Autonomous Optimization of Swimming Gait in a Fish Robot With Multiple Onboard Sensors

Wei Wang, Dongbing Gu, and Guangming Xie

***Abstract*—Autonomous gait optimization is an essential survival ability for mobile robots. However, it remains a challenging task for underwater robots. This paper addresses this problem for the locomotion of a bio-inspired robotic fish and aims at identifying fast swimming gait autonomously by the robot. Our approach for learning locomotion controllers mainly uses three components: 1) a biological concept of central pattern generator to obtain specific gaits; 2) an onboard sensory processing center to discover the environment and to evaluate the swimming gait; and 3) an evolutionary algorithm referred to as particle swarm optimization. A key aspect of our approach is the swimming gait of the robot is optimized autonomously, equivalent to that the robot is able to navigate and evaluate its swimming gait in the environment by the onboard sensors, and simultaneously run a built-in evolutionary algorithm to optimize its locomotion all by itself. Forward speed optimization experiments conducted on the robotic fish demonstrate the effectiveness of the developed autonomous optimization system. The latest results show that our robotic fish attained a maximum swimming speed of 1.011 BL/s (40.42 cm/s) through autonomous gait optimization, faster than any of the robot's previously recorded speeds.**

***Index Terms*—Autonomous optimization, central pattern generators (CPGs), gait evaluation, robotic fish, underwater navigation, underwater robots.**

I. Introduction

OVER millions of years of evolution and natural selection, animals have possessed astonishing locomotion skills and fast gait-learning abilities to survive in the natural environment, thereby providing a great variety of sources of inspiration for robotics [1]–[3]. Similar to animals, an ideal biomimetic robot should interact with its surroundings, learn from the environments, and evolve locomotion controllers to adapt to its circumstances. As one of the most fundamental research issues of biomimetic robots, locomotion control has been extensively investigated in the literature. Previous studies have significantly advanced the locomotion control of biomimetic robots, and comparatively stable and flexible dynamic gaits have been generated in various biomimetic robots, such as fish robot [4]–[10], salamander robot [11], and dolphin robot [12].

Gait optimization, as one of the essential survival abilities for biomimetic robots, has been drawing more and more attentions in the field of robotics in recent decades [13]. The majority of the studies on gait optimization have focused on terrestrial robots at present, such as biped robots [14], quadruped robots [15], and modular robots [16]–[18]. By contrast, only a few studies are centered on the gait optimization of swimming robots in the literature.

Previous work for swimming gait optimization can be broadly divided into simulation-based and experiment-based. Simulation-based optimization uses a simulator or a model to optimize the swimming gait of robots. When an optimal gait is acquired in the simulation, it is later transferred to the physical robot. For example, particle swarm optimization (PSO) was adopted to search preferable parameters of the controller for the speed of the robotic fish in simulations [19], [20], which illustrates that PSO is a viable evolutionary algorithm. Learning in the simulation, however, is not always feasible because it is still difficult to precisely simulate the dynamics of water environment at present. On the other hand, experiment-based optimization employs the real robot to optimize the swimming gait. Based on a gradient-free optimization algorithm and a central pattern generator (CPG) model, an online optimization framework was presented to search for the optimal swimming and crawling gaits of an amphibious snake robot [21]. Moreover, an online genetic algorithm (GA) optimization approach was proposed to improve the capabilities of the undulatory locomotion governed by a CPG controller [22]. These experimental studies have validated that the CPGs have no problem dealing with the abrupt parameter changes during the optimization process. However, existing experiment-based optimization studies on swimming robots typically require sources external to the robot, such as an external measurement system, an auxiliary data processing center, or the assistance from experimenters [21]–[24]. To the best of our knowledge, autonomous gait optimization which is significant for the survival of swimming robots in complex aquatic environments was not tackled in the literature. Therefore, autonomous gait optimization is in urgent need of being developed and applied to swimming robots.

Manuscript received January 17, 2017; accepted March 5, 2017. This work was supported in part by the National Natural Science Foundation of China NSFC under Grant 51575005, Grant 61503008, Grant 61633002, and Grant 91648120, and in part by the China Post-Doctoral Science Foundation under Grant 2015M570013 and Grant 2016T90016. This paper was recommended by Associate Editor S. Nahavandi. *(Corresponding author: Guangming Xie.)*

W. Wang and G. Xie are with the State Key Laboratory of Turbulence and Complex Systems and the Intelligent Control Laboratory, College of Engineering, Peking University, Beijing 100871, China (e-mail: wangweiw4y4@pku.edu.cn; xiegming@pku.edu.cn).

D. Gu is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: dgu@essex.ac.uk).

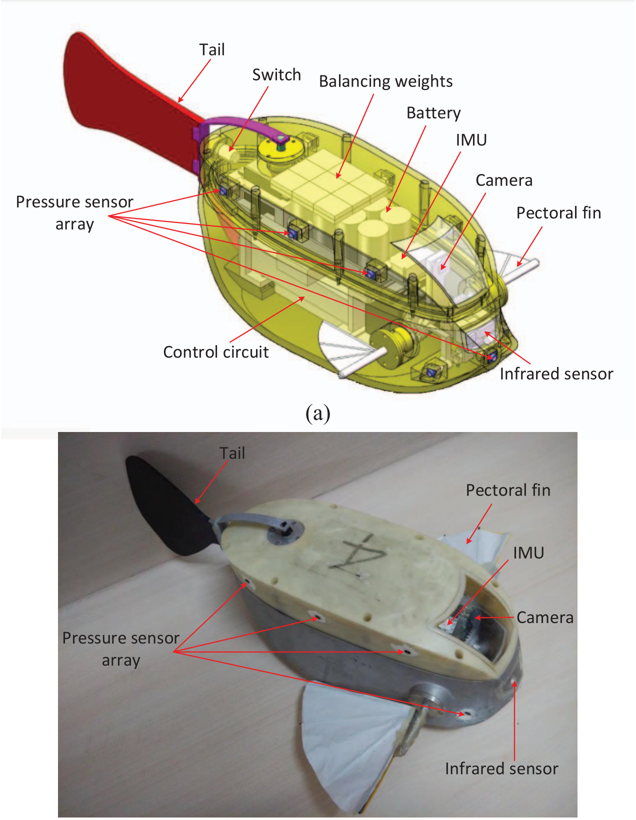
Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2017.2683524

Motivated by the aforementioned discussion and based on our previous work on the control [25] and localization of robotic fish [10], [26], this paper proposes an autonomous locomotion learning framework for underwater robots based on three components: 1) a CPG model; 2) a sensory processing center; and 3) an evolutionary algorithm referred to as PSO. In particular, the locomotion of the robotic fish is controlled by an artificial CPG model. It is inspired by the biological CPGs that consist of coupled neural networks capable of producing rhythmic oscillatory patterns while receiving simple adjustment signals from higher control centers [27]. Moreover, autonomous gait optimization requires the robot to be able to navigate and evaluate its gait quality all by itself. We develop an onboard sensory processing center to solve the underwater perception and gait evaluation problems.

Current underwater missions always require robots possess a variety of capabilities, such as high speed, high efficiency, excellent maneuverability, silent operation, and long endurance. Some of these demands are mutually exclusive and developing a versatile underwater robotic platform suitable for a multitude of missions is practically impossible. The optimization objective of this paper is to maximize the speed of the fish robot which is always required in real world exploration, probe and survey missions. We demonstrate how a swimming robot autonomously runs a built-in evolutionary algorithm (PSO) to search for the "optimal" CPG parameters that can produce the maximum speed for a given environment.

Because of the particularity of the water medium, underwater robots are confronted with considerable difficulties (for instance, many sensors and techniques are not available underwater) in motion control [28], environment identification, and gait quality evaluation, especially for those small robots with limited computational capacities and low-cost sensing devices. It is still challenging for underwater robots to achieve autonomous gait optimization. This paper aims at exploring the possibilities of autonomous gait optimization for swimming robots. We propose an autonomous gait optimization framework for underwater robots and primarily address the problems of underwater perception and gait evaluation. Compared with previous online gait optimization for swimming robots [21], [22], [24], the novelty here is the learning/optimizing with autonomy. The developed optimization system does not require the assistance of any external processing center or the experimenter, i.e., the robot is capable of evaluating its speed and assessing its environment all by itself. Thus, the gait optimization is conducted with minimal human attention. Being able to optimize the gaits autonomously is of great significance for underwater robots. Indeed, it might be one of the solutions for underwater robots to adapt their gaits to the real aquatic environments.



(b)

Fig. 1. Mechanical and hardware configurations of the fish robot. (a) Concept design. (b) Robot prototype.

The remainder of this paper is organized as follows. The robot prototype is briefly introduced in Section II. The framework of the autonomous gait optimization is proposed in Section III. Section IV presents the CPG controller of the robot. Environment perception and gait evaluation for underwater robots are tackled in Section V. The optimization algorithm is described in Section VI. Section VII shows the process of speed optimization of the robot. Gait optimization experiments and the related discussion are provided in Section VIII. Section IX concludes this paper with an outline of future work.

II. Robotic Fish Prototype

The optimization experiments are carried out with a robotic fish capable of multiple swimming modes, such as swimming forwards/backwards, turning, and pitching. This section describes the mechanics, electronics, sensors, and the CPG controller of the robot. The present robot is an improved version and holds the following improvements compared with the old version [25]: 1) high-fidelity shape; 2) good watertight design; 3) high-performance processor; 4) large-torque servomotor; and 5) multiple sensors.

A. Mechanical Design

As shown in Fig. 1, the robot consists of a well-streamlined shape, a pair of pectoral fins, and one caudal fin. It is inspired by the yellow-spotted boxfish in nature. The upper hull is made of acrylonitrile butadiene styrene resin leaving a transparent window for the view of the camera. The bottom uses 6061 aluminum alloy. A transparent window is left on the lower hull for the view of the infrared sensor. Static and dynamic sealing are designed to ensure longstanding waterproofness. A sealing ring is entangled on each driving shaft to keep out

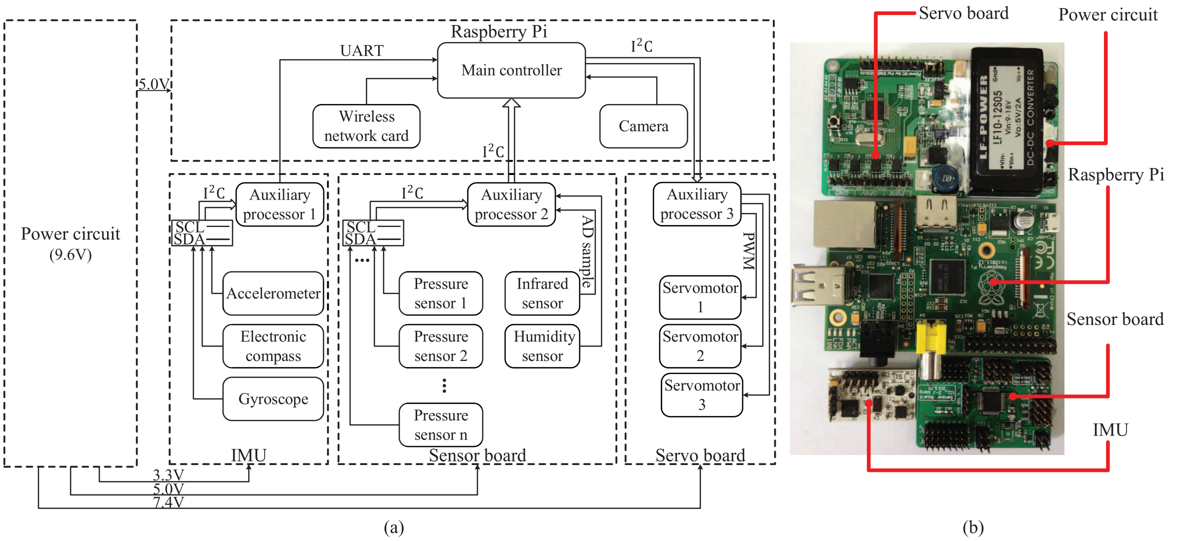


Fig. 2. Circuit control system of the robot. (a) Hierarchical electrical block diagram of the system which contains five parts: power circuit, Raspberry Pi, IMU, sensor board, and servo board. (b) Photograph of the printed circuits.

TABLE I
TECHNICAL SPECIFICATIONS OF THE ROBOT PROTOTYPE

Items	Characteristics
Dimension(L × W × H)	∼ 400 mm× 140 mm × 142 mm
Total mass	∼ 3.1 kg
Drive mode	DC servomotors (12.9 kg·cm)
Onboard sensors	Camera, IMU, pressure sensor, and infrared sensor
Power supply	9.6 V rechargeable Ni-MH batteries
Operation time	∼ 5 h
Control mode	Autonomous/Wireless mode

the water when the shaft is rotating. An O-ring is attached to the outer sleeve of the shaft to block the chink between the driving device and the hull.

B. Electronics and Sensors

Diversified sensors including camera, inertial measurement unit (IMU), pressure sensor array, and infrared sensor are installed on the robot, as shown in Figs. 1 and 2. The Logitech Webcam C110 is adopted to acquire high-resolution surroundings. The developed IMU is arranged in parallel with the robot body principal axes to monitor the yaw, pitch, and roll of the robot with a sampling rate of 50 Hz. Pressure sensors are employed to evaluate the speed of the robot. The pressure sensor CP131 from Consensic, Inc. is adopted. It has been fully calibrated and provides a 0.1 Pa resolution and a 30–120 kPa scale range. The infrared sensor GP2D12 is used to measure the distance between the robot and the border of the tank. Raspberry Pi is adopted as the main controller to fulfill the real-time requirements of the optimization. Moreover, three 32-bit auxiliary processors, STM32F103, are used for multisensor data preprocessing, locomotion control, and attitude calculation, respectively. A hierarchical hardware system is therefore generated as presented in Fig. 2. The robot is operated by a Linux system (Debian), and the specifications are listed in Table I.

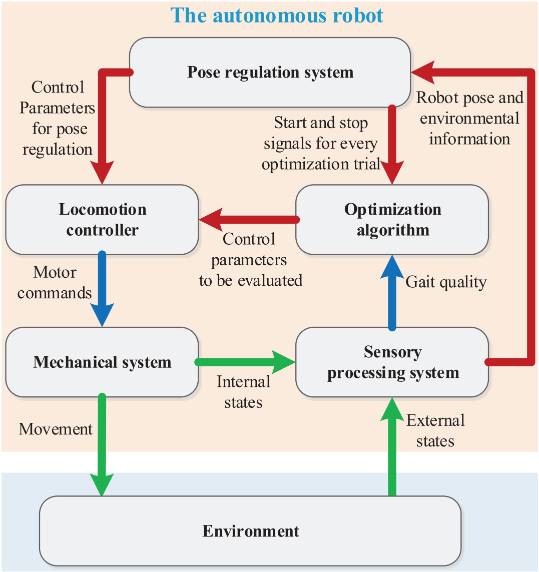

The autonomous robot
Pose regulation system
Control Parameters for pose regulation
Start and stop signals for every optimization trial
Robot pose and environmental information
Locomotion controller
Optimization algorithm
Control parameters to be evaluated
Motor commands
Gait quality
Mechanical system
Internal states
Sensory processing system
Movement
External states
Environment

Fig. 3. Proposed gait optimization framework. The arrow direction indicates the information flow, such as the commands and robot states.

III. FRAMEWORK OF AUTONOMOUS GAIT OPTIMIZATION

The primary purpose of gait optimization is to obtain the optimal parameters of the locomotion controller regarding the speed, efficiency, or flexibility. Then, the robot will generate the highest qualities of gait in a given environment. In general, the robot should be provided with several critical functional modules to achieve autonomous gait optimization. As shown in Fig. 3, an autonomous gait optimization framework is proposed for underwater robots. The framework contains six essential elements: 1) mechanical system; 2) locomotion controller; 3) sensory processing system; 4) optimization algorithm; 5) pose regulation system; and 6) the environment. In this paper, we stress on the locomotion controller and the sensory processing system for identifying the robot's nearby aquatic environment and for assessing the speed of the robot.

The mechanical system refers to the actuators, motors, fins, and legs of the robot. It receives the motor commands from the locomotion controller and produces a corresponding motion (named as the gait) acting on the external environment. The sensory processing system includes onboard sensors and the related navigation and gait assessment algorithms. It monitors the internal physical states (e.g., the gait quality, the robot position, and orientation) and the external environmental states (e.g., the obstacles and boundaries), thereby providing critical information for the optimization algorithm and pose regulation system. The locomotion controller is responsible for producing various motion commands according to the control parameters. It can receive both candidate parameters from the optimization algorithm and preparatory parameters for pose regulation from the pose regulation system. By acquiring the feedback from the sensory processing system, the pose regulation system can always adjust the robot pose to prepare for the execution of the candidate gait. The pose regulation system orders the start and the stop of the execution of candidate gait in each trial. The gait optimization algorithm can be an evolutionary algorithm, such as PSO and GA [15], [22], [23], or a gradient-free optimization method, such as Powell's method [17], [21]. After each trial, the optimization algorithm receives the evaluation of the candidate gait from the sensory processing center and creates new candidate gait.

IV. Central Pattern Generator Model

CPG models benefit rhythmic motion control of bio-inspired robots, such as bipedal robots [29], robotic fish [25], [30], and salamander robot [11]. Many artificial CPG models have been proposed in the literature, such as Matsuoka's [31] CPG model, Ijspeert *et al.*'s [11] CPG model, and Zhou and Low's [7] CPG model. We simplify the CPG model in [11] by linearizing the phase oscillator and strictly prove its stability in this paper. As shown in Fig. 4, the CPG controller consists of three coupled oscillators and takes the form

$$\dot{a}_i = \alpha_i(A_i - a_i) \tag{1a}$$

$$\dot{x}_i = \beta_i(X_i - x_i) \tag{1b}$$

$$\dot{\phi}_i = 2\pi f_i + \sum_{j\in T_i} \mu_{ij}\left(\phi_j - \phi_i - \varphi_{ij}\right) \tag{1c}$$

$$\theta_i = x_i + a_i\cos(\phi_i) \tag{1d}$$

where a_i, x_i, and ϕ_i are the state variables and denote the amplitude, offset, and phase of the ith oscillator, respectively. θ_i is the output of the ith oscillator. f_i, A_i, and X_i are the control parameters for the desired frequency, amplitude, and offset of the ith oscillator. φ_{ij} is the control parameter for the desired phase difference between the ith oscillator and the jth oscillator. μ_{ij} is a constant coefficient denoting the coupling strength between the ith oscillator and the jth oscillator. α_i and β_i are constant coefficients affecting the convergence rate of the CPG controller. T_i is the set of the neighbors of the ith oscillator. Subscripts $i = 1$, 2, and 3, respectively, represent the left pectoral fin, right pectoral fin, and tail of the robot.

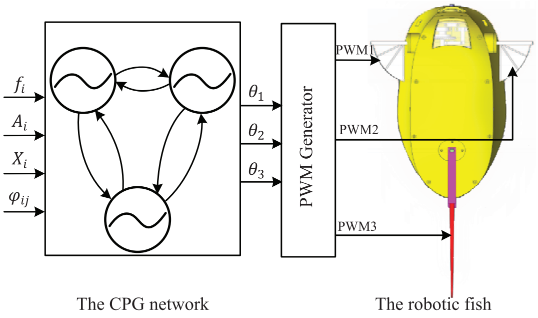


Fig. 4. CPG network of robotic fish.

We first put forward three assumptions as follows for the stability proof of the CPG model: 1) $\alpha_i = \alpha$, $\beta_i = \beta$, and $\mu_{ij} = \mu$, where $\alpha \in \mathbb{R}^+$, $\beta \in \mathbb{R}^+$, and $\mu \in \mathbb{R}^+$; 2) $\varphi_{ij} = \varphi_j - \varphi_i$, where $\varphi_i \in \mathbb{R}$ and $\varphi_j \in \mathbb{R}$; and 3) $f_i = f$, where $f \in \mathbb{R}$. Equations (1a) and (1b) define the dynamics of the amplitude and offset of oscillator i, and the solutions is derived as follows:

$$a_i(t) = A_i + (A_{i0} - A_i)e^{-\alpha(t-t_0)} \tag{2}$$

$$x_i(t) = X_i + (X_{i0} - X_i)e^{-\beta(t-t_0)}. \tag{3}$$

It is clear that $a_i(t)$ and $x_i(t)$ will converge exponentially to A_i and X_i from any initial states, respectively.

By setting $z_i = \phi_i - \varphi_i$, (1c) will convert to

$$\dot{z}_i = 2\pi f + \sum_{j\in T_i} \mu\left(z_j - z_i\right). \tag{4}$$

According to the algebraic graph theory, (4) can be translated to the matrix form

$$\dot{\mathbf{z}} = -L\mathbf{z} + 2\pi f\mathbf{1} \tag{5}$$

where $\mathbf{z} = [z_1\ z_2 \ldots\ z_N]^T$, $\mathbf{1}$ donates an $N \times 1$ column vector of all ones, and $L = (l_{ij})_{N\times N}$ donates the Laplacian matrix of the CPG network and takes the form [32]

$$l_{ij} = \begin{cases} (N-1)\mu, & i = j \\ -\mu, & i \neq j. \end{cases} \tag{6}$$

Since, μ is real, L will be a real, symmetric, and semidefinite positive matrix. It has N non-negative eigenvalues: $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$. L can be further diagonalized as

$$T^T LT = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ldots & \\ & & & \lambda_N \end{bmatrix} \tag{7}$$

where T is an orthogonal matrix satisfying $TT^T = I$, and I is the identity matrix. By setting $\mathbf{z} = T\mathbf{y}$, (5) is rewritten as follows:

$$\dot{\mathbf{y}} = -T^{-1}LT\mathbf{y} + 2\pi fT^{-1}\mathbf{1}. \tag{8}$$

T can be further expressed as $T = [\eta_1\ \eta_2 \ldots\ \eta_N]$ where η_i is a column vector. Hence, (9) can be expressed as follows:

$$\dot{y}_i = -\lambda_i y_i + 2\pi f\eta_i^T\mathbf{1}. \tag{9}$$

Each oscillator is affected by the other $N-1$ oscillators, then the graph is complete. From (6), the sum of every row of L will

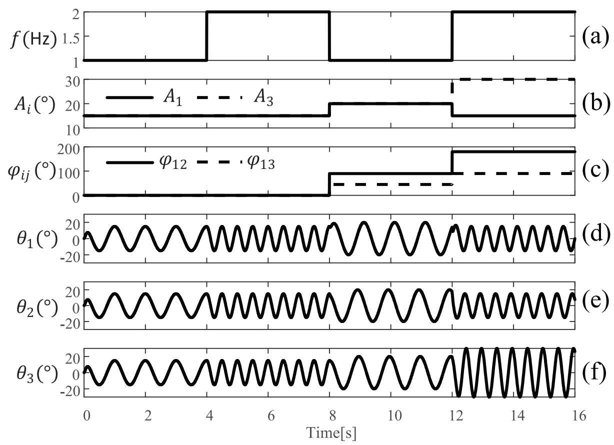


Fig. 5. Effects of changing the parameters of the CPG network. In this simulation, we set $A_1 = A_2$ and $X_i = 0$. (a) Beating frequency. (b) Oscillation amplitude. (c) Phase difference. (d)–(f) CPG outputs. At $t = 4$ s, the frequency is changed abruptly to 2.0 Hz. At $t = 8$ s, A_1 and A_3 are changed to 20° and φ_{12} and φ_{13} are changed to 90° and 45°, respectively. At $t = 12$ s, A_1 and A_3 are set, respectively, to 15° and 30°, and φ_{12} and φ_{13} are set, respectively, to 180° and 90°.

be zero, which indicates that L always has a zero eigenvalue, $\lambda_1 = 0$. According to the properties of orthogonal matrix, two results will be achieved: 1) $\mathbf{1} = \sqrt{N}\eta_1$ and 2) $\eta_i^T\eta_1 = 0$ where $i \neq 1$. Then, (9) is solved as follows:

$$y_i = \begin{cases} 2\pi f\sqrt{N}t + y_i(0), & i = 1 \\ e^{-\lambda_i t}y_i(0), & i = 2, 3, \ldots, N \end{cases} \tag{10}$$

where $y_i(0)$ is the initial state of y_i. $y_i(i \geq 2)$ will attenuate exponentially to zero as $t \to \infty$. Then, z_i converges to $2\pi ft + z_1(0)$ where $z_1(0) = (1/\sqrt{N})y_1(0)$ and

$$\phi_i = z_i + \varphi_i \to 2\pi ft + z_1(0) + \varphi_i \tag{11}$$

$$\phi_j = z_j + \varphi_j \to 2\pi ft + z_1(0) + \varphi_j. \tag{12}$$

Therefore, $\phi_j - \phi_i \to \varphi_{ij}$ as $t \to \infty$ and (1c) is stable.

The above derivation proves that the CPG can exert limit cycle behavior. It adapts to any rapid parameter change and converges to the modified oscillation after a short transient period. An illustrative example of how the CPG reacts to the parameter changes is shown in Fig. 5. it is evident that when the parameters abruptly changed the oscillator can still converge smoothly to the new limit cycle without discontinuities.

The CPG model typically acts as a type of motion generator shaped by a set of control parameters $\{f, A_i, X_i, \varphi_{ij}\}$. At present, the parameter tuning of the CPG controller is not solved satisfactorily. CPG parameters are usually adjusted by hand, which is difficult, time-consuming and inefficient. This paper attempts to optimize the CPG parameters in an autonomous manner. Indeed, autonomous gait optimization requires the robot to perceive its near environment and evaluate its swimming gait all by itself. These problems are tackled in this paper and presented in the following section.

V. Autonomous Perception and Gait Evaluation

Since, the robot optimizes its gaits in the confined space, the robot should first be able to know where it is and where the boundaries are. Moreover, the robot should assess the quality of its gaits by itself. Compared with terrestrial robots, underwater robots are confronted with more difficulties in identifying their near environments and evaluating their swimming gait qualities, especially for the small robots with limited computational capacities and low-cost sensing devices. The reason why we use low-cost sensors is twofold. First, these cheap devices are affordable for most of the current robots. The developed system can be then transferred to applications more smooth. Second, low-cost devices usually have worse performance than the expensive sensors. Hence, the developed algorithms with low-cost devices will have better adaptability. Note that the environment perception and speed evaluation algorithms constitute the sensory processing system of the proposed gait optimization framework.

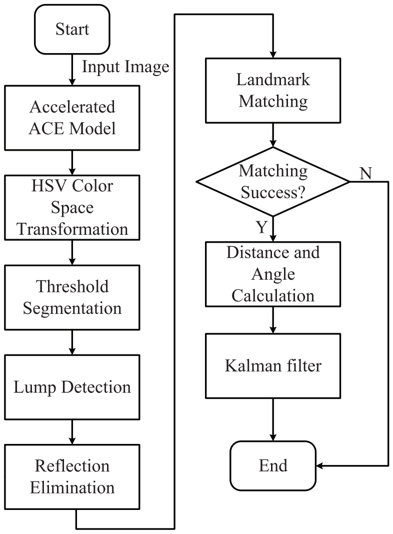


Fig. 6. Flow diagram of the proposed underwater image processing algorithm.

A. Aquatic Environment Perception

The robot speed is estimated by the pressure sensors, and the precision is affected by the duration of the evaluation. Considering the limitation of the water tank in the laboratory, we have to order the robot activities to ensure that the robot can execute the trial as long as possible. Vision is always served as one of the primary sources to assist robot control [33], [34]. Therefore, the robot roughly estimates its position by the onboard camera and two landmarks are employed to improve the identification. In particular, we proposed an online image processing algorithm to help the robot assess its position. Moreover, we developed a tiny IMU to monitor the robot orientation and adopted an infrared sensor to detect the boundaries of the environment.

1) Underwater Image Processing: The proposed online image processing algorithm is shown in Fig. 6. Here, we address the distance and relative angle calculation. More details can be referred to our related work [10], [35]. Based on the estimated distance d and the relative angle γ, the robot's position can be roughly achieved [35].

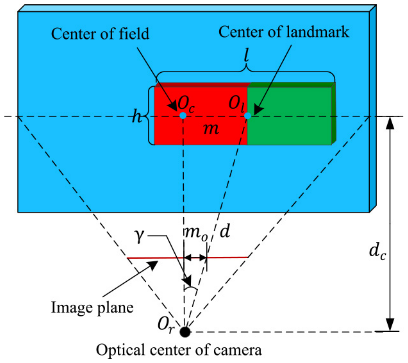


Fig. 7. Schematic of the distance and angle calculation.

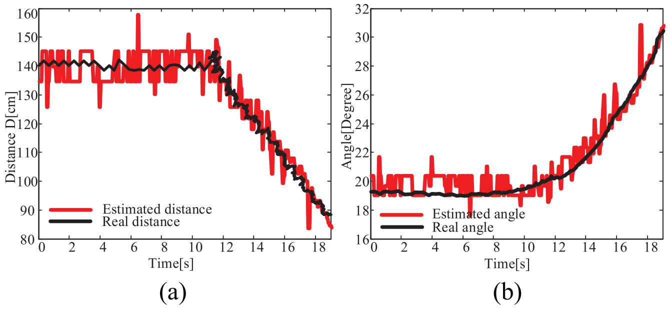
As shown in Fig. 7, the distance d and the angle γ can be calculated with the following forms:

$$d = \sqrt{{d_c}^2 + m^2} = \sqrt{\left(\frac{d_f h}{h_o}\right)^2 + \left(\frac{m_o h}{h_o}\right)^2} \tag{13}$$

$$\gamma = \arctan\frac{m}{d_c} \tag{14}$$

where d_c the distance between the optical center of the camera and the center of the field, m the distance between the center of the field and the center of the landmark, m_o the distance between the center of field and the center of the landmark in the image plane, h the height of the landmark, and h_o the height of the landmark on the image plane. d_f denotes the focal length of the camera obtained by $d_f = d_{\min}L_0/h$ where L_0 denotes the height of the image plane in pixels. $d_{\min}$ denotes the distance between the camera and the landmark when the landmark fully fills in the image plane in the longitudinal direction. h is the height of the landmark. The unit of h_o, m_o, d_f, and L_0 is pixel and the unit of m, h, and $d_{\min}$ is meter. Fig. 8 shows the estimated distance and angle when the robot is swimming. The real distance between the robot and the landmark is obtained by the tracking platform developed by our laboratory [36]. The experiment is performed as follows. First, the robot catches the sight of a valid landmark at a distance of 140 cm. Then it swims toward the landmark with an approximate speed of 8 cm/s at $t = 12$ s. Five experiments are conducted to determine the accuracy of d and ϕ. The accuracy of d is 3.52 cm ± 1.2 cm, and the accuracy of ϕ is 2.36° ±0.96° while the robot is moving.

In summary, the robot can assess the distance d and relative angle γ through the proposed image processing. Because the position of the landmark is predefined, the robot can localize itself accurately by the use of onboard camera and IMU [10].



(a) (b)

Fig. 8. Evaluated (a) distance and (b) angle.

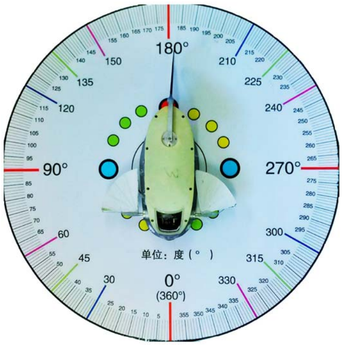


Fig. 9. Calibrating setup for the IMU.

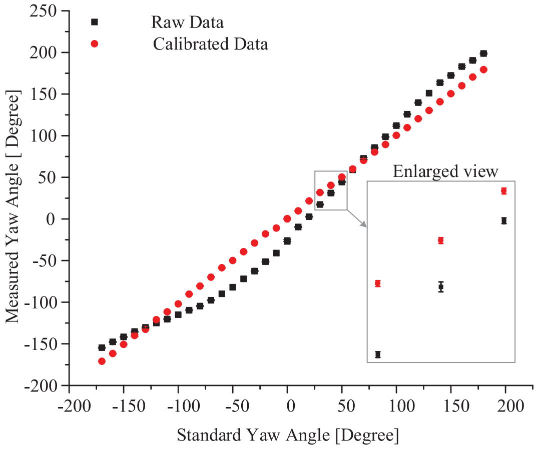


Fig. 10. Calibrated orientation results of the robot. Error bars are shown but may be obscured by data symbols.

2) Robot Orientation Perception: The robot can adjust itself to a correct direction range by the use of IMU and swim as long as possible in the narrow water space. The direction-cosine-matrix (DCM) algorithm is used to calculate the attitudes including yaw, pitch, and roll [37]. According to the DCM algorithm, the accuracy of the yaw angle primarily depends on the compass. However, the compass is susceptible to the magnetic disturbances from the robot, such as the servomotor, battery, and the other circuits. The orientation error sometimes can reach 30° on the robot. Such a significant error largely affects the robot navigation. Assuming the magnetic fields around the robot are constant, we propose an efficient calibration method. The calibration scene is illustrated in Fig. 9, where the robot is placed at the center of an azimuth disc. The measured yaw angle and the real value are recorded each 10° in the experiment. The tests are conducted at four positions of the robot's active region to eliminate the environmental impact. The experiments are repeated five times at each position. The calibrated orientation is acquired by the piecewise data fitting method. As shown in Fig. 10, the maximum error decreases approximately to 2° after the calibration. The calibrated direction is regarded as the true direction of the robot.

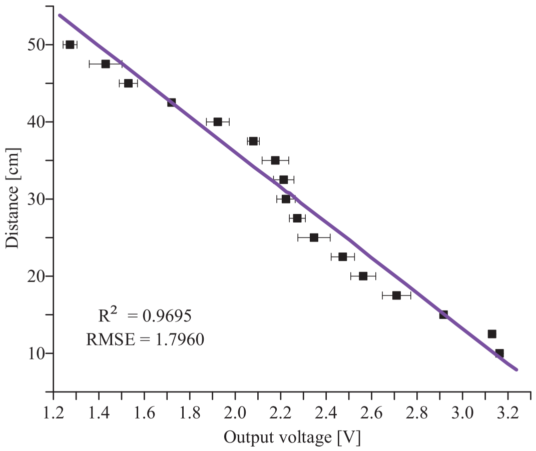


Fig. 11. Measured distance by the infrared sensor. Error bars are shown but may be obscured by data symbols.

3) Boundary Detection: The robot should stop the candidate gait correctly to avoid possible damages and disturbances to the gait evaluation. An infrared sensor is used to monitor the distance between the robot and the tank boundary. The detecting range becomes smaller when the infrared sensor works underwater. We need to calibrate the infrared sensor under the water. The experiment is conducted as follows. First, the sensor is fixed underwater and faces to the wall of the tank. Then every 2.5 cm, the output voltage of the infrared sensor is recorded. The experiments are repeated ten times. As depicted in Fig. 11, the relationship between the measured distance and the output voltage can be derived by using a linear regression model

$$d_{\text{inf}} = -22.26U_{\text{inf}} + 79.94 \tag{15}$$

where d_{inf} denotes the evaluated distance and U_{inf} denotes the output voltage.

B. Swimming Gait Quality Evaluation

Swimming speed is optimized to show the effectiveness of the proposed autonomous gait optimization approach. At present, few small underwater robots can accurately evaluate their speeds online. We use pressure sensors to tackle the problem of speed evaluation for small underwater robots.

If the robot is towed horizontally in the still water, the pressure sensor at the tip of the nose measures the stagnation pressure p_s according to Bernoulli equation. The stagnation pressure is equal to the sum of the static pressure p_{sw} and the dynamic pressure

$$p_s = p_{sw} + \frac{1}{2}\rho v_f \tag{16}$$

where ρ is the density of water and v_f is the towed velocity. v_f can be derived from (16) by the use of pressure sensor at the tip of the nose when the robot is towed. However, we find that this relation does not hold for freely swimming robotic fish because rotary and lateral movements also exist [38]. We therefore use the data-driven method to derive the relationship between the pressure readings and the robot speed.

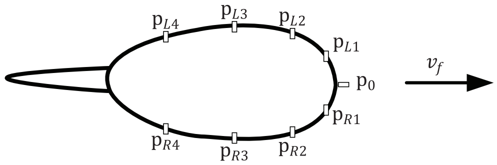


Fig. 12. Distribution of nine pressure sensors around the body of the robot (from top view).

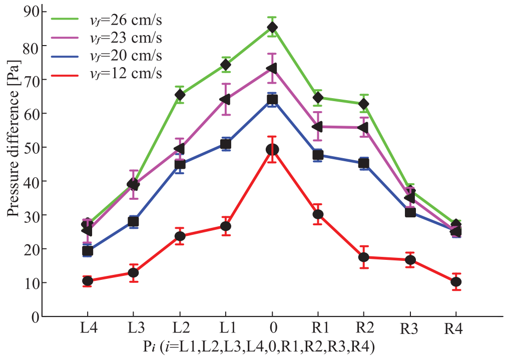


Fig. 13. Pressure readings when the robot was freely swimming forwards at different speeds.

The depth of the robot largely affects the pressure according to the pressure formula in a liquid. Hence, the depth of the pressure sensor (namely the depth of the robot) should be constant change while the robot is swimming. We set $X_i = 0$ $(i = 1,\ 2,\ 3)$ to preclude any forces in the vertical plane and set $A_1 = A_2$ to produce forward swimming [38]. Moreover, the buoyancy of the robot is trimmed to be greater than the gravity of the robot. These configurations guarantee that the robot always swims in the horizontal plane during experiments. Furthermore, IMU data shows that pitch and roll oscillatory motions do exist but have very small amplitudes (typically less than 1°). The oscillation center is nearly zero. Therefore, we can ignore the pressure changes caused by the effects of pitch and roll motions in the average sense.

Fig. 12 shows the distribution of the pressure sensors on the robot. We recorded the pressure readings of these nine sensors when the robot was swimming forwards at different speeds in the tank. The experiment was conducted as follows. The robot is started to swim from one side of the reservoir and its speed was precisely estimated by the developed vision tracking platform [36]. The robot was stopped when it approached the opposite side of the tank. The speed derived by the tracking software is regarded as the real speed of the robot. The pressure data in the accelerating phase was removed. The experiments were repeated five times for each speed. five times. The measurements in still water were taken as a reference (0 Pa). Fig. 13 shows that the pressure differences increased with the swimming speed. The foremost sensor (p_0) always measured the highest pressure. The pressure decreased gradually toward the sides. We employed the average ($\bar{p}$) of p_0, p_{L1}, and p_{R1} as an indicator of the robot speed.

Assuming the quadratic relationship between the speed and the pressure still holds, the identified model is expressed as

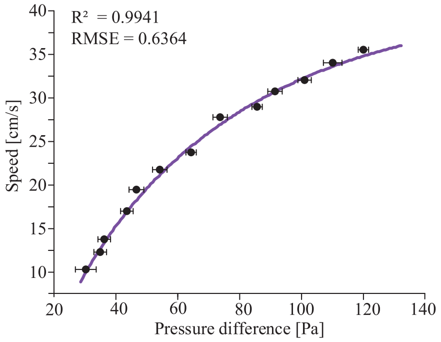


Fig. 14. Relationship between the swimming speed of the robot and the sensor readings.

$v_f = a_p\sqrt{\bar{p}} + b_p\bar{p} + c_p$. Fig. 14 shows the average pressure $\bar{p}$ at different speeds. According to the data in the figure, the relationship between the speed and the pressure is derived

$$v_f = 10.585\sqrt{\bar{p}} - 0.367\bar{p} - 36.786 \tag{17}$$

where v_f denotes the freely swimming speed of the robot. The unit of v_f is centimeter per second while the unit of $\bar{p}$ is Pa. The pressure sensor readings are online recorded when the robot is carrying out the candidate gait and the speed of the robot for relevant CPG parameters was online evaluated by (17).

VI. Optimization Algorithm

The function we want to optimize is the speed $v_f(\vec{s})$ of the robot, where $\vec{s}$ is the parameter vector containing the CPG parameters to be optimized, i.e., the frequency f, the amplitude A_i, the phase difference φ_{ij}, and the oscillation offset X_i. Since, we do not have any gradient information for $v_f(\vec{s})$, the gradient-based methods are avoided. Moreover, as we want to find a global optimum for the speed, Powell's methods are probably inappropriate because these methods have a high risk of convergence to a locally optimal solution. Furthermore, the convergence time is critical in the context since the robot has a limited battery life. The algorithm we chose is the PSO method. PSO a population-based evolutionary algorithm and can always converge rapidly in developing the better locomotive gaits of many kinds of robots, such as biped robots [39], swimming robots [19], [20], and modular robots [18].

For completeness, a brief review of the PSO algorithm is described here. PSO is inspired by the social behavior of bird flocking and fish schooling. It is initially proposed by Kennedy and Eberhart [40]. It is a kind of population-based evolutionary algorithm, exploring optimal solutions through a population of particles. The position $\mathbf{x}_i$ and velocity $\mathbf{v}_i$, respectively, represent the parameter values and the search direction of particle i. The iteration equation takes the form

$$\mathbf{v}_i^{t+1} = w\mathbf{v}_i^t + c_1 U_1^t(\mathbf{p}_i - \mathbf{x}_i^t) + c_2 U_2^t(\mathbf{p}_g - \mathbf{x}_i^t) \tag{18}$$
$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \tag{19}$$

where $\mathbf{p}_i$ is the personal best-known position of particle i and $\mathbf{p}_g$ is the global best-known position of all particles, t is the current generation of particles and $t+1$ is the next generation.

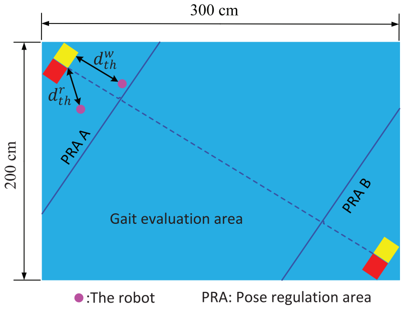


Fig. 15. Sketch map of the experimental scene.

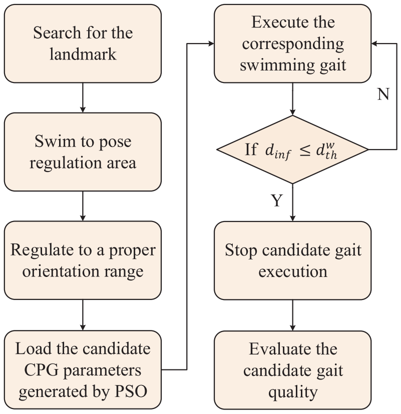

Search for the landmark
Swim to pose regulation area
Regulate to a proper orientation range
Load the candidate CPG parameters generated by PSO
Execute the corresponding swimming gait
If $d_{inf} \le d_{th}^w$
N
Y
Stop candidate gait execution
Evaluate the candidate gait quality

Fig. 16. Flow chart of one evaluation trial during optimization experiment.

c_1 and c_2 are the cognitive factor and the social factor, respectively. U_1 and U_2 are two pseudo-random numbers from 0–1. w denotes the inertia weight and it is introduced to improve the convergence speed of PSO.

VII. Autonomous Speed Optimization Process

This section describes how an underwater robot searches the parameters of CPG controller for the maximum speed autonomously in a confined water environment.

A sketch map of the optimization process is shown in Fig. 15. It should be noted that a trial in this paper is referred to that a candidate gait with a set of control parameters is correctly started, carried out, stopped, and evaluated by the robot. As the space limitation of the laboratory, the experiment is conducted in a 300 cm × 200 cm tank. Two landmarks are placed at the diagonal corners to improve the environmental identification. Once the gait optimization task is initialized, the procedure of each trial is the same until a stop condition is met. A flowchart of the procedure for each trial is depicted in Fig. 16, and detailed description of the procedure is presented below.

A. Search for the Landmark

Either the optimization task is started or a trial is finished, the robot will search for the landmark in the tank. If the robot finds the landmark through the aforementioned image processing, it will go to the pose regulation area (PRA) with the help of the landmark. Otherwise, the robot turns itself until it catches sight of the landmark. Turning motion is achieved by oscillating all the fins to reduce the turning radius. More specifically, the robot oscillates its tail with a nonzero offset and simultaneously oscillates its paired pectoral fins with an 180° offset difference to achieve turning motion.

B. Swim to Pose Regulation Area

If the robot detects the landmark, it swims to the landmark slowly. The tracking strategy is keeping the landmark in the middle of the robot's view, meaning that the angle γ defined in Section V-A1 should be zero. This angle can be minimized by adjusting the tail offset of the robot. We use a potential difference controller to minimize the angle

$$X_3 = -k_p\gamma - k_d\dot{\gamma} \tag{20}$$

where X_3 stands for the tail offset of the CPG model in Section IV. At the same time, the distance d between the robot and landmark defined in Section V-A1 is also monitored by the robot. If the distance is smaller than the threshold distance d_{th}^{r}, the robot reaches the PRA.

C. Regulate to Proper Orientation Range

After the robot arrives at the PRA, it will search for the right direction range by spinning in place to ensure that the robot can execute the candidate gait as long as possible. Spinning motion is achieved by oscillating the paired pectoral fins with an 180° offset difference. Specifically, if the robot is in PRA A, it will spin slowly until the orientation ψ satisfies $\psi_{\min}^{A} \leq \psi \leq \psi_{\max}^{A}$; if the robot is in PRA B, it will spin until ψ satisfies $\psi_{\min}^{B} \leq \psi \leq \psi_{\max}^{B}$.

D. Load the Candidate Parameters Generated by PSO

If the robot orientation reaches the correct direction range, it will load a set of candidate parameters (produced by PSO) into the CPG controller.

E. Execute the Candidate Gait

Next, the robot conducts the candidate gait in the gait optimization area until a stop signal is satisfied. Pressure sensor readings are recorded while swimming. These pressure readings will be used to assess the gait quality.

F. Stop the Candidate Gait

In the confined environment, each trial is arrested felicitously to avoid the wall effects on the gait evaluation as well as the damages to the robot. The distance d_{inf} measured by the infrared sensor is adopted as a stop signal for each trial. Concretely, the candidate gait is stopped if d_{inf} satisfies $d_{\text{inf}} \leq d_{\text{th}}^{w}$, where d_{th}^{w} is a threshold distance. Stopping the candidate gait means that the robot stops moving as well as recording the pressure data.

G. Evaluate the Gait Quality

After the candidate gait is stopped, the robot speed is assessed online by the pressure sensor readings according to (17). Typically, the robot can reach steady state in 3 s. Hence, we remove the pressure data in the first 3 s for ensuring the accuracy of the evaluation. In the end, the evaluated swimming speed serves as the fitness/value of the function for the candidate CPG parameters in the PSO algorithm.

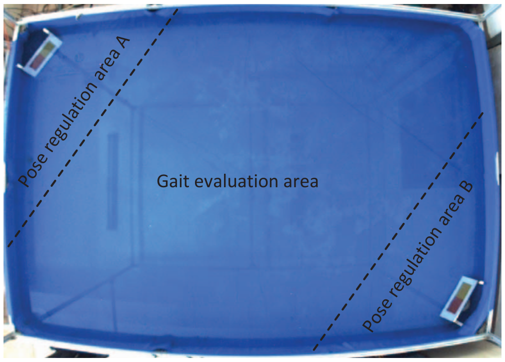


Fig. 17. Swimming tank with two landmarks in the experiment.

VIII. Results and Discussion

In this section, the speed optimization experiments with the robotic fish were carried out to validate the feasibility of the developed autonomous gait optimization system.

A. Experimental Description

The experiments are conducted in a tank with the size of 300 cm × 200 cm × 30 cm, as shown in Fig. 17. According to the force analysis of the robot [38], forward swimming can be achieved by equating the amplitude of pectoral fins ($A_1 = A_2$) and zeroing the offsets of all the fins ($X_1 = X_2 = X_3 = 0$). After that, the parameters to be optimized change to $red\vec{s} = \{A_1, A_3, \varphi_{12}, \varphi_{13}, f\}$. Typically, the frequency f does not need to be optimized [21] since previous studies indicate that the robot speed increases linearly with the frequency approximately for a wide range of the frequency [25], [41]. Moreover, multiple optimal speeds may satisfy various demands in practical applications. Therefore, the parameters to be optimized simplify to $\vec{s} = \{A_1, A_3, \varphi_{12}, \varphi_{13}\}$. The optimization experiments were conducted at $f = 1$ Hz, $f = 1.5$ Hz, $f = 2$ Hz, and $f = 2.5$ Hz in this paper. Considering of the convergence speed and the battery life, the number of the particle is selected as $N_p = 10$ and the number of the iteration is $N_g = 10$.

The first generation is created by random selection of the CPG parameters. The ranges of the parameters are determined by the mechanical restriction and the power limitation of the servomotor. Table II presents the ranges of the CPG parameters in the experiments. The maximum searching speed for each CPG parameter is determined by a large number of tentative experiments. The maximum searching speeds for A_1, A_3, φ_{12}, and φ_{13} are, respectively, $V_{A_1}^{\max} = (A_1^{\max} - A_1^{\min})/2$, $V_{A_3}^{\max} = (A_3^{\max} - A_3^{\min})/2$, $V_{\varphi_{12}}^{\max} = (\varphi_{12}^{\max} - \varphi_{12}^{\min})/3$, and $V_{\varphi_{13}}^{\max} = (\varphi_{13}^{\max} - \varphi_{13}^{\min})/3$.

At the earlier stage, we manually searched for the optimal CPG parameters in an exhaustive way. The sketch map of the experiments is shown in Fig. 18 and the experiment process is described as follows. Through wireless control, we modulated the robot to the corner of the tank and then commanded the robot to perform forward swimming. The speed was precisely evaluated by the vision tracking platform [36]. Finally, the robot was stopped to avoid possible wall effects when it approached the edge of the tank. Each

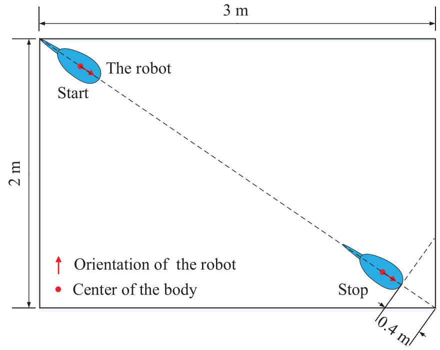


Fig. 18. Sketch map of the manual experiments for best CPG parameter.

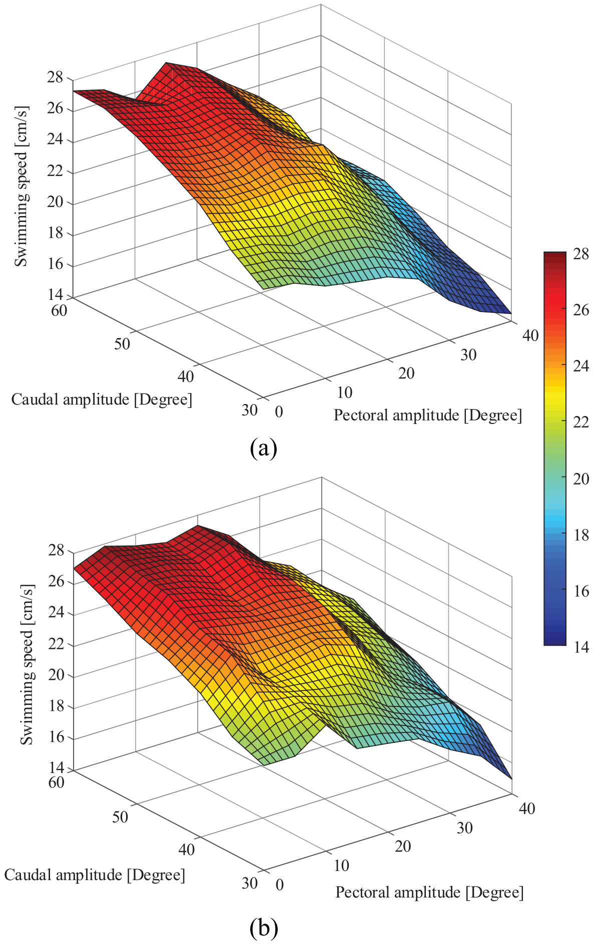
TABLE II
RANGES OF THE CPG PARAMETERS

Frequency	1 Hz	1.5 Hz	2 Hz	2.5 Hz
$A_1(^\circ)$	$0\sim30$	$0\sim30$	$0\sim30$	$0\sim30$
$A_3(^\circ)$	$0\sim60$	$0\sim50$	$0\sim40$	$0\sim30$
$\varphi_{12}(^\circ)$	$-180\sim180$	$-180\sim180$	$-180\sim180$	$-180\sim180$
$\varphi_{13}(^\circ)$	$-180\sim180$	$-180\sim180$	$-180\sim180$	$-180\sim180$

set of CPG parameters was evaluated in five trials, and its final speed was the average of the five scores. We fixed the frequency and phase difference of the CPG parameters in the experiments to reduce the number of the trials. In particular, we searched for the maximum speed for two groups of CPG parameters: 1) $f = 1$ Hz, $X_i = 0^\circ$, $\varphi_{12} = 0^\circ$, $\varphi_{13} = 0^\circ$, $A_1(^\circ) \in \{0, 5, 20, 25, 20, 25, 30, 35, 40\}$, and $A_3(^\circ) \in \{30, 35, 40, 45, 50, 55, 60\}$ and 2) $f = 1$ Hz, $X_i = 0^\circ$, $\varphi_{12} = 180^\circ$, $\varphi_{13} = 90^\circ$, $A_1(^\circ) \in \{0, 5, 20, 25, 20, 25, 30, 35, 40\}$, and $A_3(^\circ) \in \{30, 35, 40, 45, 50, 55, 60\}$. It took 630 trials to complete these experiments and took more than 10 h. Fig. 19 shows the obtained speeds with the relevant control parameters. The maximum speed for the first group of CPG parameters is 27.34 cm/s with the parameters $\{A_1 = 0^\circ, A_3 = 60^\circ\}$ and the maximum speed for the second group is 27.85 cm/s with the parameters $\{A_1 = 5^\circ, A_3 = 60^\circ\}$. Assuming the maximum speed is sought from the following integral parameter spacing 10° $\{A_1 \in [0^\circ, 30^\circ], A_3 \in [30^\circ, 60^\circ], \varphi_{12} \in [-180^\circ, 180^\circ]$, and $\varphi_{13} \in [-180^\circ, 180^\circ]\}$, the evaluations would totally take about 72 days for a particular beating frequency. It is obvious that an efficient autonomous gait optimization system is extremely required.

B. Autonomous Gait Optimization Results

As mentioned above, the autonomous optimization experiments were conducted for four frequencies: 1) $f = 1$ Hz; 2) $f = 1.5$ Hz; 3) $f = 2$ Hz; and 4) $f = 2.5$ Hz. The robot navigated in the small tank, evaluated its gaits and optimized its speeds all by itself. The results were recorded online in an SD card by the onboard Linux operating system. The optimization took 100 ($N_p \times N_g$) evaluations for each frequency.



(a)

(b)

Fig. 19. Manually evaluated speeds at $f = 1$ Hz for phase differences (a) $\varphi_{12} = 0^\circ$, $\varphi_{13} = 0^\circ$ and (b) $\varphi_{12} = 180^\circ$, $\varphi_{13} = 90^\circ$, respectively.

Fig. 20 shows the individual, best and average fitness scores (i.e., the speed) for the population of particles. For each generation, the best score represents the highest speed of the ten particles, and the average score stands for the average speed of the ten particles. Both best and average fitness scores increase gradually. At $f = 1$ Hz, the algorithm sought out the highest speed of 31.21 cm/s (0.7803 BL/s) with the corresponding parameters $\{A_1 = 5.3^\circ, A_3 = 60.0^\circ, \varphi_{12} = 134.7^\circ, \varphi_{13} = -115.3^\circ\}$. The "best" speed here is higher than that of the speed (27.85 cm/s) found by an exhaustive way. One possible reason is that the phase differences contribute the speed, but they are fixed in the manual experiments. Similarly, the algorithm developed the best speeds of 37.40 cm/s (0.9350 BL/s), 40.42 cm/s (1.011 BL/s), and 34.37 cm/s (0.8593 BL/s), respectively, for $f = 1.5$ Hz, $f = 2.0$ Hz, and $f = 2.5$ Hz with the corresponding optimal parameters $\{A_1 = 8.9^\circ, A_3 = 48.9^\circ, \varphi_{12} = 146.2^\circ, \varphi_{13} = -70.4^\circ\}$, $\{A_1 = 8.1^\circ, A_3 = 39.8^\circ, \varphi_{12} = 139.7^\circ, \varphi_{13} = -98.0^\circ\}$, and $\{A_1 = 9.6^\circ, A_3 = 29.9^\circ, \varphi_{12} = 35.0^\circ, \varphi_{13} = 131.3^\circ\}$. It can be seen that the best and average scores have approached to the optimal speeds from the sixth generation/iteration. It indicates that the algorithm is searching in the right direction and would discover the best value ultimately. These results demonstrate the effectiveness of the developed autonomous gait optimization system for underwater robots. It is interesting that the optimal speed increases along with the beating frequency from 1 to 2 Hz, which is consistent with the previous experimental and theoretical studies [25], [38], [41]. However, the optimal speed decreases from

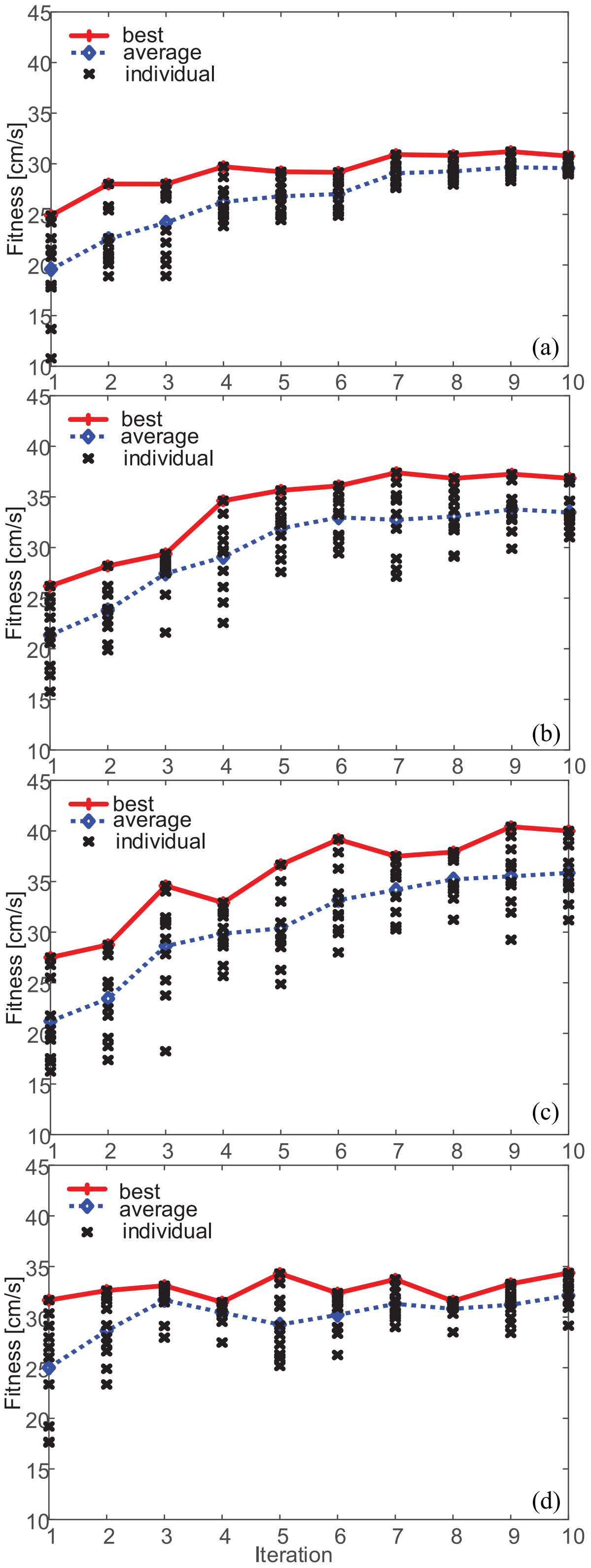


Fig. 20. Individual, best and average fitness scores in the optimization experiments. (a) $f = 1$ Hz. (b) $f = 1.5$ Hz. (c) $f = 2$ Hz. (d) $f = 2.5$ Hz.

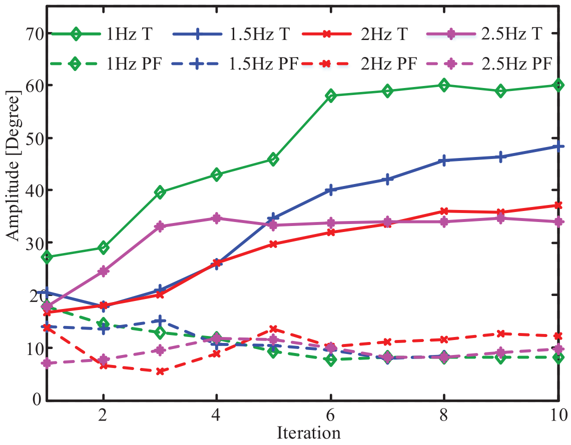


Fig. 21. Average values of the fin amplitude in the experiments. In the figure, “T” represents tail and “PF” represents pectoral fin.

2 to 2.5 Hz. Strouhal number (St) is an important dimensionless quantity observed in nature. Triantafyllou *et al.* [42] argued that fishes exhibit maximum propulsive efficiency in the interval $0.25 < \mathrm{St} < 0.35$. We calculate the Strouhal number of the tail for the four frequencies. They are 0.277, 0.302, 0.317, and 0.363, respectively, for 1, 1.5, 2, and 2.5 Hz. It indicates that the swimming efficiency of the robotic fish should be less inefficient at the frequency of 2.5 Hz.

The evolution of the fin amplitude is further analyzed. As shown in Fig. 21, it is remarkable that the tail amplitude gradually gets larger while the magnitude of the pectoral fins gets smaller along with the evolution. This partly confirms the biology-inspired concept that fish primarily rely on the caudal fin for propulsion in fast swimming and employ the paired fins for maneuvering and stabilization [43]–[45]. One possible reason could be that the large-amplitude movements of pectoral fins disturb the mechanism of force generation of the tail. This speculation is also supported by the experimental results in Fig. 19 that the robot speed declines as the amplitude of the pectoral fins get larger. Moreover, this phenomenon emerging on our boxfish-like robot is consistent with the behavior of its natural counterpart that boxfish always beats its tail fin while adducting its pectoral fins in fast swimming [46]. The fact that the tail is the main power in fish swimming has been well recognized and demonstrated in previous studies [25], [47]. However, it is the first time to the best of our knowledge that this swimming mechanism is explored from an evolutionary perspective.

C. Discussion

We have developed an autonomous learning framework containing a CPG, a sensory processing center and an evolutionary algorithm for underwater robots. The main contributions are as follows.

1) The autonomous optimization framework for fast learning in the water.
2) The algorithms of navigation and gait evaluation for underwater robots.
3) The onboard implementation of a CPG controller that offers an ideal substrate for gait generation and online optimization.

We have strictly proved the stability of the presented CPG model. As established in Section IV, the CPG model can smoothly adapt to parameter changes and synchronize to the desired phase difference between oscillators. This characteristic makes the CPG controller an ideal building block that offers the robot to run the learning algorithm in parallel with the CPG in real time. The experiments have demonstrated that the robot is able to optimize its swimming speeds autonomously. The approach is significantly faster than that of the systematic gait evaluations. For instance, a systematic exploration of the 4-D parameter space with ten steps for each parameter requires 10 000 evaluations in our experiments, whereas the learning method can achieve a very similar result with only 100 evaluations. Moreover, the optimal gaits also confirm a concept that fish rely on the caudal fin for propulsion in fast swimming, which indicates an alternative approach to the study of swimming mechanism of aquatic animals. Multiple onboard sensors and related algorithms are collectively formed a sensory processing center helping the robot to navigate and assess its gaits with great autonomy. The adopted sensors are low-cost and consumes little energy. Therefore, the proposed method is easy to be implemented on small underwater robots. Moreover, pressure sensors are adopted to estimate the speed of the underwater robot, which provides a new solution for the speed evaluation of small underwater robots. The current data-driven method for speed evaluation has limitations for general application, and we are now investigating the model-based approach.

The proposed approach is more practical for underwater robots in contrast with the simulation-based optimization in [19] and [20] as mentioned in the introduction. We believe that the empirical approach is the only reliable way for the gait optimization of underwater robots because the unstructured water environment cannot be modeled accurately at present. Moreover, the proposed approach does not need the assistance from the external processing centers or the experimenters compared with the experiment-based optimization methods in [21]–[23]. It can be performed with minimal human attention, and therefore, has the highest autonomy. The pair of the landmarks is used to guide the robot to carry out the candidate gait as long as possible in the confined space. The two landmarks could be removed if the robot is operated in a wide water space. The robot can just perform the candidate gait for a period while avoiding the boundaries of the water area.

IX. Conclusion

In this paper, we have developed an autonomous gait optimization system for underwater robots for the first time. The optimization system is mainly based on three components: 1) an artificial CPG controller; 2) a sensory processing center; and 3) an evolutionary algorithm. The CPG controller is responsible for generating locomotion gaits. Environment perception and gait evaluation are first tackled by the sensory processing center with the help of multiple onboard sensors. In particular, the camera, IMU, and infrared sensor are collectively adopted to help the robot navigate in the environment. The pressure sensors are introduced to evaluate the gait quality of the underwater robot. The forward speed optimization experiments have demonstrated the effectiveness of the proposed autonomous optimization approach. Through the optimization experiments, the robot has achieved a maximum speed of 1.011 BL/s. Fast swimming ability is not enough for underwater robots surviving in an unknown water environment. Other survival skills, such as the swimming efficiency and flexibility are equally crucial to underwater robots. Therefore, we will improve our system to optimize the swimming flexibility and efficiency of the robotic fish in our future work.

Acknowledgment

The authors would like to thank Z. Cheng for his contributions in carrying out the experiments and L. Li and C. Wang for their valuable discussions. They would also like to thank the anonymous reviewers for their valuable comments and suggestions on improving this paper.

References

[1] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: A review," *Neural Netw.*, vol. 21, no. 4, pp. 642–653, 2008.

[2] F. Liu, K.-M. Lee, and C.-J. Yang, "Hydrodynamics of an undulating fin for a wave-like locomotion system design," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 3, pp. 554–562, Jun. 2012.

[3] J. Yu, M. Tan, J. Chen, and J. Zhang, "A survey on CPG-inspired control models and system implementation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 441–456, Mar. 2014.

[4] J. Yu, M. Tan, S. Wang, and E. Chen, "Development of a biomimetic robotic fish and its control algorithm," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 4, pp. 1798–1810, Aug. 2004.

[5] H. Hu, J. Liu, I. Dukes, and G. Francis, "Design of 3D swim patterns for autonomous robotic fish," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Beijing, China, 2006, pp. 2406–2411.

[6] Y. Hu, W. Zhao, and L. Wang, "Vision-based target tracking and collision avoidance for two autonomous robotic fish," *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp. 1401–1410, May 2009.

[7] C. Zhou and K. H. Low, "Design and locomotion control of a biomimetic underwater vehicle with fin propulsion," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 1, pp. 25–35, Feb. 2012.

[8] F. Zhang, J. Thon, C. Thon, and X. Tan, "Miniature underwater glider: Design and experimental results," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 1, pp. 394–399, Feb. 2014.

[9] Z. Su, J. Yu, M. Tan, and J. Zhang, "Implementing flexible and fast turning maneuvers of a multijoint robotic fish," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 1, pp. 329–338, Feb. 2014.

[10] W. Wang and G. Xie, "Online high-precision probabilistic localization of robotic fish using visual and inertial cues," *IEEE Trans. Ind. Electron.*, vol. 62, no. 2, pp. 1113–1124, Feb. 2015.

[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, no. 5817, pp. 1416–1420, 2007.

[12] J. Yu, M. Wang, M. Tan, and J. Zhang, "Three-dimensional swimming," *IEEE Robot. Autom. Mag.*, vol. 18, no. 4, pp. 47–58, Dec. 2011.

[13] D. Gong, J. Yan, and G. Zuo, "A review of gait optimization based on evolutionary computation," *Appl. Comput. Intell. Soft Comput.*, vol. 2010, Jan. 2010, Art. no. 413179.

[14] M. Oliveira, V. Matos, C. P. Santos, and L. Costa, "Multi-objective parameter CPG optimization for gait generation of a biped robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Karlsruhe, Germany, 2013, pp. 3130–3135.

[15] G. S. Hornby, S. Takamura, T. Yamamoto, and M. Fujita, "Autonomous evolution of dynamic gaits with two quadruped robots," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 402–410, Jun. 2005.

[16] D. Marbach and A. J. Ijspeert, "Online optimization of modular robot locomotion," in *Proc. IEEE Int. Conf. Mechatronics Autom.*, vol. 1. 2005, pp. 248–253.

[17] A. Sproewitz, R. Moeckel, J. E. R. O. Maye, and A. J. Ijspeert, "Learning to move in modular robots using central pattern generators and online optimization," *Int. J. Robot. Res.*, vol. 27, nos. 3–4, pp. 423–443, 2008.

[18] R. Moeckel *et al.*, "Gait optimization for roombots modular robots—Matching simulation and reality," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Tokyo, Japan, 2013, pp. 3265–3272.

[19] I.-B. Jeong, C.-S. Park, K.-I. Na, S. Han, and J.-H. Kim, "Particle swarm optimization-based central patter generator for robotic fish locomotion," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, New Orleans, LA, USA, 2011, pp. 152–157.

[20] Z. Wu, J. Yu, and M. Tan, "CPG parameter search for a biomimetic robotic fish based on particle swarm optimization," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2012, pp. 563–568.

[21] A. Crespi and A. J. Ijspeert, "Online optimization of swimming and crawling in an amphibious snake robot," *IEEE Trans. Robot.*, vol. 24, no. 1, pp. 75–87, Feb. 2008.

[22] C. Zhou and K. H. Low, "On-line optimization of biomimetic undulatory swimming by an experiment-based approach," *J. Bionic Eng.*, vol. 11, no. 2, pp. 213–225, Apr. 2014.

[23] C. Wang, G. Xie, L. Wang, and M. Cao, "CPG-based locomotion control of a robotic fish: Using linear oscillators and reducing control parameters via PSO," *Int. J. Innov. Comput. Inf. Control*, vol. 7, no. 7B, pp. 4237–4249, 2011.

[24] J. C. Kahn and J. L. Tangorra, "Application of a micro-genetic algorithm for gait development on a bio-inspired robotic pectoral fin," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Tokyo, Japan, 2013, pp. 3784–3789.

[25] W. Wang, J. Guo, Z. Wang, and G. Xie, "Neural controller for swimming modes and gait transition on an ostraciiform fish robot," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Wollongong, NSW, Australia, 2013, pp. 1564–1569.

[26] J. Zhang, W. Wang, G. Xie, and H. Shi, "Camera-IMU-based underwater localization," in *Proc. 33rd Chin. Control Conf. (CCC)*, Nanjing, China, 2014, pp. 8589–8594.

[27] F. Delcomyn, "Neural basis of rhythmic behavior in animals," *Science*, vol. 210, no. 4469, pp. 492–498, 1980.

[28] R. Cui, C. Yang, Y. Li, and S. Sharma, "Adaptive neural network control of AUVs with control input nonlinearities using reinforcement learning," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2016.2645699.

[29] G. Endo, J. Morimoto, T. Matsubara, J. Nakanishi, and G. Cheng, "Learning CPG-based biped locomotion with a policy gradient method: Application to a humanoid robot," *Int. J. Robot. Res.*, vol. 27, no. 2, pp. 213–228, 2008.

[30] J. Yu, R. Ding, Q. Yang, M. Tan, and J. Zhang, "Amphibious pattern design of a robotic fish with wheel-propeller-fin mechanisms," *J. Field Robot.*, vol. 30, no. 5, pp. 702–716, 2013.

[31] K. Matsuoka, "Mechanisms of frequency and pattern control in the neural rhythm generators," *Biol. Cybern.*, vol. 56, nos. 5–6, pp. 345–353, 1987.

[32] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1465–1476, Sep. 2004.

[33] Z. Liu, F. Wang, and Y. Zhang, "Adaptive visual tracking control for manipulator with actuator fuzzy dead-zone constraint and unmodeled dynamic," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 10, pp. 1301–1312, Oct. 2015.

[34] F. Sun, C. Liu, W. Huang, and J. Zhang, "Object classification and grasp planning using visual and tactile sensing," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 7, pp. 969–979, Jul. 2016.

[35] W. Wang and G. Xie, "An adaptive and online underwater image processing algorithm implemented on miniature biomimetic robotic fish," in *Proc. 19th World Congr. Int. Federation Autom. Control*, Cape Town, South Africa, 2014, pp. 7598–7603.

[36] L. Shao and G. Xie, "Real-time tracking of moving objects on a water surface," in *Proc. Int. Conf. Mechatronics Autom.*, Chengdu, China, 2012, pp. 2114–2119.

[37] W. Premerlani and P. Bizard, "Direction cosine matrix IMU: Theory," *DIY DRONE: USA*, 2009. [Online]. Available: http://lars.mec.ua.pt/public/LAR%20Projects/RescueRobotics/2009_MauroSimoes/REFERENCIAS2/DCMDraft2.pdf

[38] W. Wang, G. Xie, and H. Shi, "Dynamic modeling of an ostraciiform robotic fish based on angle of attack theory," in *Proc. IEEE World Congr. Comput. Intell. (WCCI)*, 2014, pp. 3944–3949.

[39] C. Niehaus, T. Rofer, and T. Laue, "Gait-optimization on a humanoid robot using particle swarm optimization," in *Proc. 2nd Workshop Humanoid Soccer Robots IEEE-RAS 7th Int. Conf. Humanoid Robots*, 2007, pp. 1–7.

[40] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Perth, WA, Australia, 1995, pp. 1942–1948.

[41] Y. Hu, W. Zhao, G. Xie, and L. Wang, "Development and target following of vision-based autonomous robotic fish," *Robotica*, vol. 27, no. 7, pp. 1075–1089, 2009.

[42] G. S. Triantafyllou, M. S. Triantafyllou, and M. A. Grosenbaugh, "Optimal thrust development in oscillating foils with application to fish propulsion," *J. Fluids Struct.*, vol. 7, no. 2, pp. 205–224, 1993.

[43] M. Sfakiotakis, D. M. Lane, and J. B. C. Davies, "Review of fish swimming modes for aquatic locomotion," *IEEE J. Oceanic Eng.*, vol. 24, no. 2, pp. 237–252, Apr. 1999.

[44] J. E. Colgate and K. M. Lynch, "Mechanics and control of swimming: A review," *IEEE J. Oceanic Eng.*, vol. 29, no. 3, pp. 660–673, Jul. 2004.

[45] E. G. Drucker and G. V. Lauder, "A hydrodynamic analysis of fish swimming speed: Wake structure and locomotor force in slow and fast labriform swimmers," *J. Exp. Biol.*, vol. 203, pp. 2379–2393, Aug. 2000.

[46] J. R. Hove, L. M. O'Bryan, M. S. Gordon, P. W. Webb, and D. Weihs, "Boxfishes (Teleostei: Ostraciidae) as a model system for fishes swimming with many fins: Kinematics," *J. Exp. Biol.*, vol. 204, no. 8, pp. 1459–1471, 2001.

[47] R. Ding, J. Yu, Q. Yang, and M. Tan, "Dynamic modelling of a CPG-controlled amphibious biomimetic swimming robot," *Int. J. Adv. Robot. Syst.*, vol. 10, pp. 1–11, Apr. 2013.

Wei Wang received the B.E. degree in automatic control from the University of Electronic Science and Technology of China, Chengdu, China, in 2010, and the Ph.D. degree in biorobotics from Peking University, Beijing, China, in 2016.

He is a Post-Doctoral Researcher with the SENSEable City Laboratory and the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA. His current research interests include theory and application of robotics for design, modeling, control and interaction, biorobots, underwater robots, robot control, multirobot control, machine learning, and collective behavior and their application to navigation and environmental sensing.

Dongbing Gu received the B.Sc. and M.Sc. degrees in automatic control from the Beijing Institute of Technology, Beijing, China, in 1985 and 1988, respectively, and the Ph.D. degree in robotics from the University of Essex, Colchester, U.K., in 2004.

He was a Lecturer, an Associate Professor, and a Professor with the Changchun Institute of Optics and Fine Mechanics, Changchun, China, from 1988 to 1999. He was an Academic Visiting Scholar with the Department of Engineering Science, University of Oxford, Oxford, U.K., from 1997 to 1998. In 2000, he joined the University of Essex, as a Lecturer, where he has been a Professor with the School of Computer Science and Electronic Engineering, since 2012. He has published over 200 papers in international conferences and journals. His research has been supported by the Royal Society, EPSRC, EU, and industries. His current research interests include robotics, autonomous systems, navigation and control, mapping and localization, cooperative control, and machine learning.

Dr. Gu is an Editorial Board Member of *Cognitive Computation* and *Frontiers in Robotics and AI: Multirobot Systems*.

Guangming Xie received the B.S. degrees in applied mathematics and electronic and computer technology and the M.E. and Ph.D. degrees in control theory and control engineering from Tsinghua University, Beijing, China, in 1996, 1998, and 2001, respectively.

He was as a Post-Doctoral Research Fellow with the Center for Systems and Control, Department of Mechanics and Engineering Science, Peking University, Beijing, China, from 2001 to 2003. He is currently a Full Professor of Dynamics and Control with the College of Engineering, Peking University, where he joined the Center, as a Lecturer, in 2003. His current research interests include smart swarm theory, multiagent systems, multirobot cooperation, biomimetic robot, switched and hybrid systems, and networked control systems.

Dr. Xie is an Associate Editor of *Scientific Reports*, the *International Journal of Advanced Robotic Systems*, and *Mathematical Problems in Engineering*, and an Editorial Board Member of the *Journal of Information and Systems Science*.