International Journal of Innovative Computing, Information and Control
Volume **7**, Number **7(B)**, July **2011**

ICIC International ©2011 ISSN 1349-4198
pp. **4237–4249**

CPG-BASED LOCOMOTION CONTROL OF A ROBOTIC FISH: USING LINEAR OSCILLATORS AND REDUCING CONTROL PARAMETERS VIA PSO

Chen Wang[1,2], Guangming Xie[1,3], Long Wang[1] and Ming Cao[2]

[1]State Key Laboratory of Turbulence and Complex Systems
Intelligent Control Laboratory
Center for Systems and Control
College of Engineering
Peking University
No. 5, Yiheyuan Road, Haidian District, Beijing 100871, P. R. China
{ wangchen; xiegming; longwang }@pku.edu.cn

[2]Faculty of Mathematics and Natural Sciences, ITM
University of Groningen
Nijenborgh 4, 9747 AG Groningen, The Netherlands
{ chen.wang; m.cao }@rug.nl

[3]School of Electrical and Electronics Engineering
East China Jiaotong University
Changbei Developing Zone, Nanchang 330013, P. R. China

Received May 2010; revised November 2010

Abstract. *The aim of the present study is to investigate the locomotion control of a robotic fish. To achieve this goal, we design a control architecture based on a novel central pattern generator (CPG) and implement it as a system of coupled linear oscillators. This design differs significantly from the usual CPG models in which nonlinear oscillators are commonly used. While our CPG keeps all the basic features of its biological counterparts and is capable of producing coordinated patterns of rhythmic activity, thanks to the linearity of the oscillators used, the computational costs of the CPG are greatly reduced and all the structural parameters can be selected easily. In addition to the proposed CPG model, the complete control architecture in our study also contains a transition layer, which is used to transform higher level control commands into accessible inputs to the CPG. Moreover, particle swarm optimization (PSO) is implemented to reduce the control parameters. As a result, as few as two control parameters, including the frequency for speed control and the offset of the motors for direction control, are sufficient for the whole locomotion control implementation. Additionally, a transition layer makes the locomotion control implementation simple and straightforward. Results from both simulation and experiment demonstrate the efficiency of the proposed CPG-based locomotion control approach.*

Keywords: Robotic fish, Locomotion control, Central pattern generator (CPG), Particle swarm optimization (PSO), Linear oscillator, Transition layer

1. **Introduction.** Organisms have probably existed in the world for hundreds of millions of years. Their perfect physical structures and excellent locomotion properties emerging from the continuous long-period evolution fascinate all researchers who hope to design better mobile robots. Recent developments in bionics, materials, computation, electronics and fabrication technologies offer researchers an unprecedented opportunity to design novel mobile robots based on inspiration from animals.

Among all kinds of aquatic organisms, fish is always paid more attention to and often imitated to design underwater robots. The swimming modes of fish can roughly be divided into four categories: anguilliform mode, carangiform mode, ostraciiform mode and labriform mode [1]. According to this classification, a variety of biomimetic underwater robots have been constructed. Most of them are designed based on the anguilliform swimming mode and the carangiform mode, such as a lamprey-like robot [2], a salamander-like robot [3] and the well-known RoboTuna [4]. However, few underwater robots implement the ostraciiform or the labriform modes, with the box-fish-like robot BoxyBot being an example [5].

To replicate the outstanding locomotion skills of fish, locomotion control mechanism plays a key role in designing a robotic fish. The main difficulty of the locomotion control comes from how to coordinate multiple degrees of freedom of a robot and how to generate a real-time locomotion pattern.

In the past, the control mechanisms of fish robots were commonly based on model approaches [6, 7] and sine approaches [8, 9].

Model-based approaches use kinematic [6] or dynamic models [7] of the animal or the robot to design mechanisms for locomotion control. In these approaches, once the kinematic or dynamic models turn out to be inaccurate, or in some cases even wrong, the performance of the controllers will be unreliable. The high computational cost is another drawback of model-based approaches.

Sine-based approaches use simple sine functions to generate traveling waves. Usually, the propulsive wave traverses the fish-like robot body in a direction opposite to the overall movement and at a speed greater than the overall swimming speed to propel the robot. Compared with model-based approaches, sine-based approaches require lower computational costs and are thus convenient for online gait generation. Another advantage of sine-based approaches is that important quantities, such as frequency and amplitude, are explicitly defined and easily to be controlled. However, it is noteworthy that online modifications of the parameters of the sine function will result in discontinuity of the output signals, leading to jerky locomotion.

Recently, central pattern generator (CPG) based methods are increasingly used in controlling a variety of different types of robots and different modes of locomotion. As biologically inspired approaches, CPGs are essential building blocks for the locomotion neural circuits found in both invertebrates and vertebrates [10].

A major feature of CPGs is the capability of producing coordinated patterns of rhythmic activity without any rhythmic inputs from sensory feedback or high-level control signals [11, 12]. There have been a number of projects using CPGs for controlling legged robots [13], amphibian robots [3, 14], swimming robots [5, 15], etc.

In [3, 5, 14], Ijspeert proposed a novel CPG model for locomotion control of robots, which is a system of coupled nonlinear amplitude-controlled phase oscillators. This CPG model is not to model a particular biological system or only replicate biological principles at an abstract level. The main outcome of this novel model that differs from other CPG-based models is that its limit cycle behavior has an analytical solution with explicit frequency, amplitude and phase lag parameters, which can be used as control parameters [14].

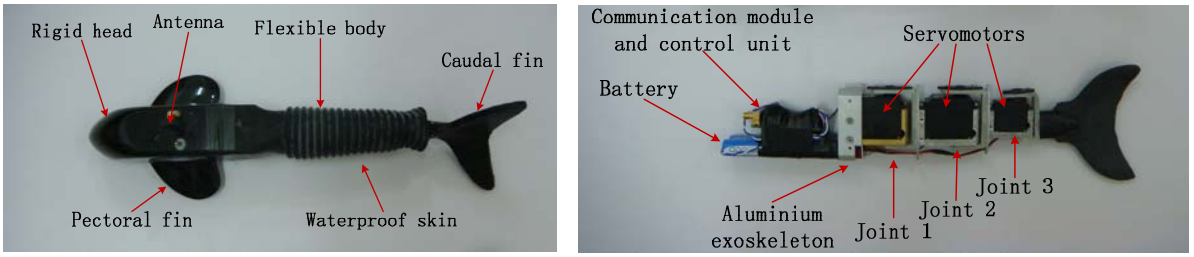
Based on the novel model described above, in the present study, we set out to design a much simpler CPG model for locomotion control of robots. The key point in our CPG model is to use linear oscillators instead of nonlinear ones. Although only simple oscillators are adopted, we show that the performance of our CPG model is similar to that of Ijspeert's. Moreover, the structural parameters in our model can be selected more reasonably and easily according to the request of the dynamic performance of the CPG.

A complete control architecture, mainly composed of the proposed CPG model and a transition layer, is built up. The transition layer is used to transform higher level control commands into the specific control inputs to the CPG for producing suitable coordinated patterns of rhythmic activity. Furthermore, in order to reduce the control parameters, a particle swarm optimization (PSO) based method is used to find an optimal point in the parameter space, where a maximum speed can be achieved. As a result, the speed control is implemented by simply modulating the frequency of the joints. A large number of numerical simulations and physical experiments are performed to test our CPG model and locomotion control method. The performance is reliable and the application is easy and simple.

The rest of the paper is structured as follows: in Section 2, we describe the robotic fish and the experimental platform, which is used to verify our theoretic results; a locomotion control architecture, which is based on our CPG model and a transition layer, is presented in Section 3; the PSO method, which is used to design the transition layer of the locomotion control method, is described in Section 4; the experimental results are shown in Section 5; finally, Section 6 concludes the paper.

2. **Robotic Fish and Experimental Platform.**

Our CPG-based locomotion method is verified by a fish-like robot. Both the robot and the experimental platform are developed by ourselves [16, 17, 18].



(a) (b)

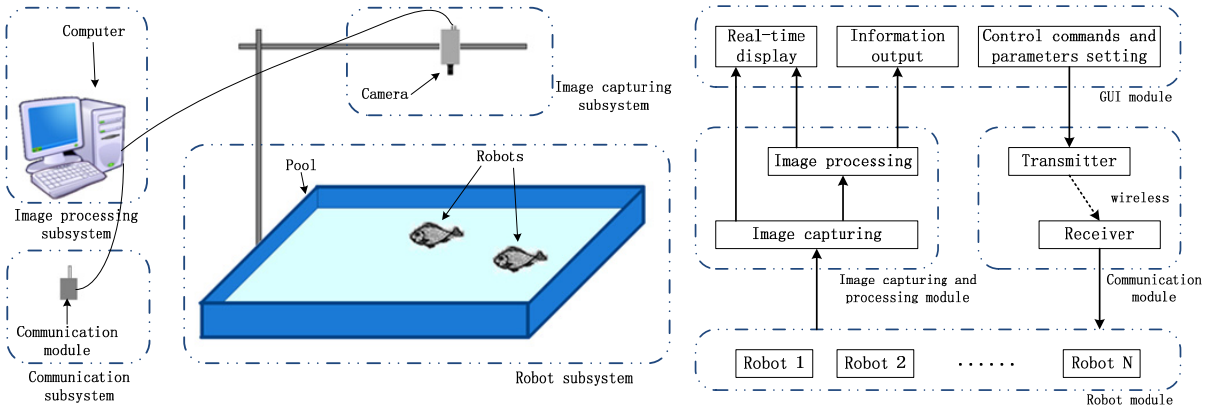
FIGURE 1. Mechanical configuration of a biomimetic robotic fish, (a) prototype of the fish (top view), (b) internal structure of the fish (side view)

2.1. **Robotic fish.**

As shown in Figure 1, the robotic fish consists of three parts: a streamlined head, a flexible body and a caudal fin. The rigid head, made of fiberglass, accommodates an onboard control unit, a duplex wireless serial port communication module and a set of rechargeable batteries. The battery pack is placed at the bottom of the head to lower the center of gravity so that the static vertical stability of the robot can be achieved (Figure 1(b)). A pair of pectoral fins is fixed on both sides of the head to ensure the roll stability of the robot during swimming. This pair of fins is immovable, so the locomotion of the robot is limited to a horizontal plane. The flexible part of the robot is composed of three joints which are concatenated by aluminium exoskeletons. Each joint is connected to a R/C servomotor, which is used to change the deflection angle of this joint. The rubber caudal fin, fixed on the third joint, is like a crescent. The special shape enhances the swimming performance of the robotic fish. To be waterproof, a tailor-made waterproof rubber tube covers the robot from head to tail. The total length of the robotic fish is $40cm$. In order to make the robot swimming just below the water surface, we inject

proper amount of air so that the density of the robot is just a little smaller than that of the water.

2.2. **Experimental platform.** The experimental platform is used to estimate real-time pose information of the fish-like robot (the position, the direction and the speed in this paper). Figure 2(a) depicts the hardware system which consists of an upper computer, an overhead camera, a communication module and a robot subsystem.



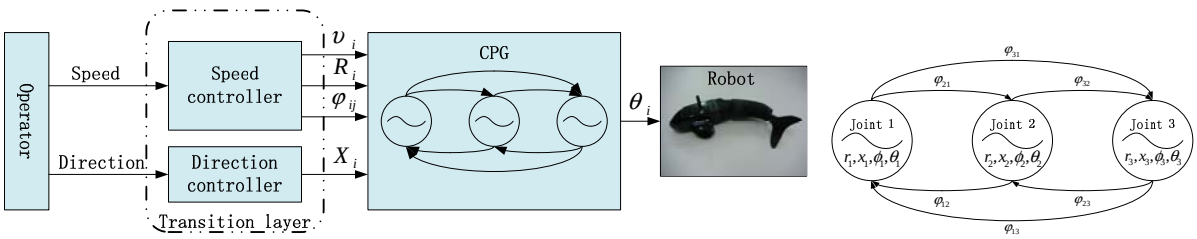
(a) (b)

FIGURE 2. Experimental platform, (a) hardware system, (b) architecture of software platform

Images of the pool are captured by the overhead camera per $40ms$ and then sent to the upper computer, where images are processed effectively to achieve the pose information of the robots. Through the wireless communication module, the upper computer sends control commands to robot and receives feedback information from the robot. The schematic diagram of the software system architecture is shown in Figure 2(b). It consists of four modules: GUI (Graphics User Interface) module, robot module, image capturing and processing module and communication module. Through the GUI, one can input control commands and locomotion parameters. Then, these signals are sent to the robot by the communication module in real time. In the image capturing and processing module, the pose information of the robot is calculated and displayed in the GUI window in real time.

3. **Locomotion Control.** The complete locomotion control architecture is shown in Figure 3(a). The transition layer, implemented at an upper computer, transforms simple control commands to specific CPG control parameters, and sends them wirelessly to the robot. The CPG-based model which is implemented at the robot can generate the control signals to adjust deflection angles of the joints.

3.1. **CPG model.** Our CPG model is a system of N coupled linear oscillators, where N is the number of the body joints and $N \geqslant 2$. In other words, there is a one-to-one correspondence between oscillators of the CPG-based model and joints of the robotic fish.



(a) (b)

FIGURE 3. (a) Diagram of the locomotion control architecture, (b) structure of the CPG-based model used in the robotic fish

The ith oscillator is implemented as follows:

$$\ddot{r}_i(t) = \alpha_i[\alpha_i(R_i - r_i(t)) - 2\dot{r}_i(t)] \tag{1}$$

$$\ddot{x}_i(t) = \beta_i[\beta_i(X_i - x_i(t)) - 2\dot{x}_i(t)] \tag{2}$$

$$\ddot{\phi}_i(t) = \sum_{j=1, j\neq i}^{N} \mu_{ij}\left[\mu_{ij}\left(\phi_j(t) - \phi_i(t) - \varphi_{ij}\right) - 2\left(\dot{\phi}_i(t) - 2\pi\upsilon_i\right)\right] \tag{3}$$

$$\theta_i(t) = x_i(t) + r_i(t)\cos(\phi_i(t)) \tag{4}$$

where the state variables $r_i(t)$, $x_i(t)$ and $\phi_i(t)$ represent, respectively, the amplitude, the offset and the phase of the ith oscillator at time t. The variable $\theta_i(t)$ is the output of oscillator i, i.e., the desired deflection angle of the corresponding joint i at time t. The parameters υ_i, R_i and X_i are the desired frequency, amplitude and offset of the oscillator i respectively. And the parameter φ_{ij} is the desired phase bias between oscillators i and j, which satisfies the following conditions: i) $\varphi_{ii} = 0$; ii) $\varphi_{ij} = -\varphi_{ji}$ and iii) $\varphi_{ik} + \varphi_{kj} = \varphi_{ij}$. Moreover, α_i, β_i and μ_{ij} are structural parameters that affect the transient dynamics. To simplify the model, we let $\alpha_i = \alpha$, $\beta_i = \beta$, $\mu_{ii} = 0$ (i.e., all oscillators have no self-couplings); $\mu_{ij} = \mu$ for $i \neq j$. The values of α, β and μ are positive and need to be designed later. In this paper, we use the same desired frequency $\upsilon_i = \upsilon$ for all oscillators (i.e., joints). Finally, the CPG model (1)-(4) can be written as follows:

$$\ddot{r}_i = -\alpha^2(r_i - R_i) - 2\alpha\dot{r}_i \tag{5}$$

$$\ddot{x}_i = -\beta^2(x_i - X_i) - 2\beta\dot{x}_i \tag{6}$$

$$\ddot{\phi}_i = -\mu^2 \sum_{j=1, j\neq i}^{N} (\phi_i - \phi_j - \varphi_{ji}) - 2(N-1)\mu\left(\dot{\phi}_i - 2\pi\upsilon\right) \tag{7}$$

$$\theta_i = x_i + r_i\cos(\phi_i). \tag{8}$$

To change the locomotion behavior of the robotic fish, the input signals (i.e., the desired frequency, amplitude and offset) are usually in the form of step signals. Thus, we focus on the performance of the step response of the proposed CPG-model in Subsections 3.2 and 3.3.

3.2. **Stability analysis.** Equations (5) and (6) are *critically damped second order linear systems*. Consider the general form of such a system:

$$\frac{1}{\omega^2}\ddot{y}(t) + \frac{2}{\omega}\dot{y}(t) + y(t) = u(t) \tag{9}$$

where $y(t)$ and $u(t)$ represent, respectively, the output and the input. The positive scalar ω is the damping coefficient of the system. Step response of the system changes monotonously with the exponential law, i.e., the output $y(t)$ asymptotically and monotonically converges to the input $u(t)$. Therefore, the amplitude r_i and the offset x_i of the oscillator converge, respectively, to the desired amplitude R_i and offset X_i asymptotically and monotonically.

In Equation (7), the phase ϕ_i is not only determined by oscillator i itself but also correlated with other oscillators. We say all oscillators achieve consensus if, for all $\phi_i(0)$ and $\dot{\phi}_i(0)$ and all i, $j = 1, \ldots, N$, $\phi_i(t) - \phi_j(t) \to \varphi_{ji}$ and $\dot{\phi}_i(t) \to 2\pi\upsilon$, as $t \to \infty$. We can prove that the subsystem (7) achieves consensus asymptotically for any $\mu > 0$. In other words, the phase bias between ϕ_j and ϕ_i converges to the desired phase bias φ_{ij} asymptotically. We omit the proof due to lack of space [18]. Equation (8) is designed to generate the rhythmic output signal θ_i, which is completely determined by the amplitude r_i, the offset x_i and the phase ϕ_i. Since all the amplitude r_i, the offset x_i and the phase ϕ_i can asymptotically and monotonically converge to their desired values from any initial conditions, the output θ_i of oscillator i is modulated smoothly. Based on the above regulations, oscillator i converges to the following limit cycle from arbitrary initial conditions asymptotically.

$$\theta_i^{\infty}(t) = X_i + R_i \cos(2\pi\upsilon t + \phi_0 + \varphi_{1i}), \quad i = 1, \ldots, N \tag{10}$$

where ϕ_0 is a constant depending on the initial conditions of the system.

3.3. **Structure parameters selection.** During the following discussion, we use $N = 3$ considering our three-joint robotic fish (Figure 3(b)). To make the locomotion performance of the robotic fish closer to those of real fish, suitable structure parameters of the CPG model must be designed.

After a large amount of numerical simulations, we observe that with the same α, the performances of the transient period are quite different when different values of υ are used. And the parameters β and μ have the similar results. Therefore, we assume that the values of the three parameters will change with the frequency υ to achieve better performance of the transient period. We re-analyze the transient performance of the unit step response of the general system (9). Unfortunately, we omit details for lack of space [18]. By simulation comparison and theoretical analysis, the parameters α, β and μ are selected according to the following equations:

$$\alpha = 11.68\upsilon \tag{11}$$

$$\beta = 11.68\upsilon \tag{12}$$

$$\mu = 5.84\upsilon. \tag{13}$$

Based on the above parameters selection, only four kinds of parameters (υ, φ_{ij}, R_i and X_i) need to be set when applied to the CPG model described by Equations (5)-(8), (11)-(13) to generate the desired traveling wave for swimming.

3.4. **Transition layer.** For convenience of controlling the locomotion of a robotic fish, we proposed a transition layer, composed of a direction controller and a speed controller, to reduce the complexity of the control commands.

3.4.1. *Direction controller.* The locomotion direction is influenced mainly by the three offset X_i ($i = 1, 2, 3$). Here, we use the offset to present the difference between left and right amplitudes of each oscillator. Thus, we use a simple direction controller described by $X_i = X$ ($i = 1, 2$) and $X_3 = 0$ which make $X(rad)$ be the only command for direction control. The equation $X_3 = 0$ is used to make the last joint of the robot have no offset

with the second joint to ensure enough power for propelling the robot. The command $X > 0$ means turning right and $X < 0$ means turning left.

3.4.2. *Speed controller.* The locomotion speed (only discuss swimming forwards in this section) is influenced jointly by the frequency υ, the amplitude R_i, the phase bias φ_{ij} and even the dynamic environment. Combined activities of all these factors make the design of the speed controller difficult. We do a lot of experiments with the robotic fish to identify how the speed changes when these parameters change. The observation is that the speed of locomotion monotonically increases with the frequency when all the parameters satisfied the physical limitation of servomotors. However, the amplitude R_i and phase bias φ_{ij} are shown a complex non-monotone influence on the speed of locomotion.

We therefore choose the frequency υ as the only command for speed control, and design a piecewise linear function $[R_i, \varphi_{ij}] = f_{speed}(\upsilon)$ as the speed controller.

The frequency is limited in $[0.5Hz, 3.0Hz]$ with the step size of $0.25Hz$. For each frequency, the PSO method is used to optimize the other five parameters (R_1, R_2, R_3, φ_{12} and φ_{13}) to obtain the maximum speed (see more details in Section 4).

The piecewise linear function, speed controller, is the linear interpolations between these optima acquired by PSO method. Based on this speed controller, all the specific control inputs of the CPG model (υ, R_i, φ_{ij}) can be obtained by a single control command υ. And the speed controller ensures that the outputs of the proposed CPG model remain close to the fastest gait for each frequency.

With this transition layer, the direction and the speed of the robotic fish can be simply controlled and modified by modulating the offset X (resp. the frequency υ) for direction (resp. for speed).

4. **Parameter Optimization via PSO.** Particle swarm optimization inspired by social behavior of bird flocking or fish schooling is a kind of stochastic optimization technique [19]. It was originally proposed in 1995 [20]. Similar to the Genetic Algorithm (GA), the PSO method searches for optimal solutions through iterations of a population of individuals, which are called a swarm of particles in PSO. However, the crossover and mutation operation are replaced with moving inside the solution space decided by the so-called velocity of each particle. Many published researches were made on the use of PSO method for developing gaits with better locomotion performance of quadruped robots [21, 22], biped robots [23, 24], modular robots [25], etc.

The PSO method has proved to be faster computed and easier implemented compared with many other optimization algorithms such as GA [26], and be effective in solving many global optimization problems [25, 27, 28, 29].

To improve convergence performance of PSO methods, inertia weight is introduced into PSO [30]. Furthermore, the inertia weight is dynamically adjusted in the PSO evolution process in order to balance global searching ability and convergence rate. This revised PSO with changing inertia weight is usually called Adaptive PSO (APSO).

In this paper, we used the APSO method to optimize the speed of the robotic fish.

The APSO iteration equation is given as follows:

$$v_{ij}^{k+1} = wv_{ij}^{k} + c_1 r_{1j}^{k} \left[p_{ij}^{k} - x_{ij}^{k}\right] + c_2 r_{2j}^{k} \left[p_{gj}^{k} - x_{ij}^{k}\right] \tag{14}$$

where w is the inertia weight (see [21] for details).

In this paper, we use a swarm of 10 particles. The state of particle i is a vector $(R_{i1}, R_{i2}, R_{i3}, \varphi_{i12}, \varphi_{i13})$ in a five-dimensional space, where R_{i1}, R_{i2} and R_{i3} present, respectively, the amplitude of each joint of robotic fish, φ_{i12} is the phase bias between joint 1 and joint 2, and φ_{i13} between joint 1 and joint 3.

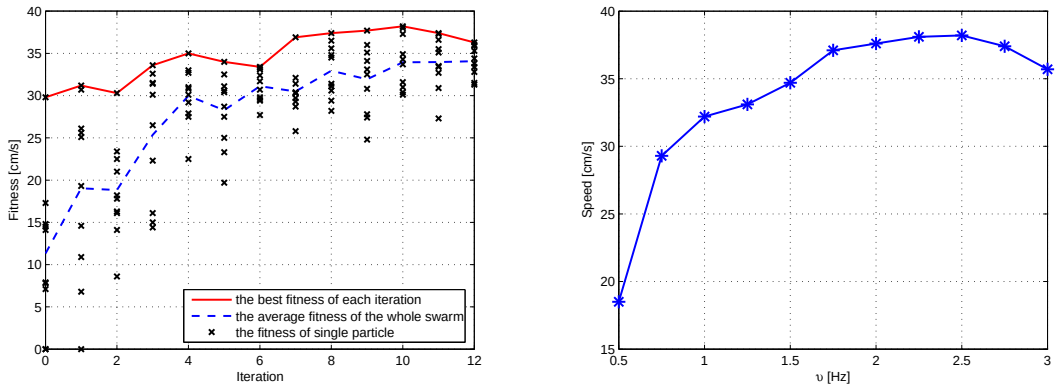
The spectrum of particle is determined by the physical limitation of servomotors. We use $c_1 = 1.8$ and $c_2 = 2$ to expedite the search process. The inertia weight w is described as the following equation.

$$w = 0.9 - \frac{0.9}{M} \times n, \quad n = 1, 2, \ldots, M \tag{15}$$

where n presents the iteration, and M is the maximum number of iterations (using $M = 12$ in this case). In this way, both global search in a broaden area at the beginning and intensive search in a currently effective area at the end can be realized.

The speed of the robotic fish is considered to be the fitness. However, the exact objective function cannot be given because the complexity dynamics result in having no accurate simulator for the robotic fish. Thus, we perform the APSO method on real robot within the experimental platform described above. To be more accurate, each result is the average of 5 speed measurements.

Using this APSO method described above, optimization experiments with the real robot are performed for frequencies from $0.5Hz$ to $3.0Hz$ with a step of $0.25Hz$. For instance, at the fixed frequency $\upsilon = 2.5Hz$, the other five controller parameters of the proposed CPG model ($R_1, R_2, R_3, \varphi_{12}$ and φ_{13}) have been optimized using the APSO method. Figure 4(a) shows the iteration process. The average fitness of all particles is converging quite



(a) (b)

FIGURE 4. (a) Results of the optimization for swimming at $\upsilon = 2.5Hz$. (b) Optimization results of speed for the frequencies between $0.5Hz$ and $3.0Hz$ (with a step of 0.25Hz). Each data point is the average of 5 speed measures.

fast from iteration 1 to 4. Then, the average fitness rises slowly but firmly until around iteration 10. The curve rises up to the maximum fitness (i.e., the forwards speed of the robotic fish) of $38.2cm/s$ (0.955 *body lengths*$/s$) at iteration 10. The parameter values for the best fitness are $R_1 = 0.42rad$, $R_2 = 0.31rad$, $R_3 = 0.40rad$, $\varphi_{12} = -0.70rad$ and $\varphi_{13} = -1.41rad$. Thus, we get a high-speed gait of the robot in a short time by using the above-mentioned APSO method.

Similarly, a series of optimization processes are taken to obtain the maximum speed and the corresponding parameters for each chosen frequency. The results of the optimization are shown in Figure 4(b). The maximum speed monotonically increases with the frequency υ up to $2.5Hz$. However, the speed begins to decrease when the frequency υ still increases after $2.5Hz$. The reason is that the amplitude of the servomotors is physical limited when

the frequency is too high. Thus, the maximum value of the swimming speed is $38.2cm/s$ (0.955 *body lengths*$/s$) when $\upsilon = 2.5Hz$. As a result, the speed controller of the transition layer can be obtained using the proposed method in Subsection 3.4.2.

5. **Simulations and Experiments.** For verifying the performance of the proposed CPG-based locomotion control method in Section 3, some simulations and experiments are completed in this section.

5.1. **Comparing with Ijspeert's model.** We first simulate our CPG model and Ijspeert's model [5] when the phase biases are modified. We can see that both of two models converge to the new limit cycle smoothly and quickly (Figure 5). The results show that the performance of our CPG model within linear differential equations is similar to that of Ijspeert's model [5] where nonlinear differential equations are used.

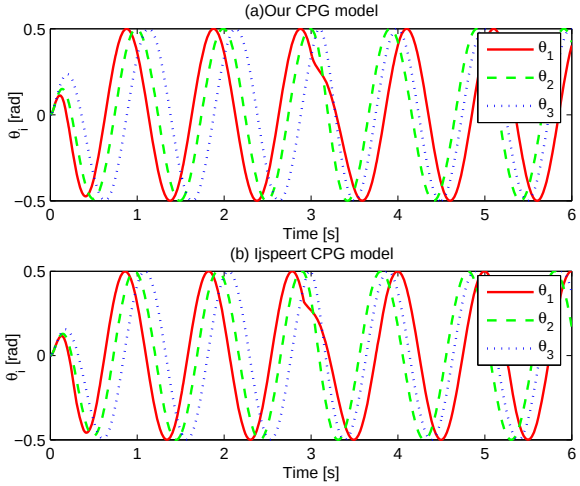


FIGURE 5. Comparing with Ijspeert's CPG model. (a) Output of our CPG model with the parameters $\upsilon = 1.0Hz$, $R_i = 0.5rad$, $X_i = 0$; (b) Output of the Ijspeert's model with the parameters of which are as follows: $\omega_i = 2\pi rad/s$, $R_i = 0.5rad$, $X_i = 0$, $a_r = a_x = 20rad/s$ $(i = 1, 2, 3)$ and $w_{ij} = 5/s(i \neq j)$. The phase biases $(\varphi_{12}, \varphi_{13})$ change from $(-0.7, -1.6)rad$ to $(1.2, 0.7)rad$ at $t = 3s$ (for our model) and at $t = 2.92s$ (for Ijspeert's model). The difference of time points is to make the mutation produced in the same position of the waveforms for convenience of comparison.

5.2. **Frequency variation.** We then simulate the proposed CPG model when the frequency changes (Figure 6). In this experiment, we take $R_i = 0.5rad$, $X_i = 0$ $(i = 1, 2, 3)$ and $(\varphi_{12}, \varphi_{13}) = (-0.7, -1.6)rad$. It is shown that when the frequency υ changes, the corresponding output still keep continuous and smoothly (Figure 6).

5.3. **Locomotion behaviors variation.** Based on the designed transition layer, locomotion behaviors of the robotic fish are divided into two basic categories. *Swimming forwards*, we set $\upsilon \in [0.5, 3]Hz$ and $X = 0$ so that the CPG model can generate traveling wave from head to tail to make the robot swim forwards. *Turning*, we set $\upsilon \in [0.5, 3]Hz$ and $X \neq 0$ so that the robot turns. When $X > 0$ the robot turns right, and when $X < 0$ it turns left.

Based on the above basic categories, we compose four typical locomotion behaviors to verify the performance of the CPG model: swimming forwards slowly, swimming forwards

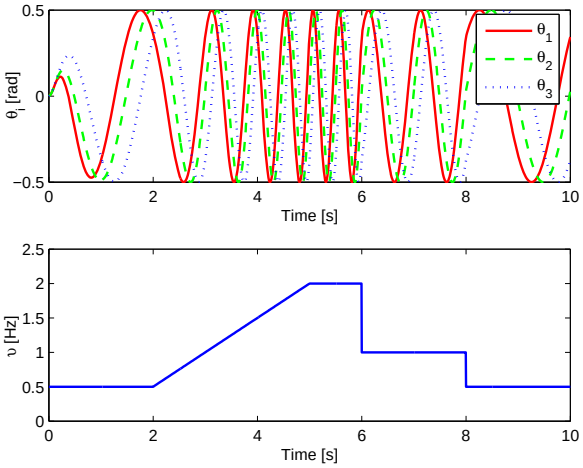


FIGURE 6. Output signals of the CPG model when frequency changes

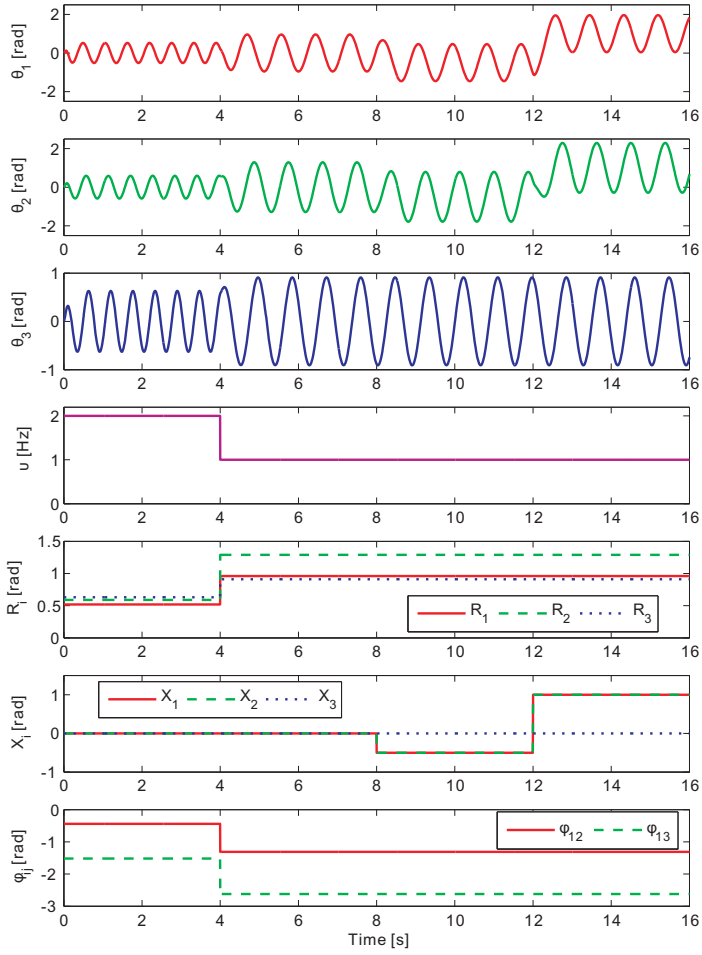
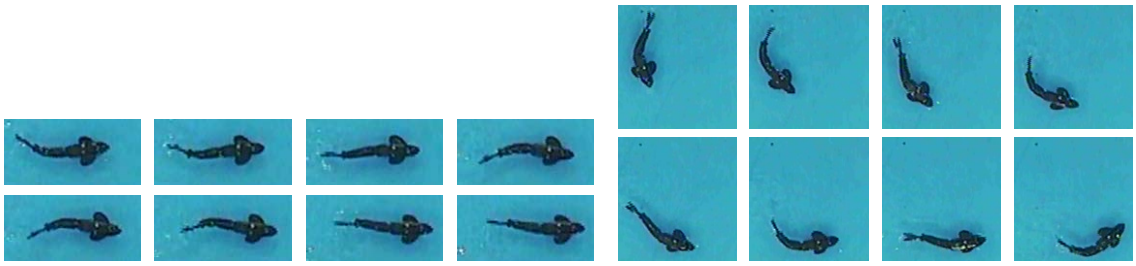


FIGURE 7. Sequence of different locomotion behaviors. The figure shows output signals of our CPG model while the locomotion model are changing. See text for details.

fast, turning right sharply and turning left gently (Figure 7). In the sequence shown in Figure 7, the CPG makes transitions in the order that swimming forwards fast ($0 \leqslant t \leqslant 4s$), swimming forwards slowly ($4 < t \leqslant 8s$), turning left gently ($8 < t \leqslant 12s$) and turning

right sharply ($12 < t \leqslant 16s$). We generate different locomotion behaviors by changing the control commands υ and X, then the specific control inputs of the CPG model (υ, R_i, X_i and φ_{ij}) are obtained by the designed transition layer. Although these parameters are changed abruptly, the output signals of all oscillators are still continuous and smooth.

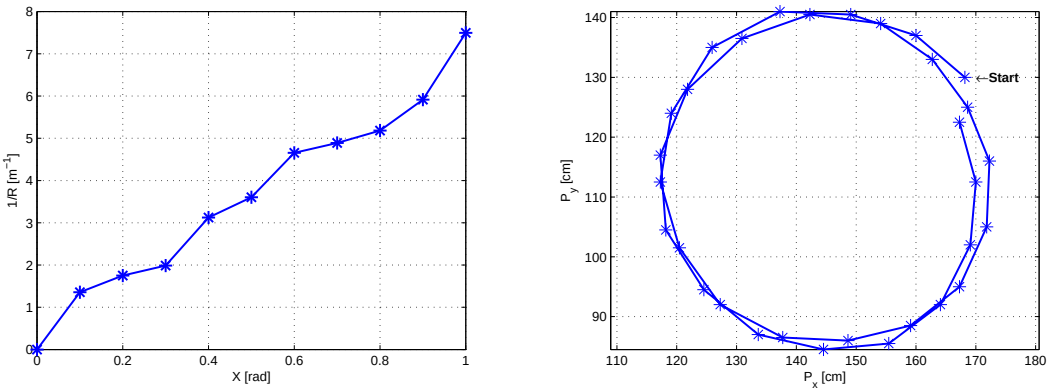


(a) (b)

FIGURE 8. (From left to right, from top to bottom.) (a) The robotic fish is swimming forwards with the control commands $\upsilon = 1Hz$ and $X = 0$. The time step between the snapshots is $0.125s$. (b) The robotic fish is turning left at the control command for direction: $X = -0.5rad$. The control command for speed is set to $\upsilon = 1.0Hz$. The time step between the snapshots is $0.5s$.

The swimming forwards gait of the robotic fish is shown in Figure 8(a) with the control commands $\upsilon = 1.0Hz$ for speed and $X = 0$ for direction.

When the control command X is not zero, the fish robot will swim on a circle. The experiment results are shown in Figure 9(a). It is easy to see that the radius R of turning circle monotonically increases when the command X changes from 0 to $1rad$. The smaller radius (R) results in the sharper turning. In Figure 9(a), it is shown that the sharpest turning is $R = 13.34cm$ (0.334 *body lengths/s*) when $|X| = 1rad$. Figure 8(b) shows



(a) (b)

FIGURE 9. (a) Control of direction. Each data point is the average of 5 direction measures. (b) Trajectory of the robot with the control commands $X = -0.5rad$ for direction and $\upsilon = 1.0Hz$ for speed. The position of the robot is obtained by the experimental platform per $0.4s$.

the turning gait of the robotic fish with the control commands $v = 1.0Hz$ for speed and $X = -0.5rad$ for direction. And the swimming trajectory of the robot is shown in Figure 9(b).

6. **Conclusion.** In this paper, a novel CPG-based locomotion control method has been proposed and used for a robotic fish. The proposed CPG model, which is a system of coupled linear oscillators, has some unique advantages. First, linear differential oscillators have been used instead of nonlinear ones to make the CPG model easy to be implemented. Second, the adaptive structural parameters have ensured satisfying dynamic performance. And explicitly presented parameters in the CPG model have brought clarity in applications. The experimental results have shown that our CPG model is well implemented in locomotion control of a three-joint robotic fish. We believe that our model could be applied to all biomimetic multi-joint underwater robots with link structures.

We are currently extending our work in the following directions. First, the stability performance of the locomotion has been ignored in this paper because we choose the locomotion speed as the only optimization goal. We are testing some optimization methods to optimize the parameters for better stability and transient performance. Second, the optimization has depended on the experimental platform because the robotic fish has no sensor. Some pose sensors are being integrated into the robot body. Then, the locomotion controller may be implemented on board, and on-line optimization can be easily applied. Third, the proposed CPG model is open-loop. Some posture information is being added as coupling terms to obtain a CPG with feedback. Additionally, based on the designed transition layer, higher-level intelligent control methods such as neural network based control or fuzzy control may be easily realized for the robotic fish. We are also interested in controlling a formation of robotic fish, and some preliminary work can be found in [31].

Acknowledgment. This work was supported by the National Natural Science Foundation of China (NSFC) under Contracts 10972003, 60736022 and 60774089 and by grants from the Dutch Organization for Scientific Research (NWO) and the Dutch Technology Foundation (STW). The authors would also like to thank the anonymous reviewers for their comments and suggestions, which have improved the presentation of the paper.

REFERENCES

[1] 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, no.2, pp.237-252, 1999.

[2] A. J. Ijspeert, J. Hallam and D. Willshaw, Evolving swimming controllers for a simulated lamprey with inspiration from neurobiology, *Adaptive Behavior*, vol.7, no.2, pp.151-172, 1999.

[3] 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.

[4] M. S. Triantafyllou and G. S. Triantafyllou, An efficient swimming machine, *Scientific American*, vol.272, pp.64-70, 1995.

[5] 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, no.1-2, pp.3-13, 2008.

[6] J. Ostrowski and J. Burdick, Gait kinematics for a serpentine robot, *Proc. of the 1996 IEEE International Conference of Robotics and Automation*, pp.1294-1299, 1996.

[7] K. McIsaac and J. Ostrowski, Motion planning for anguilliform locomotion, *IEEE Transactions on Robotics and Automation*, vol.19, no.4, pp.637-625, 2003.

[8] K. H. Low, Mechatronics and buoyancy implementation of robotic fish swimming with modular fin mechanisms, *Proc. of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol.221, no.3, pp.295-309, 2007.

[9] D. P. Tsakiris, M. Sfakiotakis, A. Menciassi, G. L. Spina and P. Dario, Polychaete-like undulatory robotic locomotion, *Proc. of the 2005 IEEE International Conference on Robotics and Automation*, pp.3018-3023, 2005.

[10] A. J. Ijspeert, Central pattern generators for locomotion control in animals and robots: A review, *Neural Networks*, vol.21, no.4, pp.642-653, 2008.
[11] F. Delcomyn, Neural basis for rhythmic behaviour in animals, *Science*, vol.210, no.4469, pp.492-498, 1980.
[12] S. Grillner, Neural control of vertebrate locomotion – Central mechanisms and reflex interaction with special reference to the cat, *Feedback and Motor Control in Invertebrates and Vertebrates*, pp.35-56, 1985.
[13] H. Kimura, Y. Fukuoka and A. H. Cohen, Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts, *The International Journal of Robotics Research*, vol.26, no.5, pp.475-490, 2007.
[14] A. Crespi and A. J. Ijspeert, Online optimization of swimming and crawling in an amphibious snake robot, *IEEE Transactions on Robotics*, vol.24, no.1, pp.75-87, 2008.
[15] W. Zhao, J. Yu, Y. Fang and L. Wang, Development of multi-mode biomimetic robotic fish based on central pattern generator, *Proc. of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.3891-3896, 2006.
[16] J. Shao, L. Wang and J. Yu, Development of an artificial fish-like robot and its application in cooperative transportation, *Control Engineering Practice*, vol.16, no.5, pp.569-584, 2008.
[17] J. Shao, L. Wang and J. Yu, Development of multiple robotic fish cooperation platform, *International Journal of Systems Science*, vol.38, no.3, pp.257-268, 2007.
[18] Intelligent Control Laboratory, Control and optimization of robotic fish, *Technical Report*, Peking University, 2010.
[19] C. Reynolds, Flocks, herds, and schools: A distributed behavioral model, *Computer Graphics*, vol.21, no.4, pp.25-34, 1987.
[20] J. Kennedy and R. C. Eberhart, Particle swarm optimization, *IEEE International Conference on Neural Networks*, pp.1942-1948, 1995.
[21] C. Rong, Q. Wang, Y. Huang, G. Xie and L. Wang, Autonomous evolution of high speed quadruped gaits using particle swarm optimization, *Lecture Notes in Artificial Intelligence*, vol.5399, pp.259-270, 2009.
[22] D. He, Q. Wang, C. Rong and G. Xie, Generating high-speed three-dimensional dynamic quadruped walking using an evolutionary search, *Proc. of the 2009 International Conference on Informatics in Control, Automation and Robotics*, pp.167-172, 2009.
[23] C. Niehaus, T. Röfer and T. Laue, Gait optimization on a humanoid robot using particle swarm optimization, *Proc. of the 2nd Workshop on Humanoid Soccer Robots in Conjunction with the 2007 IEEE-RAS International Conference on Humanoid Robots*, 2007.
[24] M. Hebbel, R. Kosse and W. Nistico, Modeling and learning walking gaits of biped robots, *Proc. of the Workshop on Humanoid Soccer Robots of the 2006 IEEE-RAS International Conference on Humanoid Robots*, pp.40-48, 2006.
[25] Y. Bourquin, *Self-Organization of Locomotion in Modular Robots*, Master Thesis, University of Sussex, Brighton, U.K., 2005.
[26] J.-F. Chang, A performance comparison between genetic algorithms and particle swarm optimization applied in constructing equity portfolios, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(B), pp.5069-5079, 2009.
[27] Y. Taguchi, H. Nakano, A. Utani, A. Miyauchi and H. Yamamoto, A competitive particle swarm optimization for finding plural acceptable solutions, *ICIC Express Letters*, vol.4, no.5(B), pp.1899-1904, 2010.
[28] C. Karakuzu, Parameter tuning of fuzzy sliding mode controller using particle swarm optimization, *International Journal of Innovative Computing, Information and Control*, vol.6, no.10, pp.4755-4770, 2010.
[29] R.-J. Kuo, C. C. Huang and T.-L. Hu, Normal vector-controlled particle swarm optimization algorithm for solving bi-level linear programming problem, *ICIC Express Letters*, vol.4, no.5(A), pp.1417-1424, 2010.
[30] Y. Shi and R. C. Eberhart, A modified particle swarm optimizer, *IEEE International Conference on Evolutionary Computation*, pp.69-73, 1998.
[31] C. Wang, M. Cao and G. Xie, Antiphase formation swimming for autonomous robotic fish, *Proc. of the 18th World Congress of the International Federation of Automatic Control*, Milan, Italy, 2011.