequency, amplitude and phase lag data from master, solving equations for each CPG, and encoding gait signals obtained from (5.28) into PWM signals to control the servo motor.

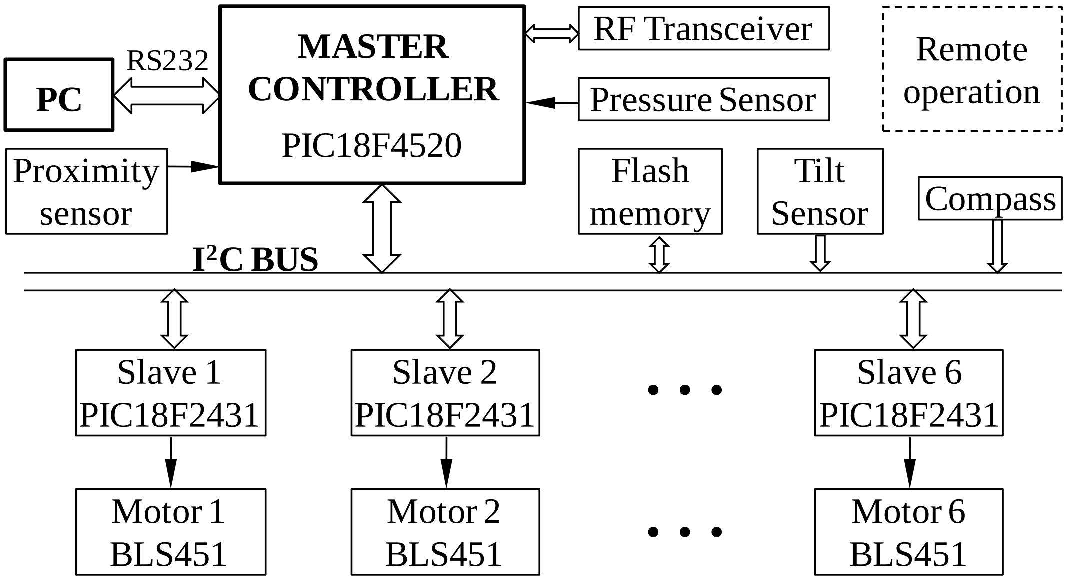


Fig. 5.18 Control hardware hierarchy for RoMan-II.

Experiments were conducted to test the swimming performance under CPG-based control. The testing includes several aspects: (i) generation of non-harmonic swimming gait; (ii) smooth accelerating by increasing the nominal oscillation frequency fin rays; (iii) left turning motion by increasing q; and (iv) left pivot turning when q >1. The control signals for flapping, speed changing, and turning are shown in Fig. 5.19. Variable f and q are two control inputs. Their variations against time are plotted in Fig. 5.19a and Fig. 5.19b, respectively. By solving Eqs. (5.28)-(5.34), the six gait control signals can be obtained, which is illustrated by Fig. 5.19c.

Initially, the CPG model generates a non-harmonic gait pattern. Both fins produce a backward traveling wave and the robot performs forward swimming (t=0 s to15 s). The wave number is selected to 0.5 to mimic ray-like swimming. A stable phase lag can be maintained as indicated by the dashed lines in Fig. 5.19c. It is seen that θ_1 leads θ_2 and θ_3 in the left fin. It is the same case for the right side. At t=5s, as the nominal flapping frequency becomes gradually larger, the oscillation smoothly gets faster and the robot performs accelerating.

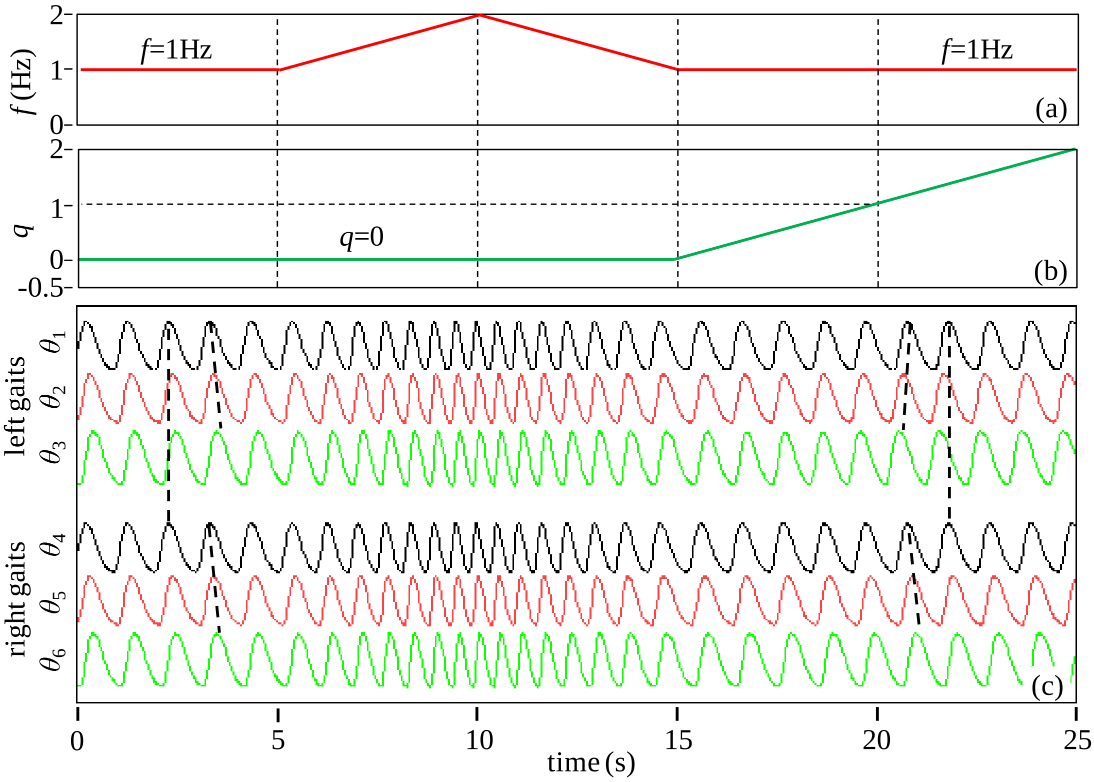


Fig. 5.19 Illustrated experimental results (a) control input f; (b) control input q; and (c) the experimental swimming gait control signals generated by the CPG model. The six signals, $\theta_1 - \theta_6$, are obtained by solving (5.28)-(5.34). The amplitudes are normalized to 1. Other parameters are: τ'=4, τ=20, n=6, α=0.33, ε=2.5 and k=10.

During t=10 s to 15 s, the robot performs decelerating as the nominal frequency becomes smaller. In the region of t=15 s to 20 s, $\varphi_{d,L}$ is gradually decreased under control of q and the robot performs left turning. The results show that the $\varphi_{d,R}$ keeps unchanged, whereas $\varphi_{d,L}$ becomes smaller until $\varphi_{d,L}$ =-φ_d. In this period, the robot turns to the left. The turning radius is controlled by q. During t=20s to 25s, the left fin produces forward traveling wave as θ_3 leads θ_2 and θ_1 and the right fin still produce backward traveling wave. The robot now performs left pivot turning. Because oscillators 1 and 4 are locked in phase with mutual coupling, these two oscillators always evolve simultaneously. Fig. 5.20 shows flapping motions of RoMan-II in forward swimming testing. In experiments, smooth gait transitions are obtained. The maximum steady forward swimming speed reaches 0.8 BL/s.

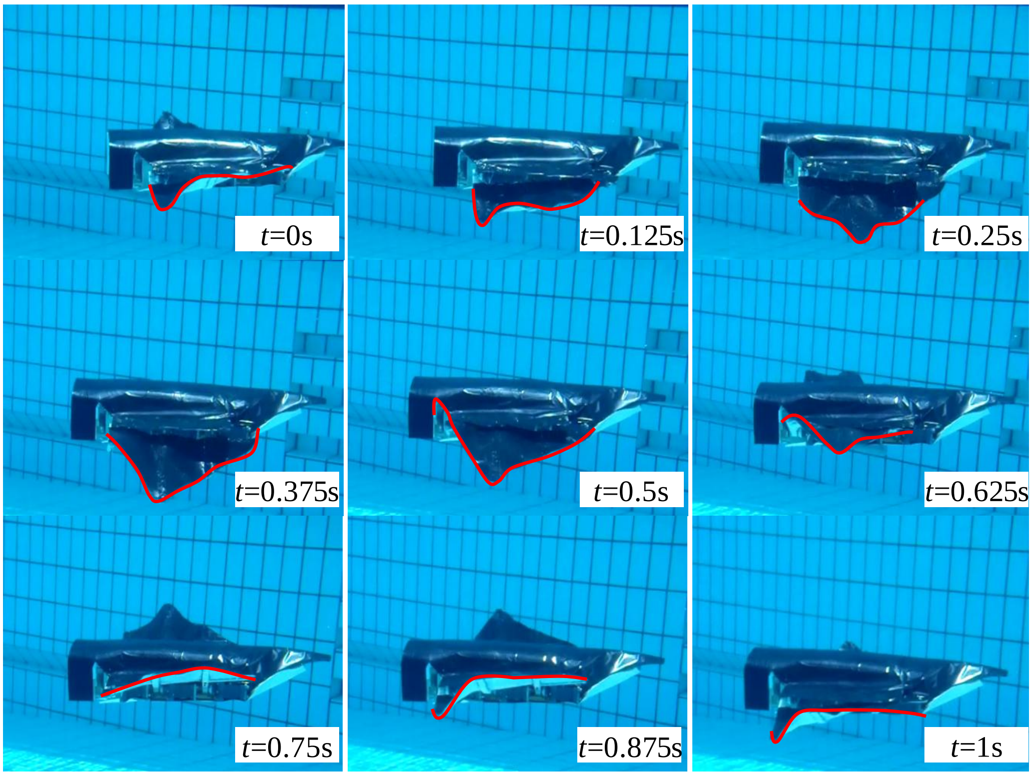


Fig. 5.20 Snapshots of RoMan-II in swimming locomotion in one second (f = 1 Hz). The robot performs down stoke flapping during t = 0 s to 0.375 s and upstroke flapping during t = 0.375 s to 0.875 s. Red curves indicate the profile of the left fin edge.

Turning motion of the robot was tested in experiments to find the turning radius. According to the analysis in Section 5.3.2, the turning motion can be obtained by unbalanced thrust of two fins. The testing results of right turning experiments are listed in Table 5.2.

Turning is achieved by adjusting the motion of the right fin. The motion of the left fin remains unchanged. In cases 1 & 2, both fins produce backward propagating wave and the robot swims forward. As the right fin generates weaker thrust due to lower amplitude (case 1) or lower frequency (case 2) or both (case 3), the robot also performs right steering. In case 4, as the two fin produces waves propagating to the opposite direction, the robot should perform right pivot turning with zero turning radius. However, experiment shows a small radius exists. It may be caused by imperfect asymmetry of mechanical designs of two fins.

Table 5.2 Turing radius test of RoMan-II

Case	**Motion**	**Left Fin**			**Right Fin**			**Turning Radius** (m)
		A (deg)	f (Hz)	Wave Dir.	A (deg)	f (Hz)	Wave Dir.	
1	turn right	20	1	Backward	10	1	Backward	0.84
2	turn right	20	1	Backward	20	0.5	Backward	0.71
3	turn right	20	1	Backward	0	0	Backward	0.05
4	right pivot turn	20	1	Backward	20	1	Forward	0.01

5.4 Summary

Kinematic modeling of swimming gaits based on CNLOs for locomotion control of multi-DOF fish robots are formulated in this chapter. The emphasis is place on the analytical derivation of coupling terms used to maintain the stable phase differences among oscillators. Those coupled oscillators can be regarded as artificial CPGs that control the rhythmic motion of biomimetic propulsors. The applications of the CPG model in swimming motion control of fish robots are investigated and tested on two different multi-DOF fish robots. Experiments show that the CPG based method has advantages in terms of the on-line generation of swimming gaits and on-line gait transitions. This method is expected to be a framework for modeling and control of swimming locomotion.

Although the hydrodynamic aspects of the locomotion, for example, the adaptation to actual flow situation and the energy efficient swimming, are not included, the method discussed in the chapter provides potentials in performance improvement of those aspects. Firstly, the method provides a simple way to on-line switch motion patterns, which can be achieved by tuning corresponding parameters as illustrated in Fig. 5.13 and Fig. 5.19. Secondly, the evolving of the gaits maintains smooth and continuous as those parameters vary. These features make the closed-loop swimming control possible. For example, feedback control laws can be developed to tune the parameters in Fig. 5.13 and Fig. 5.19. In this case, sensory feedback of water environment is incorporated into the modeling framework, which will be useful to achieve autonomous or adaptive locomotion. The station-holding control on multi-DOF fish robots is such an example.

In experiments on NKF-II and RoMan-II, the amplitudes of all oscillators are set to constant values. Those predefined values are adopted just to illustrate how the model works. The model allows all amplitudes can be different and independently controlled. The real-time tuning of those amplitudes may influence the swimming efficiency, which will be investigated in the following chapters.

Chapter 6 Robust Tracking of Swimming Gaits

Upon implementation on real robots, it is expected that gait control signals can be tracked by actuators of robots accurately. For example, when it comes to testing what kind of motion pattern is superior over others in experiments, one basic precondition is that the robot system can exactly perform the desired motion pattern. Without this precondition, it cannot be accepted that the experiment data obtained are true reflections of the specific motion pattern. Because of the un-modeled hydrodynamics and the uncertainties among parameters of mathematical model of fish robot, it is critical for the motorized robot system to track the periodical gait control signals in water environment. The oscillatory gait motion results in reactive force from water, which brings continuously varying disturbance to the control system. A robust motion control system is necessary to track the gait signal and reject disturbance. This motor-level control problem will be addressed in this chapter.

Two methods are applied to solve this problem. Firstly, the disturbance is treated as a velocity damping with a constant coefficient. Usually, PID controllers are applied as the tracking controllers [28, 129, 174] for such linear system. In the second method, the disturbance from the reactive force will be modeled and incorporated into the controller design. A robust tracking system will be developed for the actuators of a fish robot to track the periodical input and rejecting disturbance caused by the interaction force with water. The equations of the robust tracking controller will be derived in this chapter.

The two-DOF fish robot (NAF-II) actuated by an electric DC motor is adopted to illustrate the design process of the robust gait tracking controller. The tracking controller for fish robots with multiple actuators can be achieved by duplicating the same design process. Firstly, limitations of normal PID tracking controller will be discussed in Section 6.1. As an improvement, the robust tracking controller will be formulated in Section 6.2. The robust tracking controller is tested on NAF-II, which will be presented in Section 6.3.

6.1 PID Controller for Swimming Gait Tracking

A PID controller can be applied to regulate the gait tracking. However, it is found in our experiments [7] that the tracking control system is not robust when the frequency or amplitude of gait control signal is changed because those changes will cause significant variation of the disturbance and make the assumption of the velocity damping unreliable, which will be discussed in this section.

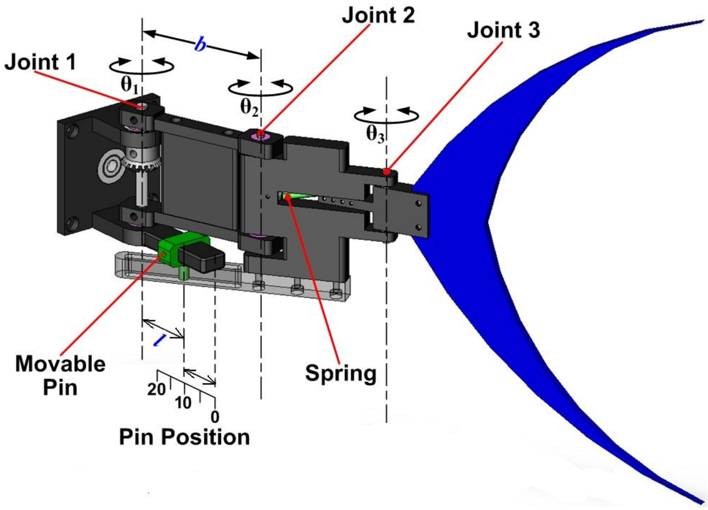
6.1.1 Motion Control Model of a Biomimetic Propulsor

Electric motors are commonly adopted robotic as actuators of biomimetic propulsor [15, 81]. The driven force/torque provided by motors is applied on the join of linkages to drive the fin or body segment. For each joint, the motor, the linkage and the related motion transmission mechanism consist of a small mechatronic system, which is the plant to be controlled. A multi-DOF swimming machine is made up of several such small systems. The fish robot, NAF-II, is such a system. The thrust of the robot is contributed by the oscillatory motion of the caudal fin. As the tail oscillates in water, the caudal fin pushes water aside and causes the distortion of water, which results in the reactive force. The force is related to the geometry of fin, the oscillation speed and the feature of fluid, as shown in Fig. 6.1.

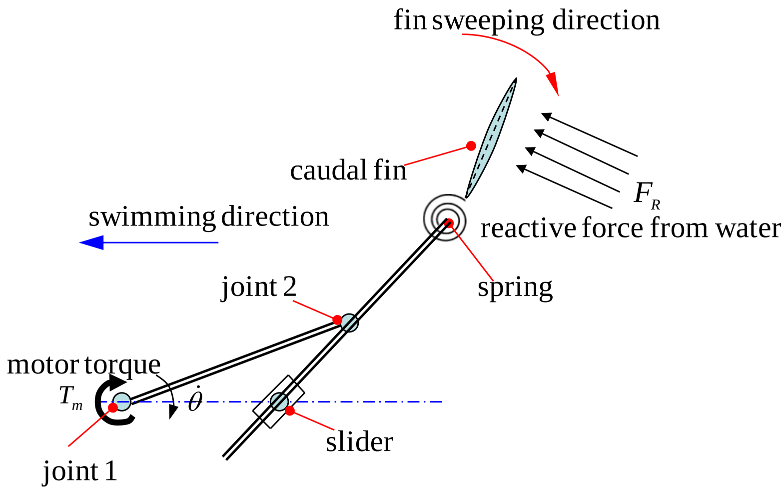
The water reactive force can be treated as a damping in the controller design for the tail fin. In order to obtain a motion control model, it is necessary to know the value of the damping coefficient (*b*). This coefficient is related to the geometry of caudal, the oscillation speed, the feature of fluid, and the attack angle of caudal fin to water flow. Here we use the following equation to evaluate the reactive force [175].

$$F_R = \frac{1}{2}\rho C_d S V_s^2 \tag{6.1}$$

where ρ is the flow density, C_d is the drag coefficient, and S is the characteristic cross-sectional area of the caudal fin and V_s is the sweeping speed of caudal fin. The reactive force varies periodically with the oscillation of the tail.



(a)



(b)

Fig. 6.1 (a) Tail fin mechanism of NAF-II and (b) the schematic of tail linkages and the water reactive force (top view of NAF-II).

It can be assumed that the force acts as a velocity damping to the control system, which is equivalent to a resistance torque (T_w) on motor shaft (joint 1 in Fig. 6.1b) and given by

$$T_w = b\dot{\theta} \approx \frac{RF_d}{G} \tag{6.2}$$

where R (= 0.07m) is the distance between the joint 1 and the center of the torsion spring, G is the gear ratio. To obtain a good performance, the precise evaluation of this damping coefficient is necessary. The value of damping can be obtained through experiments in the apparatus shown in

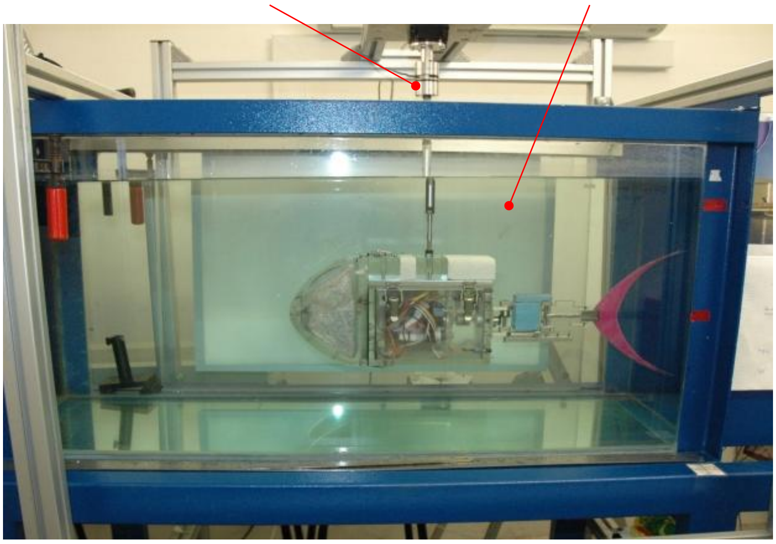


Fig. 6.2 Water tunnel and force measurement configuration [81].

An ATI-Gamma^{TM} six-axis load-cell transducer is utilized to measure forces generated by the fin motion. The transducer reacts to the applied forces and torques from the fin motion. By combining Eqs. (6.1) and (6.2), the damping coefficient can be obtained

$$b \approx \frac{R^2 F_d}{G V_s} \tag{6.3}$$

The range of this coefficient is 0.7514~1.12 mN s/rad. A DC motor can be treated as a linear actuator [146]. The dynamics of the motor-linkage caudal mechanism can be given as follows:

$$T_m = K_m I = (J\ddot{\theta} + b\dot{\theta}) / G \tag{6.4}$$

$$V_{in} - K_e \frac{\dot{\theta}}{G} = L_m \frac{dI}{dt} + IR_m \tag{6.5}$$

where T_m is the output torque of the motor, K_m is the torque constant, θ is the angular position of tail, I is the electric current through the armature, J is the equivalent moment of inertia of the tail mechanism, V_{in} is the supply voltage, K_e is the back electromotive force coefficient, L_m is the inductance of motor, and R_m is the resistance of motor armature. For the current mechanical system, the response delay of motor caused by the inductance of armature is much insignificant compared with the one caused by the inertia of mechanical components. Therefore the influence of the inductance can be neglected. Base on this assumption, by applying Laplace transformation to Eqs. (6.4) and (6.5), the open loop transfer function can be obtained:

$$\frac{\theta(s)}{\theta_d(s)} \approx \frac{GK_m}{s(JRs + bR + K_m K_e)} = \frac{K_o}{s(\tau s + 1)} \tag{6.6}$$

where K_o is the open loop gain, τ is the mechanical time constant, s is the Laplace operator and they are given by:

$$\tau = \frac{RJ}{bR + K_e K_m} \tag{6.7}$$

$$K_o = \frac{GK_m}{bR + K_e K_m} \tag{6.8}$$

$$K_e = K_m \tag{6.9}$$

The parameters of motor are listed in Table 6.1. The closed-loop tracking can be described by the block diagram in Fig. 6.3.

Table 6.1 Parameters of motion control model at nominal supply voltage.

Nominal Voltage V_{in} (V)	15
Torque constant K_m (mN m/A)	14.3
Terminal Resistance R_m (ohm)	0.902
Terminal inductance L_m (mH)	0.0882
Tail Inertia J (gcm^2)	87.3
Nominal Gear Ratio G	180/π/103
Mechanical time constant τ (ms)	8.2

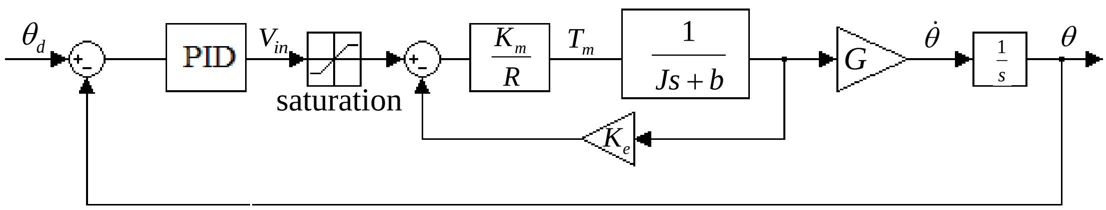


Fig. 6.3 Block diagram of tail motion control with PID controller.

In short, the open loop transfer function of the gait tracking control system can be given as follows:

$$\frac{\theta(s)}{\theta_d(s)} = \frac{K_o}{s(\tau s + 1)} \tag{6.10}$$

where K_o is the open loop gain, τ is the mechanical time constant, s is the Laplace operator. The tracking control loop can be described by the block diagram in Fig. 6.4.

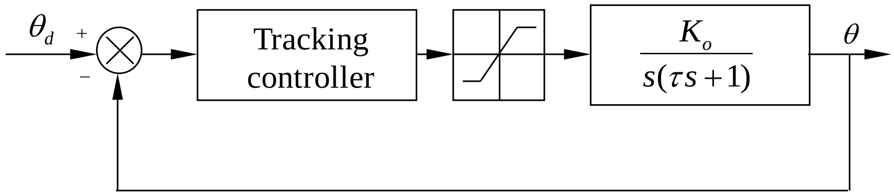


Fig. 6.4 Closed-loop gait tracking control.

6.1.2 Limitation of PID Controller

Design of a PD controller (Integral term is absent because the control plant already contains an integrator) is detailed in Appendix A. The PD controller works well to the step input (refer to Appendix A). When it is used to tracking control of periodical gait signals, such good performance is limited to control signals with certain parameters. Fig. 6.5 shows the simulation results of the tracking to different gait control inputs by using the PD controller. The controller is designed under parameter of case 1. Red curve denotes sine-pattern input signals. Good tracking performance can be observed. In case 2 and 3 where the amplitude and frequency are changed respectively, the tracking performance deteriorates.

In the design of a PD controller, the external reactive force from water is treated as linear damping to the control system. This assumption is only an approximation to the actual circumstance. The water reactive force is proportional to the square of the sweeping velocity and the damping coefficient varies periodically along with the periodical gait control input. If the input signal changes or the evaluation of damping coefficient is not accurate, the controller cannot provide high performance regulation. One way to solve this problem is to design a PD controller for every input case. However, because there are numerous combinations of input frequency and amplitude in practical application, a dynamic programmed PD controller can hardly be developed.

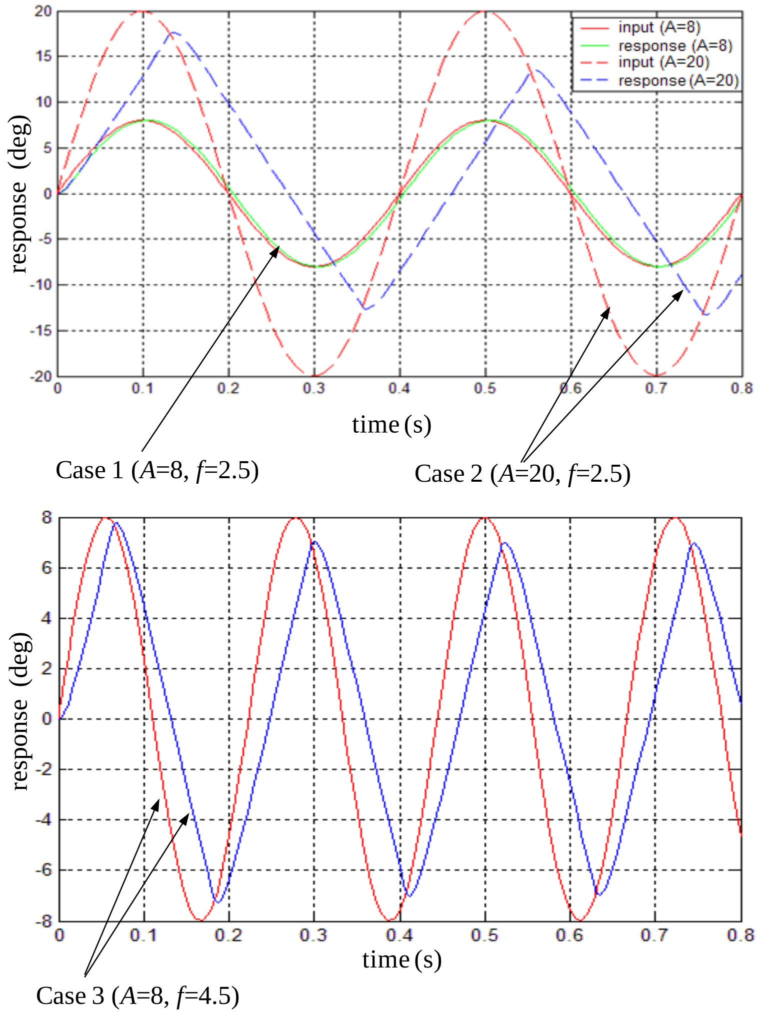


Fig. 6.5 Tracking performances for different gait inputs: tracking response by changing input amplitude (case 2) and influence from change of frequency (case 3).

6.2 Design of Robust Tracking Controller

Since the problem is caused by the unknown input and the inaccurate evaluation of the damping coefficient, a controller able to eliminate those factors can be developed by modeling them in the controller design. This controller should be able to compensate the change of the gait input. It also can provide robustness for the system to reject the negative effect caused by the inaccuracy of parameters. To achieve this aim, two principles should be noted in the design: 1) the control system should know the change of the gait input. That is to say, the controller will provide the system capability of predicting the input in some extent. The model of the gait input can be incorporate into the controller design to obtain such prediction. 2) The information of the water reactive force should be incorporated into the controller design, which requires more accurate modeling of the force. The design of such a controller is discussed in this section.

6.2.1 Modeling of the Disturbance

The value of the reactive force can be obtained in experiments. As the fish tail oscillates, F_R can be calculated from readings of force sensor. One experiment result is plotted in Fig. 6.6. The data was obtained under gait input frequency f= 2.5Hz and A=8°. To eliminate noises, the output of the force sensor was filtered through a fourth order low pass Butterworth filter. It can be seen that the reactive force is also periodical and it shares the same frequency with the gait control signal. Due to the spring effect of the caudal fin mechanism, the phase of F_R may not be the same as θ. The motor torque and the reactive force have effect on the tail mechanism together. The latter can be looked as a disturbance to the system. According to Fig. 6.6, this disturbance has the similar form as the gait control input. Since the frequency of the gait control signal is known, a sinusoidal function is used to describe it, which is given by:

$$T_R \propto F_R = A_R \sin(2\pi ft + \varphi_R) \tag{6.11}$$

where A_R is the amplitude of reactive force, and φ_R is the phase difference between the gait input and the water reactive force. The frequency f is the one of the gait control signal.

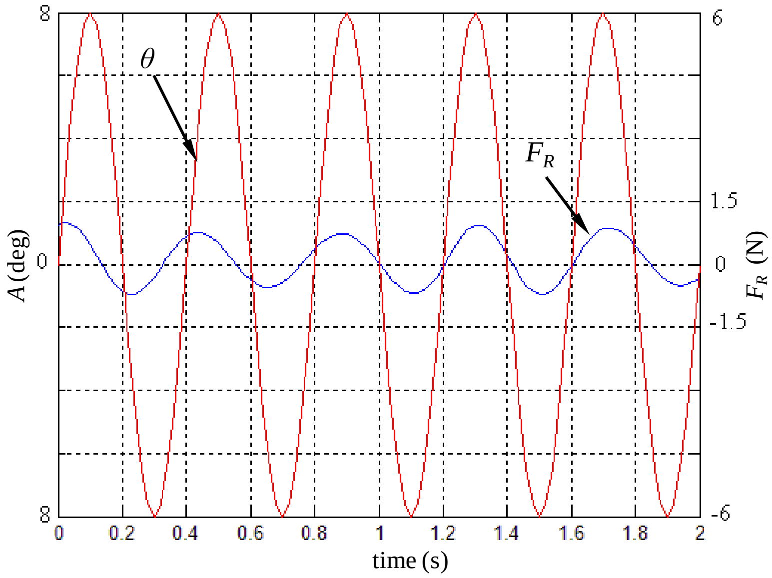


Fig. 6.6 Gait control signal and the water reactive force acting on the caudal fin.

6.2.2 Design of a Robust Gait Tracking Controller

The purpose of the measurement shown in Fig. 6.6 is only to verify that the reactive force can also be modeled as a periodical signal like the gait control signal. The amplitude and the phase lag are not necessarily known variables. Actually, in the following deriving of the robust controller, it can be seen that it is not necessary to know the value of the amplitude and the phase lag. The robust controller can handle the effect due to these unknowns.

(a) Formulation of controller

The approach based on error space analysis and introduced in [147] will be employed to design the robust tracking system for NAF-II. Usually, a feedback controller can be designed in the system state space. In the controller design for a tracking problem, a new state space, which is called an error space and derived from original space, can be fabricated. The tracking error in the original space will be the output of the new space. A controller can be derived in the error space to achieve zero output. Then the controller can be converted back into the original space, which will reduce the tracking error. The following derivation is performed based on this idea.

In the state space model, the external disturbance, T_R, is included. According to Eq. (6.10), the state space model of the tail system can be given by

$$\dot{\Theta} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -1/\tau \end{bmatrix} \cdot \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ K_o \end{bmatrix} u + \mathbf{B'} T_R = \mathbf{A}\Theta + \mathbf{B}u + \mathbf{B'} T_R \tag{6.12}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \mathbf{C}\Theta \tag{6.13}$$

where $\mathbf{\Theta}$ is the system state vector, **A** is the state matrix, **B** is the input matrix, u is system input, **B’** is the disturbance matrix, y is the system output, and **C** is the output matrix. Here, the disturbance is incorporated into the control system as part of the system and the gait control problem is solved in an error space. To build the error space, firstly the tracking error is defined by

$$e = \theta_d - y \text{ i.e. } \theta_d = e + y \tag{6.14}$$

By differentiating both sides of Eq. (6.14) two times, we obtain

$$\ddot{\theta}_d = \ddot{e} + \ddot{y} \tag{6.15}$$

The reference gait control input satisfy the following differential equation:

$$\ddot{\theta}_d + \omega^2\theta_d = 0 \tag{6.16}$$

where $\omega=2\pi f$. Substituting Eqs. (6.14) and (6.15) into (6.16), we obtain:

$$(\ddot{e}+\ddot{y})+\omega^2(e+y)=0 \tag{6.17}$$

By rearranging Eq. (6.17), explicit input θ_d can be eliminated from those equations and the following relationship can be obtained:

$$\ddot{e}+\omega^2 e=-(\ddot{y}+\omega^2 y)=-\mathbf{C}(\ddot{\Theta}+\omega^2\Theta) \tag{6.18}$$

For further derivation, new state γ is introduced. It is defined as

$$\boldsymbol{\gamma}=\ddot{\Theta}+\omega^2\Theta=\begin{bmatrix}\gamma_1\\ \gamma_2\end{bmatrix} \tag{6.19}$$

With this definition, the Eq. (6.18) can be replaced by

$$\ddot{e}=-\mathbf{C}\boldsymbol{\gamma}-\omega^2 e=-\boldsymbol{\gamma}_1-\omega^2 e \tag{6.20}$$

The state equation of γ can then be derived as

$$\begin{aligned}\dot{\gamma}&=\dddot{\Theta}+\omega^2\dot{\Theta}\\&=\mathbf{A}\ddot{\Theta}+\mathbf{B}\ddot{u}+\mathbf{B'}\ddot{T}_R+\omega^2(\mathbf{A}\Theta+\mathbf{B}u+\mathbf{B'}T_R)\\&=\mathbf{A}(\ddot{\Theta}+\omega^2\Theta)+\mathbf{B}(\ddot{u}+\omega^2 u)+\mathbf{B'}(\ddot{T}_R+\omega^2 T_R)\\&=\mathbf{A}\boldsymbol{\gamma}+\mathbf{B}\mu+\mathbf{B'}(\ddot{T}_R+\omega^2 T_R)\end{aligned} \tag{6.21}$$

where a new definition is introduced:

$$\mu=\ddot{u}+\omega^2 u \tag{6.22}$$

Considering Eq. (6.11), the disturbance satisfies the same differential equation as the gait control signal, which is

$$\ddot{T}_R+\omega^2 T_R=0 \tag{6.23}$$

Substitute (6.22) into (6.21) and obtain

$$\dot{\gamma}=\mathbf{A}\gamma+\mathbf{B}\mu=\begin{bmatrix}\gamma_2\\ -\gamma_2/\tau\end{bmatrix}+\begin{bmatrix}0\\ K_o\end{bmatrix}\mu \tag{6.24}$$

It can be noticed that in the state Eq. (6.24), the reference input signal θ_d and the disturbance T_R become implicit. Eqs. (6.20)-(6.24) show information about the relationship between tracking error e and the newly introduced states. Then the problem of tracking reference gait input θ_d and rejecting the disturbance T_R in original system can be converted into a new problem where the tracking error e

becomes the output of this new state space system. A suitable control law can be designed in the error space. And the new control law regulates this error space system in the order that the error tends to zero as time gets large. If the states of the new error space system can be defined as:

$$\varepsilon = \begin{bmatrix} e & \dot{e} & \gamma_1 & \gamma_2 \end{bmatrix}^T \tag{6.25}$$

The error space can then be defined by

$$\dot{\varepsilon} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega^2 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1/\tau \end{bmatrix} \cdot \begin{bmatrix} e \\ \dot{e} \\ \gamma_1 \\ \gamma_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ K_o \end{bmatrix} \mu \tag{6.26}$$

A full order state feedback controller is applied to the error space system. The control law is given by

$$\mu = 0 - \begin{bmatrix} k_4 & k_3 & k_2 & k_1 \end{bmatrix} \cdot \begin{bmatrix} e \\ \dot{e} \\ \gamma_1 \\ \gamma_2 \end{bmatrix} \tag{6.27}$$

where k_i (i=1, 2, 3, 4) is the feedback gains. In this error space, in order to get desired dynamics and zero tracking error, values of feedback gains can be obtained through pole placing method. Theoretically, when the system performance is defined by the desired closed-loop poles, a response with zero steady state error can be achieved by pole placing method [147]. The feedback gains in Eq. (6.27) can then be computed through Ackermann's formula [147].

The above design of control law is conducted for the error space. The expression of control law for the original gait control system in terms of the actual system state can then by derived. By combining Eqs. (6.24)-(6.27), we obtain:

$$\mu = \ddot{u} + \omega^2 u = -(k_4 e + k_3 \dot{e}) - \begin{bmatrix} k_2 & k_1 \end{bmatrix} \cdot (\ddot{\Theta} + \omega^2 \Theta) \tag{6.28}$$

Rearrange Eq. (6.28) and combine like terms:

$$(u + \mathbf{K}\Theta)^{(2)} + \omega^2 (u + \mathbf{K}\Theta) + (k_4 e + k_3 \dot{e}) = 0 \tag{6.29}$$

where $\mathbf{K} = [k_2, k_1]$.

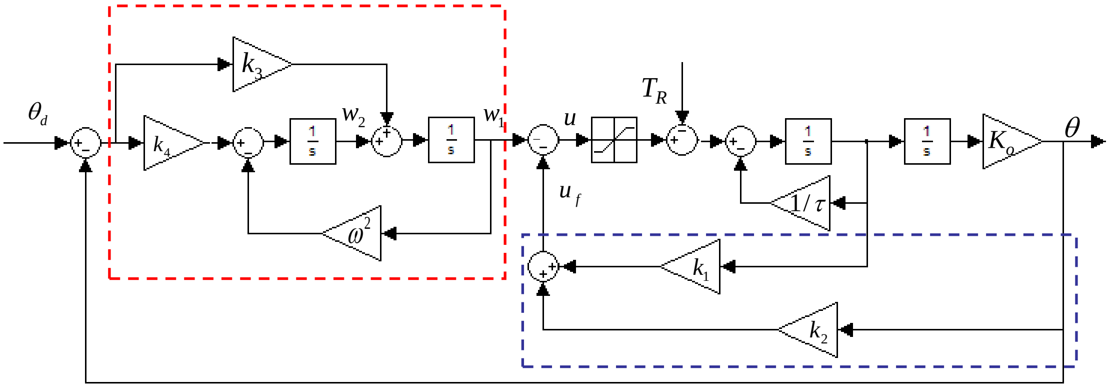


Fig. 6.7 Block diagram of the closed-loop system with a robust controller.

Eq. (6.29) defines the dynamics of the robust controller. It can be seen that that the system is made up of two parts. One part is the full order state feedback controller for the motor-linkage system and the other part is a feed forward regulator which can be given by the following transfer function between output of the controller x_1 and its input e:

$$\frac{w_1(s)}{e(s)} = -\frac{k_3 s + k_4}{s^2 + \omega^2} \tag{6.30}$$

Its state space interpretation is

$$\dot{\mathbf{W}} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \mathbf{W} + \begin{bmatrix} k_3 \\ k_4 \end{bmatrix} e \tag{6.31}$$

where $\mathbf{W}=(w_1, w_2)^T$ is the state of the controller. By using this controller, the overall feedback control system can be described by the block diagram depicted in Fig. 6.7.

It can be seen from Fig. 6.7 that the overall control system comprises three loops. The inner loop includes the full-order feedback control loop of the plant and the feed-forward controller loop. The outer loop is an output feedback loop. The controller is actually an oscillator which shares the same frequency as the input signal and therefore as that of the disturbance. This characteristic ensures the tracking control system can accurately perform the periodical gait control input under disturbance with the same frequency. According to Fig. 6.7, the transfer function of the full-order feedback controller is

$$\frac{u_f(s)}{\theta(s)} = k_2 + \frac{k_1}{K_o} s \tag{6.32}$$

According to Eq. (6.30) and (6.32), the control law of this robust tracking system can be given as

$$u(s) = -\frac{k_3 s + k_4}{s^2 + \omega^2} e(s) - (k_2 + \frac{k_1}{K_o} s)\theta(s) \tag{6.33}$$

The total open-loop system is:

$$G(s) = \frac{-k_3 K_0 (s + k_4 / k_3)}{(s^2 + \omega^2)[s^2 + (k_1 + 1/\tau)s + k_2 K]} \tag{6.34}$$

The transfer function of the total closed-loop system is:

$$\frac{\theta(s)}{\theta_d(s)} = \frac{-k_3 K_0 (s + \frac{k_4}{k_3})}{s^4 + (k_1 + \frac{1}{\tau})s^3 + (k_2 K_0 + \omega^2)s^2 + [\omega^2(k_1 + \frac{1}{\tau}) + k_3 K_0]s + K_0(\omega^2 k_2 + k_4)} \tag{6.35}$$

(b) Simulation of the Robust Tracking

The theoretical analysis shows that the robust control scheme can bring about a zero-error response to the periodical gait control input. It can be reflected by equations (6.21)-(6.23). In equation (6.21), the disturbance term $\mathbf{B}'(\ddot{T}_R + \omega^2 T_R)$ will be zero as the external disturbance T_R satisfied the second-order differential equation defined in Eq. (6.23). This indicates that the external disturbance is eliminated in the error space and it will not affect the tracking performance if the robust tracking controller is applied. In the error space analysis, the tracking error e in the original space becomes the system output. The feedback controller defined by Eq.(6.27) in the error space is designed for zero-input, which will lead to zero output if the error space system processes appropriate poles. This can be achieved by the well-known pole placement method.

To verify the design, a simulation is performed in Matlab™ Simulink. For the caudal fin prototype shown in Fig. 6.1a, the gait input signal with f=1Hz and A=20 °is tested first. The desired poles for the error space system are set to (-10±15j; -50±50j). By using Ackermann's formula, the gain vector can be obtained:

$$\begin{bmatrix} k_4 & k_3 & k_2 & k_1 \end{bmatrix} = \begin{bmatrix} -1323.9 & -126.47 & 7.2121 & -0.0079 \end{bmatrix} \tag{6.36}$$

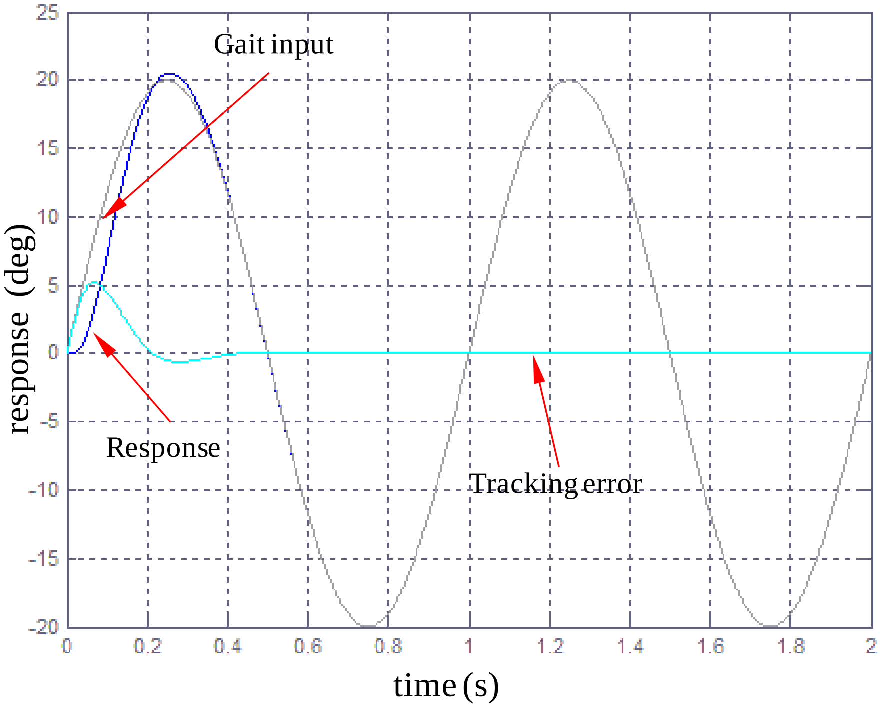


Fig. 6.8 Response to gait control input with f=1Hz and A=20 º

The closed-loop transfer function is:

$$\frac{\theta(s)}{\theta_d(s)} = \frac{1.278\times10^5(s+10.46)}{s^4+154.2s^3+7325s^2+1.339\times10^5 s+1.625\times10^6} \tag{6.37}$$

In the simulation shown by Fig. 6.8, the disturbance described by Eq. (6.11) is added into the system. This disturbance has amplitude as strong as 20% of the motor's nominal torque and a random phase lag φ_R. In such an extreme situation, the system can still tracking the input with almost zero errors after a short setting time (see Fig. 6.8). As the theoretical analysis implies, the tracking error converges to zero over time. More simulations can be performed for different gait control inputs. Fig. 6.9 shows another example where f=2.5Hz and A=12 º. The tracking system will show the similar performance for different gait control inputs.

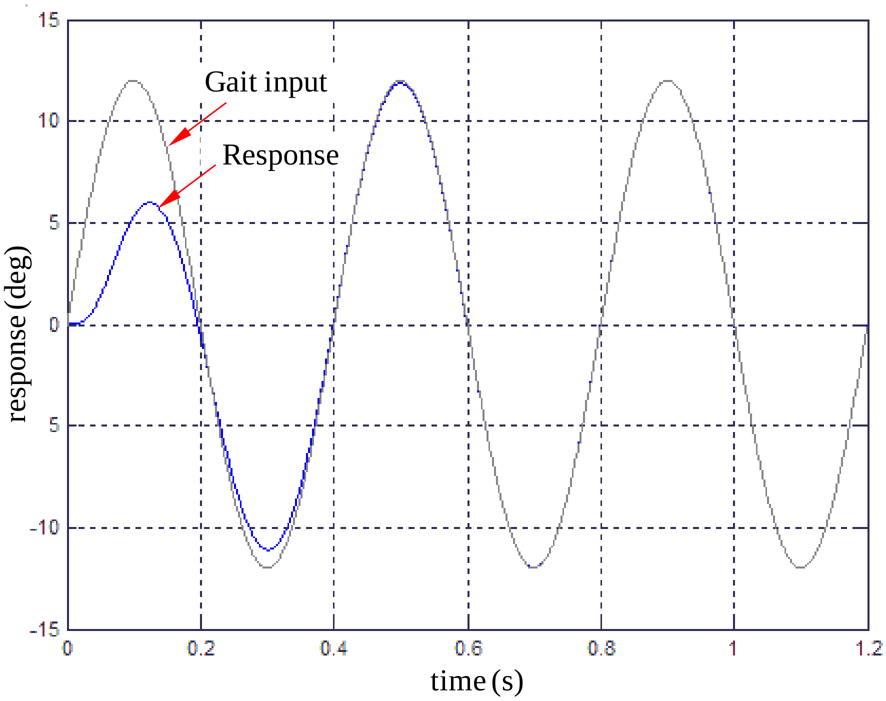


Fig. 6.9 Response to gait control input with f=2.5 Hz and A=12 º.

6.3 Experiments of Gait Tracking Control

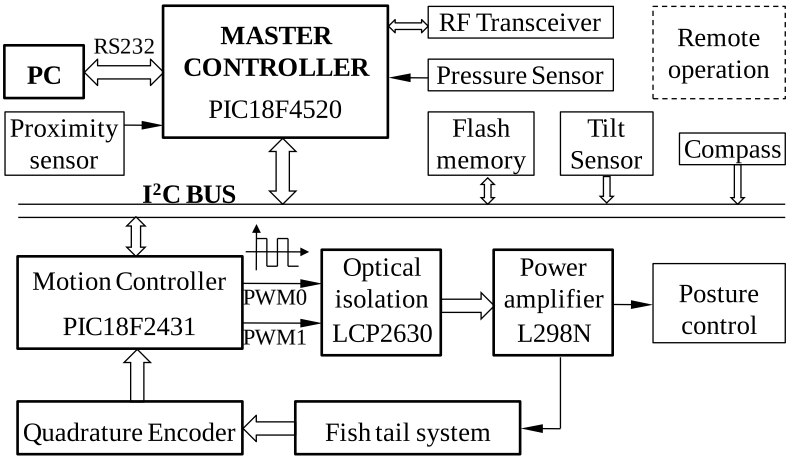
6.3.1 Control Hardware

A set of experiments has been conducted on NAF-II in a lab water tank to test the PD controller and the robust tracking controller. The electronic control of the fish prototype is implemented based on a microcontroller based real-time hardware. The main tasks of the control hardware should perform include: (i) processing the signals from various sensors such as the tilt sensor, compass, the pressure sensor, etc; (ii) communicating with host computer and remote controller; and (iii) implementing real-time motion control algorithms to drive the actuators.

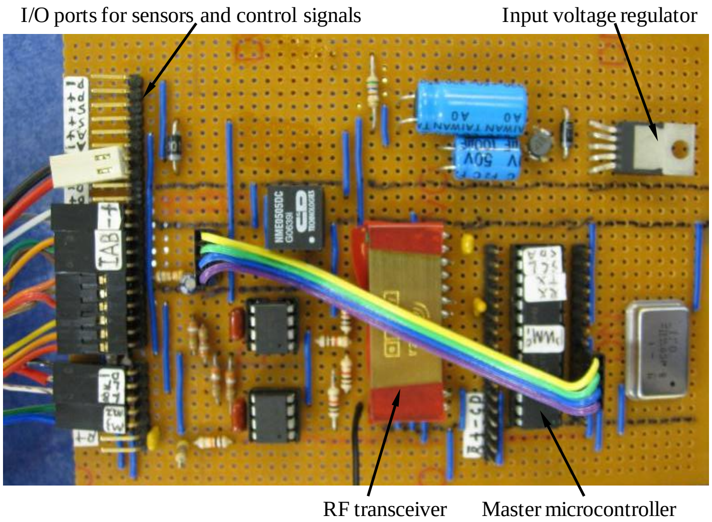
Master-slave control hardware is designed to perform these functions, as depicted in Fig. 6.10. A 8-bit micro-controller, PIC18F4520, is used as the master controller. It reads the sensor outputs and manages various tasks including sending commands, communication and task assignment. The communication with host PC is through the wireless university serial asynchronous transmission. The internal communication is based on the I^2C bus. The motion control is taken by a PIC18F2431 micro-controller.

This controller runs at 40MHz, which is fast enough to perform the calculation of control algorithms.

The controller generates complimentary 20KHz PWM signals to achieve a bi-directional motor control. The PWM signals are isolated from the power amplifier by an optical coupler, which enhances the stability of the electronic control system. The local controller PIC18F2431 performs the control algorithms and records the actual position data of tail fin. Then those data are transmitted through a RF transceiver to the host PC for further analysis.



(a)



(b)

Fig. 6.10 Control hardware of NAF-II: (a) a master-slave control hierarchy and (b) master controller circuit board.

Upon implementation of the control algorithms, the sample rate is set to 256Hz. This sample rate is fast enough to minimize the difference between the continuous and digital controller design for a DC motor driven mechanical tail system. This sample rate is chosen as the 2's integral power to make use of the shift operation in calculation, which is much faster than the normal multiplication. This technique helps to maintain the real-time property of the control system. With the sample rate, the servo cycle is approximately 4 ms. The selection of the sample rate must guarantee that the control algorithm can be fully performed in one servo cycle. The robust control algorithms are performed in PIC18F2431. The 40MHz internal CPU speed of this microcontroller ensures the computation can be finished in one cycle. The I^2C bus is used to transfer simple motion command such as start, stop, turn, and so on from master controller the motion controller. It does not involve in the computation of control algorithms. Therefore, the real-time property of the motion control will not be breached by the slow data rate on the I^2C bus.

6.3.2 Experimental Results

The controller design for tail system in previous sections is conducted in continuous-time domain. In order to implement the control algorithms in the digital control hardware, the continuous-time controller needs to be discretized. The Tustin (bilinear) transform is used to convert the controller into the discrete-time domain, which is given by:

$$s = 2 f_s \frac{z-1}{z+1} \tag{6.38}$$

where f_s is the sample rate and z is an operator in discrete-time domain. Fig. 6.11 shows the responses to the gait control function with a PD controller. The testing gait is a sine signal with f=2.5Hz and A=12°. Output 1 in Fig. 6.11 is the experiment result with theoretical controller parameters (K_p = 76.7 and K_d =0.302). It can be seen that there is obvious discrepancy between the input and the output. Significant phase lag and tracking error appears. This discrepancy may come from the imperfect modeling of the tail system.

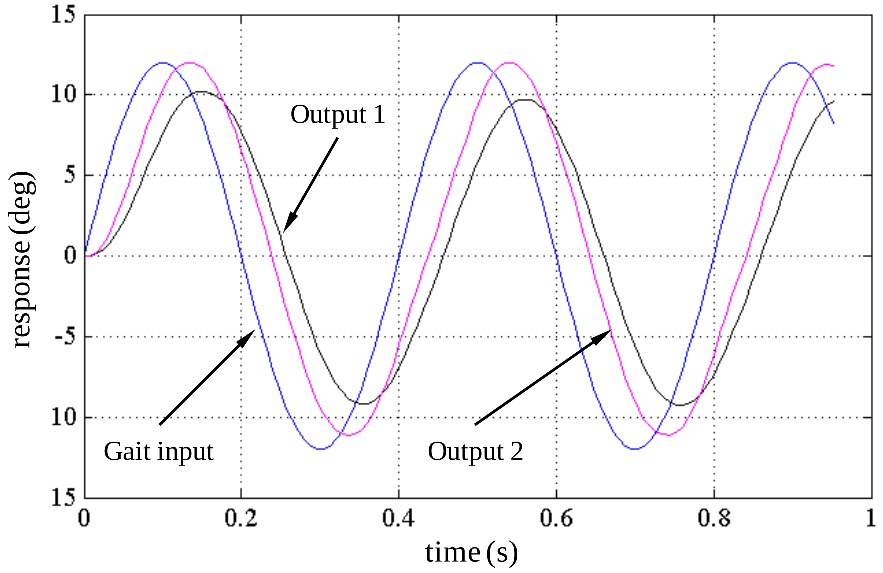


Fig. 6.11 Experiment results of PD controller with theoretical parameters (output 1) and manually adjusted parameters (output 2).

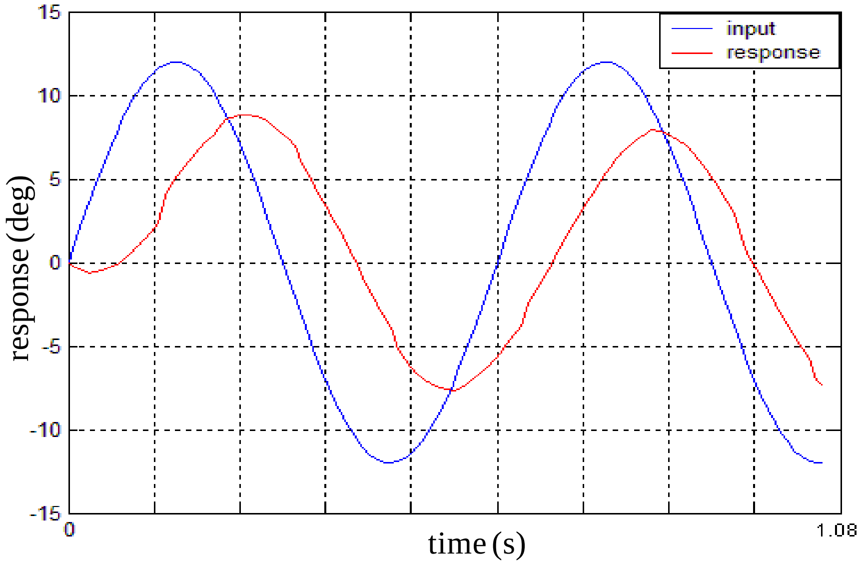


Fig. 6.12 Experiment results of PD control (K_p = 47.5, K_d =0.73, f=1.75Hz, and A=12°).

The PD controller parameters are adjusted manually upon experiments. The value of K_d is increased to reduce the phase lag and K_p is decreased to restrain large overshoots. The actual values are set to K_p = 47.5 and K_d =0.73. Accordingly, the pole and the DC gain of the fore-filter is set to zp=65.1. The response is shown by output 2 in Fig. 6.11. It can be seen that the error becomes smaller but there still is a phase lag.

In on-line locomotion control of fish robot, the frequency and amplitude of the swimming gait will not be constants. The PD controller has to be redesigned because in this case, the tracking control system may not be robust enough to handle different control signals. This can be illustrated by the experiment shown in Fig. 6.12. The PD controller designed for output 2 in Fig. 6.11 is used to tracking the gait input with *f*=1.75Hz and *A*=12°. The tracking performance becomes worse in this case as the significant difference between the input and output appears.

This problem can be avoided by using the robust tracking design. By using the Tustin transform, the discrete-time version of the robust tracking control algorithms in Eq. (6.33) can be given by the following linear difference equation:

$$u(kT) = -\sum_{j=1}^{3} a_j u(kT - jT) + \sum_{j=0}^{3} \left[b_j e(kT - jT) + c_j y(kT - jT) \right] \tag{6.39}$$

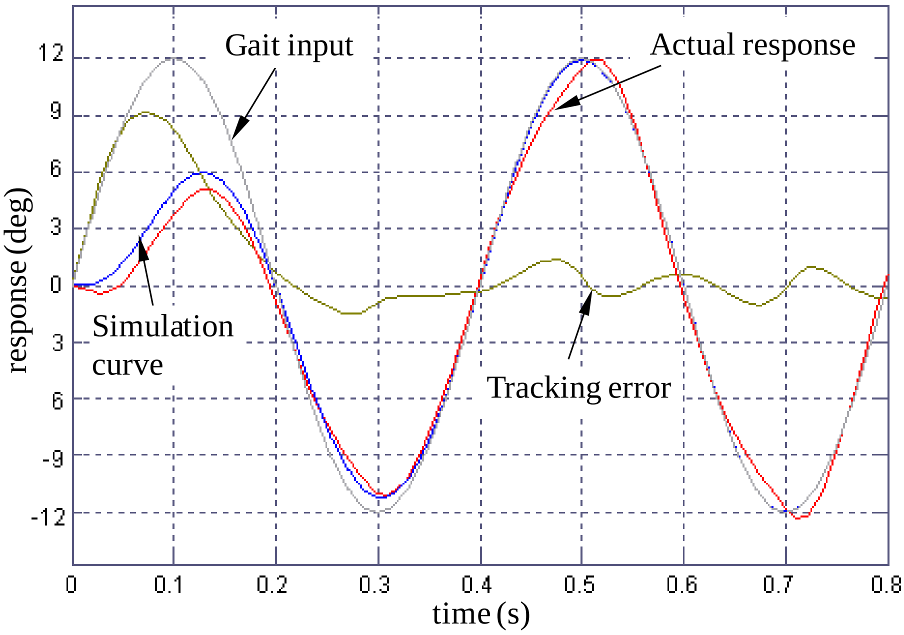
where *k*=1, 2, 3, … and the coefficients are listed in Table 6.2.

Table 6.2 Expressions of coefficients in Eq. (6.39).

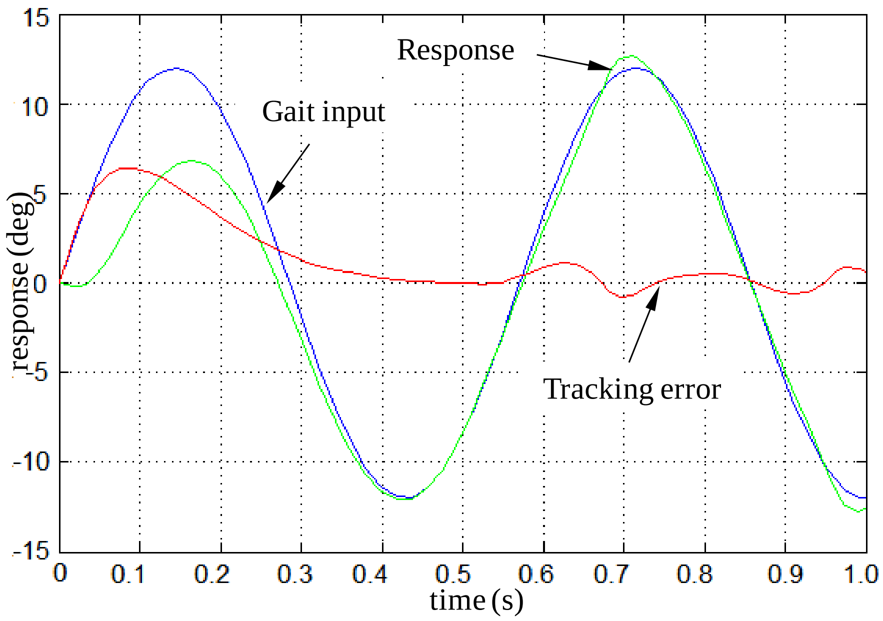
No.	*j*=0	*j*=1	*j*=2	*j*=3
a_j	0	$\frac{3\omega^2 + 4f_s^2}{\omega^2 + 4f_s^2}$	$\frac{3\omega^2 + 4f_s^2}{\omega^2 + 4f_s^2}$	1
b_j	$-\frac{k_4 + 2f_s k_3}{\omega^2 + 4f_s^2}$	$-\frac{3k_4 + 2f_s k_3}{\omega^2 + 4f_s^2}$	$\frac{2f_s k_3 - 3k_4}{\omega^2 + 4f_s^2}$	$\frac{2f_s k_3 - k_4}{\omega^2 + 4f_s^2}$
c_j	$k_2 + 2k_1 / K_o$	$k_2 - 2k_1 / K_o$	$\frac{(\omega^2 - 4f_s^2)(k_2 + 2k_1 / K_o)}{\omega^2 + 4f_s^2}$	$\frac{\omega^2(k_2 - 2k_1 / K_o)}{\omega^2 + 4f_s^2}$

Fig. 6.13 shows the experiment results for gait input with *f*=2.5Hz and *A*=12°. The desired input, the simulation result, the actual response from experiments and the actual error curve are shown. This gait input is selected for testing because it is the most critical control input that needs to be tracking for NAF-II (the fish cannot track inputs with higher frequency or amplitude because of the power limitation of the motor). In experiments that frequency is less than 2.5Hz, smaller tracking errors and shorter setting time are achieved (see Fig. 6.13). Although the ideal zero-error

response is not obtained, which differs from what the theoretical analysis predicts, the tracking error is reduced significantly by this robust control after a half period of setting.



(a)



(b)

Fig. 6.13 Experiment results of the robust tracking: (a) A=12°, f=2.5 Hz and (b) A=12°, f =1.75 Hz. The controller parameters are k_1=0.0081, k_2=5.035, k_3=-81.92, k_4=150.6, and f_s=256 Hz.

Because the input structure is modeled into the controller (the inner oscillator loop shown in Fig. 6.7), the controller can compensate the negative effects caused by the change of input and varying external disturbance. Therefore, the robustness of the whole tail motion control system is enhanced. Additionally, an analytical solution to the robust tracking problem is given, which makes it easier to program the control algorithms dynamically. The controller parameters of the algorithm can be automatically changed by analytical equations according to the variations of the gait input. It is not necessary to try different parameters.

6.4 Summary

The robust tracking discussed in this chapter is in the motor control level and it does not affect the locomotion performance of a fish robot. The purpose of designing such a controller is to guarantee that an actuator can exactly follow the command signal. Experiments show that the robust tracking design can effectively help the robotic fish system to perform swimming gait with small tracking error. Because the input structure is modeled into the controller, the controller can compensate the negative effects brought by the change of input and varying external disturbance caused by this change. Therefore, the robustness of the whole tail motion control system is enhanced.

Additionally, the chapter illustrates an analytical solution to the robust tracking problem, which makes it possible for dynamic programming of the control algorithms. The controller parameters can be automatically generated by analytical equations according to the change of gait control signals. Note that in this chapter the gait control signals are limited within the periodical form, which is associated with the straight-line swimming of the fish robot. The controller is not suitable for tracking some transient swimming gaits like those in burst. And the water is assumed to be still. Flow situation can also influence the tracking performance. The robust tracking in those conditions can be topics in our future studies.

Chapter 7 On-line Optimization of Swimming Locomotion

The swimming performances, especially the speed and efficiency, are always main concerns when it comes to the design of a robotic fish system. In this chapter, the application of CPGs is further investigated in swimming control of multi-DOf fish robots in order to achieve high performance locomotion. The swimming gaits generated by CPGs will ensure that fish robots can swim, but will not guarantee the good performance swimming motions. The reason is that behaviors of CPGs are completely governed by the self-sustained dynamics of nonlinear oscillators. The CPG model does not know any information about actual states of a robot. This can be improved by incorporating sensory feedback from robot-water interaction into the CPG model. In this way, a closed-loop control scheme will be applied to the locomotion control of fish robots. Together with the optimization techniques, the optimal locomotion of fish robots can be achieved. The Genetic Algorithm (GA) in optimization of swimming locomotion is explored together with CPGs in generation and coordination of swimming gaits for multi-DOF fish robots. Factors of many different aspects will influence the swimming performance. In this chapter, the finding of optimal body curves which are associated with optimal swimming performance is the focus [176].

The chapter is structured as follows. Firstly, a closed loop control scheme combining genetic algorithms and CPG model will be discussed in Section 7.1. In Section 7.2, GA will be adopted to find the optimal control parameters associated to the fast locomotion. On-line measurement of swimming speed is utilized to derive the fitness in GA. Similar method will be applied to find the cost-efficient locomotion in Section 7.3. The measurement of power and criteria for evaluation of swimming efficiency will be discussed. The experimental results on the propulsor prototype will be presented in Section 7.4. In Section 7.5, the optimization approach will be applied to find the optimal locomotion patterns for other two different propulsor prototypes. The results and findings in these experiments will be presented.

7.1 Gait Generation Based on CPGs and Genetic Algorithm (GA)

CPGs will be utilized to generate coordinated swimming gaits and the GA will be applied to search the optimal control inputs to the CPG model. Two separate objectives will be achieved: optimization of swimming speed and on-line optimization of swimming energy efficiency. A closed-loop control scheme is necessary to perform the on-line optimization, as depicted by Fig. 7.1. The eight-DOF biomimetic propulsor shown in Fig. 3.8 will be used to test this optimization method.

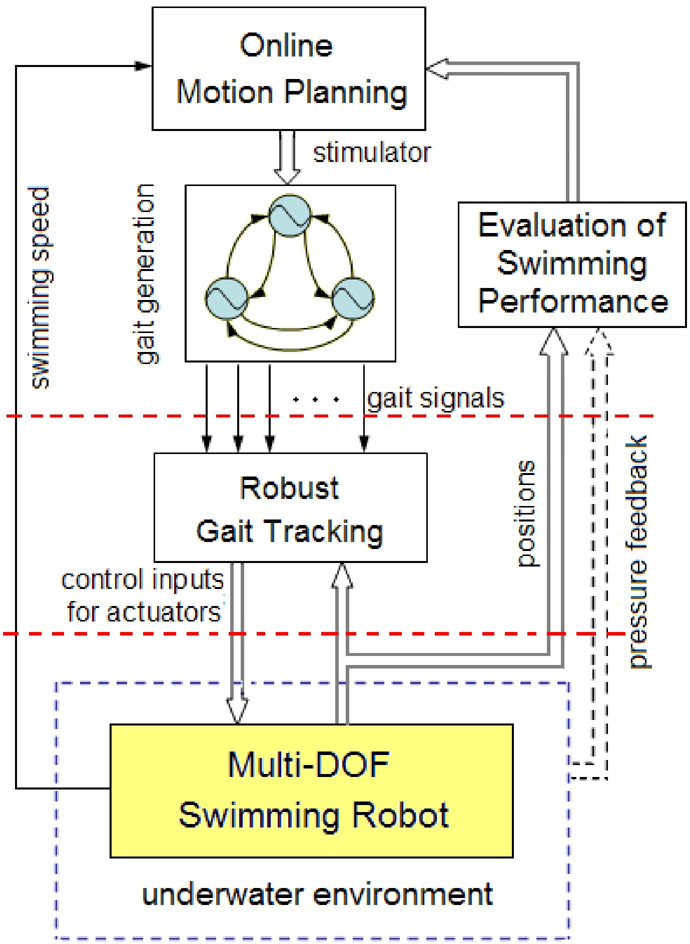


Fig. 7.1 An on-line swimming control scheme. The on-line motion planning is achived by GA. The robust gait tracking is applied to perform high performance low level actuaotor control, which has been detailed in Chapter 6 and [30].

7.1.1 Problem Definition

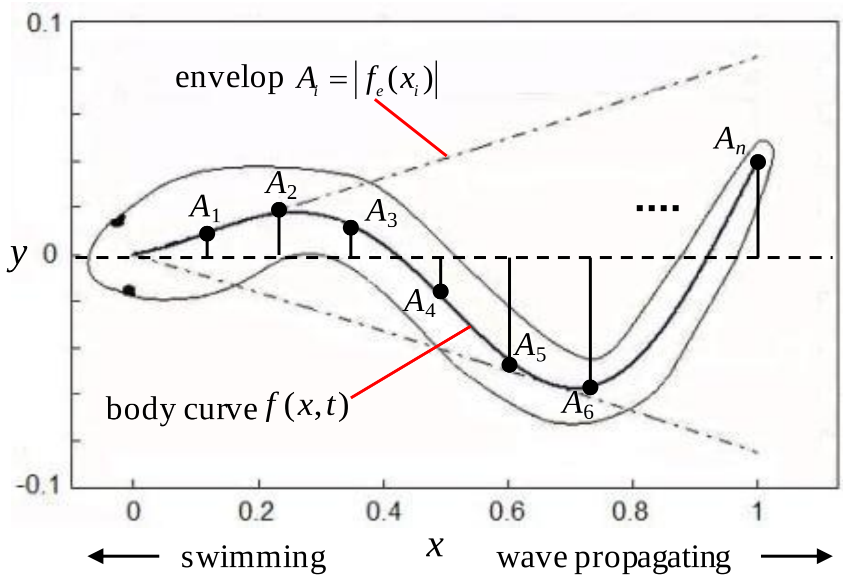


Fig. 7.2 Illustration of fish body curve and envelop during forward swimming (top view). The dark solid curve represents the midline (or vertebrate) of an undulatory swimmer. The dimensions are normalized to the body length.

There are optimal swimming fin curvatures leading to the most energy efficient swimming or fastest swimming in the undulatory locomotion [97]. A desirable control strategy should ensure all system elements of a propulsor involved in the thrust generation effectively [55, 56, 177]. First of all, it is necessary to identify what kind of factors will affect swimming performance. In a systematic perspective, many aspects should be taken into account such as the streamline of the body shape, the skin texture, and the locomotion pattern, etc. In this chapter, the focus is placed only on the control of swimming patterns as the physical properties are already given (the size, structure, material, and power transmission mechanism, etc.).

Based on Lighthill's model discussed in Chapter 3, the swimming performance is related to the envelop of the body motion and the oscillating frequency of body segments. In Chapter 3, the empirical testing with given controlling parameters is conducted to find swimming performance and a constant envelop is assumed. Such an envelop may not lead to energy-efficient swimming. In the present chapter, the controlling parameters will be explored in on-line swimming.

The envelop of the BMF is a key factor that affects the swimming performance. Lighthill proposed a parabolic envelop to explain the swimming kinematics of real fish. This assumption has also been adopted in robotics to model the kinematics of biomimetic propulsor. Many works are devoted to find the optimal values of parameters in the parabola equation [16, 178]. One question arises: will the optimal swimming pattern of a man-made propulsor share the same form as that of a live fish? In the on-line optimization of the present paper, the envelop is assumed to have an arbitrary form. Finding the form would be a goal of the proposed approach.

Biomimetic propulsors normally consist of a limited number of segments such as linkages or other types of mechanical components. These segments oscillate coordinately and collectively to fit the body curve during swimming. Achievement of swimming locomotion depends on oscillations of all segments, which is associated to two key parameters: the amplitude A and the frequency f (see Fig. 7.2). Therefore, the parameter set (S) to be solved in the on-line optimization will be

$$S = \{A_1, A_2, \dots, A_n;\ f\} \tag{7.1}$$

Note that the wavelength λ is not included in the parameter set because it is a dependent variable of amplitudes. The form of envelop can then be obtained by curve fitting as

$$f_e(x) \approx f_e(x_i)\big|_{i=1,2,\dots n} = \text{Function}(A_1, A_2, \dots, A_n) \tag{7.2}$$

7.1.2 Swimming Gait Generation by Using CPGs

The artificial CPG model discussed in previous chapters is adopted as the on-line swimming gait generator for the fish robot. The gait control parameters in CPGs (phase difference among propulsor segments, oscillation frequency, or amplitude) can be tuned in real-time according to the feedback from the environment. The swimming speed and the instantaneous power consumption will be evaluated and taken as inputs for GA to obtain the fitness of the candidate solution. Then improved system parameters will be produced and applied to CPGs until the optimal objectives are reached.

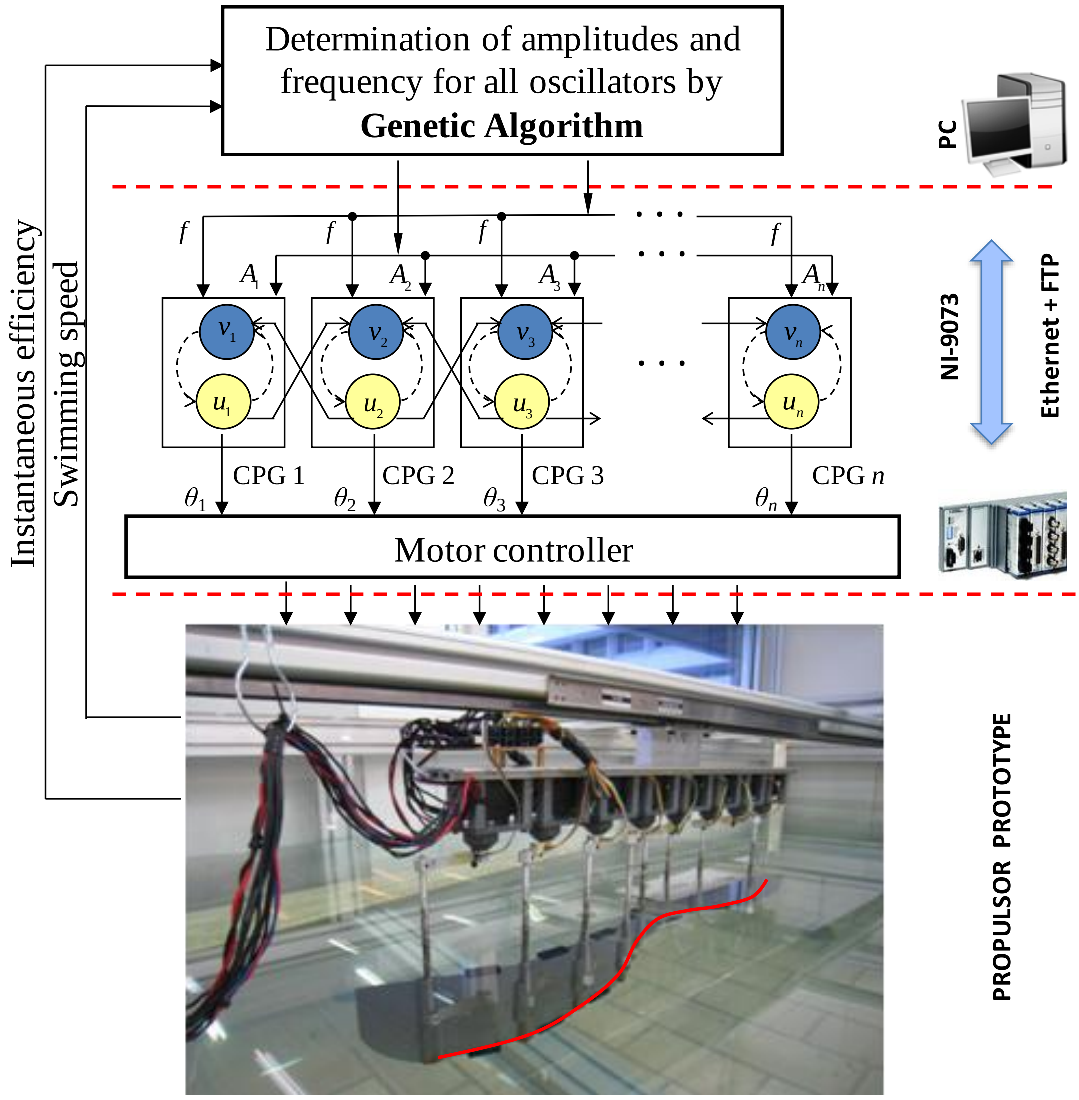


Fig. 7.3 GA-based closed loop control and configuration of oscillators (CPGs).

The structure of the closed-loop swimming control based on CPGs and GA is shown in Fig. 7.3. The swimming speed and the instantaneous power consumption will be measured by sensors. The evaluation of speed is simple and can be achieved by using readings of normal velocity sensors. Evaluation of power will be performed by taking the instant torque and angular velocities of driving cranks. The significance of this measurement will be discussed shortly. These two feedback signals help to evaluate the fitness of the associated gait control parameters.

All oscillators in the CPG model are coupled with the first oscillator. There are backward coupling between adjacent oscillators. This configuration is made because in the on-line optimization, the gait control signals keep changing, which requires the fast gait transition. The CPG model in this case can be given by:

$$\theta_i(t) = A_i u_i \tag{7.3}$$

$$\begin{bmatrix} \dot{u}_i \\ \dot{v}_i \end{bmatrix} = \begin{cases} F(\mathbf{X}_i) + \varepsilon \begin{bmatrix} 0 \\ v_2 \cos\varphi_d - u_2 \sin\varphi_d \end{bmatrix}, & i = 1 \\ F(\mathbf{X}_i) + \varepsilon \begin{bmatrix} 0 \\ u_1 \sin[(i-1)\varphi_d] + v_1 \cos[(i-1)\varphi_d] \\ +v_{i+1} \cos\varphi_d - u_{i+1} \sin\varphi_d \end{bmatrix}, & \text{otherwise} \end{cases} \tag{7.4}$$

$$\varphi_d = -\frac{2\pi m}{n} \tag{7.5}$$

7.1.3 Application of GA in Swimming Control

Genetic Algorithm (GA) will be used to tune parameters of CPGs according to on-line evaluation of swimming performance (see Fig. 7.3). GA is chosen as the optimization tool because of its good global searching capability and universality in many applications [178]. GA is also used in many other works such as [16, 108, 112] to find the optimal swimming patterns in swimming locomotion. The difference between this thesis and those works is that an on-line optimization is performed by combing the GA and CPG, i.e. the swimming gaits of fish robot keep improving as the robot swims. The robustness of the CPG based gait generation ensures that the motion is smooth and continuous when tuning control parameters.

GA is a stochastic global search method inspired by biological evolution in nature. It operates on a population of potential solutions which evolves generation by generation based on the principle of survival of the fittest. The operation of GA consists of several steps. Firstly, the system variables are randomly assigned to initial values in the solution domain. Those variables are encoded and processed by three procedures: the selection, the crossover, and the mutation, until the ending condition is satisfied. There are a set of algorithms and operations involved in GA. Those algorithms are adopted to initialize population, compute fitness of an individual, and perform selection, crossover and mutation operations. The flow chart of a commonly adopted GA is shown in Fig. 7.4.

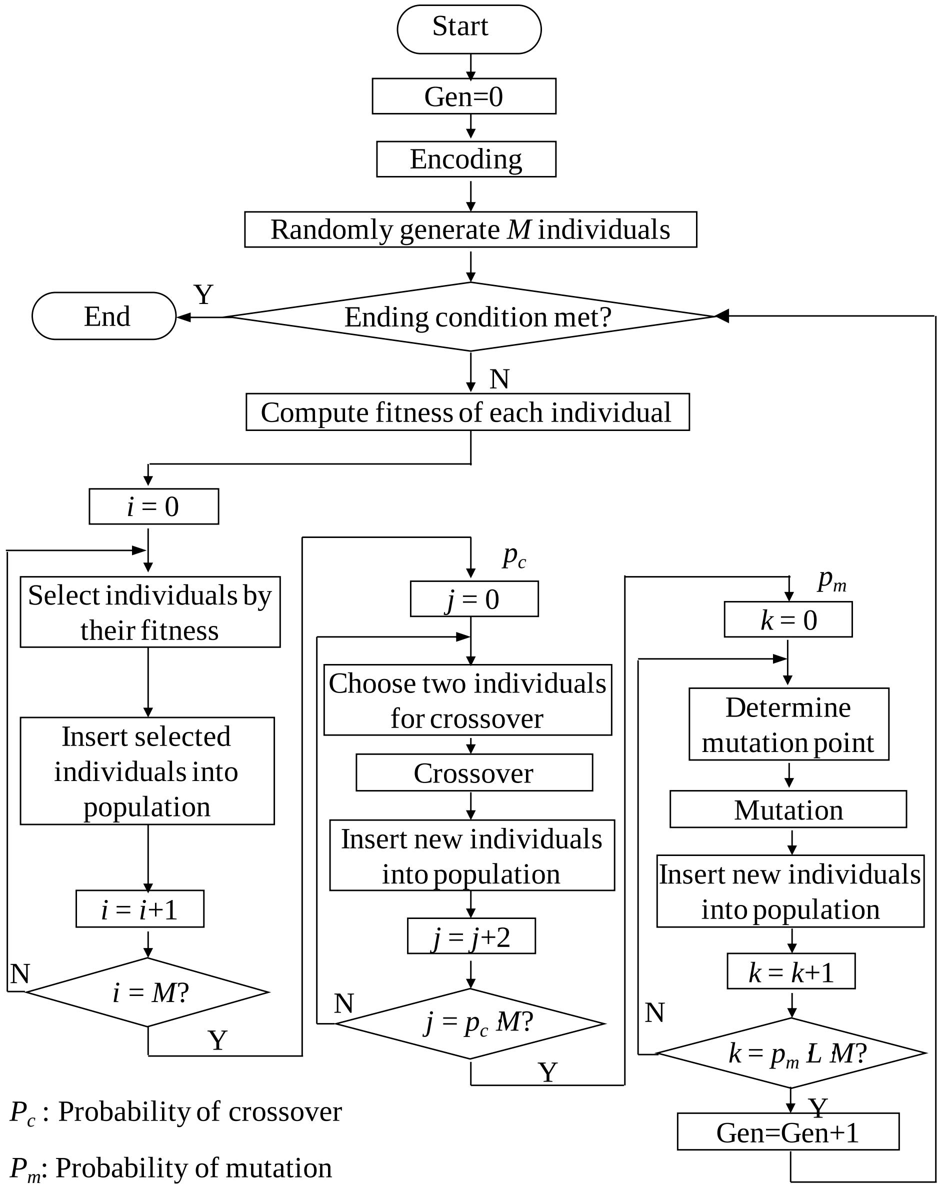


Fig. 7.4 Flow chart of Genetic Algorithm.

One solution of an optimization problem (a group of system variables) is regarded as an individual. Individuals are encoded into a population of strings, which are called chromosomes. A fitness function is introduced to evaluate the performance of each chromosome. A population will be processed by the selection, crossover and mutation operator based on the fitness of the individual in that population (see Fig. 7.4). New population is then produced used in the next iteration of the algorithm. The population is improved by repeating the same process, as illustrated by Fig. 7.1. Individuals evolve toward better solutions and those with higher fitness are more probably inherited to next generations.

7.2 Application of GA in Fast Swimming Mode

It is not likely that a fish robot reaches fastest swimming speed while consuming the least energy. Therefore, the optimal swimming of fish is either in fast swimming mode or in energy-efficient mode. The considerations in the implementation of GA in fast swimming model are discussed in this section.

7.2.1 Choosing of Fitness Function

In the optimization of speed, the robot system itself will act as the objective function in GA. The swimming speed collected from on-line measurement will be the objective of optimization. Then the swimming speed can be an indicator showing how well individuals have performed in the search domain. This is a raw measure of fitness. A fitness function is applied to transform the value of objective into a measure of relative fitness [179]. Here the individual fitness defined by the ratio between individual's raw performance (steady swimming speed U) and the whole population's performance is adopted, which is given by

$$FIT = \frac{U_i}{\sum_{i=1}^{m} U_i} \tag{7.6}$$

where FIT is the fitness function, U_i is the swimming speed, i.e. the raw performance of the i^{th} individual, and m is the population size.

7.2.2 Representation of Population

The eight brushless servo motors of the fish robot will be controlled by eight outputs of CPGs (see Fig. 7.3). In the CPG model, oscillators require totally nine parameters as the system inputs (eight amplitudes and a frequency). All oscillators share the same frequency in order to maintain the continuity of the body wave. Those parameters will be encoded into chromosome and processed by GA.

It is observed from real fish that the waveform of fish with elongated body shape shows a gradual increase in amplitudes from head to tail. This wave profile is proved efficient in producing of thrust [2]. This phenomenon implies that the amplitudes of all actuators are subjected to

$$0 \le A_1 \le A_2 \le \cdots \le A_n \le A_{\max} \tag{7.7}$$

where A_{max} is the maximum allowable amplitude of actuators. Usually, handling constraints will make the exercise of GA more complicated. In the present study, however, the constraint is preferable because it can decrease the dimension of the search space and hence significantly reduce the number of generations. Real-valued genes are employed in the GA instead of the commonly used single-level binary string. Real-valued encoding can increase the level of possible exploration in the search space without adversely affecting the convergence of GA [180]. In our applications, it also makes the handling of constraints easier.

7.2.3 Selection, Crossover and Mutation

Selection is a process of determining the number of offspring that an individual will produce. Stochastic universal sampling algorithm (SUSA) is adopted to perform the selection operation [181]. Unlike the selection operator, crossover can produce new chromosomes that have some parts of both parent's genetic material. Single-point discrete recombination algorithm is used as in crossover operation for real-valued chromosome [180]. The generation of new chromes is subjected to constraints given in Eq. (7.7). Normally fewer new individuals will be produced after selection and crossover than the size of the original population due to the generation gap. Those new individuals will be tested in the robot to obtain their fitness and reinserted into the old population to maintain the size of the population and replace the least fit members. In GA, mutation is used to randomly modify the elements in a chromosome, which is acts as a safety net to recover good elements that may be lost through the action of selection and crossover [181]. The mutation algorithm adopted from the Breeder GA [180] is adopted to process the real-valued mutation. The mutation may be benefit from the using of real-coded chromosomes for the higher mutation rates than binary-coded ones.

7.3 Application of GA in Efficient Swimming Mode

When GA is used to find a cost-efficient locomotion mode, the objective function, feedback of environment, and fitness function will be different from those in the fast swimming mode. The evaluation of swimming cost and implementation issues are discussed in this section.

7.3.1 Measurement of Power in Locomotion of Fish Robot

The swimming efficiency of a real fish is defined as the ratio of the actual power (contributed to the propulsion) to the total amount of the power consumed by fish. The total energy used during steady swimming is commonly referred to the Active Metabolic Rate (AMR) [182]. In the robotics area, there is a similar measure of swimming efficiency, which is defined by the ratio between useful power (output) and the total power input [108]. Hence, the useful power refers to what a fish robot obtains to move forward is given by

$$P_u = F_T U \tag{7.8}$$

Mechanically, the total power (P_t) comprises four parts: the useful power (P_u), the power loss in motion transmission from the actuators to the propulsor (P_l), the power used to drive the propulsor itself (P_p), and the power wasted in the wake (P_w). Accordingly, the efficiency can be defined by

$$\eta = \frac{P_u}{P_t} = \frac{P_u}{P_u + P_l + P_p + P_w} \tag{7.9}$$

There is no standard form to measure the swimming efficiency of fish. During the swimming, the propulsive wave traverses the fish body in a direction opposite to forward movement. The swimming speed, U, is smaller than the wave propagation speed, U_w. Therefore, the swimming efficiency of a real fish can be defined as the ratio U/U_w. This definition has long been used as an indication of swimming efficiency [8].

The information reflected by this efficiency is more on kinematics aspect. For an actual fish robot, the effective use of power will be more concerned since the robot could be expected to act as a useful underwater vehicle. Then the ratio between the thrust power and the effective input power could be a better measure of efficiency [108]. Here the effective input power comprises two parts: the thrust power and the power shed into the wake ($P_t = P_u + P_w$). This ratio is the so-called Froude efficiency index [40]. It has been used as a measure of energy efficiency in many off-line based optimization applications. In on-line motion planning, accurate value of thrust power can hardly be obtained because it is difficult to differentiate thrust and drag in a swimming body [40]. Usually, it is assumed that in steady swimming, the thrust is balanced by the drag and the latter is easier to be measured. The Froude efficiency can then be simplified as

$$\eta = \frac{P_u}{P_t} = \frac{P_u}{P_u + P_w} = \frac{F_T U}{F_T U + P_w} \tag{7.10}$$

According to Fig. 7.5, because the instantaneous reactive force on each driving crank can be obtained by force sensors, the real-time measurement of thrust could be possible, thus enabling the Froude efficiency available.

7.3.2 Evaluation of Swimming Efficiency: Economic Perspective

The power used in swimming will be measured to evaluate the energy efficiency. For a motorized bionic propulsor, the total power can be measured by the algebraic product of the input voltage and the electric current. The swimming performance can then be reflected by the ratio of thrust power over the total input power. Two parts of the total power will be wasted by the mechanisms: the power consumed in motion transmission from the actuators to the propulsor and the power used to drive the propulsor itself. In a rough estimation, these two parts can be assumed to have constant values or take constant proportions of the total input power. This assumption, however, is not accurate enough for on-line motion planning, because they are often influenced by factors, such as the quality of motors, imperfect mechanical design of fish robot, property of materials of the robot, etc. In the present research, the actual output power of motor in body-water interaction is measured.

Force sensors are employed to measure the actual reactive forces imposed on each driving crank by water. As shown by Fig. 7.5, the driving crank can be modeled as a cantilever. As the crack oscillates, the reactive force from water will cause the bending of the cantilever. Strain gauge based force sensors are used to measure the bending and thus obtain the torque on the crank. The total instantaneous power, $P(t)$, can then be obtained by

$$P(t) = \sum_{i=1}^{n} \left| T_{R,i}(t)\dot{\theta}_i(t) \right| \tag{7.11}$$

where $T_{R,i}(t)$ is the measured torque caused by reactive force on the i^{th} crank and $\dot{\theta}_i(t)$ is the angular velocity of the crank.

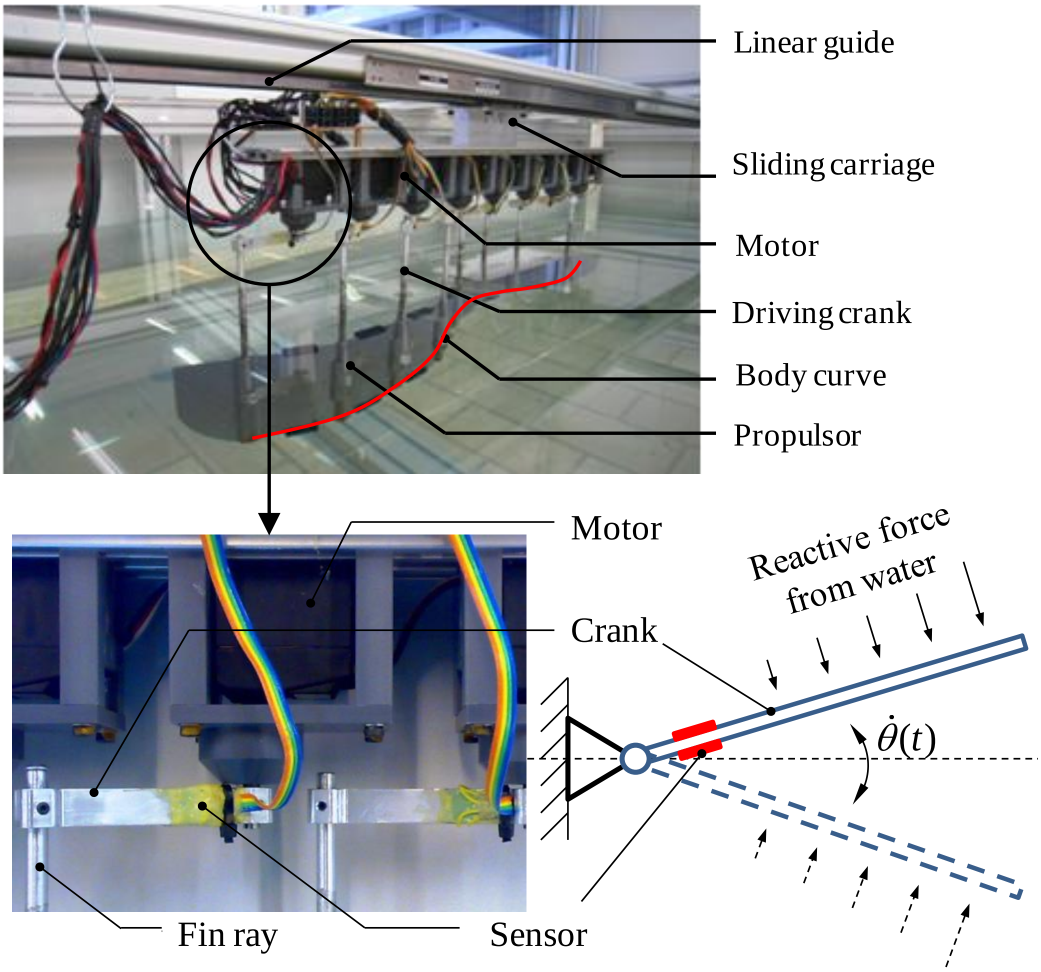


Fig. 7.5 Torque measurement at each crank.

As the sliding plates are made up of light materials, the energy loss due to the inertia force is insignificant. The instantaneous power for all cranks can be measured in real-time. Therefore, they can be used as feedback signal in closed-loop control [3, 5]. The specifications of force sensors can be found in Appendix C.

In the efficient swimming mode, it is always preferred achieving the fast swimming speed while consuming energy as little as possible. Finding such an economical way of swimming is of value for actual robots. In this point of view, it is natural to take the speed (in Body Length per seconds) per unit power as an indicator showing how well the input energy is converted into the kinetic energy of robot. This dimensional measure, I_E, can be given by

$$I_E = \frac{U/L}{\overline{P}} = \frac{U}{L\int_0^{1/f} f\sum_{i=1}^{n}\left|T_{r,i}(t)\dot{\theta}_i(t)\right|\mathrm{d}t} \tag{7.12}$$

In the efficient swimming mode, Eq. (7.12) will be the objective function of GA. The aim is to find the largest possible I_E in the locomotion. A similar fitness function as that in the fast swimming mode is adopted, which is given by

$$FIT = \frac{I_{E,k}}{\sum_{k=1}^{m} I_{E,k}} \tag{7.13}$$

7.4 Experimental Results

One motivation of the present work comes from the consideration that the optimal performance obtained by offline-based approaches might not be the best performance a fish robot might possess. This hypothesis has been verified by experiments. On-line locomotion optimization has been implemented on the fish robot. Experiment results will be presented in this section.

7.4.1 Experiment Setup

In experiments, the eight-DOF propulsor prototype performs straight-line swimming on the linear guide. The control parameters are updated by GA in every three periods of crank oscillations. In the first two periods, the robot accelerates until the steady speed is reached. This two-period time slot is also for evolving of CPG outputs. Then in the third period, the averaged velocity and averaged power over this period will be calculated based on data from eight force sensors and the velocity sensor. Once these values are available, the GA is performed to obtain the improved population. Afterwards, this new population will be applied to the CPG model and a new round of testing will start. The whole testing process can be explained by the flow chart in Fig. 7.6. The parameter set is

$$S = \{A_1, A_2, \ldots, A_8;\ f\} \tag{7.14}$$

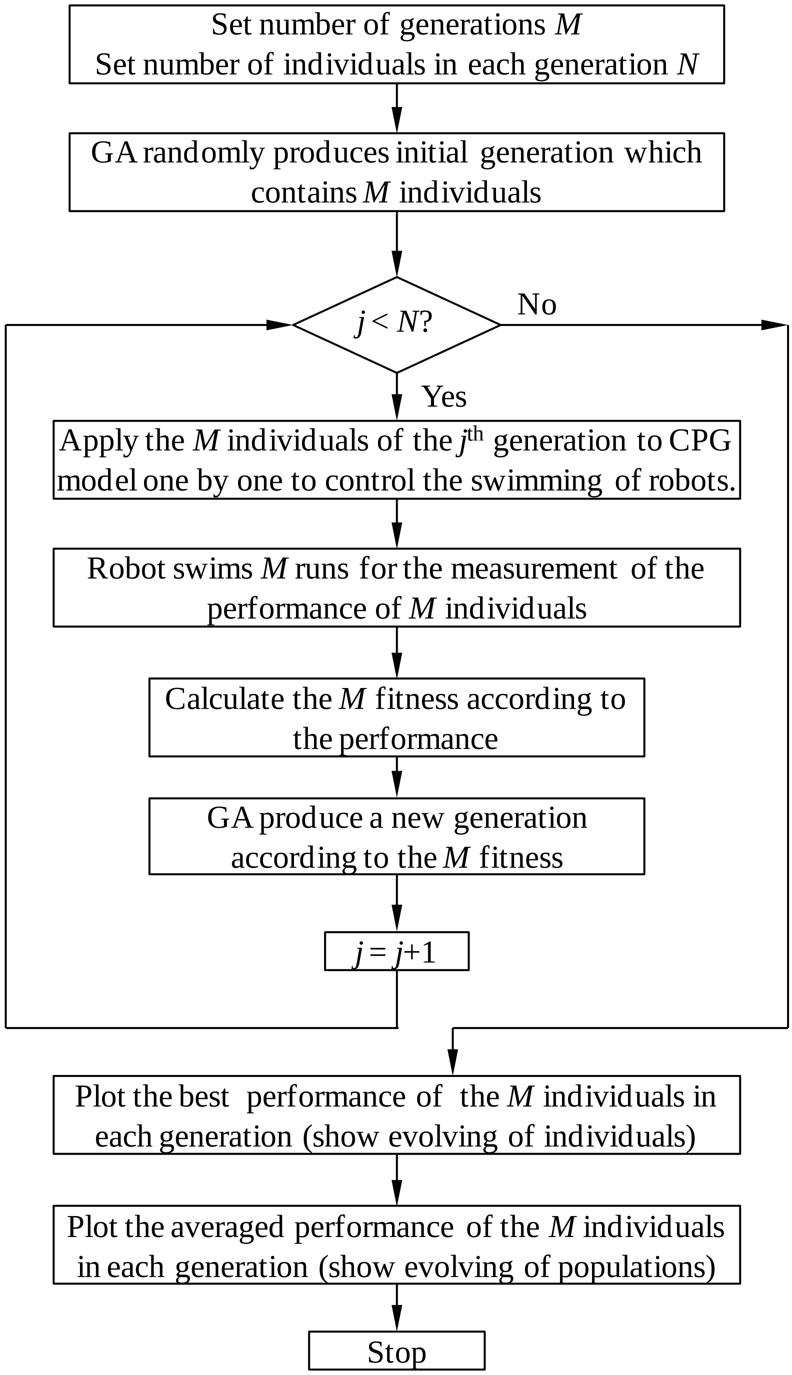


Fig. 7.6 Flow chart of the optimization process in experiments.

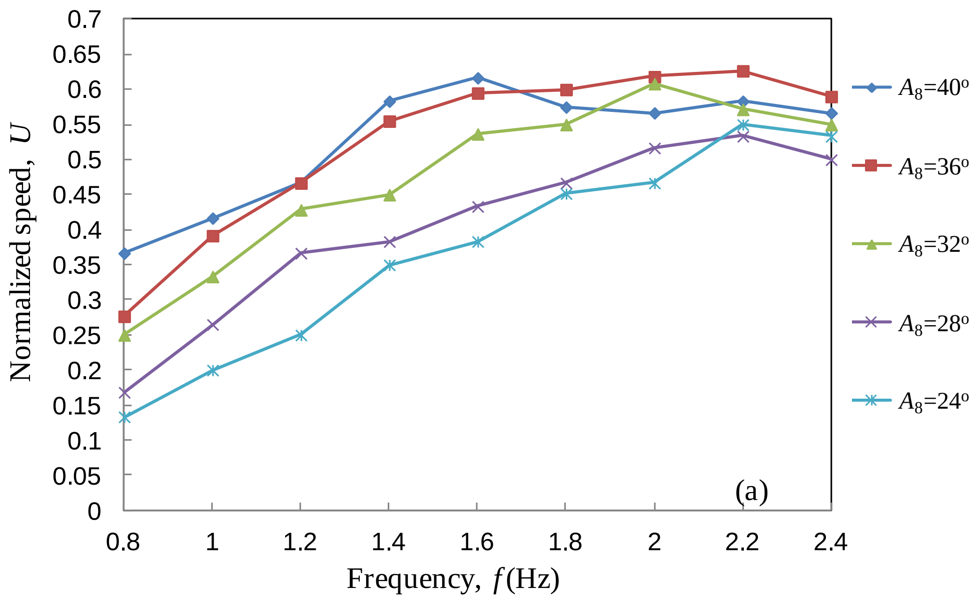
As many generations in GA will run before an optimum is found, the robot has to swim in the straight path for quite a long distance. For example, if 50 generations are performed, each generation contains 10 individuals. The total traveling distance will be at least 200 m. In the lab water tank, the robot has to perform reciprocating movements on the linear guide because the guide is not long enough.

On-line optimization of swimming gaits for a multi-DOF fish robot requires intensive computing costs because the performing of GA and solving of nonlinear differential equations of CPGs must be fulfilled within a short servo cycle. As shown in Fig. 7.3, a PC is utilized to perform GA and the CPGs are parallel handled in an embedded controller. The hardware ensures the real-time property of the whole control system. Equations in the CPG model are solved in LabVIEW™ development environment and implemented with the CompactRIO™ embedded controller. GA is programmed in Matlab™ software on a PC. An FTP server is built in Matlab™ for sharing data between CPG and GA.

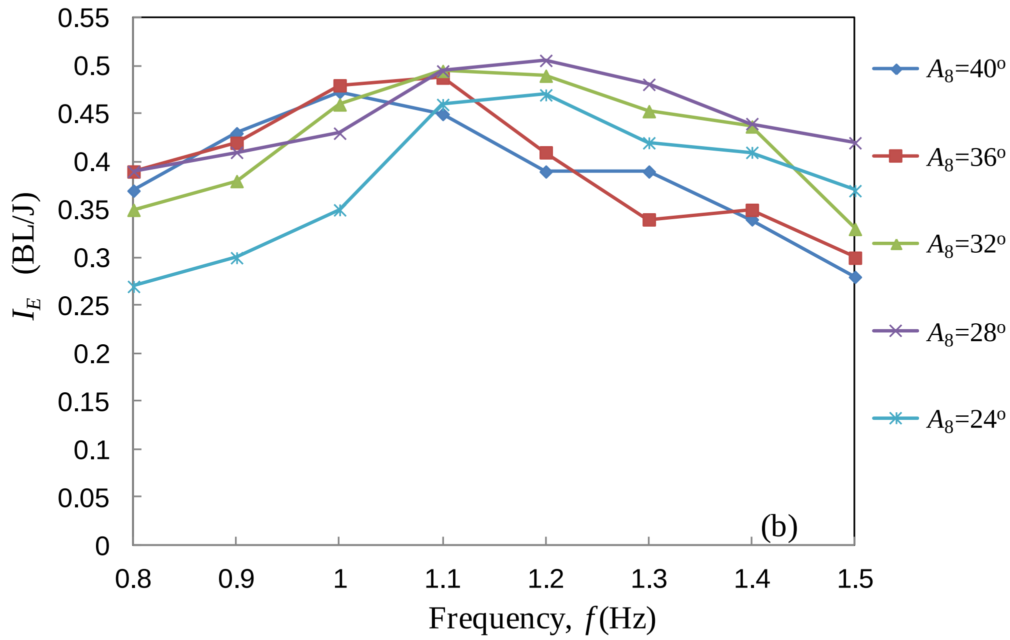
7.4.2 Performance Testing Based on Offline Approach

A common way to explore the swimming performance of a fish robot is to conduct empirical testing with different combinations of parameters. Considering the constraints given by Eq. (7.7), it can be assumed the envelop has a approximately linear form. In this case, the parameter set will be $S = \{A_8; f\}$, and the envelop can be described by a straight line equation. Amplitudes of other propulsor segments can be determined by the line equation. Such configuration can significantly reduce the number of testing runs since there are only two parameters to be explored.

Fig. 7.7a shows the swimming speed testing results. The best speed found in experiments is 0.618 BL/s (37.6 cm/s). A general trend can be seen that the swimming speed increases with the increasing of frequency. However, when frequency is larger than 2.2 Hz, the speed drops due to the power limitation of servo motors. The same experiment method has been applied to obtain efficiency testing results, which are shown in Fig. 7.7b. The most economical way of locomotion is found to have an undulation frequency f = 1.2 Hz and maximum amplitude $A_8 = 28^{o}$. The envelop and the propulsor profile during fast-mode swimming and efficient-mode swimming are illustrated in Fig. 7.8a and b, respectively.



(a)



(b)

Fig. 7.7 Off-line testing results of swimming performance: (a) swimming speed vs. frequency; and (b) energy efficiency vs. frequency. The experiments are conducted in several groups with different amplitudes. Selection of parameters is subjected to the power limitation of motors.

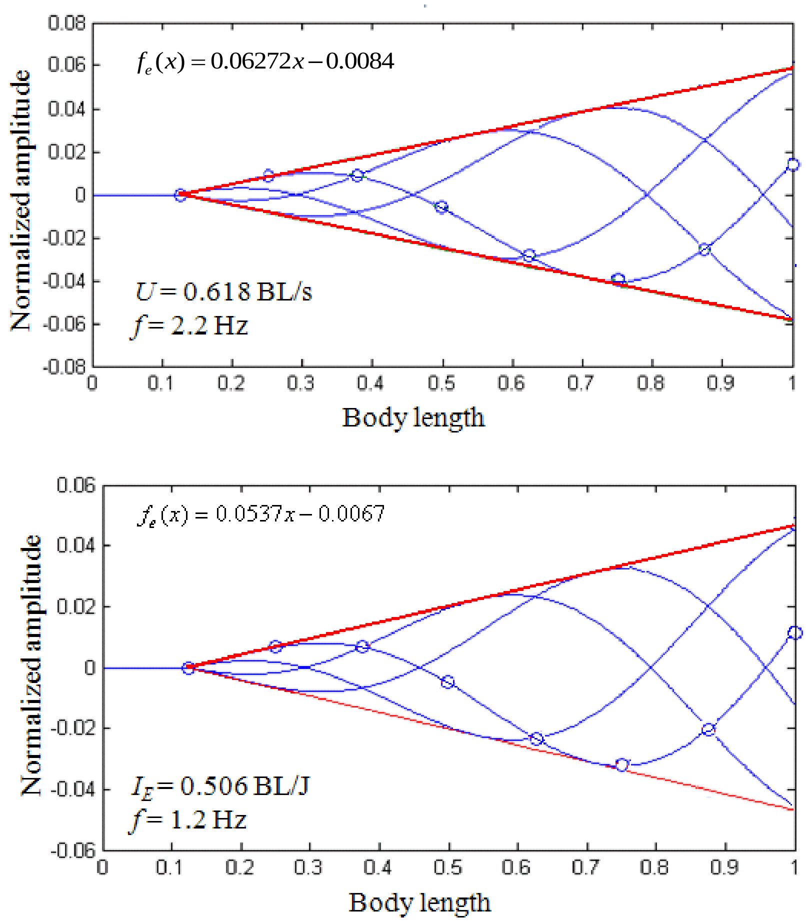


Fig. 7.8 Body curve associated to the optimal speed obtained by off-line testing.

7.4.3 On-line Searching of Optimal Swimming Speed

Because the eight cranks are equally-spaced, the phase difference between adjacent cranks share the same value ($\varphi_{d,i}=\pi/4$). The values of other constants in the CPG model follow those given in Fig. 5.13. In experiments, nine parameters are encoded into chromosome for GA. Each generation contains sixteen individuals. Generation gap is set to 0.9, which is a commonly used value in GA. Due to the mechanical limitations of the prototype, the maximum value of all amplitudes, A_{max}, is set to 45^{o} and maximum frequency is set to 2.5 Hz. Initial tests show that the maximum speed is not likely appear at frequencies lower that 1Hz. So the range of frequency is set to 1~2.5Hz. The population of the first generation is randomly created, subjected to Eq. (7.7) and the range.

The nine parameters in the parameter set Eq. (7.1) are encoded into chromosome for GA. The first generation is listed in Table 7.1. It can be seen that the objectives for those candidate solutions spread in a large range. As the GA improves the control parameters, the robot is expected to swim faster and faster. The performance of the GA, i.e. the convergence speed, is illustrated by Fig. 7.9. The red curve shows the evolving of the swimming speed of the best individual in each generation and the blue curve shows the evolving of the averaged speed of each generation. It is difficult to formally specify convergence criteria for GA as it is a stochastic search method. Fifty generations are performed in the search. From Fig. 7.9, it can be seen that the solution actually begins to converge within several generations.

Table 7.1 Individuals of the first generation in optimization of speed.

#	$A_1 \ldots A_8$ (Degree)								f (Hz)	U/L (BL/s)
1	20.1	23.2	24.3	28.0	29.9	30.0	32.4	34.6	1.9	0.552
2	17.1	21.6	22.8	24.5	27.2	31.6	35.8	39.4	1.3	0.441
3	3.0	6.2	8.5	13.9	20.4	26.8	32.0	34.6	1.1	0.281
4	0	2.1	2.5	3	5.3	6.3	6.5	15.7	1.0	0.05
5	1.5	1.5	2.2	6.9	12.6	19.3	22.4	26.60	2.1	0.460
6	10.1	11.6	18.1	18.3	25.9	27.7	33.7	36.3	1.3	0.417
7	1.5	4.0	7.7	15.4	20.2	24.8	28.4	30.5	1.3	0.340
8	0.7	3.0	4.8	7.0	12.5	15.5	20.5	25.4	1.8	0.401
9	10.8	11.9	15.6	19.7	21.3	24.5	29.4	33.5	1.8	0.552
10	14.0	19.0	27.0	29.5	30.2	30.6	34.4	38.2	1.7	0.579
11	2.4	6.8	8.7	15.3	15.5	19.7	21.9	28.0	2.4	0.587
12	16.1	18.8	24.4	26.3	29.0	35.0	36.3	41.4	2.1	0.496
13	0.7	7.1	8.2	15.5	19.9	26.3	29.7	33.9	2.2	0.587
14	8.8	15.7	16.8	16.9	21.3	21.8	29.4	31.6	2.1	0.622
15	0.2	2.2	4.1	9.5	10.6	16.2	22.8	27.9	1.5	0.346
16	0.5	1.9	7.2	12.8	14.9	20.0	25.7	32.8	1.2	0.290

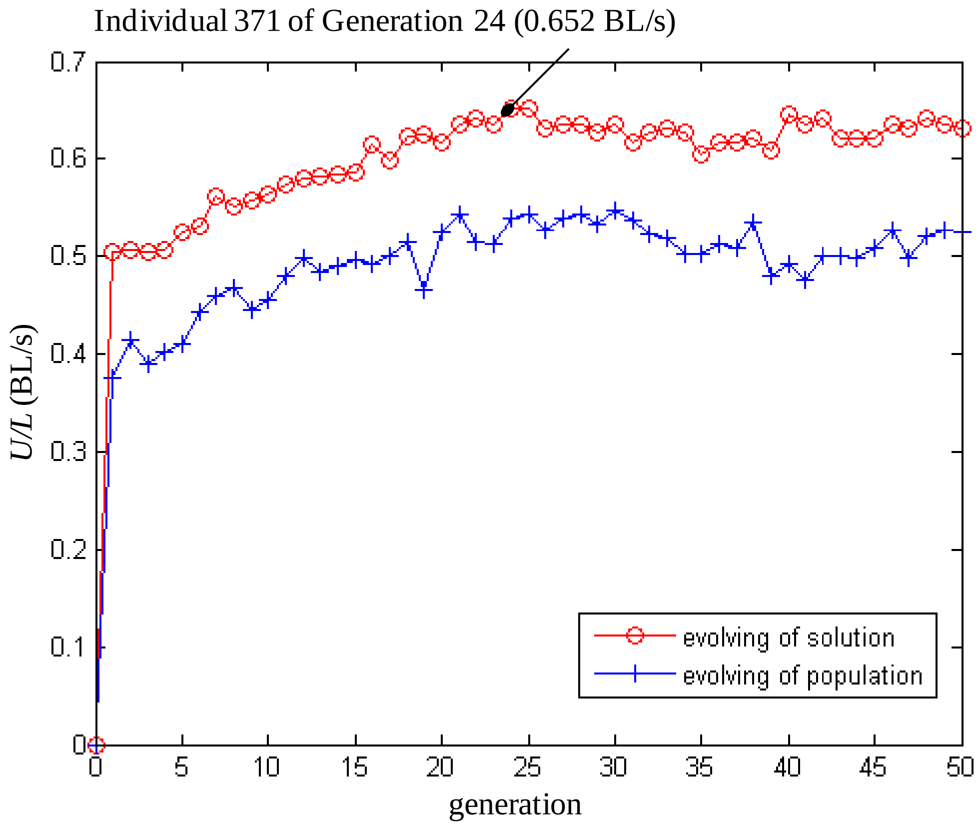


Fig. 7.9 Tracking the performance of GA in optimization of speed. Red curves represent the evolving of the best individual in 50 generations. Blue curves reflect the averaged performance of all individuals in 50 generations.

The optimal solution is the individual 371 in generation 24, which is listed in Table 7.2. The fastest speed of the fish robot is 39.1 cm/s (0.652 BL/s) and averaged power is 2.83 W. The wave number produced by the undulatory locomotion is approximate 0.88 in this case. The speed is better than the maximum speed obtained from the off-line based testing on the same prototype. In those tests, the fish robot is controlled under different combinations of parameters derived by an orthogonal experiment design [94]. The best swimming speed obtained from this statistical method is 37.4 cm/s (0.623 BL/s). The on-line optimization apparently achieves better solution.

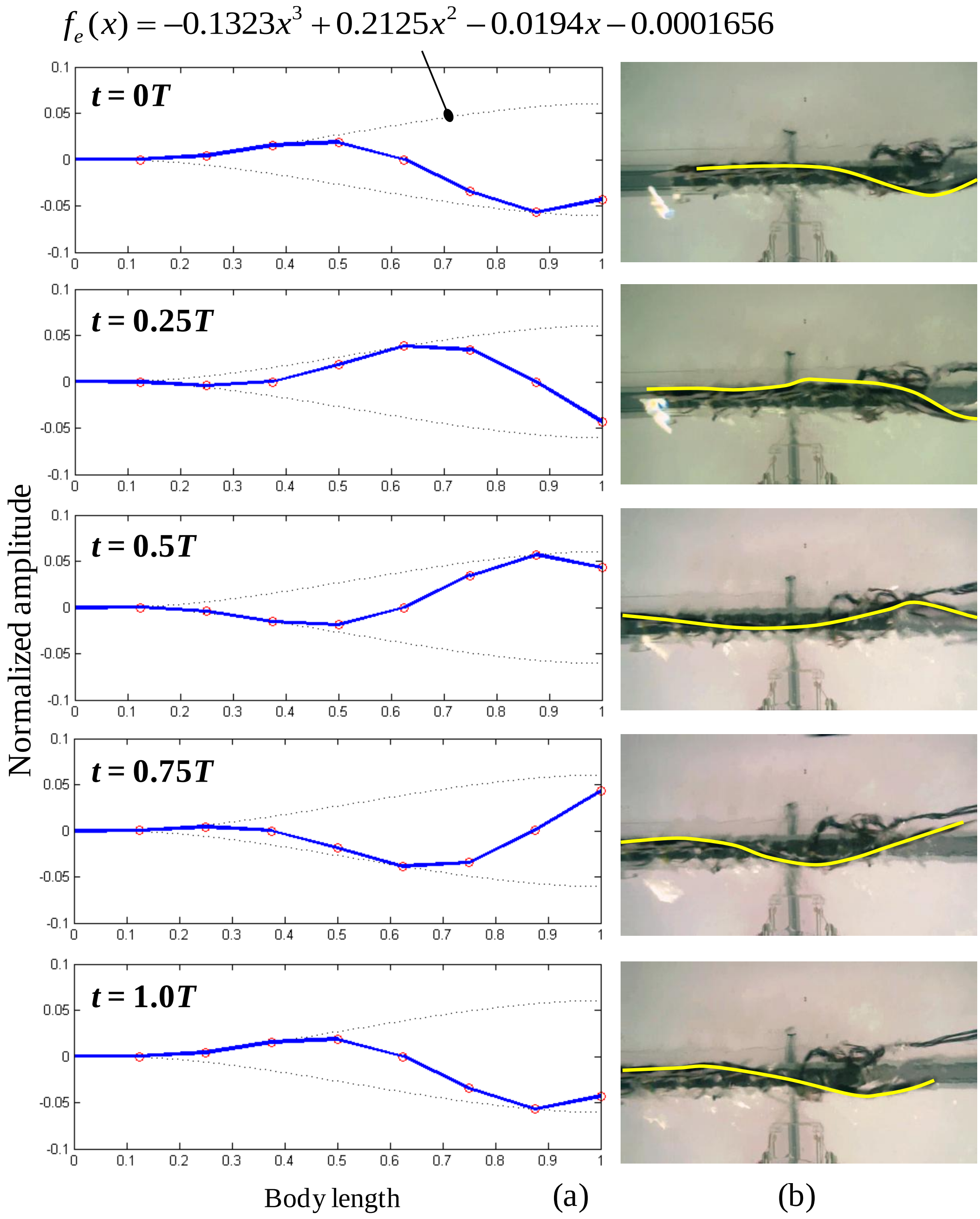
It is difficult to formally specify convergence criteria for GA, as it is a stochastic search method. Termination of iterations adopted in the present research is based on the following conditions: (i) the solution found in on-line testing is better than that found by off-line test; and (ii) no better solution appears in successive 5 generations.

If these two conditions are satisfied, the GA will stop evolving and output final results. The performance of GA, i.e. the convergence speed, is illustrated in Fig. 7.9. The red curve shows the evolving of the performances of the best individual in each generation, and the blue curve shows the evolving of the averaged performance of each generation.

In the optimization of speed, each generation contains 16 individuals and the search is performed in 50 generations (800 testing runs). It can be seen from Fig. 7.9 that the solutions actually begin to converge within several generations. The prototype performs back-and-forth straight line motion on the carriage in the experiment. The total traveling distance of the robot is approximately 1600 m. It takes 78 minutes to finish one testing. The best individual is found 41 minutes later after the online searching started.

Table 7.2 Individuals of the 24th generation in optimization of speed.

#	$A_1 \ldots A_8$ (Degree)								f (Hz)	U/L (BL/s)
369	0.0	3.8	3.8	3.8	3.8	3.8	29.4	33.9	1.34	0.177
370	0.0	2.9	8.7	16.7	21.3	29.0	35.9	36.9	2.28	0.603
371	**0.0**	**3.7**	**8.7**	**16.7**	**21.3**	**29.0**	**35.9**	**36.9**	**2.28**	**0.652**
372	0.0	3.7	10.	16.7	22.7	30.7	32.5	33.8	2.28	0.603
373	0.0	3.7	8.7	16.7	22.7	30.7	35.9	36.9	2.28	0.592
374	0.0	2.9	8.7	16.7	21.3	29.0	35.9	36.9	2.28	0.617
375	0.0	2.9	8.7	16.7	21.3	27.9	35.9	36.9	2.28	0.642
376	2.0	2.0	2.0	2.0	2.0	2.0	2.0	38.2	1.69	N/A
377	0.0	2.9	8.7	16.7	21.3	22.9	24.4	25.0	2.28	0.560
378	0.0	3.7	10.	16.1	22.7	30.7	30.9	33.8	2.18	0.560
379	0.0	3.7	8.7	16.7	21.3	29.0	35.9	43.9	2.28	0.600
380	0.0	3.7	8.7	16.7	22.7	29.0	35.9	36.9	2.23	0.592
381	0.0	3.7	8.7	16.7	21.3	29.0	35.9	36.9	2.32	0.583
382	0.0	3.7	8.7	16.7	21.3	29.0	35.9	36.9	2.18	0.622
383	0.0	2.9	8.7	16.7	21.3	27.9	35.9	36.9	2.28	0.613
384	0.0	3.7	8.7	16.7	21.3	29.0	35.9	43.9	2.28	0.587



Best speed: U = 0.652 BL/s where I_E = 0.23 BL/J, f = 2.28 Hz, T = 0.439 s

Fig. 7.10 Body curve associated to the optimal speed: (a) examples of profiles at different time in one undulation cycle (T = 0.439 s); and (b) the curves captured in the experiment at the same time instances. All dimensions are normalized for the propulsor length. The blue lines illustrate the fish body profiles at different time instances.

The profile associated to the optimal speed is described in Fig. 7.10. In this figure, data of the individual 371 in Table 7.2 is used to visually produce the propulsor's profile during its fastest swimming. The wave profile in successive five time instances in one undulation cycle is plotted. Fig. 7.10b shows the wave profiles of the testing propulsor at the same time instances in the experiments. It can be seen that the actual oscillations of cranks can well follow the control commands.

GA is used in a 3-D simulation reported in [16] and system parameter design for a robot tuna in [178]. In both cases, the envelop of body wave is modeled by a parabola curve and GA is used to find two curve parameters. Other forms of the body wave profiles are also used in locomotion control of fish robots. In the present work, the profile is not prescribed but searched by GA. The envelop can be well fitted by a third order polynomial function, $f_e(x)$, which is given by

$$f_e(x) = -0.1323x^3 + 0.2125x^2 - 0.0194x - 0.0001656 \qquad 0 \le x \le 1 \tag{7.15}$$

By considering the actual mechanism of the testing propulsor, the normalized amplitude is computed by the following equation:

$$\text{amplitude}_i = (l_i \sin A_i) / L \tag{7.16}$$

where l_i is the length of driving crank, A_i is from Table 7.2 and L is the length of the propulsor.

7.4.4 Optimization of Swimming Energy Efficiency

The optimization of swimming efficiency is similar as it in fast swimming mode, except that the objective function is given by Eq. (7.12) and fitness is evaluated by Eq. (7.13) in this case. The range of frequency is set to 0.8~2.5Hz. Other settings remain unchanged. Initial population is shown in Table 7.3.

Each generation contains 10 individuals and the search is performed in 60 generations. The online searching lasts 109 minutes and the best individual appears after the searching performed 97 minutes. The performance of the GA is shown in Fig. 7.11. The red curve shows the evolving of the largest efficiency index, I_E, in each generation and the blue curve shows the evolving of the averaged efficiency of each generation. Individual 545 in generation 55 is the optimal solution, which is listed in Table 7.4. The most efficient locomotion of the fish robot will have a speed of 15.0 cm/s (0.25 BL/s). The power is 0.424 W. It can be seen that relatively large amplitude appears in the tail part. This result is in agreement with the conclusion that most of the thrust is generated by the caudal part for efficient swimming [8].

Table 7.3 Initial individuals in optimization of efficiency.

#	$A_1 \ldots A_8$ (deg)								f (Hz)	U/L (BL/s)	P (W)	I_E (BL/J)
1	1.7	4.9	5.7	11.5	18.0	24.6	25.3	29.5	1.6	0.41	1.244	0.332
2	1.6	8.2	11.7	15.3	21.7	26.9	34.7	38.9	1.5	0.50	1.799	0.279
3	9.1	9.6	16.0	22.7	25.8	27.2	28.3	28.9	1.8	0.49	2.142	0.230
4	3.2	5.6	7.9	12.9	18.2	21.5	25.7	28.9	2.2	0.48	2.285	0.211
5	14.5	18.5	20.0	23.0	23.9	25.2	33.1	33.2	1.7	0.51	1.801	0.280
6	8.7	12.1	14.3	20.2	20.6	27.9	35.6	38.3	2.4	0.46	3.329	0.139
7	1.2	3.7	9.3	11.7	19.5	22.9	30.4	34.8	1.6	0.47	1.412	0.333
8	9.1	10.5	18.5	19.5	20.9	26.3	31.8	36.2	1.7	0.50	1.684	0.296
9	3.8	7.0	9.9	11.4	16.8	23.8	28.3	28.6	2.3	0.54	2.329	0.234
10	6.2	9.6	11.1	14.1	19.3	26.1	27.4	35.2	0.9	0.04	0.403	0.111

Table 7.4 Individuals of the 55th generation in optimization of energy efficiency.

#	$A_1 \ldots A_8$ (deg)								f (Hz)	U/L (BL/s)	P (W)	I_E (BL/J)
541	0.0	0.6	6.1	14.1	14.8	17.3	25.3	29.5	1.2	0.22	0.456	0.492
542	0.0	0.6	7.3	7.3	14.8	17.3	25.3	29.5	1.2	0.22	0.398	0.545
543	0.0	0.6	8.6	14.1	16.1	17.3	25.3	29.5	1.2	0.25	0.454	0.560
544	0.0	0.6	7.3	14.8	16.1	17.3	25.3	29.5	1.2	0.24	0.452	0.525
545	**0.0**	**0.6**	**6.1**	**14.1**	**16.6**	**17.3**	**25.3**	**29.5**	**1.2**	**0.25**	**0.424**	**0.589**
546	0.0	0.0	0.0	0.0	0.0	17.3	25.3	29.5	1.1	N/A	N/A	N/A
547	0.0	0.6	7.3	14.1	16.1	17.3	18.0	25.0	1.1	0.16	0.327	0.500
548	0.0	0.6	7.3	14.8	16.6	17.3	25.3	29.5	1.2	0.24	0.448	0.536
549	0.0	0.6	7.3	14.1	16.1	17.3	25.3	29.5	1.1	0.22	0.414	0.535
550	0.0	0.6	8.6	14.8	16.6	23.7	25.3	29.5	0.8	N/A	N/A	N/A

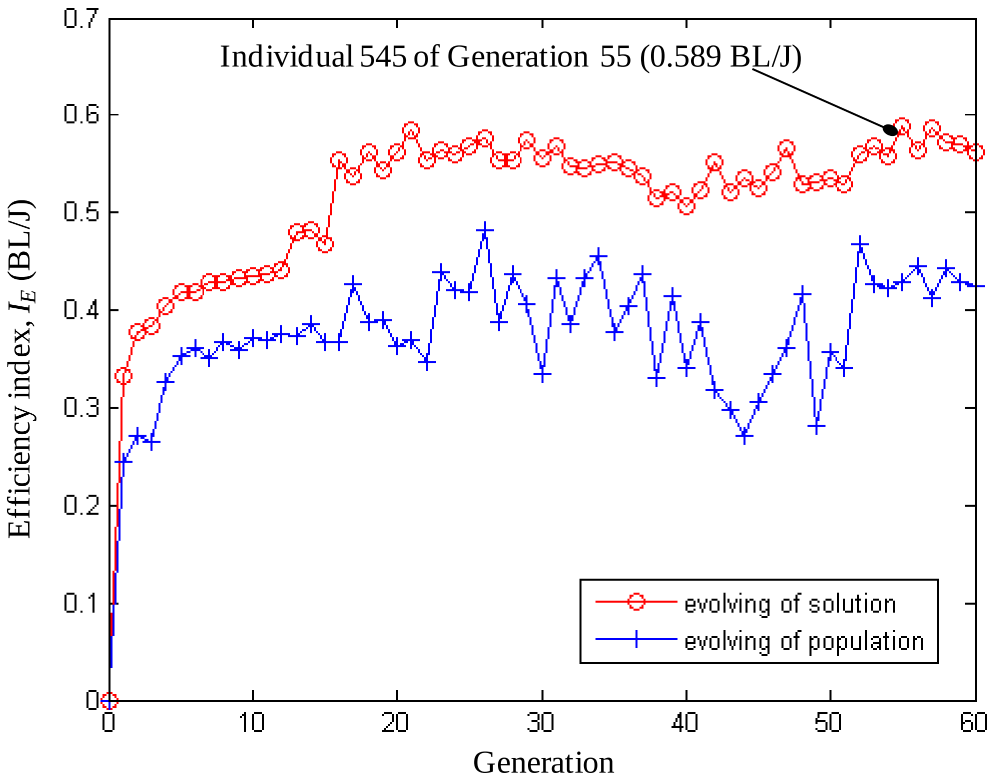


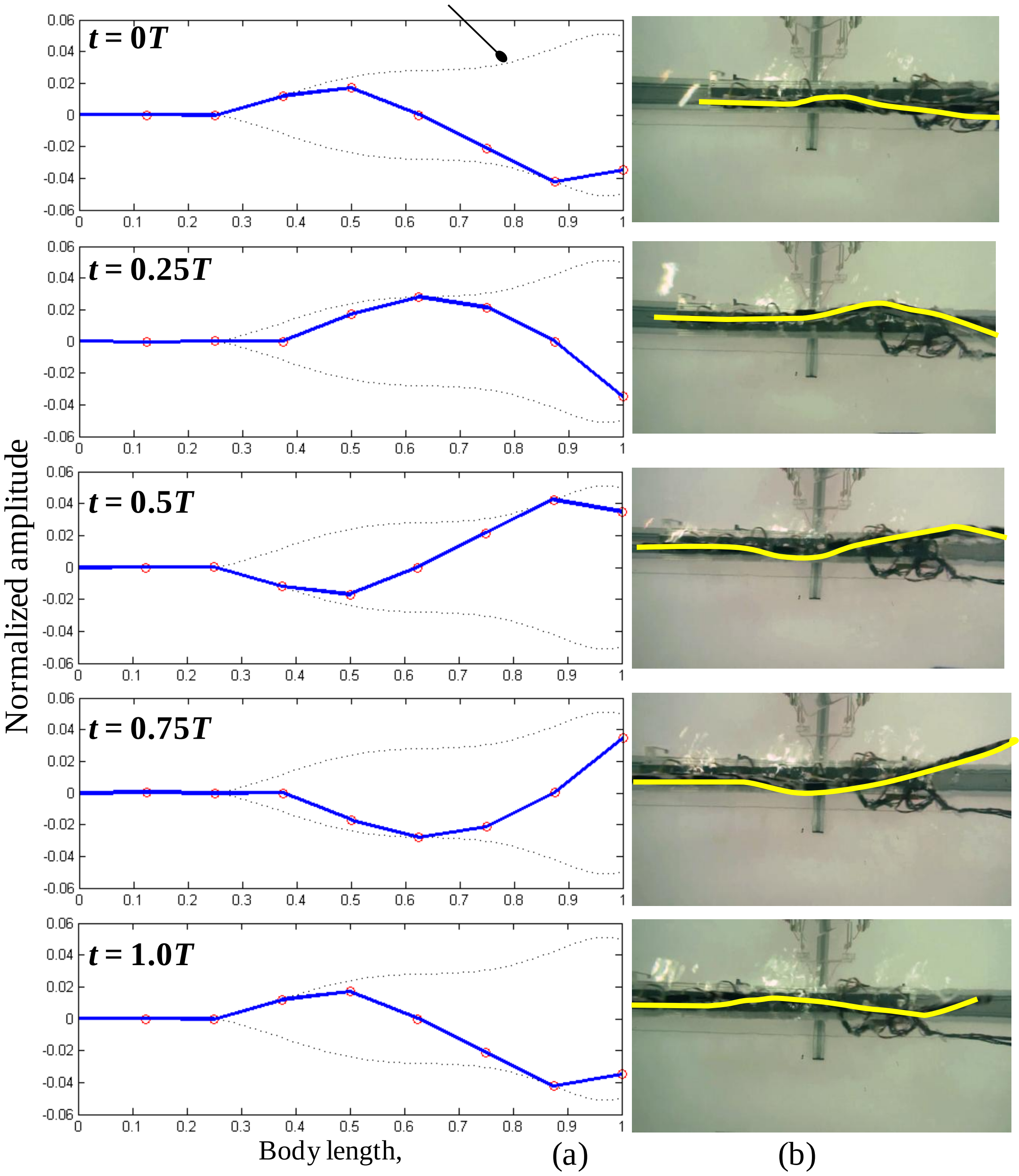
Fig. 7.11 Tracking the performance of GA in optimization of efficiency. The best individual in each generation is plotted in red square. The averaged efficiency index of 10 individuals in each generation is plotted in blue circle.

Similarly, the envelop of the body wave can be well fitted by a sixth-order polynomial, which is depicted in Fig. 7.12 and defined by

$$f_e(x) = \begin{cases} 0 & 0 \le x < 0.125 \\ -7.289x^6 + 21.9x^5 - 24.38x^4 \\ \quad +12.16x^3 - 2.5127x^2 + 0.1707x - 0.00007 & 0.125 \le x \le 1 \end{cases} \tag{7.17}$$

Note that in the online optimization for an actual robot, the evolving of the generations is totally depends on the sensor feedback. The quality of the sensor signals has a significant effect the convergence of GA. In optimization of speed, the convergence relies on only one sensor feedback, while in the optimization of energy-efficiency, the fitness function is computed by using data from 9 sensors (eight force sensors and one speed sensor), the reliability of the system is lowered. Relatively large fluctuation appears in the evolving of the generations. Therefore, the convergence is not smooth as that in speed optimization.

$$f_e(x) = -7.289x^6 + 21.9x^5 - 24.38x^4 + 12.16x^3 - 2.5127x^2 + 0.1707x - 0.00007$$



Best efficiency index: I_E = 0.589 BL/J where U = 0.25 BL/s, f = 1.2 Hz, T = 0.833s

Fig. 7.12 Body curve associated to the optimal efficiency. All dimensions are normalized for the propulsor length. The blue lines illustrate the fish body profiles at different time instances.

The experiments also show the advantages of the power measurement method presented in Section 7.3. When using electrical power consumption as (product of voltage and current) power index, the fish robot consumes 15.31 W electrical power in fast swimming mode and 7.79 W in economic swimming mode. However, by using the method given in Section 7.3, the power consumption is only 2.83 W and 0.424 W, respectively. This indicates that most of the input electrical power is wasted by the mechanisms and the motion transmission. The evaluation of efficiency based on the electrical power will not be a true reflection of the performance associated to that swimming pattern. If taking the power waste into account, it is difficult to find how the motion pattern affects the swimming performance. By using Eq. (7.11), the actual power acquired by a propulsor is obtained, which makes the experimental results in efficiency optimization stand on the firm ground.

7.5 Optimization of Locomotion for Different Propulsors

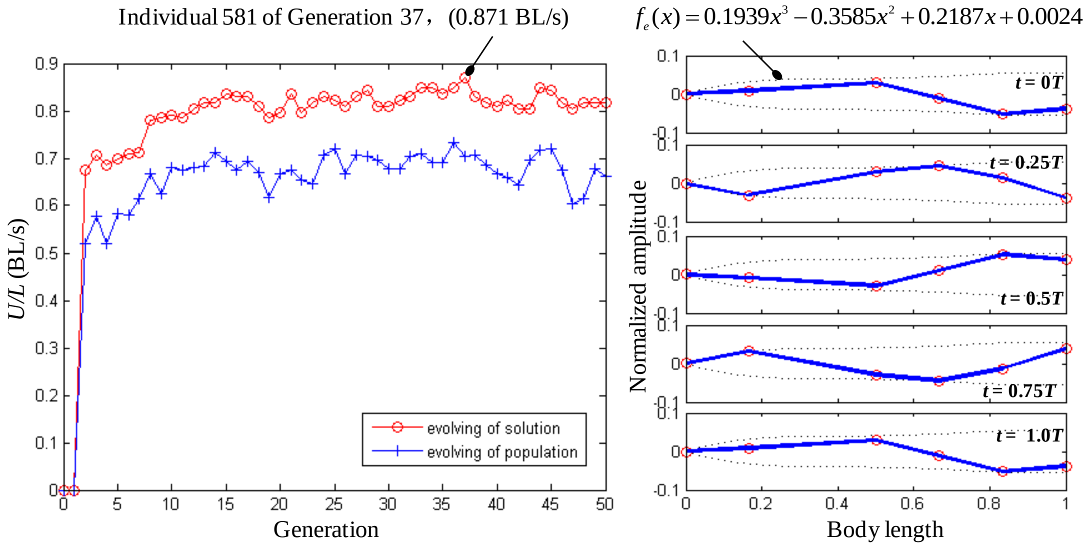
In order to assess the optimization approach discussed in this chapter, more experiments were conducted on different undulatory propulsors. These propulsors share the same mechanism as NKF-II. As mentioned in Section 2.31, modular designs are adopted in NKF-II. Therefore, a new propulsor can be obtained by simply removing or adding modules. Two more propulsors have been used in experiments, including a 6-DOF fin prototype (with 6 pairs of fin sliders) and a 4-DOF fin prototype (with 4 pairs of fin sliders). The setup of experiment device is the same as that shown in Fig. 3.8.

Each generation contains 16 individuals. The experiment process is similar to that conducted for NKF-II. The parameter sets for the two propulsors are $S=\{A_1, A_2, \ldots, A_6;\ f\}$ and $S=\{A_1, A_2, A_3, A_4;\ f\}$, respectively.

For the optimization of speed, the total number of generation in GA is chosen as 50. In the experiment, the best swimming speed found is 0.871 BL/s for the 6-DOF fin and 1.23 BL/s for 4-DOF fin. The experimental results are plotted in Fig. 7.13. The best speed in each generation is plotted with red circle in the figure. An increasing trend of the speed can be observed. Envelops of fast speed mode are:

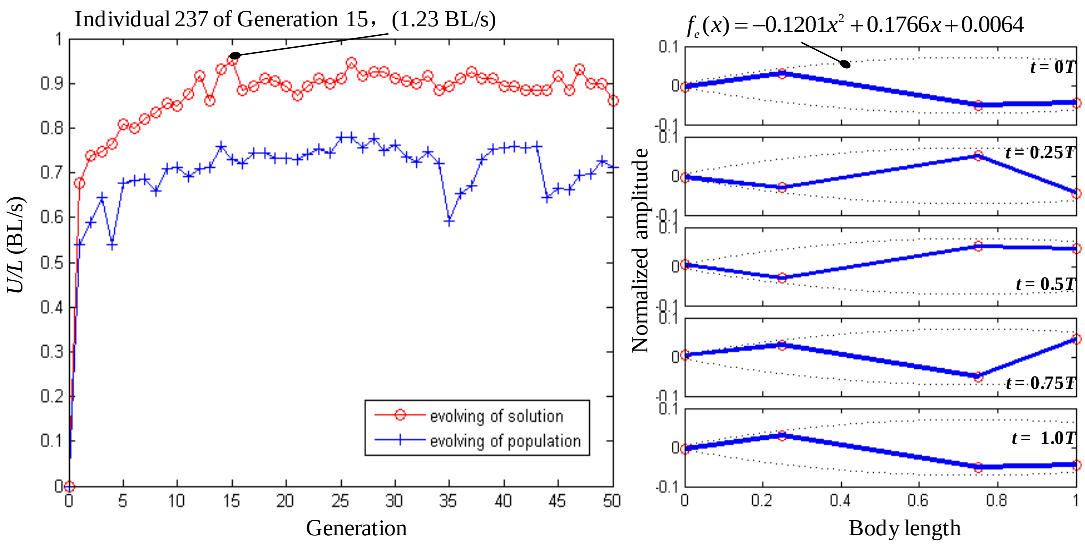
Propulsor with 6 fin pairs: $f_e(x) = 0.1939x^3 - 0.3585x^2 + 0.2187x + 0.0024$

Propulsor with 4 fin pairs: $f_e(x) = -0.1201x^2 + 0.1766x + 0.0064$



Best speed: U = 39.2 cm/s or 0.871 BL/s where f = 2.42 Hz, T = 0.413 s and I_E = 0.19 BL/J

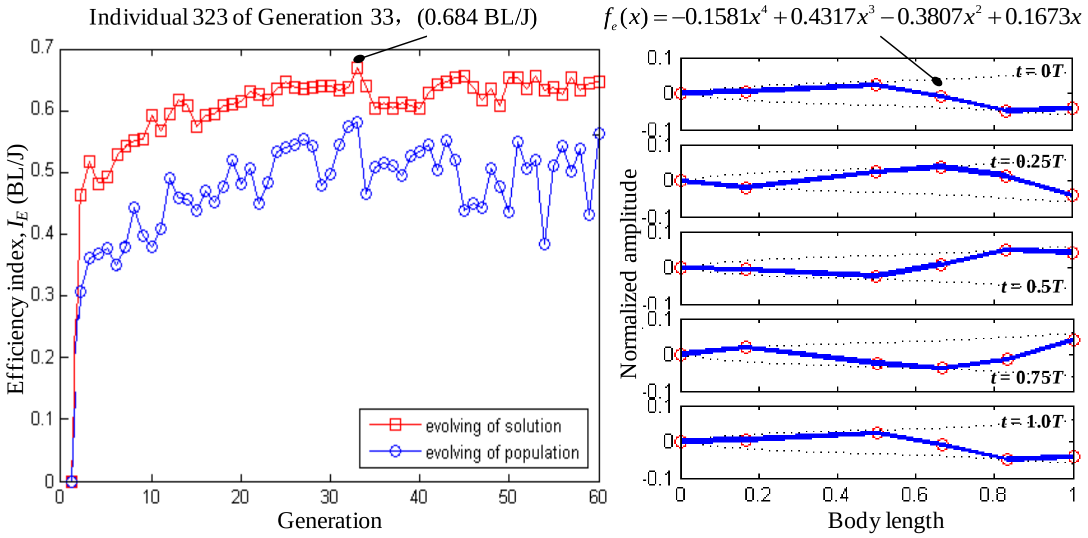
(a) The propulsor with 6 fin slider pairs



Best speed: U = 28.6 cm/s or 0.952 BL/s where f = 2.48 Hz, T = 0.403 s and I_E = 0.14 BL/J

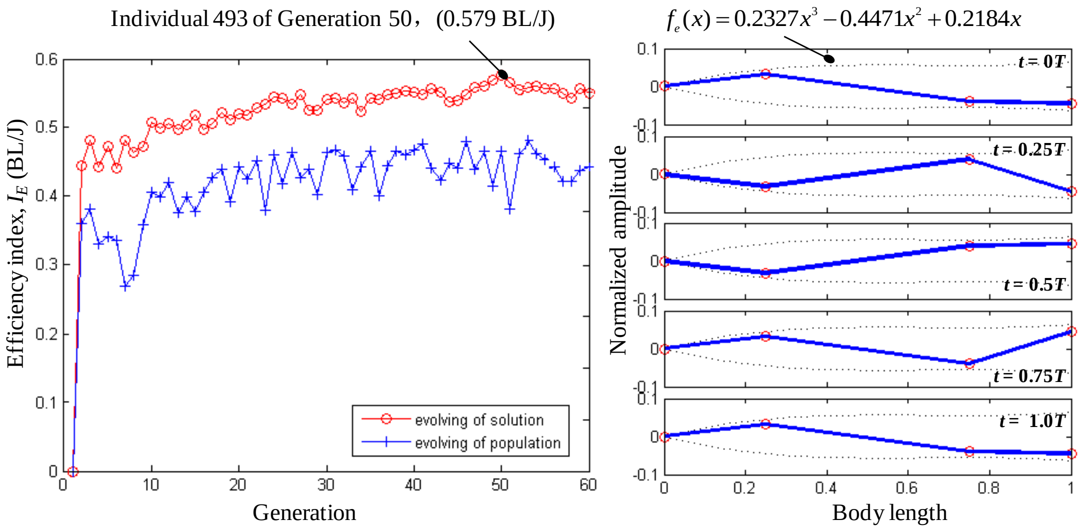
(b) The propulsor with 4 fin slider pairs

Fig. 7.13 Experimental results of the speed optimization for (a) the 6-DOF propulsor and (b) the 4-DOF propulsor. The left figures show the performance of GA in the optimization and the right figures show the optimal fin curvatures associated to the best speed at different time instances.



Best efficiency index: I_E = 0.684 BL/J where U = 30.1 cm/s or 0.67 BL/s, f = 2.42 Hz, and T = 0.413 s

(a) The propulsor with 6 fin slider pairs



Best efficiency index: I_E = 0.579 BL/J where U = 21.0 cm/s or 0. 7 BL/s, f = 1.51 Hz, and T = 0.662 s

(b) The propulsor with 4 fin slider pairs

Fig. 7.14 Experimental results of the energy efficiency optimization for (a) the 6-DOF propulsor and (b) the 4-DOF propulsor. The left figures show the performance of GA in the optimization and the right figures show the optimal fin curvatures associated to the best speed at different time instances.

For the optimization of the energy efficiency, the total number of generation in GA is chosen as 60. Each generation contains 10 individuals. The best energy efficiency index is 0.649 BL/J for the 6-DOF fin and 0.579 BL/J for the 4-DOF fin. The experimental results are plotted in Fig. 7.14. Envelops of energy-efficient mode are:

Propulsor with 6 fin pairs: $f_e(x) = -0.1581x^4 + 0.4317x^3 - 0.3807x^2 + 0.1673x$

Propulsor with 4 fin pairs: $f_e(x) = 0.2327x^3 - 0.4471x^2 + 0.2184x$

The experimental results on the three different propulsors show that each of them has its unique optimal swimming pattern to obtain faster speed or save the energy. These patterns can be different in many ways (distinct fin curvature and swimming frequency). These results strengthen the necessity of the on-line optimization of swimming locomotion as there is not a standard optimal swimming pattern that is valid for all fish robots.

The parameter sets and performance data obtained in the optimization of speed are listed in Table 7.5 for a comparison. It can be seen that by using the fin mechanism proposed in this research, the 6-DOF configuration of fin sliders yields the fastest absolute speed. This result seems to contradict a common sense that a larger propulsor (the 8-DOF one) should have the greater absolute speed because it will generate larger thrust. One should note that as more fin sliders are involved in the thrust generation, the drag from water will also increase. Therefore, the absolute speed may not be better than that with less fin sliders.

Table 7.5 Comparison of the speed optimization for different propulsors.

No.	Fin Pairs	L (cm)	$A_1 \ldots A_8$ (Degree)								f (Hz)	P (W)	I_E (cm/J)	I_E (BL/J)	U/L (BL/s)	U (cm/s)
1	8	60	0.0	3.7	8.7	16.7	21.3	29.0	35.9	36.9	2.28	2.83	**13.8**	0.23	0.652	**39.12**
2	6	45	17.8	20.2	23.5	28.2	31.8	32.9	-	-	2.42	2.85	**13.75**	0.31	0.871	**39.2**
3	4	30	25.0	34.4	35.8	43.8	-	-	-	-	2.48	3.25	**8.79**	0.29	0.952	**28.56**

The reason why 8 pairs of sliders were adopted in the design of NKF-II is that more such sliders can produce smoother body wave. If the swimming speed is a crucial criterion for the design of a fish robot, both 6-slider-pair and 8-slider-pair prototype designs are recommended as they can produce almost the same highest speed (39.12cm/s vs. 39.2 cm/s, see Table 7.5). The 4-slider-pair propulsor can produce the highest relative speed. However, it will not be recommended because its energy consumption is also high. It only advances 8.79 cm with unit power consumption, which is much shorter than the other two propulsors. By looking into the energy consumption aspect of the fast swimming mode, this recommendation is still given as both propulsors can advance almost the same distance by consuming unit power (13.8 and 13.75 cm/J, respectively).

If the energy consumption is a key issue to be considered in the design of propulsor, only the 6-DOF propulsor will be preferred. As shown in Table 7.6, it is the most energy efficient propulsor among the three. By looking into the swimming speed aspect of the energy-efficient swimming mode, this propulsor is also the preferable one because it advances 39.2 cm by consuming unit power. Therefore, if the consideration of energy efficiency is important, the propulsor should contain 6 fin slider pairs.

The conclusion that the 6-slider-pair is a preferable design for better energy efficiency can also be drawn by observing the fin curvature. It can be seen from Table 7.6 that the amplitudes of the first two cranks of the 8-slider-pair prototype are almost zero, i.e. the first two fin pairs do not move in such a mode. In this case, the 8-DOF fish prototype actually becomes a 6-DOF one. This phenomenon also supports that the 6-slider-pair should be adopted in the propulsor design for higher energy efficiency.

Table 7.6 Comparison of energy efficiency optimization for different propulsors.

No.	Fin Pairs	L (cm)	$A_1 \ldots A_8$ (Degree)	f (Hz)	P (W)	U (cm/s)	U/L (BL/s)	I_E (cm/J)	I_E (BL/J)
1	8	60	0.0 0.6 6.1 14.1 16.6 17.3 25.3 29.5	1.2	0.424	15.1	0.25	**35.6**	**0.589**
2	6	45	8.2 13.9 19.1 22.8 29.2 37.0 - -	1.42	0.769	30.12	0.67	**39.2**	**0.649**
3	4	30	23 30.9 33.7 39.8 - - - -	1.51	1.21	21.0	0.70	**17.4**	**0.579**

7.6 Summary

The application of CPG based swimming control for locomotion is further investigated in this chapter. With Genetic Algorithm optimizing parameters of CPGs according to on-line evaluation of swimming performance, a closed loop swimming control system for a multi-DOF fish robot is established to find the fastest locomotion pattern or the most efficient swimming pattern of the robot. This is a continuous work of our previous study reported in [30, 134, 135].

In the present work, the form of the body wave shape is not predefined. It is completely determined by eight amplitudes of cranks, which is the solution of optimization problem solved by GA. The experiments on the fish robot demonstrated that this consideration is necessary. It can be seen from Fig. 7.8 and Fig. 7.12 that the optimal body wave shapes of the fish robot found in on-line optimization is actually different from parabola or a linear form in both cases. The results obtained with on-line approach, the optimal swimming speed and the energy efficiency, are better than those found with off-line tests.

GA has been used to optimize the swimming locomotion in many cases. For example, it is utilized with a 3-D simulation to find the optimal locomotion patterns for the anguilliform undulation in [16]. In [178], it was adopted to find the optimal control parameter for a robot tuna. In both cases, the envelop of body wave is modeled by a parabola curve and GA is used to find two curve parameters. The physical properties of a fish robot (skins, shapes, materials, motions, etc.) can hardly be identical with those of a real fish or settings in a simulation. Therefore, it is not convincing to assume that the optimal body wave shape is also applicable for other fish robots.

The optimal body curve of a real fish is also found and explained by using CFD simulation in [16] or analysis of hydrodynamics in [176]. Ideally, those results can directly be used to guide the control of fish robots. However, the expectation is not supported by our experiments because that the analytical results do not coincide with curves shown in Fig. 7.10 and Fig. 7.12. This indicates that for an actual robot, the optimal swimming patterns may not be exactly the same as those exhibited by real fishes due to the significant physical differences in materials, structures and dimensions between the robot and real fish. In that sense, the experiment-based method discussed in this chapter will be of more value for fish robots. In future work, efforts will be put to customize the theoretical analysis method proposed in those publications for fish robots and compare the results obtained by using the two methods. These two ways are expected to benefit each other.

The results of optimization might only be restricted to the fish robot considered in this paper. For other fish robots or for a real fish, the pattern may be different from what is presented in this paper. One may argue that the results are not generic. As a matter of fact, this is the unique intention of the on-line optimization: different fish robots have their own optimal swimming patterns. The process of locomotion optimization discussed in this paper can be informative for optimal control of other types of fish robots and for understanding the mechanics of undulatory swimming. It is also hoped that the proposed model can be extended or further modified, not only for control of swimming robots but also for other types of robots having rhythmic locomotion patterns.

Chapter 8 Conclusions and Future Works

The thesis will be summarized in this chapter. Results and conclusions in previous chapters will be restated and discussed. An outlook of future work will be proposed.

8.1 Thesis Contributions

The kinematics modeling of swimming gaits for multi-DOF fish robots is discussed in this thesis. Several contributions are presented:

- Applications of Hopf nonlinear oscillator in generation of swimming gaits were investigated and implemented on fish robots. A closed-loop swimming control system is developed for a fish robot in order to achieve station-holding performance in adverse unsteady flow. Hopf oscillator was used to produce robust swimming gaits in this application. According to what we know so far, this is the first implementation of station-holding performance on a biomimetic swimming robot.
- An artificial CPG model based on CNLOs was formulated to generate on-line swimming gaits. The coupling schemes and structure configuration of oscillators of CPGs were analyzed. Applications of the CPG model in locomotion control of fish robots were discussed. The CPG model has been applied to control the locomotion of two multi-DOF fish robots (NAF-II and RoMan-II).
- A robust tracking controller was developed to regulate the tracking control of swimming gaits produced by CPGs. This controller ensures that the fish robot can perform the motions with high accuracy and therefore guarantees the reliability of the experimental results.
- On-line optimization of swimming locomotion has been conducted. By combining GA with CPGs, a closed-loop swimming control framework is established for locomotion control of multi-DOF swimming robots. The method has been tested on three propulsor prototypes. The fastest swimming pattern and most energy efficient swimming pattern were found.

8.2 Discussions

The implementation of fish-like locomotion on swimming robots is not the only goal of the research. It is expected that the swimming performance such as speed and power efficiency can also be improved through well-designed closed-loop control system. Sensory feedbacks were incorporated into the CPG model to achieve this aim. Control laws are fabricated to modulate gait patterns of fish robots, which can yield adaptive and optimal locomotion. Firstly, the mechanisms of biomimetic propulors of fish robots and sine-function based modeling of swimming gaits are discussed. This is an offline approach because those sine generators are decoupled from predefined fish body motion functions in order to maintain the coordination of motions. Gait control functions for fish robots with parallel mechanisms and serial open chain mechanism are derived, respectively. The former is achieved by simply specifying the phase difference for each generator. The latter is conducted by solving the optimal problem discussed in Chapter 3. The derived gaits have been tested on several fish robots.

The research road map characterized by the process from theories to implementations on actual robots is always preferred. The theories are expected to be able to guide the design of robot and the high performance locomotion control. However, in practical world, a perfect theory which can accurately describe the dynamics of real robots is not easily obtained. Deriving such theories is particularly difficult in robotic fishes due to several reasons: (i) the dynamics in the robot-water interaction is too complex; (ii) current theories are more or less based on certain simplification and ideal assumption which may not be satisfied by an actual fish robot. In this sense, the experiment based approaches would be important in this area. The on-line gait generation and gait modulation method discussed in this thesis establish a bridge between those data and the desired control command. Actually, the research in robotic fish area always goes in two lines: developing theories that can explain the mystery of high efficiency of fish swimming and finding solutions from experimental results. For example, the EBT, CFD simulation or the theories about the manipulating of vortices belongs to the former; the direct analysis of vortex data obtained in experiments and the method used in this thesis belongs to the latter.

Hopf oscillator was introduced to perform on-line generation of swimming gait. Different from sine generators, the frequency and amplitude of the nonlinear oscillator can be adjusted without causing jerks of motions. This robust property allowed the on-line tuning of oscillator parameters. By combining feedback control laws, the station-holding performance in adverse unsteady flow was successfully achieved on a fish robot. The application of the robust gait generator was extended to control swimming gaits of multi-DOF fish robots. In this case, the coupling connections are derived for multiple oscillators. The one way coupling between two oscillators, bidirectional coupling of two oscillators and multiple couplings to one oscillator were studied. In applications, the type of coupling was selected according to actual robot. Generally, the multiple coupling is adopted to obtain faster gait transition.

The CPG based method is chosen in this research not because it is superior over other existing methods. The reason is that CPG can provide properties required in the on-line motion planning of this research. Firstly, the control signals are not predefined but on-line generated as a robot swims. Secondly, those control signals should be coordinated even if some perturbations are imposed on the system. Thirdly, when the control signals are modulated, the coordination property is still maintained. These properties offer opportunities to modulate the gait shape even if the locomotion is on the fly. Such regulation can be made by feedback control laws derived by many well-documented classic or modern control theories. By using this way, the locomotion and the implementation of closed-loop swimming control on a fish robot would be easier.

The closed-loop swimming control was further investigated by combing Genetic Algorithm with CPG in optimization of swimming locomotion. Optimization of swimming speed and power efficiency were discussed. The optimal speed found by this on-line optimization was better than that found by off-line based approaches. The speed per unit input power was used as an indicator of cost efficiency. Under this criterion, the efficient swimming pattern was obtained in experiments. A measurement method was developed to evaluate the swimming efficiency. By using Eq. (7.11), the actual power acquired by the propulsor could be obtained, which made the experimental results in efficiency optimization stand on the firm ground. In the swimming gait tracking control, the normal PID controller was not adequate to maintain accurate tracing of gait commands due to the un-modeled hydrodynamics and the uncertainties among parameters of mathematical model of fish robot. A robust tracking controller was developed by using the error space analysis. This tracking controller was proved superior over the PID controllers in our experiments.

The significance of combining the CPG model and feedbacks in a closed-loop control system is that it enables such control system with intelligence somehow. CPGs themselves have no intrinsic adaptive capabilities to the environment of robots. The evolving of states in CPGs could be toward a direction which is beneficial for a robot or a totally unacceptable direction. Such evolving is completely decided by governing nonlinear differential equations. The feedback control system can be used to guide the evolving of CPGs in a manner useful for robots. Another significance of this method is that for a given swimming robot, it provides a way to dig its potentials. The station-holding control and the optimization of swimming locomotion on fish robots are conducted based on those considerations.

8.3 Future Works

In the thesis, several conditions are pre-assumed to simplify the problem. In most cases, the fish robot is assumed to perform steady swimming in still water. The focus of the thesis is placed only on the coupling schemes of oscillators and swimming performance regarding the speed and energy efficiency under this assumption. Kinematics models are designed to produce swimming gaits for straight-line locomotion as it is the major way a fish robot moves by. More versatile motions like burst, fast start, sharp turning, and diving are not topics of this thesis. The application of CPG based locomotion control in those aspects will be investigated in future works.

A relatively simple case of transient swimming is also studied: the station-holding performance on fish robots. This is a preliminary work in manipulation of transient movement for fish robots, which is used to illustrate the application of a nonlinear oscillator in generation of robust swimming gait and the effect of closed-loop swimming control. The condition is restricted to one-dimensional adverse unsteady flow. In actual water environment, such as river and ocean, the flow situation will be much more complicated. The station-holding control, or more generally, the trajectory tracking control in unsteady flow, will be a critical challenge. One difficulty is that the effective sensors perceiving tiny changes of flow velocity and pressure can hardly be obtained, which limits the implementation of closed-loop swimming control for fish robots. With the improvement of sensors, study on the movement manipulation in unsteady flow will be a future research direction.

Several prototypes of fish robots and the experiments are presented in this thesis. These facilities could be useful test beds for study on hydrodynamics of fish swimming. The efficient swimming may be associated to certain power distribution along the fish body. The power contributions by muscles at different parts of a real fish body are also different. Is there an analogy between the biological system and the experiment results on fish robot? What is the explanation of the power distribution in the hydrodynamics aspect? And is there any hints can be obtained from the wake structure associated to the optimal swimming pattern? These could be interesting questions to be answered in future works.

The process of locomotion optimization discussed in the thesis will be informative not only for control of swimming robots but also for other types of robots which have rhythmic locomotion patterns. For fish robots, the swimming speed and energy efficiency are selected as indications of performance. For other types of robots, the performance index can be different. The implementation of the method in more applications would be a direction in future works. Also, different types of CPGs models and their applications on locomotion control of robots are worth exploring.

In this thesis, an oscillator with harmonic output was used to control fish robots because the swimming gaits of fish exhibit such pattern. For other types of robots with rhythmic locomotion patterns, legged robots, for example, the gaits may not necessarily be subjected to harmonic patterns. The gait control for those robots can also benefit from the limit cycle of CPGs. It can be achieved by two ways: (i) train parameters of a given oscillator network (Matsuoka oscillator, Van Der Pol oscillator, for instance) by a reference pattern so as that the oscillator can produce approximate gaits; or (ii) develop a method that can map a known limit cycle to a desired limit cycle. The research in these two ways is toward to design methods of artificial CPGs, which will help the CPG based control applicable for more kinds of robots.

In this thesis and in current literatures related to the CPG based control, the topological structure and the coupling schemes of artificial CPGs have fixed forms. It is different for a real animal. When it grows up, an animal's locomotor nerve system also keeps developing according to its experience. This means a real CPG has the ability of learning and self-organizing. This remarkable ability makes the locomotion of animals highly adaptive to the environment. In this thesis, the adaptation is obtained through tuning of the parameters of fixed-form oscillator network. A more exciting way is to develop CPG models which can be trained by experiences and can automatically re-organize the structure of oscillators or coupling schemes. Developing such an intelligent CPG will be a promising future research direction.

Appendix A Design of a PD Tracking Controller for NAF-II

This appendix is included as a supplement of Chapter 6. The modeling of the motor-tail system and the PD tracking controller are discussed. In NAF-II, a DC motor provides torque, T_m, to drive the tail mechanism (see Fig. 6.1).

PID controller is used for the tracking control of tail fin, which is given by

$$\mathrm{PID} = K_p + K_i \frac{1}{s} + K_d s \tag{A.1}$$

where K_p, K_i and K_d are coefficient of proportional, integral and derivative terms respectively. The whole feedback control scheme can be depicted by the following block diagram [146]. θ_d is the control input, i.e. the gait control function derived in the previous chapter, θ is the actual angular position of joint 1 (see Fig. 2.27). The open loop system described has already contained an integrator, which can result in a zero steady state error for the step response. Hence the integral term of PID controller can be absent, i.e. $K_i = 0$. Two main requirements for the control system are presented. The overshoot, $P.O.$, is expected to be less than 5% and the setting time t_s to be less than 20ms (2% criterion) to a unit step input, which means:

$$P.O. = 100 \cdot e^{-\xi\pi/\sqrt{1-\xi^2}}\,\% < 5\% \tag{A.2}$$

$$e^{-\xi\omega_n t_s} < 0.02 \tag{A.3}$$

where ξ is the damping ratio of the closed-loop system and ω_n is the natural frequency [146]. By solving Eq. (A.2) we obtain:

$$\xi > 0.7 \tag{A.4}$$

Here ξ is set to 0.707. By combining Eq. (A.2) and (A.3), the real part of the desired closed-loop poles can be obtained:

$$\left|\mathrm{Re}(p)\right| = \xi\omega_n \geq 195.6 \tag{A.5}$$

The pole of the open-loop system is $1/\tau$=121.7. The desired closed-loop poles are p=-195.6+j195.6. The PD controller changes the structure of system by moving the pole of the open-loop system.

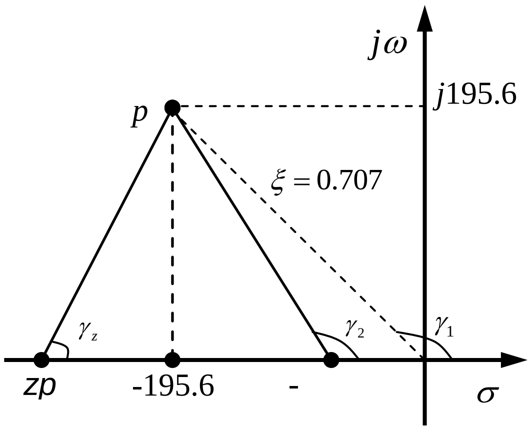


Fig. A.1 Using graphics to calculate the parameter of controller.

Many different methods have been developed to obtain the parameters of PID controller. Fig. A.1 shows a way utilizing the root locus [146]. By connecting all zeros and poles of the open-loop system to the desired closed-loop poles, the phase condition and amplitude condition can be used to obtain the parameters of the PD controller. The controller will add one zero $zp=-K_p/K_d$ to the system. According to the phase condition, we have

$$\gamma_{zp} = \gamma_1 + \gamma_2 - 180^\circ = 135^\circ + 118.3^\circ - 180^\circ = 73.3^\circ \tag{A.6}$$

$$\tan \gamma_{zp} = \frac{195.6}{zp - 195.6} \tag{A.7}$$

By combining Eq. (A.6) and (A.7), the zero can be located.

$$zp = \frac{K_p}{K_d} = 254.3 \tag{A.8}$$

Then by using amplitude condition, the gain of obtain the gain of the open loop system:

$$\left| \frac{1010.2K_d(s+228.1)}{s(s+98.35)} \right|_{s=p} = 1 \tag{A.9}$$

Solve this equation and obtain the value of K_d:

$$K_d = \frac{|p| \cdot |p+98.35|}{1010.2|p+228.1|} = 0.3016 \tag{A.10}$$

Then the proportional term can be obtained:

$$K_p = zp \cdot K_d = 76.70 \tag{A.11}$$

Hence the transfer function of PD controller can be obtained:

$$\mathrm{PD} = 76.7 + 0.3016s = 0.3016(254.3 + s) \tag{A.12}$$

The open-loop gain K_o is given by

$$K_o = \frac{K_m G}{RJ} = \frac{180 \times 14.3 \times 10^{-3}}{3.1416 \times 0.902 \times 87.3 \times 10^{-7} \times 103} = 1010.2 \tag{A.13}$$

By incorporating the controller, the feedback control becomes:

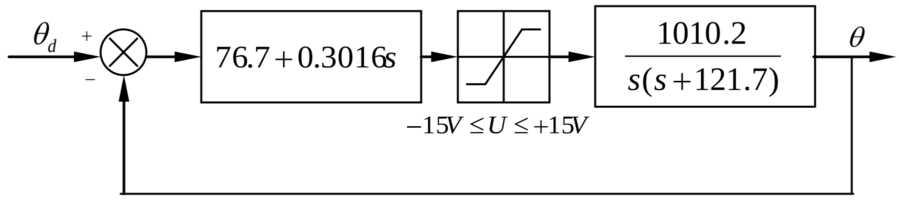


Fig. A.2 Closed-loop motion control of the fish tail with a PD controller.

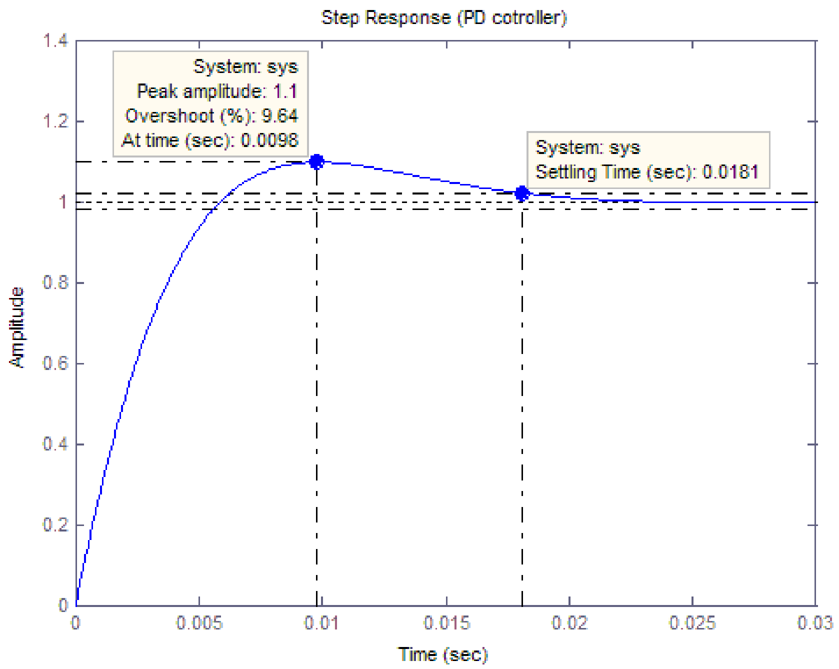


Fig. A.3 Step response of gait tracking with a PD controller.

The final closed-loop transfer function is given by:

$$\frac{\theta(s)}{\theta_d(s)} = \frac{304.7(s + 254.3)}{s^2 + 416.7s + 77480} \tag{A.14}$$

To justify and test the effect of the PD controller, a simulation of the step response is done in Matlab™ software. Fig. A.3 gives the simulation results of the PD controller. The system has quick response. By using a mere PD controller, the overshoot (9.64%) exceeds the expected 5% limit. The PD controller is designed based on the assumption that the control system has properties of a standard second order system.

It brings an extra zero (zp=-254.3) to the closed-loop control system, which changes the dominance of ideal closed-loop poles [146]. This additional zero causes the overshoot of larger step response than the desired value. A fore filter is employed to eliminate the influence of this zero. The filter's pole is set as the zero of the closed-loop system to eliminate that zero and the gain is set as the same value to make sure that this filter will not change the gain of the closed-loop system. The transfer function of the filter can be interpreted as:

$$filter(s) = \frac{254.3}{s + 254.3} \tag{A.15}$$

The modified control block diagram is shown in Fig. A.4. The solid line in Fig. A.5 illustrates its step response, from which the obvious improvement of the overshoot can be found.

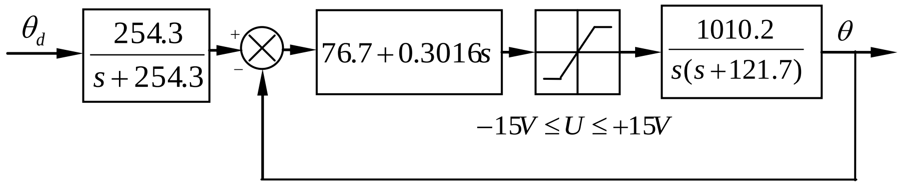


Fig. A.4 Block diagram of closed-loop system with fore filter.

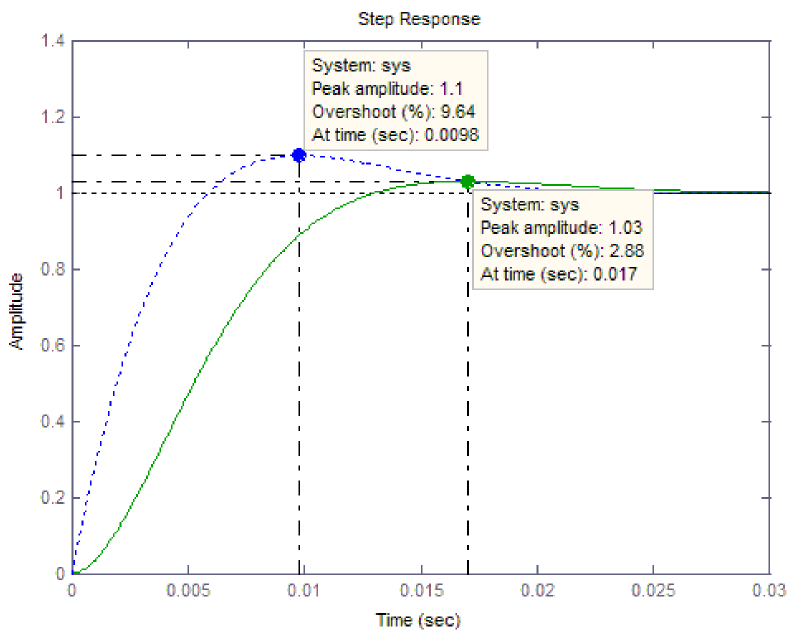


Fig. A.5 Step response with PD controller and fore filter.

Appendix B Modeling of Gaits for Fish Robots with Serial Open Chain Mechanism with Sine Generators

For the planar serial open chain design, the linkages comprising the propulsor are controlled in order to fit a body (or fin) curve defined by body motion functions. The gait control function describes the relative angles of links with respect to the adjacent link. The body motion function can be decomposed into the time-dependant gait functions for every actuator. This gait planning becomes an inverse kinematics problem, which consists of the determination of the joint angles corresponding to a posture given by the body motion function. The solution to this problem is of fundamental importance in order to transform the body motion specifications into the joint space motions, which allow the execution of body curve fitting by the chain of rigid links.

There is no standard method for the joint determination and it is difficult to find a completely analytical solution to this inverse kinematics problem. Hu and Liu [71-73] presented excellent works on swimming motion control of BCF type fish robot by using linkages mechanism to fit fish body curve. The authors proposed a method to achieve high precision curve fitting and compensation of the effect caused by swing motion of fish head. Smooth, flexible and elegant swimming motion has been obtained on their fish robot prototype. In other literatures, numeric methods are applied with a lookup table that consists of the positions of joints at different time instances. The table is pre-computed and programmed in a microcontroller. When the robotic fish swims, these positions are read from memory and act as the control input of actuators. For example, The *OscData* is I by K matrix that contains the angles of every joint of a serial open chain mechanism [174],

$$OscData[I][K] = \begin{pmatrix} \theta_{01} & \theta_{02} & \cdots & \theta_{0K} \\ \theta_{11} & \theta_{12} & \cdots & \theta_{1K} \\ \cdots & \cdots & \cdots & \cdots \\ \theta_{I-1,1} & \theta_{I-1,2} & \cdots & \theta_{I-1,K} \end{pmatrix}$$

in which I is the number of actuators, K is the number of steps per motion cycle and θ is the joint angle. This numeric method shows good performance in the gait control of in many fish prototypes. One major disadvantage of this method is that the swimming gaits are not on-line generated and it is difficult to adjust the swimming pattern when the robot swims.

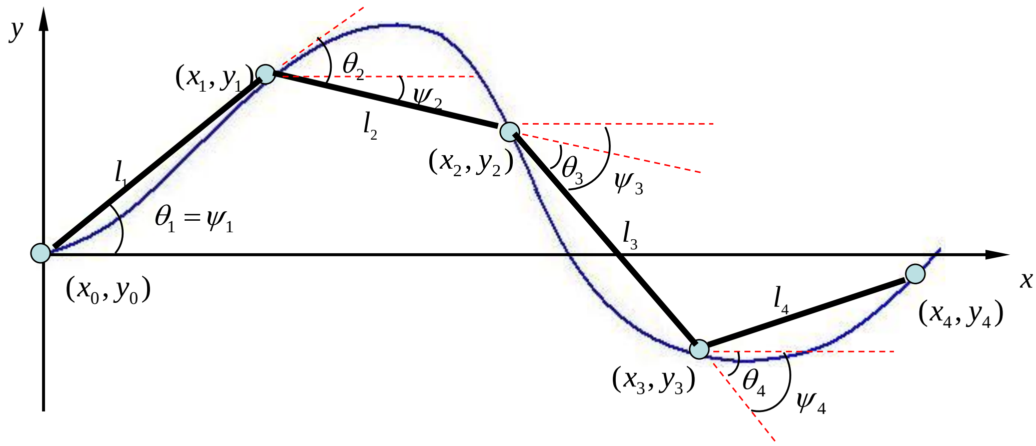


Fig. B.1 Fish body curve fitting through rigid linkages.

In this thesis, the inverse kinematics problem is analytically solved in order to change the motion parameters as the robot swims. Fig. B.1 shows a section of body curve and the fitting through four rigid links. Note that (x_i, y_i) is the position of the revolute joint in the coordinate system, $\Psi_i(x, t)$ is the absolute angle of the i^{th} link, $\theta_i(x, t)$ is the relative angle of the i^{th} link with respect to previous link, and l_i is the length of the link (i = 1, 2, 3,4,…,n). Without loss of generality, if the body length of the robotic fish is denoted by L and the total number of links is denoted by n, the following relationship can be assumed,

$$L = \sum_{i=1}^{n} l_i \approx m\lambda \tag{B.1}$$

The relative angle of the i^{th} link in Fig. B.1 can be calculated by

$$\theta_i = \psi_i - \psi_{i-1} \tag{B.2}$$

and

$$\psi_i = \arctan \frac{y_i - y_{i-1}}{x_i - x_{i-1}} \tag{B.3}$$

The expression of the relative angle of the i^{th} link can then be obtained as

$$\theta_i = \arctan \frac{y_i - y_{i-1}}{x_i - x_{i-1}} - \arctan \frac{y_{i-1} - y_{i-2}}{x_{i-1} - x_{i-2}} \qquad i=2,3,\ldots,n \tag{B.4}$$

and

$$\theta_1 = \psi_1 = \arctan\frac{y_1}{x_1} \tag{B.5}$$

It is noted that in the swimming motion the i^{th} link always rotates about the joint with previous link. As a result, the constrain equation for the position of joints can be given by

$$(y_i - y_{i-1})^2 + (x_i - x_{i-1}) = l_i^2 \tag{B.6}$$

where $y_i = f(x_i, t)$. The gesture of the fish body defined by the motion function is a two dimensional function with respect to time and position.

The waveform can be thought of a result of oscillations of all linkages in a coordinated manner. Those oscillations should be governed by gait functions of time *t*. The expression of the gait given by Eq. (B.4) is stilled coupled with the position. To find out this time-dependant function, look into an example depicted in Fig. B.2. Here, it can be assumed that the fish body generates n_λ waveform and the active part of body is fitted by three rigid links with lengths of l_1, l_2 and l_3 respectively. The oscillation frequency of linkages is set to 1 Hz. The cases at t = 0.2 s and t = 0.6 s are shown in Fig. B.2 as examples.

The three joint angels in are depicted in Fig. B.3. As those angles show obvious periodical patterns, they can be modeled by a harmonic function:

$$\theta_i(t) = \begin{cases} \arctan\dfrac{y_1}{x_1} & i = 1 \\ A_i \sin(2\pi f t + \varphi_{d,i}) & \text{otherwise} \end{cases} \tag{B.7}$$

where φ_i is the phase difference between two adjacent linkages.

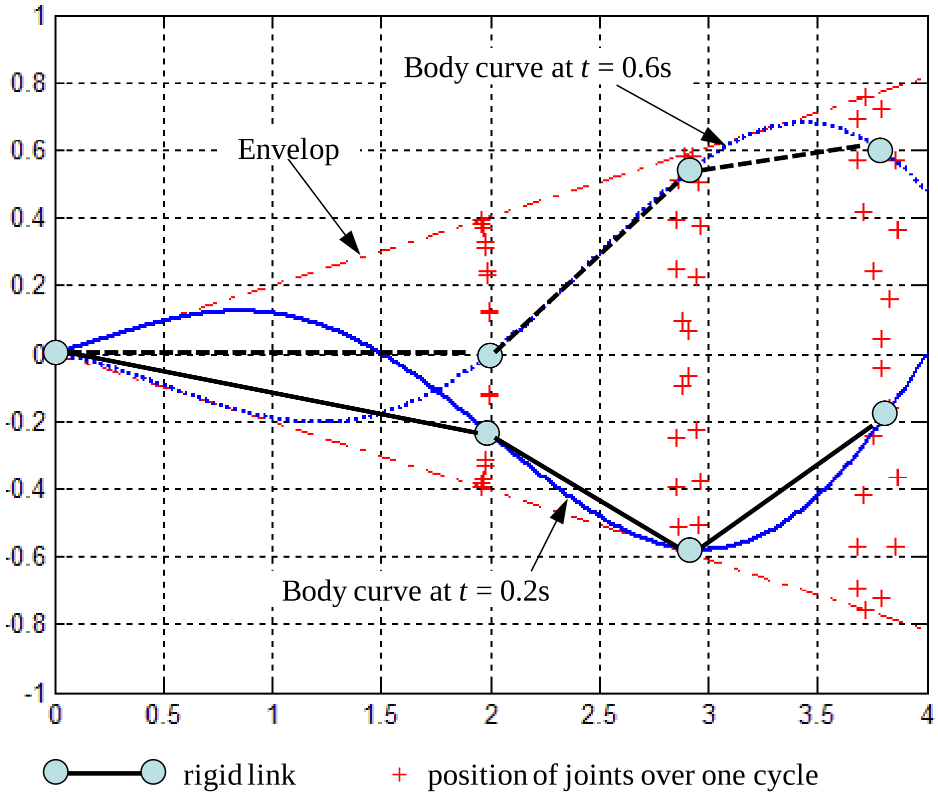


Fig. B.2 A body curve fitting example by rigid body linkages. Three links are employed. All dimensions are normalized to the body length.

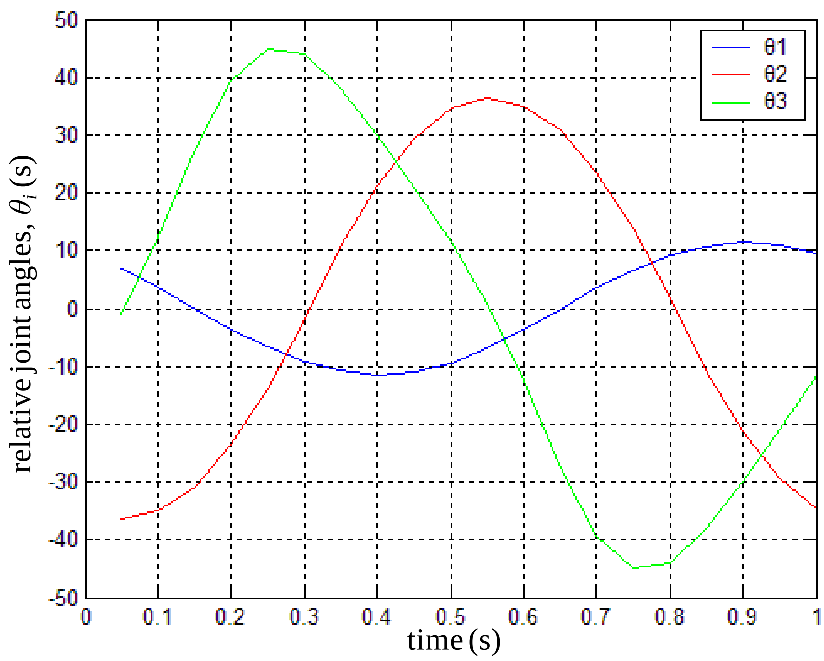


Fig. B.3 Joint angles in one cycle.

Next the joint angle can be decoupled from the position variables by using the time-dependant function given by Eq. (B.7). The values of A_i and $\varphi_{d,i}$ need to be found. It can be seen from Fig. B.3 that the Ai is the peak value of the joint angels and $\varphi_{d,i}$ is the phase difference of peaks of different joints. Therefore, the procedure of finding A_i then becomes a problem to find the maximum value of the corresponding joint angel, which can be defined by the following equation:

$$A_i = \max(\theta_i) = \max(\arctan\frac{y_i - y_{i-1}}{x_i - x_{i-1}} - \arctan\frac{y_{i-1} - y_{i-2}}{x_{i-1} - x_{i-2}}) \qquad \text{(B.8)}$$

The variables in the equation are subject to the following constrains:

$$(y_i - y_{i-1})^2 + (x_i - x_{i-1}) - r_i^2 = 0 \text{ (Joint rotation)}$$

$$y_i - f(x_i, t) = 0 \text{ (Body curve fitting)}$$

$$\theta_1 = \arctan\frac{y_1}{x_1}\sin(2\pi ft)$$

$$i = 1,2,...n \text{ and } 0 \le t < \frac{1}{f}$$

By further comparing the time of distinct links when the maximum angle value appears, the phase difference can be given by:

$$\varphi_{d,i} = \{\frac{1}{4f} - t[\max(\theta_i)]\} \times 2\pi f \qquad \text{(B.9)}$$

where $t[\max(\theta_i)]$ denotes the time as the i^{th} link reaches its maximum position with respect to previous link in one cycle.

Appendix C Specifications of Force Sensors

The cranks can be regarded as oscillating cantilever [183]. Strain gauges are used to measure the bending and therefore obtain the interaction force with water (see Fig. 7.5). Cranks are made of high strength aluminum alloy (its Young's Module is 70 GPa). Four strain gauges are mounted on the polished surface of each crank in form of Wheatstone bridge, as shown by Fig. C.1 and Fig. C.2.

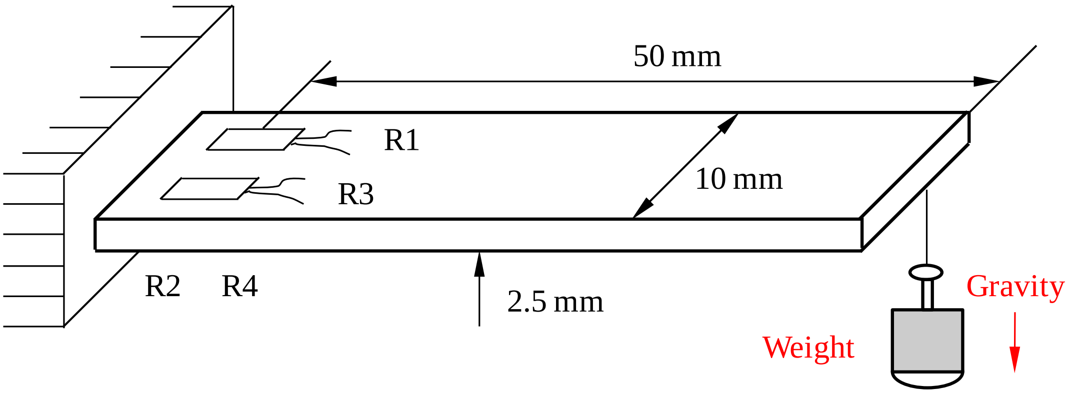


Fig. C.1 Geometries of a crank and the bonding of strain gauges.

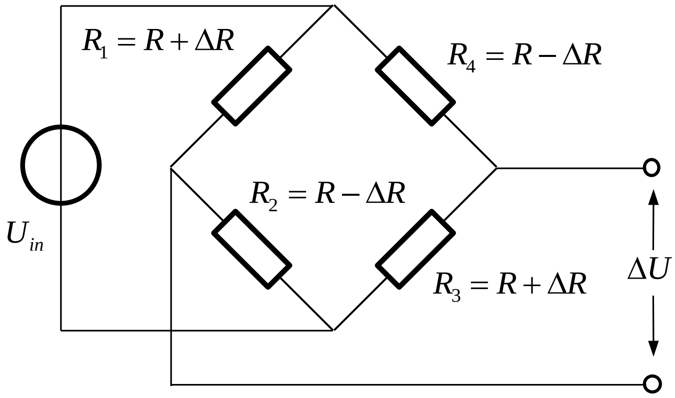


Fig. C.2 Connection of strain gauges (R1-R4) in form of Wheatstone bridge. A precision constant voltage source is applied to drive the circuit.

$$\Delta U = \frac{U}{4} \cdot \left(\frac{\Delta R_1}{R_1} - \frac{\Delta R_2}{R_2} - \frac{\Delta R_3}{R_4} - \frac{\Delta R_4}{R_4}\right) = \xi \varepsilon U_{in} \tag{C.1}$$

where U_{in} is the excitation voltage of the bridge, ΔU is the output of the Wheatstone bridge, the resistance of the strain gauges are R_1=$R2$=R_3=R_4=R, ε is the strain and ζ is the gauge factor. The specifications of the strain gauges are listed in Table C.1.

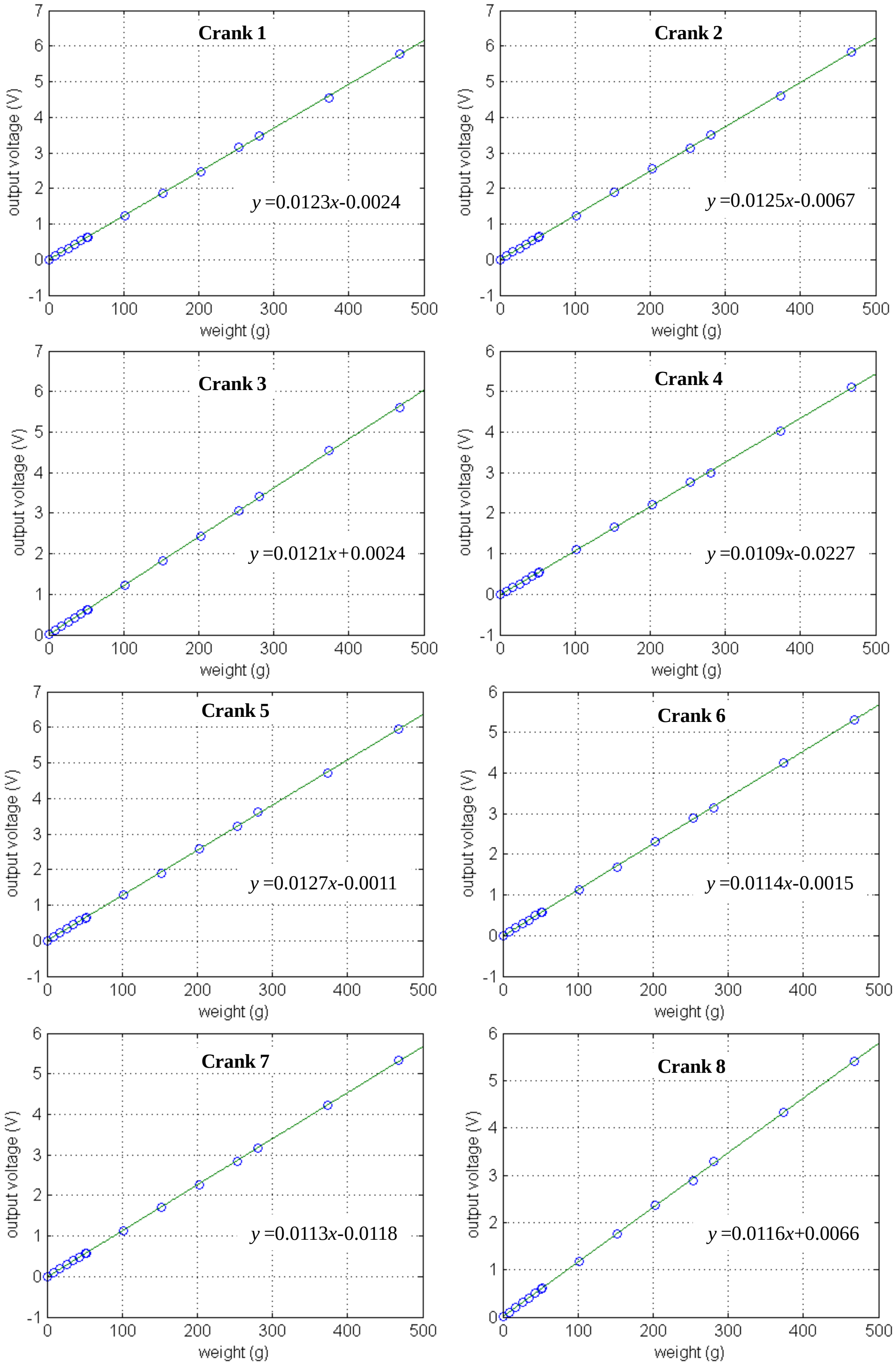


Fig. C.3 Output voltage of force sensors under different balance weights. Circle dots denote data recorded in calibration. Lines are linear fittings of experimental data. y denotes the voltage output and x is the mass of balance weights.

Table C.1 Specification of strain gauge.

Resistance, R	Gauge factor, ζ	Gauge size	Coefficient of thermal expansion
350 ± 1	$2.15 \pm 1\%$	5mm X 3mm	23.0 X $10^{-6}/^{o}$C

The theoretical output of the force sensor can be obtained by using Eq. (C.1). But the calculation may not be accurate enough owing to two major reasons: (i) the size of the elastomer (the aluminum alloy crank) can hardly be precise because of the imperfect machining; and (ii) because the mounting surface is manually polished by using sand paper and the strain gauges are manually bonded, the quality of all force sensors may not be uniform. Calibrations of those force sensors were made to find the relationship between the input force and the output readings. High precision balance weight is used to perform the static calibration for each force sensor (see Fig. C.1). The local gravity acceleration (in Singapore) is 9.7807 m/s^2. The linear calibration results are plotted in Fig. C.3. Linear regression is performed to find the output equation for each sensor. The average relative error is smaller than 0.5%. The According to the data plotted in Fig. C.3, the linearity of sensors can be calculated by

$$L_s = \frac{\Delta y_{\max}}{y_{F.S}} \times 100[F.S\%] < 0.1\% \tag{C.2}$$

where L_s is the linearity index, Δy_{max} is the maximum difference between the measured outputs of sensors and the value predicted by the regression equations given in Fig. C.3, and $y_{F.S}$ is the full scale reading of sensor output.

Appendix D Source Codes of GA

The following codes are implemented with Matlab™ 2008. It performs Genetic Algorithms used in Chapter 7. The codes are mainly revised from the GA toolbox developed by Sheffield University, Britain.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%file name: speed.m
%function: main program of GA in optimization of speed
%type: matlab *.m code
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

NIND = 2;       % number of individuals
MAXGEN = 3;   % maximum number of generations
NVAR = 10;      % dimension of the function
PRECI = 5;     % precision of variables
GGAP = 0.9;     % generation gap

% connect to FTP server
cRIO = ftp('155.69.135.151','Fisherman','fish');
ascii(cRIO);

ObjEx = zeros(NIND, 1);
dlmwrite('ObjEx.txt', ObjEx, 'precision', '%.1f','newline', 'pc');
mput(cRIO, 'ObjEx.txt');

% create Stat on local disk
Stat = 0;
dlmwrite('Stat.txt', Stat, 'precision', '%d','newline', 'pc');

% create ChromEx on local disk
% buidl field descriptor
FieldDR = [ rep([0;45],[1,NVAR-3]) [25;45] [1.0;2.5] [1.0;1.0]];

% innitialization
Chrom = CrtChrom4(NIND, FieldDR); % initial population
gen = 0;          % counter
dlmwrite('ChromEx.txt', Chrom, 'precision', '%.1f','newline', 'pc');
mput(cRIO, 'ChromEx.txt');

% obtain obj from FTP
obj = GetObj(cRIO,gen);
gen = gen+1;
record = [obj Chrom];
while gen<MAXGEN
    fit = ranking(-obj);    % caculate fitness
    SelCh = select('sus', Chrom, fit, GGAP);          % selection
    SelCh = recombin('CrossOver4', SelCh, 0.7);        % crossover
    SelCh = Mutation4(SelCh, FieldDR);                % mutation

    % re-evaluate the objective function
    dlmwrite('ChromEx.txt', SelCh, 'precision', '%.1f','newline',
'pc');
    mput(cRIO, 'ChromEx.txt');
```

```
    obj_sel = GetObj(cRIO, gen);
    [Chrom obj] = reins(Chrom, SelCh, 1, 1, obj, obj_sel);   % new
generation
    CurrentGen = [obj Chrom];
    record = [record; CurrentGen];
    gen = gen + 1;
    trace(gen,1) = max(obj);% tracing the performance
    trace(gen,2) = sum(obj)/length(obj);

    [BestObj BestIdx] = max(obj);
    BestGen(gen,:) = [Chrom(BestIdx,:) BestObj];
end

[OPT OptIdx] = max(BestGen(:,NVAR+1));
Solution = BestGen(OptIdx,:)
dlmwrite('record.txt', record, 'precision', '%.2f','newline', 'pc');

plot(trace(:,1)');
hold on;
plot(trace(:,2)', '-.');
grid;
legend('evolving of solution', 'evolving of population');

saveas(gcf, 'output', 'm');

% End of function
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%file name: CrtChrome.m
%function: create chromosome
%type: matlab *.m code
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Chrom = CrtChrom4(Nind,FieldDR);

% Check parameter consistency
   if nargin < 2, error('parameter FieldDR missing'); end
%    if nargin > 2, nargin = 2; end

 [mN, nN] = size(Nind);
   [mF, Npara] = size(FieldDR);

   if (mN ~= 1 & nN ~= 1), error('Nind has to be a scalar'); end
   if mF ~= 2, error('FieldDR must be a matrix with 2 rows'); end

% Compute Matrix with Range of variables and Matrix with Lower value
%    Range = FieldDR(2,:)-FieldDR(1,:);
%    Lower = FieldDR(1,:);
    Nvar = Npara-2;
    Chrom = zeros(Nind, Npara);
    Lower = FieldDR(1,:);
    Range = FieldDR(2,:)-FieldDR(1,:);
    for i=1:Nind
        Chrom(i,:) = rand(1,Npara).*Range+Lower;
    end
%   Chrom(:,Nvar) = rand(Nind,1)*Range(Nvar)+Lower(Nvar);
    amp_d = 8;
```

```
% Create initial population
% Each row contains one individual, the values of each variable
uniformly
% distributed between lower and upper bound (given by FieldDR)
   for i=1:Nind
       for j=Nvar-1:-1:1
          Higher(j) = Chrom(i,j+1);
          Range(j) = min( Higher(j)-Lower(j), amp_d);
          Chrom(i,j) = Higher(j)-rand()*Range(j);
       end
   end

% End of function
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%file name: GetObj.m
%function: Obtain objectives from FTP server in CompactRIO
%type: matlab *.m code
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function obj = GetObj(cRIO, gen)

% Check whether Stat=0
fprintf('Processing Generation %d \n',gen);
while dlmread('Stat.txt') == 0
    pause(1);
end
% clear Stat in local disk. then upload Stat to server
mget(cRIO, 'ObjEx.txt');
dlmwrite('Stat.txt', 0, 'precision', '%d','newline', 'pc');

% return obj downloaded from server
obj = dlmread('ObjEx.txt');
while isempty(obj)
    mget(cRIO, 'ObjEx.txt');
    pause(0.5);
    obj = dlmread('ObjEx.txt');
end
fprintf('Optimal Obj. in Generation %d is    %.3f\n',gen,
max(obj(:,1)));


% End of function
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%file name: rank.m
%function: evaluate fitness of individuals
%type: matlab *.m code
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%
% Input parameters:
% ObjV - Column vector containing the objective values of the
```

```
%                 individuals in the current population (cost values).
% RFun - (optional) If RFun is a scalar in [1, 2] linear ranking is
%         assumed and the scalar indicates the selective pressure.
%         If RFun is a 2 element vector:
%            RFun(1): SP - scalar indicating the selective pressure
%            RFun(2): RM - ranking method
%                      RM = 0: linear ranking
%                      RM = 1: non-linear ranking
% If RFun is a vector with length(Rfun) > 2 it contains the fitness
% to be assigned to each rank. It should have the same length as
% ObjV. Usually RFun is monotonously increasing. If RFun is omitted
% or NaN, linear ranking and a selective pressure of 2 are assumed.
%
% SUBPOP - (optional) Number of subpopulations
%            if omitted or NaN, 1 subpopulation is assumed
%
% Output parameters:
% FitnV     - Column vector containing the fitness values of the
%                 individuals in the current population.


function FitnV = ranking(ObjV, RFun, SUBPOP);

% Identify the vector size (Nind)
   [Nind,ans] = size(ObjV);

   if nargin < 2, RFun = []; end
   if nargin > 1, if isnan(RFun), RFun = []; end, end
   if prod(size(RFun)) == 2,
      if RFun(2) == 1, NonLin = 1;
      elseif RFun(2) == 0, NonLin = 0;
      else error('Parameter for ranking method must be 0 or 1'); end
      RFun = RFun(1);
      if isnan(RFun), RFun = 2; end
   elseif prod(size(RFun)) > 2,
      if prod(size(RFun)) ~= Nind, error('ObjV and RFun disagree');
end
   end

   if nargin < 3, SUBPOP = 1; end
   if nargin > 2,
      if isempty(SUBPOP), SUBPOP = 1;
      elseif isnan(SUBPOP), SUBPOP = 1;
      elseif length(SUBPOP) ~= 1, error('SUBPOP must be a scalar');
end
   end

   if (Nind/SUBPOP) ~= fix(Nind/SUBPOP), error('ObjV and SUBPOP
disagree'); end
% Compute number of individuals per subpopulation
   Nind = Nind/SUBPOP;
% Check ranking function and use default values if necessary
   if isempty(RFun),
% linear ranking with selective pressure 2
         RFun = 2*[0:Nind-1]'/(Nind-1);
   elseif prod(size(RFun)) == 1
      if NonLin == 1,
% non-linear ranking
```

```
         if RFun(1) < 1, error('Selective pressure must be greater
than 1');
         elseif RFun(1) > Nind-2, error('Selective pressure too
big'); end
         Root1 = roots([RFun(1)-Nind [RFun(1)*ones(1,Nind-1)]]);
         RFun = (abs(Root1(1)) * ones(Nind,1)) .^ [(0:Nind-1)'];
         RFun = RFun / sum(RFun) * Nind;
      else
         % linear ranking with SP between 1 and 2
         if (RFun(1) < 1 | RFun(1) > 2),
            error('Selective pressure for linear ranking must be
between 1 and 2');
         end
         RFun = 2-RFun + 2*(RFun-1)*[0:Nind-1]'/(Nind-1);
      end
   end;

   FitnV = [];

% loop over all subpopulations
for irun = 1:SUBPOP,
   % Copy objective values of actual subpopulation
      ObjVSub = ObjV((irun-1)*Nind+1:irun*Nind);
   % Sort does not handle NaN values as required. So, find those...
      NaNix = isnan(ObjVSub);
      Validix = find(~NaNix);
   % ... and sort only numeric values (smaller is better).
      [ans,ix] = sort(-ObjVSub(Validix));

   % Now build indexing vector assuming NaN are worse than numbers,
   % (including Inf!)...
      ix = [find(NaNix) ; Validix(ix)];
   % ... and obtain a sorted version of ObjV
      Sorted = ObjVSub(ix);

   % Assign fitness according to RFun.
      i = 1;
      FitnVSub = zeros(Nind,1);
      for j = [find(Sorted(1:Nind-1) ~= Sorted(2:Nind)); Nind]',
         FitnVSub(i:j) = sum(RFun(i:j)) * ones(j-i+1,1) / (j-i+1);
         i =j+1;
      end

   % Finally, return unsorted vector.
      [ans,uix] = sort(ix);
      FitnVSub = FitnVSub(uix);

   % Add FitnVSub to FitnV
      FitnV = [FitnV; FitnVSub];
end


% End of function
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%file name: select.m
%function: perform selection operation
```

```
%type: matlab *.m code
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function select(hndl)
ud = getscribeobjectdata(hndl.HGHandle);
MLObj = ud.ObjectStore;
select(MLObj);


% End of function
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%file name: recombin.m
%function: recombine pairs of individuals (crossover)
%type: matlab *.m code
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Syntax:  NewChrom = recombin(REC_F, OldChrom, RecOpt, SUBPOP)
%
% Input parameters:
% REC_F - String containing the name of the recombination or
%         crossover function
% Chrom - Matrix containing the chromosomes of the old
%         population. Each line corresponds to one individual
% RecOpt - (optional) Scalar containing the probability of
%          recombination/crossover occurring between pairs of
%          individuals. If omitted or NaN, 1 is assumed
% SUBPOP - (optional) Number of subpopulations
%          if omitted or NaN, 1 subpopulation is assumed
%
% Output parameter:
% NewChrom  - Matrix containing the chromosomes of the population
%             after recombination in the same format as OldChrom.

function NewChrom = recombin(REC_F, Chrom, RecOpt, SUBPOP);


% Check parameter consistency
   if nargin < 2, error('Not enough input parameter'); end

   % Identify the population size (Nind)
   [Nind,Nvar] = size(Chrom);

   if nargin < 4, SUBPOP = 1; end
   if nargin > 3,
      if isempty(SUBPOP), SUBPOP = 1;
      elseif isnan(SUBPOP), SUBPOP = 1;
      elseif length(SUBPOP) ~= 1, error('SUBPOP must be a scalar');
end
   end

   if (Nind/SUBPOP) ~= fix(Nind/SUBPOP), error('Chrom and SUBPOP
disagree'); end
   Nind = Nind/SUBPOP;  % Compute number of individuals per
subpopulation
```

```
   if nargin < 3, RecOpt = 0.7; end
   if nargin > 2,
      if isempty(RecOpt), RecOpt = 0.7;
      elseif isnan(RecOpt), RecOpt = 0.7;
      elseif length(RecOpt) ~= 1, error('RecOpt must be a scalar');
      elseif (RecOpt < 0 | RecOpt > 1), error('RecOpt must be a
scalar in [0, 1]'); end
   end


%Select individuals of one subpopulation and call low level function
   NewChrom = [];
   for irun = 1:SUBPOP,
      ChromSub = Chrom((irun-1)*Nind+1:irun*Nind,:);
      NewChromSub = feval(REC_F, ChromSub, RecOpt);
      NewChrom=[NewChrom; NewChromSub];
   end

% End of function
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%file name: mutation4.m
%function: perform real-value mutation operation
%type: matlab *.m code
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Syntax:  NewChrom = mutbga(OldChrom, FieldDR, MutOpt)
%
% Input parameter:
% OldChrom  - Matrix containing the chromosomes of the old
%              population. Each line corresponds to one individual.
% FieldDR   - Matrix describing the boundaries of each variable.
%  MutOpt    - Vector containing mutation rate and shrink value
%  MutOpt(1): MutR - number containing the mutation rate -
%  probability for mutation of a variable if omitted or NaN, MutR =
%  1/variables per individualis assumed MutOpt(2): MutShrink -
%  (optional) number for shrinking the mutation range in the range
%  [0 1], possibility to shrink the range of the mutation depending
%  on, for instance actual generation. If omitted or NaN, MutShrink
%  = 1 is assumed
%
% Output parameter:
% NewChrom - Matrix containing the chromosomes of the population
%                 after mutation in the same format as OldChrom.

function NewChrom = Mutation3(OldChrom, FieldDR, MutOpt);

% Check parameter consistency
   if nargin < 2,  error('Not enough input parameter'); end

   % Identify the population size (Nind) and the number of variables
(Nvar)
   [Nind,Nvar] = size(OldChrom);

   [mF, nF] = size(FieldDR);
   if mF ~= 2, error('FieldDR must be a matrix with 2 rows'); end
```

```
   if Nvar ~= nF, error('FieldDR and OldChrom disagree'); end

   if nargin < 3, MutR = 1/Nvar; MutShrink = 1;
   elseif isempty(MutOpt), MutR = 1/Nvar; MutShrink = 1;
   elseif isnan(MutOpt), MutR = 1/Nvar; MutShrink = 1;
   else
      if length(MutOpt) == 1, MutR = MutOpt; MutShrink = 1;
      elseif length(MutOpt) == 2, MutR = MutOpt(1); MutShrink =
MutOpt(2);
      else, error(' Too many parameter in MutOpt'); end
   end

   if isempty(MutR), MutR = 1/Nvar;
   elseif isnan(MutR), MutR = 1/Nvar;
   elseif length(MutR) ~= 1, error('Parameter for mutation rate must
be a scalar');
   elseif (MutR < 0 | MutR > 1), error('Parameter for mutation rate
must be a scalar in [0, 1]'); end

   if isempty(MutShrink), MutShrink = 1;
   elseif isnan(MutShrink), MutShrink = 1;
   elseif length(MutShrink) ~= 1, error('Parameter for shrinking
mutation range must be a scalar');
   elseif (MutShrink < 0 | MutShrink > 1),
      error('Parameter for shrinking mutation range must be a scalar
in [0, 1]');
   end

% Matrix with range values for every variable
   Range = rep(0.5 * MutShrink *(FieldDR(2,:)-FieldDR(1,:)),[Nind
1]);

% zeros and ones for mutate or not this variable, together with
Range
   Range = Range .* (rand(Nind,Nvar) < MutR);

% compute, if + or - sign
   Range = Range .* (1 - 2 * (rand(Nind,Nvar) < 0.5));

% used for later computing, here only ones computed
   ACCUR = 5;
   Vect = 2 .^ (-(0:(ACCUR-1))');
   Delta = (rand(Nind,ACCUR) < 1/ACCUR) * Vect;
   Delta = rep(Delta, [1 Nvar]);

% perform mutation
   variation = Range.*Delta;
   NewChrom(:,Nvar-2)=OldChrom(:,Nvar-2) + variation(:,Nvar-2);
   for i=1:Nind
       for j=Nvar-3:-1:1
           MutedCh = OldChrom(i,j)+variation(i,j);
           if MutedCh<= NewChrom(i,j+1) NewChrom(i,j) = MutedCh;
           elseif OldChrom(i,j)<=NewChrom(i,j+1) NewChrom(i,j) =
OldChrom(i,j);
           else NewChrom(i,j) = NewChrom(i,j+1);
           end
       end
```

```
   end
   NewChrom(:,Nvar-1) = OldChrom(:,Nvar-1) + variation(:,Nvar-1);
   NewChrom(:,Nvar) = OldChrom(:,Nvar) + variation(:,Nvar);

% Ensure variables boundaries, compare with lower and upper
boundaries
   NewChrom = max(rep(FieldDR(1,:),[Nind 1]), NewChrom);
   NewChrom = min(rep(FieldDR(2,:),[Nind 1]), NewChrom);

% End of function
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%file name: reins.m
%function: recombine pairs of individuals to produce new ones
%type: matlab *.m code
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Input parameters:
% Chrom - Matrix containing the individuals (parents) of the current
%              population. Each row corresponds to one individual.
% SelCh - Matrix containing the offspring of the current
%              population. Each row corresponds to one individual.
% SUBPOP - (optl) Number of subpopulations
%              if omitted or NaN, 1 subpopulation is assumed
% InsOpt - (optl) Vector containing the insertion method parameters
%          ExOpt(1): Select - number indicating kind of insertion
%          0 - uniform insertion
%          1 - fitness-based insertion
%          if omitted or NaN, 0 is assumed
%          ExOpt(2): INSR - Rate of offspring to be inserted per
%          subpopulation (% of subpopulation) if omitted or NaN, 1.0
%          (100%) is assumed
% ObjVCh - (optional) Column vector containing the objective values
%          of the individuals (parents - Chrom) in the current
%          population, needed for fitness-based insertion
%          saves recalculation of objective values for population
% ObjVSel - (optl) Column vector containing the objective values
% of the offspring (SelCh) in the current population, needed for
% partial insertion of offspring,saves recalculation of objective
% values for population
%
% Output parameters:
% Chrom - Matrix containing the individuals of the current
%         population after reinsertion.
% ObjVCh - if ObjVCh and ObjVSel are input parameter, than column
%         vector containing the objective values of the individuals
%         of the current generation after reinsertion.

% Check parameter consistency
   if nargin < 2, error('Not enough input parameter'); end
   if (nargout == 2 & nargin < 6), error('Input parameter missing:
ObjVCh and/or ObjVSel'); end

   [NindP, NvarP] = size(Chrom);
   [NindO, NvarO] = size(SelCh);
```

```
   if nargin == 2, SUBPOP = 1; end
   if nargin > 2,
      if isempty(SUBPOP), SUBPOP = 1;
      elseif isnan(SUBPOP), SUBPOP = 1;
      elseif length(SUBPOP) ~= 1, error('SUBPOP must be a scalar');
end
   end

   if (NindP/SUBPOP) ~= fix(NindP/SUBPOP), error('Chrom and SUBPOP
disagree'); end
   if (NindO/SUBPOP) ~= fix(NindO/SUBPOP), error('SelCh and SUBPOP
disagree'); end
   NIND = NindP/SUBPOP;  % Compute number of individuals per
subpopulation
   NSEL = NindO/SUBPOP;  % Compute number of offspring per
subpopulation

   IsObjVCh = 0; IsObjVSel = 0;
   if nargin > 4,
      [mO, nO] = size(ObjVCh);
      if nO ~= 1, error('ObjVCh must be a column vector'); end
      if NindP ~= mO, error('Chrom and ObjVCh disagree'); end
      IsObjVCh = 1;
   end
   if nargin > 5,
      [mO, nO] = size(ObjVSel);
      if nO ~= 1, error('ObjVSel must be a column vector'); end
      if NindO ~= mO, error('SelCh and ObjVSel disagree'); end
      IsObjVSel = 1;
   end

   if nargin < 4, INSR = 1.0; Select = 0; end
   if nargin >= 4,
      if isempty(InsOpt), INSR = 1.0; Select = 0;
      elseif isnan(InsOpt), INSR = 1.0; Select = 0;
      else
         INSR = NaN; Select = NaN;
         if (length(InsOpt) > 2), error('Parameter InsOpt too long');
end
         if (length(InsOpt) >= 1), Select = InsOpt(1); end
         if (length(InsOpt) >= 2), INSR = InsOpt(2); end
         if isnan(Select), Select = 0; end
         if isnan(INSR), INSR =1.0; end
      end
   end

   if (INSR < 0 | INSR > 1), error('Parameter for insertion rate
must be a scalar in [0, 1]'); end
   if (INSR < 1 & IsObjVSel ~= 1), error('For selection of offspring
ObjVSel is needed'); end
   if (Select ~= 0 & Select ~= 1), error('Parameter for selection
method must be 0 or 1'); end
   if (Select == 1 & IsObjVCh == 0), error('ObjVCh for fitness-based
exchange needed'); end

   if INSR == 0, return; end
```

```
   NIns = min(max(floor(INSR*NSEL+.5),1),NIND);    % Number of
offspring to insert

% perform insertion for each subpopulation
   for irun = 1:SUBPOP,
      % Calculate positions in old subpopulation, where offspring
are inserted
         if Select == 1,     % fitness-based reinsertion
            [Dummy, ChIx] = sort(-ObjVCh((irun-1)*NIND+1:irun*NIND));
         else                % uniform reinsertion
            [Dummy, ChIx] = sort(rand(NIND,1));
         end
         PopIx = ChIx((1:NIns)')+ (irun-1)*NIND;
      % Calculate position of Nins-% best offspring
         if (NIns < NSEL),   % select best offspring
            [Dummy,OffIx] = sort(ObjVSel((irun-1)*NSEL+1:irun*NSEL));
         else
            OffIx = (1:NIns)';
         end
         SelIx = OffIx((1:NIns)')+(irun-1)*NSEL;
      % Insert offspring in subpopulation -> new subpopulation
         Chrom(PopIx,:) = SelCh(SelIx,:);
         if (IsObjVCh == 1 & IsObjVSel == 1), ObjVCh(PopIx) =
ObjVSel(SelIx); end
   end

% End of function
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%file name: power1.m
%function: main program for optimization of energy efficiency
%type: matlab *.m code
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
NIND = 10;         % number of individuals
MAXGEN = 60;     % maximum number of generations
NVAR = 10;         % dimension of the function
PRECI = 5;       % precision of variables
GGAP = 0.9;        % generation gap

% connect to FTP server
cRIO = ftp('155.69.135.151','Fisherman','fish');
ascii(cRIO);

ObjEx = zeros(NIND, 1);
dlmwrite('ObjEx.txt', ObjEx, 'precision', '%.2f','newline', 'pc');
mput(cRIO, 'ObjEx.txt');

% create Stat on local disk
Stat = 0;
dlmwrite('Stat.txt', Stat, 'precision', '%d','newline', 'pc');
% create ChromEx on local disk
% buidl field descriptor
FieldDR = [ rep([0;40],[1,NVAR-3]) [25;40] [0.8;2.5] [1.0;1.0]];

% innitialization
```

```
Chrom = CrtChrom4(NIND, FieldDR); % initial population
gen = 0;          % counter
dlmwrite('ChromEx.txt', Chrom, 'precision', '%.2f','newline', 'pc');
mput(cRIO, 'ChromEx.txt');

% obtain obj from FTP
data = GetObj(cRIO,gen);
obj = data(:,1);
print_obj = obj'
gen = gen+1;
ChRec = [obj Chrom];
PowerRec = data;

while gen<MAXGEN
    fit = ranking(-round(obj));    % caculate fitness
    SelCh = select('sus', Chrom, fit, GGAP);            % selection
    SelCh = recombin('CrossOver4', SelCh, 0.7);         % crossover
    SelCh = Mutation4(SelCh, FieldDR);                  % mutation

    % re-evaluate the objective function
    dlmwrite('ChromEx.txt', SelCh, 'precision', '%.2f','newline',
'pc');
    mput(cRIO, 'ChromEx.txt');
    data_sel = GetObj(cRIO, gen);
    obj_sel = data_sel(:,1);

    tmp1 = obj;
    [Chrom obj] = reins(Chrom, SelCh, 1, 1, tmp1, obj_sel);   % new
generation
    ChRec = [ChRec; [obj Chrom]];
    [data tmp2] = reins(data, data_sel, 1, 1, tmp1, obj_sel);   %
new generation
    PowerRec = [PowerRec; data];

    gen = gen + 1;
    trace(gen,1) = max(obj);% tracing the performance
    trace(gen,2) = sum(obj)/length(obj);

    [BestObj BestIdx] = max(obj);
    BestGen(gen,:) = [Chrom(BestIdx,:) BestObj];
end

[OPT OptIdx] = max(BestGen(:,NVAR+1));
Solution = BestGen(OptIdx,:)
dlmwrite('ChRec.txt', ChRec, 'precision', '%.2f','newline', 'pc');

plot(trace(:,1)');
hold on;
plot(trace(:,2)', '-.');
grid;
legend('evolving of solution', 'evolving of population');
saveas(gcf, 'output', 'm');
% End of function
```

List of Publications

Journal articles:

[1] C. Zhou and K. H. Low, "Design and Locomotion Control of a Biomimetic Underwater Vehicle with Fin Propulsion," *IEEE/ASME Transactions on Mechatronics*, vol. 17, issue 1, 2011, in press.

[2] C. Zhou and K. H. Low, "Better Endurance and Load Capacity: An Improved Design of Manta Ray Robot (RoMan-II)," *Journal of Bionic Engineering,* vol. 7, pp. S137-S144, 2010.

[3] K. H. Low, C. Zhou, and Y. Zhong, "Gait Planning for Steady Swimming Control of Biomimetic Fish Robots," *Advanced Robotics,* Vol. 23, No. 7-8, pp. 805-829, 2009.

[4] K. H. Low, C. Zhou, G. G. L. Seet, J. Yu, "Learning from Gymnotiform Swimmers – Design and Implementation of Robotic Knifefish NKF-II," *International Journal of Information Acquisition,* Vol. 5, No. 2, pp. 137-147, 2008.

Conference articles:

[1] C. Zhou and K. H. Low, "Optimization of Swimming Locomotion for Fish Robots with Multi-actuation," in *Proceedings* of *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Phuket Island, Thailand, 2011, pp. 2120-2125.

[2] K. H. Low, G. Seet, C. Zhou, S. Bi, Y. Cai, "Improvement and Testing of a Robotic Manta Ray (RoMan-III)," in *Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics (ROBIO),* Phuket Island, Thailand, 2011, pp. 1730-1735.

[3] C. Zhou and K. H. Low, "Modeling and Control of Flapping Motion for Robotic Manta Ray via Central Pattern Generators," presented at *the 1st International Conference on Applied Bionics and Biomechanics (ICABB),* Venice, Italy, 2010.

[4] C. Zhou and K. H. Low, “Study and Implementation of Station-holding Performance on a Fish Robot in Adverse Unsteady Flow,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 2010, pp. 928-933.

[5] C. Zhou and K. H. Low, “Kinematic modeling framework for biomimetic undulatory fin motion based on coupled nonlinear oscillators,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 2010, pp. 934-939.

[6] C. Zhou, K. H. Low and C. W. Chong, “An Analytical Approach for Better Swimming Efficiency of Slender Fish Robots Based on Lighthill’s Model,” in *The 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Guilin, China, 2009, pp. 1651-1656.

[7] C. Zhou and K. H. Low, “Locomotion Planning for Biomimetic Robotic Fish with Multi-Joint Actuation,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS*), St. Louis, USA, 2009, pp. 2132-2137.

[8] C. Zhou, C. W. Chong, Y. Zhong, and K. H. Low, "Robust Gait Control for Steady Swimming of a Carangiform Fish Robot," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Singapore, 2009, pp. 100-105.

[9] C. W. Chong, Y. Zhong, C. Zhou, and K. H. Low, "Can the Swimming Thrust of BCF Biomimetics Fish Be Enhanced?" in *The 2008 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Bangkok, Thailand, 2008, pp. 437-442.

[10] K. H. Low, C. Zhou, T. W. Ong, J. Yu, “Modular Design and Initial Gait Study of an Amphibian robotic turtle,” in the 2007 *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Sanya, China, 2007, pp. 535-540.

References

[1] A. Azuma, *The Biokinetics of Flying and Swimming, Second Edition*. Reston: American Institute of Aeronautics and Astronautics, Inc., 2006.

[2] G. V. Lauder, E. J. Anderson, J. Tangorra, and P. G. A. Madden, "Fish biorobotics: Kinematics and hydrodynamics of self-propulsion," *Journal of Experimental Biology,* vol. 210, pp. 2767-2780, 2007.

[3] L. Lionel, *Underwater Robots Part I: current systems and problem pose.*: Pro Literatur Verlag, Germany / ARS, Austria, 2006.

[4] P. R. Bandyopadhyay, "Trends in biorobotic autonomous undersea vehicles," *IEEE Journal of Oceanic Engineering,* vol. 30, pp. 109-139, 2005.

[5] L. Lionel, *Underwater Robots Part II: Existing Solutions and Open Issues*: Pro Literatur Verlag, Germany / ARS, Austria, 2006.

[6] M. S. Triantafyllou and G. S. Triantafyllou, "An efficient swimming machine," *Scientific American,* vol. 272, pp. 64-70, 1995.

[7] K. Mohseni, "Pulsatile vortex generators for low-speed maneuvering of small underwater vehicles," *Ocean Engineering,* vol. 33, pp. 2209-2223, 2006.

[8] M. Sfakiotakis, D. M. Lane, and J. B. C. Davies, "Review of fish swimming modes for aquatic locomotion," *IEEE Journal of Oceanic Engineering,* vol. 24, pp. 237-252, 1999.

[9] T. R. Asplund, *The Effects of Motorized Watercraft on Aquatic Ecosystems*, 2000.

[10] F. E. Fish and J. J. Rohr, "Review of dolphin hydrodynamics and swimming performance," *SPAWARS System Center Technical Report 1801,* 1999.

[11] A. J. Ijspeert, A. Crespi, D. Ryczko, and J. M. Cabelguen, "From swimming to walking with a salamander robot driven by a spinal cord model," *Science,* vol. 315, pp. 1416-1420, 2007.

[12] K. H. Low, C. Zhou, and Y. Zhong, "Gait planning for steady swimming control of biomimetic fish robots," *Advanced Robotics,* vol. 23, pp. 805-829, 2009.

[13] N. Kato, Y. Ando, T. Shigetomi, and T. Katayama, "Biology-inspired precision maneuvering of underwater vehicles (part 4)," *International Journal of Offshore and Polar Engineering,* vol. 16, pp. 195-201, 2006.

[14] R. L. McLaughlin and D. L. Noakes, "Going against the flow: an examination of the propulsive movements made by young brook trout in streams," *Can. J. Fish. Aquat. Sci.,* pp. 853-860, 1998.

[15] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: A review," *Neural Networks,* vol. 21, pp. 642-653, 2008.

[16] S. Kern and P. Koumoutsakos, "Simulations of optimized anguilliform swimming," *Journal of Experimental Biology,* vol. 209, pp. 4841-4857, 2006.

[17] F. E. Fish and G. V. Lauder, "Passive and active flow control by swimming fishes and mammals," in *Annual Review of Fluid Mechanics* vol. 38, ed, 2006, pp. 193-224.

[18] G. V. Lauder and E. D. Tytell, "Hydrodynamics of undulatory propulsion," *Fish Biomechanics,* vol. 23, pp. 425-468, 2006.

[19] X. Li, Y. Wu, X. Lu, and X. Yin, "Measurements of fish's wake by PIV," *Optical Technology and Image Processing for Fluids and Solids Diagnostics 2002,* vol. 5058, pp. 139-145, 2003.

[20] E. G. Drucker and G. V. Lauder, "Locomotor forces on a swimming fish: Three-dimensional vortex wake dynamics quantified using digital particle image velocimetry," *Journal of Experimental Biology,* vol. 202, pp. 2393-2412, 1999.

[21] E. D. Tytell and G. V. Lauder, "The hydrodynamics of eel swimming: I. Wake structure," *Journal of Experimental Biology,* vol. 207, pp. 1825-1841, 2004.

[22] S. Grillner, "Biological Pattern Generation: The Cellular and Computational Logic of Networks in Motion," *Neuron,* vol. 52, pp. 751-766, 2006.

[23] P. Arena, "A mechatronic lamprey controlled by analog circuits," presented at the the 9th IEEE Mediterranean Conference on Control and Automation (MED'01), Dubrovnick, Croatia, 2001.

[24] R. Yuste, J. N. MacLean, J. Smith, and A. Lansner, "The cortex as a central pattern generator," in *Nature Reviews Neuroscience* vol. 6, ed, 2005, pp. 477-483.

[25] S. Grillner, P. Wallen, L. Brodin, and A. Lansner, "Neuronal network generating locomotor behavior in lamprey: Circuitry, transmitters, membrane properties, and simulation," *Annual Review of Neuroscience,* vol. 14, pp. 169-199, 1991.

[26] D. Zhang, D. Hu, L. Shen, and H. Xie, "Design of an artificial bionic neural network to control fish-robot's locomotion," *Neurocomputing,* vol. 71, pp. 648-654, 2008.

[27] Z. Lu, S. Ma, B. Li, and Y. Wang, "3D locomotion of a snake-like robot controlled by cyclic inhibitory CPG model," in *IEEE International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 3897-3902.

[28] A. Crespi, D. Lachat, A. Pasquier, and A. J. Ijspeert, "Controlling swimming and crawling in a fish robot using a central pattern generator," *Autonomous Robots,* vol. 25, pp. 3-13, 2008.

[29] J. Buchli, F. Iida, and A. J. Ijspeert, "Finding resonance: Adaptive frequency oscillators for dynamic legged locomotion," in *IEEE International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 3903-3909.

[30] C. Zhou, C. W. Chong, Y. Zhong, and K. H. Low, "Robust Gait Control for Steady Swimming of a Carangiform Fish Robot " in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Singapore, 2009, pp. 100 - 105.

[31] C. M. Breder, "The locomotion of fishes," *Zoologica,* vol. 4, pp. 159-297, 1926.

[32] C. C. Lindsey, "Form, function, and locomotory habits in fish," *Fish Physiology,* vol. 7, pp. 1-100, 1978.

[33] P. W. Webb, "Form and function in fish swimming," *Scientific American,* vol. 251, pp. 58-68, 1984.

[34] P. W. Webb, "Body form, locomotion and foraging in aquatic vertebrates," *American Zoologist,* vol. 24, pp. 107-120, 1984.

[35] P. R. Last and J. D. Stevens, *Sharks and rays of Australia (Second Edition)*: CSIRO Publishing, 2009.

[36] T. P. L. Brower, "Design of a manta ray inspired underwater propulsive mechanism for long range, low power operation," Master Thesis in the Department of Mechanical Engineering, Tufts University, Neatherlands, 2006.

[37] L. J. Rosenberger, "Pectoral fin locomotion in batoid fishes: Undulation versus oscillation," *Journal of Experimental Biology*, vol. 204, pp. 379-394, 2001.

[38] W. Klausewitz, "Der lokomotionsmodus der flugelrochen (*myliobatoidei)*," *Natur-Museum und Forschungsinstitute Senckenberg,* pp. 111-117, 1963.

[39] K. D. von Ellenrieder, K. Parker, and J. Soria, "Fluid mechanics of flapping wings," *Experimental Thermal and Fluid Science,* vol. 32, pp. 1578-1589, 2008.

[40] W. W. Schultz and P. W. Webb, "Power requirements of swimming: Do new methods resolve old questions?," *Integrative and Comparative Biology,* vol. 42, pp. 1018-1025, 2002.

[41] M. S. Triantafyllou, G. S. Triantafyllou, and D. K. P. Yue, "Hydrodynamics of fishlike swimming," *Annual Review of Fluid Mechanics,* vol. 32, pp. 33-53, 2000.

[42] M. S. Triantafyllou, F. S. Hover, A. H. Techet, and D. K. P. Yue, "Review of Hydrodynamic Scaling Laws in Aquatic Locomotion and Fishlike Swimming," *Applied Mechanics Reviews,* vol. 58, pp. 226-237, 2005.

[43] Q. Zhu, M. J. Wolfgang, D. K. P. Yue, and M. S. Triantafyllou, "Three-dimensional flow structures and vorticity control in fish-like swimming," *Journal of Fluid Mechanics*, vol. 468, pp. 1-28, 2002.

[44] S. D. Kelly, R. J. Mason, C. T. Anhalt, R. M. Murray, and J. W. Burdick, "Modeling and experimental investigation of carangiform locomotion for control," in *Proceedings of the 1998 American Control Conference,* Philadelphia, USA, 1998, pp. 1271-1276.

[45] R. Bhaskaran and L. Collins. *Introduction to CFD Basics*.

[46] M. Bozkurttas, H. Dong, R. Mittal, J. Tangorra, I. Hunter, G. V. Lauder, and P. Madden, "CFD based analysis and design of biomimetic flexible propulsors for autonomous underwater vehicles," in *Collection of Technical Papers - 37th AIAA Fluid Dynamics Conference,* Miami, USA, 2007, pp. 1482-1489.

[47] S. N. Singh, A. Simha, and R. Mittal, "Biorobotic AUV maneuvering by pectoral fins: Inverse control design based on CFD parameterization," *IEEE Journal of Oceanic Engineering,* vol. 29, pp. 777-785, 2004.

[48] S. Sarkar and K. Venkatraman, "Numerical simulation of thrust generating flow past a pitching airfoil," *Computers and Fluids,* vol. 35, pp. 16-42, 2006.

[49] M. Bozkurttas, H. Dong, R. Mittal, P. Madden, and G. V. Lauder, "Hydrodynamic performance of deformable fish fins and flapping foils," in *Collection of Technical Papers - 44th AIAA Aerospace Sciences Meeting*, 2006, pp. 16743-16753.

[50] E. J. Stamhuis and J. J. Videler, "Quantitative flow analysis around aquatic animals using laser sheet particle image velocimetry," *Journal of Experimental Biology,* vol. 198, pp. 283-294, 1995.

[51] J. M. Anderson, "Vorticity control for efficient propulsion," Department of Ocean Engineering, Ph.D Dissertation, MIT, 1996.

[52] E. G. Drucker and G. V. Lauder, "Wake dynamics and fluid forces of turning maneuvers in sunfish," *Journal of Experimental Biology,* vol. 204, pp. 431-442, 2001.

[53] M. Sfakiotakis, D. M. Lane, and B. C. Davies, "An experimental undulating-fin device using the Parallel Bellows Actuator," in *Proceedings - IEEE International Conference on Robotics and Automation (ICRA)*, Seoul, South Korea, 2001, pp. 2356-2362.

[54] Y. Toda, N. Sogihara, Y. Sanada, and M. Danno, "The motion of a fish-like under-water vehicle with two undulating side fins," in *Proc. of the 3rd Int. Sym. on Aero Aqua Bio-mechanisms (ISABMEC 2006)*, Okinawa, Japan, 2006.

[55] M. A. MacIver, E. Fontaine, and J. W. Burdick, "Designing future underwater vehicles: Principles and mechanisms of the weakly electric fish," *IEEE Journal of Oceanic Engineering,* vol. 29, pp. 651-659, 2004.

[56] M. Epstein, J. E. Colgate, and M. A. Maciver, "Generating thrust with a biologically-inspired robotic ribbon fin," in *IEEE International Conference on Intelligent Robots and Systems*, Beijing, 2006, pp. 2412-2417.

[57] K. H. Low, G. L. G. Seet, and C. Zhou, "Biomimetic design and workspace study of compact and modular undulating fin body segments," in *Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation (ICMA)*, Harbin, China, 2007, pp. 129-134.

[58] K. H. Low, C. Zhou, G. G. L. Seet, and J. YU, "Learning from gymnotiform swimmers–design and implementation of robotic knifefish NKF-II," *International Journal of Information Acquisition,* vol. 5, pp. 137-147, 2008.

[59] Y. Toda, K. Fukui, S. Uto, and N. Tanaka, "Fundamental study on swimming of a fish-like body with two undulating side fins," presented at the Second Int. Symp. on Aqua Bio-mechanism, Japan, 2003.

[60] K. H. Low and A. Willy, "Biomimetic motion planning of an undulating robotic fish fin," *Journal of Vibration and Control,* vol. 12, pp. 1337-1359, 2006.

[61] Z. Wang, Y. Wang, J. Li, and G. Hang, "A micro biomimetic manta ray robot fish actuated by SMA," in *the 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Guilin, China, 2009, pp. 1809-1813.

[62] C. Zhou and K. H. Low, "Locomotion planning of biomimetic robotic fish with multi-joint actuation," in *the 2009 IEEE/RSJ International Conference onIntelligent Robots and Systems (IROS)*, St. Louis, USA, 2009, pp. 2132-2137.

[63] Y. Xu, G. Zong, S. Bi, and J. Gao, "Initial development of a flapping propelled unmanned underwater vehicle (UUV)," in *the 2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Sanya, China, 2007, pp. 524-529.

[64] J. Gao, S. Bi, Y. Xu, and C. Liu, "Development and design of a robotic manta ray featuring flexible pectoral fins," in *the 2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Sanya, China, 2007, pp. 519-523.

[65] P. Valdivia y Alvarado, S. Chin, R. W. Larson, A. Mazumdar, and K. Youcef-Toumi, "A soft body under-actuated approach to multi degree of freedom biomimetic robots: A stingray example," presented at the the 3rd IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics, Tokyo, Japan, 2010.

[66] M. Nagai, *Thinking Fluid Dynamics with Dolphins*. Tokyo: Ohmsha, Ltd., 2002.

[67] http://web.mit.edu/towtank/www/Pike/photos.html.

[68] http://web.mit.edu/towtank/www/Tuna/Tuna1/tuna1.html.

[69] J. M. Anderson, M. S. Triantafyllou, and P. A. Kerrebrock, "Concept design of a flexible-hull unmanned undersea vehicle," in *Proceedings of the International Offshore and Polar Engineering Conference*, Hawaii, USA, 1997, pp. 82-88.

[70] J. M. Anderson and P. A. Kerrebrock, "The vorticity control unmanned undersea vehicle (VCUUV) - An autonomous vehicle employing fish swimming propulsion and maneuvering," in *Proc. 10th Int. Symp. Unmanned Untethered Submersible Technology*, New Hampshire, USA, 1997, pp. 189-195.

[71] H. Hu, J. Liu, I. Dukes, and G. Francis, "Design of 3D swim patterns for autonomous robotic fish," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 2406-2411.

[72] J. Liu and H. Hu, "Biological Inspiration: From Carangiform Fish to Multi-Joint Robotic Fish," *Journal of Bionic Engineering*, vol. 7, pp. 35-48.

[73] J. Liu and H. Hu, "A methodology of modelling fish-like swim patterns for robotic fish," in *Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, ICMA 2007*, 2007, pp. 1316-1321.

[74] J. Ayers, C. Wilbur, and C. Olcott, "Lamprey robots," in *Proc. Int. Symp. Aqua Biomechanisms*, 2000, pp. 1-6.

[75] G. Dogangil, E. Ozcicek, and A. Kuzucu, "Modeling, simulation, and development of a robotic dolphin prototype," in *the 2005 IEEE International Conference on Mechatronics and Automation (ICMA)*, Niagara Falls, Canada, 2005, pp. 952-957.

[76] Y. Hu, W. Zhao, and L. Wang, "Vision-based target tracking and collision avoidance for two autonomous robotic fish," *IEEE Transactions on Industrial Electronics*, vol. 56, pp. 1401-1410, 2009.

[77] D. Xia, W. Chen, and J. Liu, "Using biological spring to improve propulsive efficiency for fishlike robot," in *Proceedings of 2008 IEEE International Conference on Mechatronics and Automation (ICMA)*, Kagawa, Japan, 2008, pp. 137-142.

[78] K. A. McIsaac and J. P. Ostrowski, "Motion planning for anguilliform locomotion," *IEEE Transactions on Robotics and Automation*, vol. 19, pp. 637-652, 2003.

[79] K. A. McIsaac and J. P. Ostrowski, "A geometric approach to anguilliform locomotion: Simulation and experiments with an underwater eel robot," *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 1, pp. 2843-2848, 1999.

[80] K. H. Low, S. Prabu, and A. P. Pattathil, "Initial prototype design and development of hybrid modular underwater vehicles," in *2006 IEEE International Conference on Robotics and Biomimetics, ROBIO 2006*, Kunming, 2006, pp. 311-316.

[81] C. W. Chong, Y. Zhong, C. Zhou, and K. H. Low, "Can the swimming thrust of BCF biomimetic fish be enhanced?," presented at the the 2008 IEEE International Conference on Robotics and Biomimetics (ROBIO), Bangkok, Thailand, 2008.

[82] C. Zhou, K. H. Low, and C. W. Chong, "An Analytical Approach for Better Swimming Efficiency of Slender Fish Robots Based on Lighthill's Model," in *the 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Guilin, China, 2009, pp. 1651-1656.

[83] Y. Zhong, D. Zhang, C. Zhou, C. W. Chong, T. Hu, S. L., and K. H. Low, "Better endurance and load capacity: an underwater vehicle inspired by manta ray," presented at the the Fourth International Symposium on Aero Aqua Bio-Mechanisms, Shanghai, China, 2009.

[84] C. Zhou and K. H. Low, "Better Endurance and Load Capacity: An Improved Design of Manta Ray Robot (RoMan-II)," *Journal of Bionic Engineering,* vol. 7, pp. S137-S144, 2010.

[85] F. Liu, K. M. Lee, and C. J. Yang, "Hydrodynamics of an Undulating Fin for a Wave-Like Locomotion System Design," *IEEE/ASME Transactions on Mechatronics,* to be published.

[86] C. Zhou and K. H. Low, "Better Endurance and Load Capacity: An Improved Design of Manta Ray Robot (RoMan-II)," *Journal of Bionic Engineering,* vol. 7, pp. S137-S144, 2010.

[87] Y. Zhong, D. Zhang, C. Zhou, C. W. Chong, T. Hu, L. Shen, and K. H. Low, "Better Endurance and Load Capacity: An underwater Vehicle Inspired by Manta Ray," presented at the The Fourth International Symposium on Aero Aqua Bio-Mechanisms, ISABMEC, Shanghai, China, 2009.

[88] S. Guo, T. Fukuda, and K. Asaka, "A new type of fish-like underwater microrobot," *IEEE/ASME Transactions on Mechatronics,* vol. 8, pp. 136-141, 2003.

[89] C. Zheng, S. Shatara, and T. Xiaobo, "Modeling of Biomimetic Robotic Fish Propelled by An Ionic Polymer-Metal Composite Caudal Fin," *Mechatronics, IEEE/ASME Transactions on,* vol. 15, pp. 448-459, 2010.

[90] K. A. Harper, M. D. Berkemeier, and S. Grace, "Modeling the dynamics of spring-driven oscillating-foil propulsion," *IEEE Journal of Oceanic Engineering,* vol. 23, pp. 285-296, 1998.

[91] Y. Fu and W. G. Price, "Interactions between a partially or totally immersed vibrating cantilever plate and the surrounding fluid," *Journal of Sound and Vibration,* vol. 118, pp. 495-513, 1987.

[92] C.-C. Liang, C.-C. Liao, Y.-S. Tai, and W.-H. Lai, "The free vibration analysis of submerged cantilever plates," *Ocean Engineering,* vol. 28, pp. 1225-1245, 2001.

[93] K. H. Low, "Parametric Study of Modular and Reconfigurable Robotic Fish with Oscillating Caudal Fin Mechanisms," in *the 2007 International Conference on Mechatronics and Automation*, Harbin, China, 2007, pp. 123-128.

[94] K. H. Low, C. W. Chong, and C. Zhou, "Performance Study of a Fish Robot Propelled by a Flexible Caudal Fin," in *2010 IEEE International Conference on Robots and Automation (ICRA),* Alaska, USA, 2010, pp. 90-95.

[95] K. H. Low and C. W. Chong, "Parametric study of the swimming performance of a fish robot propelled by a flexible caudal fin," *Bioinspiration & Biomimetics,* vol. 5, p. 046002, 2010.

[96] J. J. Magnuson, "Locomotion by scombrid fishes: Hydromechanics, morphology and behavior," *Fish Physiology,* vol. 7, pp. 239-313, 1978.

[97] E. D. Tytell, "The hydrodynamics of eel swimming II. Effect of swimming speed," *Journal of Experimental Biology,* vol. 207, pp. 3265-3279, 2004.

[98] J. Pratt, C. M. Chew, A. Torres, P. Dilworth, and G. Pratt, "Virtual model control: an intuitive approach for bipedal locomotion," *International Journal of Robotics Research,* vol. 20, pp. 129-143, 2001.

[99] K. Singh and T. J. Pedley, "The hydrodynamics of flexible-body maneuvers in swimming fish," *Physica D: Nonlinear Phenomena,* vol. 237, pp. 2234-2239, 2008.

[100] J. E. Colgate and K. M. Lynch, "Mechanics and control of swimming: A review," *IEEE Journal of Oceanic Engineering,* vol. 29, pp. 660-673, 2004.

[101] M. J. Lighthill, "Note on the swimming of slender fish," *J. Fluid Mech.,* vol. 9, pp. 305-317, 1960.

[102] J.-D. Liu and H. Hu, "Biologically inspired behaviour design for autonomous robotic fish," *International Journal of Automation and Computing,* vol. 3, pp. 336-347, 2006.

[103] J. Yu, L. Liu, L. Wang, M. Tan, and D. Xu, "Turning control of a multilink biomimetic robotic fish," *IEEE Transactions on Robotics,* vol. 24, pp. 201-206, 2008.

[104] Z. Chen, S. Shatara, and X. Tan, "Modeling of biomimetic robotic fish propelled by an ionic polymermetal composite caudal fin," *IEEE/ASME Transactions on Mechatronics,* vol. 15, pp. 448-459, 2010.

[105] X. Tan, M. Carpenter, J. Thon, and F. Alequin-Ramos, "Analytical modeling and experimental studies of robotic fish turning," in *Proceedings - IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, USA, 2010, pp. 102-108.

[106] M. Anton, Z. Chen, M. Kruusmaa, and X. Tan, "Analytical and computational modeling of robotic fish propelled by soft actuation material-based active joints," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, 2009, pp. 2126-2131.

[107] H. Hu, J. Liu, I. Dukes, and G. Francis, "Design of 3D swim patterns for autonomous robotic fish," in *IEEE International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 2406-2411.

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

[109] T. Hu, L. Shen, L. Lin, and H. Xu, "Biological inspirations, kinematics modeling, mechanism design and experiments on an undulating robotic fin inspired by Gymnarchus niloticus," *Mechanism and Machine Theory,* vol. 44, pp. 633-645, 2009.

[110] G. S. Triantafyllou, M. S. Triantafyllou, and M. A. Grosenbaugh, "Optimal Thrust Development in Oscillating Foils with Application to Fish Propulsion," *Journal of Fluids and Structures,* vol. 7, pp. 205-224, 1993.

[111] D. Bernal, K. A. Dickson, R. E. Shadwick, and J. B. Graham, "Review: Analysis of the evolutionary convergence for high performance swimming in lamnid sharks and tunas," *Comparative Biochemistry and Physiology - Part A: Molecular & Integrative Physiology,* vol. 129, pp. 695-726, 2001.

[112] H. S. Kim, B. R. Lee, T. Q. Vo, and Q. B. Truong, "A study on optimization of fish robot velocity using genetic algorithm," in *ICSMA 2008 - International Conference on Smart Manufacturing Application*, Goyang-si, South Korea, 2008, pp. 441-446.

[113] M. R. Dimitrijevic, Y. Gerasimenko, and M. M. Pinter, "Evidence for a spinal central pattern generator in humans," *Neuronal Mechanisms for Generating Locomotor Activity*, pp. 360-376, 1998.

[114] S. Grillner, "Neurobiological bases of rhythmic motor acts in vertebrates," *Science*, vol. 228, pp. 143-149, 1985.

[115] J. Duysens and H. W. A. A. Van De Crommert, "Neural control of locomotion; Part 1: The central pattern generator from cats to humans," *Gait and Posture,* vol. 7, pp. 131-141, 1998.

[116] T. G. Brown, "The Factors in Rhythmic Activity of the Nervous System," *proceedings of the Roual Society of London. Series B,* vol. 85, pp. 278-289, 1912.

[117] A. H. Cohen, P. J. Holmes, and R. H. Rand, "The nature of the coupling between segmental oscillators of the lamprey spinal generator for locomotion: A mathematical model," *Journal of Mathematical Biology,* vol. 13, pp. 345-369, 1982.

[118] A. H. Cohen and P. Wallen, "The neuronal correlate of locomotion in fish. 'Fictive swimming' induced in an in vitro preparation of the lamprey spinal cord," *Experimental Brain Research,* vol. 41, pp. 11-18, 1980.

[119] R. Jung, T. Kiemel, and A. H. Cohen, "Dynamic behavior of a neural network model of locomotor control in the lamprey," *Journal of Neurophysiology,* vol. 75, pp. 1074-1086, 1996.

[120] P. L. Varkonyi, T. Kiemel, K. Hoffman, A. H. Cohen, and P. Holmes, "On the derivation and tuning of phase oscillator models for lamprey central pattern generators," *Journal of Computational Neuroscience,* vol. 25, pp. 245-261, 2008.

[121] S. Grillner, T. Deliagina, O. Ekeberg, A. El Manira, R. H. Hill, A. Lansner, G. N. Orlovsky, and P. Wallen, "Neural networks that co-ordinate locomotion and body orientation in lamprey," *Trends in Neurosciences,* vol. 18, pp. 270-279, 1995.

[122] M. L. Shik, F. V. Severin, and G. N. Orlovskii, "Control of walking and running by means of electrical stimulation of the mid-brain," *Biophysics,* vol. 11, pp. 756-765, 1966.

[123] T. Bem, J. M. Cabelguen, O. Ekeberg, and S. Grillner, "From swimming to walking: A single basic network for two different behaviors," *Biological Cybernetics,* vol. 88, pp. 79-90, 2003.

[124] L. Righetti and A. J. Ijspeert, "Design methodologies for central pattern generators: an application to crawling humanoids," in *Proceedings of robotics: Science and systems*, Philadelphia, USA, 2006, pp. 191-198.

[125] L. Righetti, J. Buchli, and A. J. Ijspeert, "Dynamic Hebbian learning in adaptive frequency oscillators," *Physica D: Nonlinear Phenomena,* vol. 216, pp. 269-281, 2006.

[126] J. Buchli, "Engineering Limit Cycle Systems: Adaptive Frequency Oscillators and Applications to Adaptive Locomotion Control of Compliant Robots," Ph.D Thesis in the Institute of Bioengineering, Swiss Federal Institute of Technology at Lausanne, Switzerland, 2007.

[127] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization: A Universal Concept in Nonlinear Science*: Cambridge University Press, 2001.

[128] P. G. Drazin, *Nonlinear Systems*. Cambridge: Cambridge University Press, 1992.

[129] W. Zhao, J. Yu, Y. Fang, and L. Wang, "Development of multi-mode biomimetic robotic fish based on central pattern generator," in *IEEE International Conference on Intelligent Robots and Systems,* Beijing, China, 2006, pp. 3891-3896.

[130] M. Wang, J. Yu, M. Tan, and J. Zhang, "Design and implementation of a novel CPG-based locomotion controller for robotic dolphin," in *The 8th World Congress onIntelligent Control and Automation (WCICA)* Jinan, China, 2010, pp. 1611-1616.

[131] M. K. Habib, K. Watanabe, and K. Izumi, "Biped locomotion using CPG with sensory interaction," in *IEEE International Symposium on Industrial Electronics*, Seoul, Korea, 2009, pp. 1452-1457.

[132] C. Tang, S. Ma, B. Li, and Y. Wang, "A cubic CPG model for snake-like robot to adapt to environment," in *2010 IEEE International Conference on Information and Automation, ICIA 2010*, Harbin, China, 2010, pp. 24-29.

[133] R. Ding, J. Yu, Q. Yang, M. Tan, and J. Zhang, "CPG-based dynamics modeling and simulation for a biomimetic amphibious robot," in *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Guilin, China, 2009, pp. 1657-1662.

[134] C. Zhou and K. H. Low, "Kinematic modeling framework for biomimetic undulatory fin motion based on coupled nonlinear oscillators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 2010, pp. 934-939.

[135] C. Zhou and K. H. Low, "Study and Implementation of Station-holding Performance on a Fish Robot in Adverse Unsteady Flow," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 2010, pp. 928-933.

[136] M. Wang, J. Yu, and M. Tan, "Modeling neural control of robotic fish with pectoral fins using a CPG-based network," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, Shanghai, China, 2009, pp. 6502-6507.

[137] K. Seo, S. J. Chung, and J. J. E. Slotine, "CPG-based control of a turtle-like underwater vehicle," *Autonomous Robots*, vol. 28, pp. 247-269, 2010.

[138] C. Liu, Q. Chen, and J. Zhang, "Coupled Van Der Pol oscillators utilised as Central pattern generators for quadruped locomotion," in *Control and Decision Conference, 2009. CCDC '09. Chinese*, Guilin, China, 2009, pp. 3677-3682.

[139] C. Wilbur, W. Vorus, Y. Cao, and S. N. Currie, "A lamprey-based undulatory vehicle," *Neurotechnology for Biomimetic Robots*, pp. 285-296, 2002.

[140] M. Sfakiotakis, D. P. Tsakiris, and K. Karakasiliotis, "Polychaete-like pedundulatory robotic locomotion," in *Proceedings - IEEE International Conference on Robotics and Automation*, Roma, Italy, 2007, pp. 269-274.

[141] W. Zhao, Y. Hu, L. Zhang, and L. Wang, "Design and CPG-based control of biomimetic robotic fish," *Control Theory & Applications, IET*, vol. 3, pp. 281-293, 2009.

[142] S. Gay, S. Degallier, U. Pattacini, A. Ijspeert, and J. S. Victor, "Integration of vision and central pattern generator based locomotion for path planning of a non-holonomic crawling humanoid robot," in *The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 2010, pp. 183-189.

[143] L. Lapierre and B. Jouvencel, "Path following control for an eel-like robot," in *Oceans 2005 - Europe*, Brest, France, 2005, pp. 460-465.

[144] M. Sfakiotakis and D. P. Tsakiris, "Biomimetic centering for undulatory robots," *International Journal of Robotics Research*, vol. 26, pp. 1267-1282, 2007.

[145] C. Tapia, T. Hoinville, and R. Chellali, "Locomotion through Karman Street Using Adaptive Oscillator," in *Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009. COMPUTATIONWORLD '09. Computation World:*, Athens, 2009, pp. 188-191.

[146] R. C. Dorf and R. H. Bishop, *Modern Control Systems (8th Edition)*: Addison Wesley Longman Inc., 1998.

[147] G. F. Franklin and J. D. Powell, *Feedback Control of Dynamic Systems*: Prentice-Hall Inc., 2002.

[148] C. Stefanini, G. Orlandi, A. Menciassi, Y. Ravier, G. La Spina, S. Grillner, and P. Dario, "A mechanism for biomimetic actuation in lamprey-like robots," in *Proceedings of the First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, Pisa, Italy, 2006, pp. 579-584.

[149] K. A. Morgansen, V. Duindam, R. J. Mason, J. W. Burdick, and R. M. Murray, "Nonlinear control methods for planar carangiform robot fish locomotion," in *Proceedings - IEEE International Conference on Robotics and Automation*, Seoul, 2001, pp. 427-434.

[150] E. Yang, T. Ikeda, and T. Mita, "Nonlinear tracking control of a nonholonomic fish robot in chained form," in *SICE Annual Conference*, Denver, USA, 2003, pp. 1260-1265.

[151] M. Nakashima, N. Ohgishi, and K. Ono, "A study on the propulsive mechanism of a double jointed fish robot utilizing self-excitation control," *JSME International Journal, Series C: Mechanical Systems, Machine Elements and Manufacturing*, vol. 46, pp. 982-990, 2003.

[152] I. Ariyanto, T. Kang, and Y. J. Lee, "Identification of a fish-like robot using inertial sensors," *Key Engineering Materials*, vol. 326-328 II, pp. 1439-1442, 2006.

[153] R. L. McMasters, C. P. Grey, J. M. Sollock, R. Mukherjee, A. Benard, and A. R. Diaz, "Comparing the mathematical models of Lighthill to the performance of a biomimetic fish," *Bioinspiration and Biomimetics*, vol. 3, p. available online, 2008.

[154] M. J. Lighthill, "Large-Amplitude Elongated-Body Theory of Fish Locomotion," in *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 1971, pp. 125-138.

[155] S. Saimek and P. Y. Li, "Motion planning and control of a swimming machine," *International Journal of Robotics Research*, vol. 23, pp. 27-53, 2004.

[156] N. Kato, "Control performance in the horizontal plane of a fish robot with mechanical pectoral fins," *IEEE Journal of Oceanic Engineering*, vol. 25, pp. 121-129, 2000.

[157] J. Yu and Y. Qiu, "On the Efficiency of Robot Fish," in *26th Chinese Control Conference*, Hunan, China, 2007, pp. 203 - 206.

[158] J. Liu and H. Hu, "Biological Inspiration: From Carangiform Fish to Multi-Joint Robotic Fish," *Journal of Bionic Engineering*, vol. 7, pp. 35-48, 2010.

[159] P. W. Webb, "Station-Holding by three Species of Benthic Fishes," *J Exp Biol*, vol. 145, pp. 303-320, 1989.

[160] R. L. Carlson and G. V. Lauder, "Living on the bottom: Kinematics of benthic station-holding in darter fishes (percidae: Etheostomatinae)," *Journal of Morphology*, vol. 271, pp. 25-35, 2009.

[161] Y. Yang, N. Chen, C. Tucker, S. Pandya, D. Jones, and C. Liu, "Biomimetic flow sensing using artificial lateral lines," in *ASME International Mechanical Engineering Congress and Exposition, Proceedings*, Seattle, WA, 2008, pp. 1331-1338.

[162] A. Flock and D. M. K. Lam, "Neurotransmitter synthesis in inner ear and lateral line sense organs," *Nature*, vol. 249, pp. 142-144, 1974.

[163] A. Alvarez, A. Caiti, and R. Onken, "Evolutionary path planning for autonomous underwater vehicles in a variable ocean," *IEEE Journal of Oceanic Engineering,* vol. 29, pp. 418-429 2004.

[164] M. Soulignac, "Feasible and Optimal Path Planning in Strong Current Fields," *IEEE Transactions on Robotics,* vol. 27, pp. 89-98, 2011.

[165] J. J. Collins and S. A. Richmond, "Hard-wired central pattern generators for quadrupedal locomotion," *Biological Cybernetics,* vol. 71, pp. 375-385, 1994.

[166] Y. Zhong, C. W. Chong, C. Zhou, and K. H. Low, "Performance Predict Model for BCF Biomimetics Robotic Fish," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM),* Singapore, 2009, pp. 1230-1235

[167] B. W. Verdaasdonk, H. F. J. M. Koopman, and F. C. T. V. D. Helm, "Energy efficient and robust rhythmic limb movement by central pattern generators," *Neural Networks,* vol. 19, pp. 388-400, 2006.

[168] A. Visioli, "Fuzzy logic based set-point weight tuning of PID controllers," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans.,* vol. 29, pp. 587-592, 1999.

[169] J. Buchli and A. J. Ijspeert, "Distributed central pattern generator model for robotics application based on phase sensitivity analysis," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 3141, ed, 2004, pp. 333-349.

[170] M. W. Hirsch, S. Smale, and R. L. Devaney, *Differential Equations, Dynamical Systems, and An Introduction to Chaos (2nd Edition)*: Elsevier, USA, 2003.

[171] B. M. K. Anuar and T. Yasuno, "Moving control of quadruped hopping robot using adaptive CPG networks," presented at the 2010 IEEE Conference on Robotics Automation and Mechatronics (RAM), Singapore, 2010.

[172] F. D. Libera, T. Minato, H. HIshiguro, and E. Menegatti, "Direct programming of a central pattern generator for periodic motions by touching," *Robot. Auton. Syst.,* vol. 58, pp. 847-854, 2010.

[173] F. Cellier and E. Kofman, *Continuous System Simulation*: Springer Verlag, 2006.

[174] J. Z. Yu, E. K. Chen, S. Wang, and M. Tan, "Motion control algorithms for a free-swimming biomimetic robot fish," *Zidonghua Xuebao/Acta Automatica Sinica,* vol. 31, pp. 537-542, 2005.

[175] K. Singh and T. J. Pedley, "The hydrodynamics of flexible-body manoeuvres in swimming fish," *Physica D: Nonlinear Phenomena,* vol. 237, pp. 2234-2239, 2008.

[176] E. D. Tytell, C.-Y. Hsu, T. L. Williams, A. H. Cohen, and L. J. Fauci, "Interactions between internal forces, body stiffness, and fluid environment in a neuromechanical model of lamprey swimming," *Proceedings of the National Academy of Sciences,* vol. 107, pp. 19832-19837, 2010.

[177] Y. Toda, M. Danno, K. Sasajima, and H. MIki, "Model Experiment on the Squid-Like Under-Water Vehicle with two Side Fins," presented at the The Fourth International Symposium on Aero Aqua Bio-Mechanisms, ISABMEC, Shanghai, China, 2009.

[178] D. Barrett, M. Grosenbaugh, and M. Triantafyllou, "Optimal control of a flexible hull robotic undersea vehicle propelled by an oscillating foil," in *Proceedings of the IEEE Symposium on Autonomous Underwater Vehicle Technology*, 1996, pp. 1-9.

[179] K. A. De Jong, "Analysis of the Behavior of a Class of Genetic Adaptive Systems," Ph.D, Department of Computer and Communication Sciences, University of Michigan, USA, 1975.

[180] K. Deep and M. Thakur, "A new mutation operator for real coded genetic algorithms," *Applied Mathematics and Computation*, vol. 193, pp. 211-230, 2007.

[181] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*: Springer-Verlag, 1996.

[182] J. Ohlberger, G. Staaks, and F. Holker, "Swimming efficiency and the influence of morphology on swimming costs in fishes," *Journal of Comparative Physiology B: Biochemical, Systemic, and Environmental Physiology*, vol. 176, pp. 17-25, 2006.

[183] C. Zhou, "Optimal Design of Active Wire Bonding Tool," Master thesis (in Chinese), , School of Mechanical and Power Energy Engineering, Shanghai Jiao Tong University, Shanghai, China, 2006.