SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2020, Vol. 63 192206:1–192206:14
https://doi.org/10.1007/s11432-019-2671-y

Controlling the depth of a gliding robotic dolphin using dual motion control modes

Jian WANG[1,2], Zhengxing WU[1], Min TAN[1] & Junzhi YU[1,3*]

[1]*State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing* 100190, *China;*
[2]*University of Chinese Academy of Sciences, Beijing* 100049, *China;*
[3]*State Key Laboratory for Turbulence and Complex System, Department of Mechanics and Engineering Science, BIC-ESAT, College of Engineering, Peking University, Beijing* 100871, *China*

Received 21 July 2019/Accepted 2 September 2019/Published online 24 July 2020

Abstract This paper investigates the performance of the dual mode, namely flipper mode and central pattern generator (CPG) mode, for controlling the depth of a gliding robotic dolphin. Subsequent to considering the errors in dynamic models, we propose a depth control system that combines the line-of-sight (LOS) method with an adaptive control approach (ACA) to deal with uncertainties in the model parameters. First, we establish a full-state dynamic model to conduct simulations and optimize the parameters used in later aquatic experiments. Then, we use the LOS method to transform the control target from the depth to the pitch angle and employ the ACA to calculate the control signal. In particular, we optimize the ACA's control parameters using simulations based on our dynamic model. Finally, our simulated and experimental results demonstrate not only that we can successfully control the robotic dolphin's depth, but also that its performance was better than that of the CPG-based control, thus indicating that we can achieve three-dimensional motion by combining flipper-based and CPG-based control. The results of this study suggest valuable ideas for practical applications of gliding robotic dolphins.

Keywords gliding robotic dolphin, depth control, dual motion, adaptive control approach

Citation Wang J, Wu Z X, Tan M, et al. Controlling the depth of a gliding robotic dolphin using dual motion control modes. Sci China Inf Sci, 2020, 63(9): 192206, https://doi.org/10.1007/s11432-019-2671-y

1 Introduction

Over the recent decades, many researchers have been drawn to exploring bio-inspired robots, thus leading to the development of a large number of bionic robotic prototypes [1–5]. Among these, robotic dolphins have recently received great interest, since dolphins have become talented swimmers with excellent locomotion abilities after a long natural selection process [6,7]. Robotic dolphins can thus realize astonishing locomotion patterns and are highly maneuverable. For example, Yu et al. designed two robotic dolphins that can perform 360° frontflips, backflips [8], and leaps [9]. However, robotic dolphins also consume large amounts of energy due to their propulsive mechanisms that requires motors to create oscillating motions. In contrast, underwater gliders aim to save energy [10–12]. Many generations of these have been developed, establishing both theoretical foundations and engineering experience in this research area [13, 14]. Thus, to reduce energy consumption, the first gliding robotic dolphin, which combines a robotic dolphin with an underwater glider, has been designed [15], and a 1.5 m-long prototype has been developed [16], using a sliding-mode observer-based heading control with flippers [17].

* Corresponding author (email: junzhi.yu@ia.ac.cn)

Depth control has always been an important task for autonomous underwater vehicles (AUVs), since it is the basis of both path tracking and planning and is needed to accomplish trickier tasks in complex underwater environments. There are many ways to achieve depth control, but they can be divided into three main categories: adjusting a movable slider inside the robot; changing the flipper deflection angles; and changing the net buoyancy. Shen et al. [18] controlled the depth of a robotic dolphin using a fuzzy proportional-integral-derivative (PID) controller to adjust a slider. Ranganathan et al. [19] designed a hybrid depth controller that combines a PID controller, a linear quadratic regulator (LQR), and a sliding-mode controller (SMC) with a variablebuoyancy system. Yu et al. [20] proposed a sliding-mode fuzzy controller (SMFC) for a robotic dolphin, and was able to control its depth with a steady-state error of less than 0.5 cm by relying on the flippers.

Each of these three methods has its pros and cons. Using slider-based mass adjustment, the pitch moment is generated by changing the center of gravity, which has the advantages of producing a clear effect and resisting interference due to the use of internal forces. However, mechanical limitations and over-reactions may lead to longer transition times [21]. Controlling the net buoyancy has almost the same disadvantages as using a slider, and produces smaller effects since the generated forces/moments are relatively small, but it can yield substantial energy savings [22]. Finally, strategies based on changing the flipper deflection angles have been widely employed in recent years due to their rapid responses and precise control [23]. However, since the pitch moments are generated by hydrodynamic forces, which are closely related to the swimming speed, they may not meet the requirements if the forward speed is too slow. Thus, velocity control is also needed [20], which increases the control system's complexity. This makes controlling depth purely through the gliding robotic dolphin's propulsion mechanism a considerable problem.

This paper focuses on the issues involved in controlling the depth of a gliding robotic dolphin by comparing the performance of two motion modes, namely flipper mode and central pattern generator (CPG) mode. To the best of our knowledge, this has not previously been discussed. Our paper makes two main contributions.

First, we present a control framework that achieves depth control by combining line-of-sight (LOS) with an adaptive control approach (ACA), improving the ACA by selecting suitable control parameters. Specifically, we optimize the controller's control parameters based on extensive offline simulations of a full-state dynamic model, and then apply these directly in aquatic experiments.

Second, we use simulations to compare the effectiveness of the two control modes, finding that controlling the CPG offset improves the robot's performance. We also carry out aquatic experiments, whose results not only show that we can successfully control the gliding robotic dolphin's depth with good accuracy and short transition times, but also confirm that CPG mode is more suitable for depth control. Our results suggest that we can free the flippers from having to control the pitch moment to focus on yaw control, potentially enabling us to decouple the control of three-dimensional (3D) motion.

The remainder of this paper is structured as follows. Section 2 describes the gliding robotic dolphin's overall mechanical design and prototype. Next, we derive a full-state dynamic model in Section 3. Then, Section 4 discusses the control systems, including the LOS, CPG, and ACA methods. Section 5 discusses and analyzes the results of our simulations and aquatic experiments. Finally, Section 6 summarizes our conclusion and plans for future work.

2 Mechanical design of the gliding robotic dolphin

As illustrated in Figure 1, the gliding robotic dolphin has a streamlined shape, modelled after a killer whale, to reduce water resistance and obtain a better lift-to-drag ratio. Unlike previous gliding robotic dolphins [15, 17], ours utilizes a more compact design, and replaces the external oil bladder with a water injector to make net buoyancy adjustments faster. In addition, the robot's shell is constructed from polyethylene. Table 1 gives details of our robotic dolphin's mechanical and electrical parameters. It has the following three main compartments.

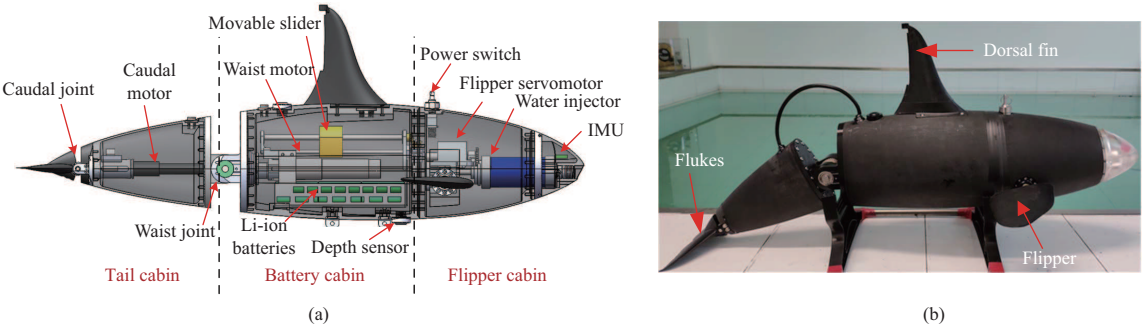

Figure 1 (Color online) Mechanical structure of the gliding robotic dolphin, showing the (a) conceptual design and (b) prototype.

Table 1 Technical parameters of the gliding robotic dolphin

Parameter	Value
Size ($L \times W \times H$)	~0.83 m×0.38 m×0.33 m
Total mass	~8.86 kg
Power supply	Li-ion battery (DC 24.6 V, 3200 mAh)
Number of the body joints	4
Drive mode	RE13, RE16, RE30, EC16, Hitec HS7980
Onboard sensors	IMU (JY901), depth sensor (MS5837)
Controller	STM32F407

• Flipper compartment. This mainly comprises an inertial measuring unit (IMU), two flipper servomotors, a water injector, and an injector motor. The IMU is a nine-axis inertial sensor that enables us to acquire the robot's attitude angles after calibrating the magnetic field. The servomotors (Hitec HS-7980) can both provide high torque and respond sufficiently rapidly for real-time control. The cylindrical injector is made of aluminum alloy to prevent deformation. By moving an internal piston, water can be sucked in or pushed out, effectively changing the net buoyancy. The piston is driven by a DC brush motor (Maxon RE16). This motor's small size and strong torque mean it can easily overcome friction between the piston and the cylinder wall to move the piston. However, its maximum rated speed is relatively low, which may result in slow responses.

• Battery compartment. This features a motor-driven movable slider, a Li-ion battery pack, a waist motor, and a depth sensor. The movable slider weighs 350 g, and its center of gravity is slightly higher than that of the robot, providing a more significant pitch moment. It is driven by a small, high-speed DC brush motor (Maxon RE13). The battery pack consists of 18 Li-ion batteries, connected in series and parallel to provide high currents. The waist motor, a 60-W DC motor (Maxon RE30) with a rated torque of up to 85.6 mNm, is fixed above the battery pack. In addition, a depth sensor (TE Connectivity MS5837) is affixed to the bottom of the compartment to provide depth data. The electronic control systems, including the microcontroller unit (MCU) and power management modules, are also installed in this compartment.

• Tail compartment. This is mainly given over to a brushless caudal motor (Maxon EC16) with a maximum rated speed of 39400 rpm, enabling the caudal joint to oscillate at very high frequencies.

3 Dynamic model

To provide depth control, we need to analyze the relationship between the control signal and pitch angle. To do this, we now present a brief derivation of our full-state dynamic model, based on our previous work [24]. Figure 2 illustrates the coordinate frames used, namely the inertial frame $C_g = o_g x_g y_g z_g$, body frame $C_b = o_b x_b y_b z_b$, and rotatable surface frames $C_i = o_i x_i y_i z_i$, where $i = w, t, l, r$ denote the waist, flukes, left flipper, and right flipper, respectively.

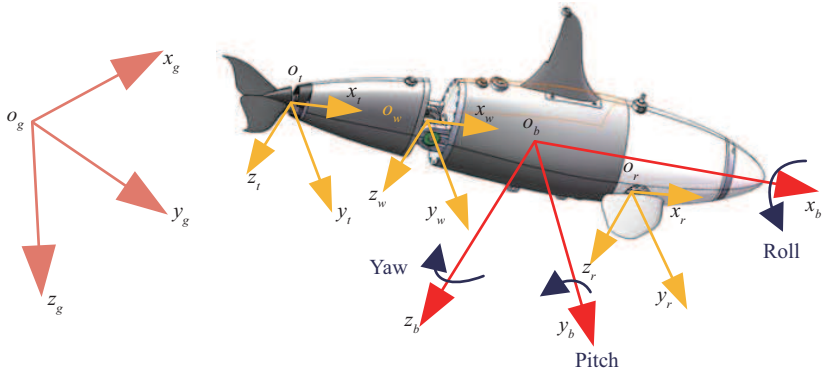

Figure 2 (Color online) Coordinate systems used in the model, showing the inertial, body, and fin frames.

Next, we define the translational and angular velocities with respect to (w.r.t.) the body frame as $U_b = (U_{bx}, U_{by}, U_{bz})^{\mathrm{T}}$ and $\Omega_b = (\Omega_{bx}, \Omega_{by}, \Omega_{bz})^{\mathrm{T}}$, respectively. In addition, we define the full velocity vector as $V_b = (U_b^{\mathrm{T}}, \Omega_b^{\mathrm{T}})^{\mathrm{T}}$. Then, the robotic dolphin's kinematics can be expressed as

$$\begin{aligned} {}^{g}\dot{P}_b &= {}^{g}U_b = {}^{g}R_b U_b, \\ {}^{g}\dot{R}_b &= {}^{g}R_b \hat{\Omega}_b, \end{aligned} \tag{1}$$

where ${}^{g}R_b$ and ${}^{g}P_b$ are the rotation matrix and position vector of C_b w.r.t. C_g, respectively. Similarly, the rotatable surfaces have their own kinematics, expressed as

$$\begin{aligned} V_i &= {}^{i}H_b V_b + \delta_i \quad (i = w, l, r), \\ V_t &= {}^{t}H_w V_w + \delta_t, \end{aligned} \tag{2}$$

where ${}^{i}H_b$ $(i = w, l, r)$ and ${}^{t}H_w$ are 6×6 transformation matrices, and the vectors δ_i denote the angular velocities. Then, based on Newton's law, we can derive the full-state dynamic model as

$$\begin{aligned} &M_b \dot{V}_b = -\Gamma_{cb} + \Gamma_{hb} + \Gamma_{wb} + \Gamma_{lb} + \Gamma_{rb} + G_b + \Gamma_m + \Gamma_j, \\ &{}^{b}H_w M_w \dot{V}_w = {}^{b}H_w(-\Gamma_{cw} + \Gamma_{hw} + \Gamma_{bw} + \Gamma_{tw}), \\ &{}^{b}H_t M_t \dot{V}_t = {}^{b}H_t(-\Gamma_{ct} + \Gamma_{ht} + \Gamma_{wt}), \\ &{}^{b}H_i M_i \dot{V}_i = {}^{b}H_i(-\Gamma_{ci} + \Gamma_{hi} + \Gamma_{bi}) \quad (i = l, r). \end{aligned} \tag{3}$$

Here, $G_b = (G_n, \tau_n)^{\mathrm{T}}$ denotes the net buoyancy force and moment, Γ_m and Γ_j are the forces and moments generated by the movable slider and water injector, respectively, and the M_i $(i = b, w, t, l, r)$ are the total inertia matrices. In addition, Γ_{hi} represents the hydrodynamic force and moment for component i, and Γ_{bi} is the corresponding external body force. Similarly, Γ_{ib} denotes the external force of component i on the body, and likewise for Γ_{wt} and Γ_{tw}. In order to unify all calculations in the body frame, both sides of the equation should be multiplied by ${}^{b}H_i$.

To simplify the calculations, we define two new variables ξ_t and ξ_i as

$$\xi_t = {}^{t}\dot{H}_w V_w \quad \text{and} \quad \xi_i = {}^{i}\dot{H}_b V_b \quad (i = w, l, r).$$

Now, we can obtain the speed derivatives of the velocities from (2), substitute them in the left-hand sides of (3), and obtain the summation. In this way, we obtain the final dynamic model, as follows:

$$M\dot{V}_b = -\Pi_e + \Pi_c + \Pi_h + \Pi_g + \Gamma_m + \Gamma_j, \tag{4}$$

where

$$M = \sum_{i=b,w,t,l,r} {}^{b}H_i M_i {}^{i}H_b,$$

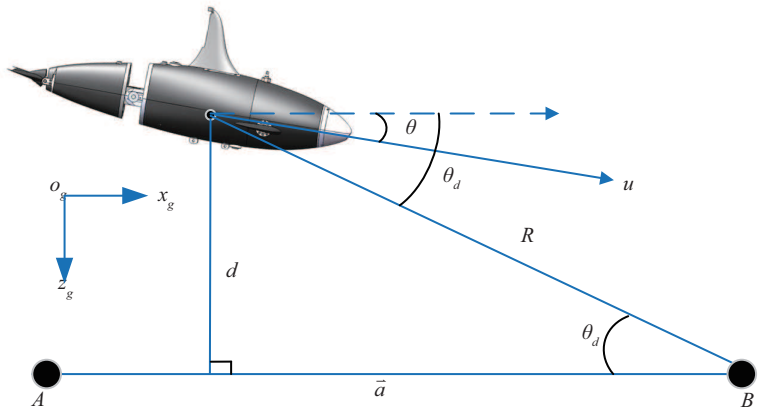

Figure 3 (Color online) Diagram showing our LOS implementation.

$$\Pi_c = -\sum_{i=b,w,t,l,r} {}^bH_i\Gamma_{ci},$$
$$\Pi_h = \sum_{i=b,w,t,l,r} {}^bH_i\Gamma_{hi},$$
$$\Pi_g = G_b,$$
$$\Gamma_m = m_m \begin{pmatrix} 2\hat{\dot{P}}_m\Omega_b - \ddot{P}_m \\ \hat{P}_m(2\hat{\dot{P}}_m\Omega_b - \ddot{P}_m) \end{pmatrix},$$
$$\Gamma_j = m_j \begin{pmatrix} 2\hat{\dot{P}}_j\Omega_b - \ddot{P}_j \\ \hat{P}_j(2\hat{\dot{P}}_j\Omega_b - \ddot{P}_j) \end{pmatrix}.$$

4 Adaptive depth control

4.1 LOS method

In this paper, we aim to make the robot swim at the desired depth by relying on dolphin-like motions. Generally, with sufficient propulsion and a suitable lift-to-drag ratio, such a robot can easily dive with a stable pitch angle. Thus, we can shift the control target from the depth to the pitch angle. Another benefit of controlling the pitch angle is that the tracking path becomes smoother, since the robot's body attitude cannot change sharply.

In principle, depth control is a two-dimensional issue, and can be simplified for identifying tracking points at the desired depth. We can then employ LOS guidance to map the desired tracking points to pitch angles. As shown in Figure 3, given the target depth d, we consider a vector $\vec{a}$ that is perpendicular to z_g. Then, taking the robot's centroid as the center, we draw a circle with radius R that intersects the vector $\vec{a}$ at the points A and B. Then, we select the point B as the target point. Based on the real-time depth z, we can then obtain the target pitch angle as

$$\theta_d = \arctan\left(\frac{d_e}{\|\vec{a}\|}\right), \tag{5}$$

where

$$\begin{cases} d_e = d - z, \\ d_e^2 + \|\vec{a}\|^2 = R^2. \end{cases}$$

Here, $\|\cdot\|$ indicates the Euclidean norm.

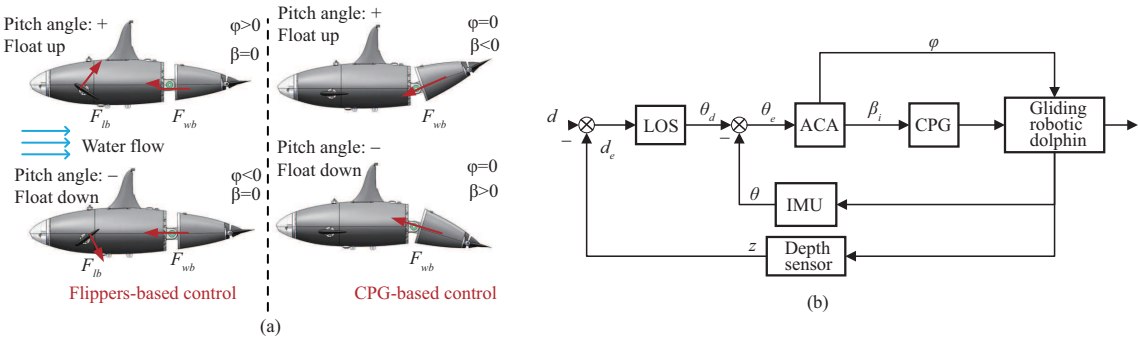

Figure 4 (Color online) Illustration of the control system, showing (a) the control signals for both modes and (b) the control framework.

4.2 Control system

4.2.1 *CPG model*

In this paper, we use a CPG model to control the robotic dolphin's swimming mode, as this can effectively ensure the control signals produce smooth motion transitions, even in the face of sudden parameter changes. This CPG model can be expressed as [25]

$$\begin{cases} \dot{\alpha}_i = 2\pi f_i + \sum\limits_{j\in\Upsilon(i)} \omega_{ij}\sin\left(\alpha_j - \alpha_i - \phi_{ij}\right), \\ \ddot{r}_i = a_i\left(\dfrac{a_i}{4}\left(A_i - r_i\right) - \dot{r}_i\right), \\ x_i = r_i\left(1 + \cos\alpha_i\right), \end{cases} \tag{6}$$

where r_i and α_i denote the amplitude and phase of the i-th oscillator, respectively. In addition, ϕ_{ij} and ω_{ij} represent the phase difference and weight between the i-th and the j-th oscillator, respectively, a_i is a strictly positive constant, A_i and f_i indicate the intrinsic amplitude and frequency, Υ_i denotes the set of oscillators from which the i-th oscillator receives inbound couplings, and x_i represents the model's output.

Since the CPG model cannot change the offset of the output signals when we employ the one-sided CPG output to save the start-up times, we should also add an offset angle β_i based on the output angle φ_i calculated by x_i, as follows:

$$x_i = \varphi_i + \beta_i. \tag{7}$$

4.2.2 *Control variables*

In order to produce diving and surfacing motions, the robotic dolphin needs to create suitable pitch moments. We therefore selected two control variables, namely the flipper deflection angle ψ and the CPG model offset β_i. By controlling these two signals, the robot can adjust its pitch moment to float up or down. Figure 4(a) illustrates the different control options and the corresponding forces.

Specifically, by changing the deflection angle ψ of the flippers, the water flow can result in upward or downward forces, which can in turn produce positive or negative pitch moments as of the center of gravity changes. The dolphin can also use the offset β_i to change the direction of the propulsive forces from its body and flukes. Clearly, a non-zero β_i means F_{wb} is no longer horizontal, but rather points obliquely upward or downward. Therefore, we combine the LOS and CPG control methods to create the proposed control algorithm framework, as illustrated in Figure 4(b). This enables the depth control target to be reached via an iterative process.

4.3 Controller design

Since our control target is the pitch angle, we describe the dolphin's angular motion by the following model [26]:

$$\dot{\theta}_{k+1} = (p_k \ q_k) \begin{pmatrix} \dot{\theta}_k \\ u_k \end{pmatrix}, \tag{8}$$

where

$$\begin{cases} \begin{pmatrix} p_k \\ q_k \end{pmatrix} = \chi_k + \zeta_k = \begin{pmatrix} a_k \\ b_k \end{pmatrix} + \begin{pmatrix} \zeta_k^0 \\ \zeta_k^1 \end{pmatrix} = \begin{pmatrix} \hat{a}_k + \tilde{a}_k \\ \hat{b}_k + \tilde{b}_k \end{pmatrix} + \begin{pmatrix} \zeta_k^0 \\ \zeta_k^1 \end{pmatrix}, \\ \|\zeta_k\| \leqslant \lambda \eta_k. \end{cases}$$

Here, $\dot{\theta}_k$ represents the rate of pitch angle change and u_k is the control variable that provides the pitch moment. Here, u_k is either ψ or β_i. Next, $\hat{a}_k$ and $\hat{b}_k$ are adaptive parameters, and $\tilde{a}_k$ and $\tilde{b}_k$ are the differences between the real and estimated values. In addition, ζ_k denotes the parameter perturbation, which is usually due to external disturbances and is thus generally random. However, it still remains bounded in a suitable range, so we assume it is less than $\lambda\eta_k$, where η_k is a positive scalar value that we will estimate and λ is a positive, manually adjusted parameter that controls the rate at which η_k changes. We also define $q_{\min}$ as the minimum value of q_k, and assume that $q_k > 0$ due to the sign of the variable u_k, i.e., $q_k > q_{\min} > 0$.

The pitch angle is related to its rate of change via

$$\dot{\theta}_{k+1} = \theta_{k+1} - \theta_k, \tag{9}$$

and substituting this into (8) gives us the final form

$$\theta_{k+1} = (1 + p_k)\theta_k - p_k\theta_{k-1} + q_k u_k. \tag{10}$$

By setting the reference pitch angle $\theta_{k+1}^r = \theta_d$ and inverting (10), we can express the control signal u_k as follows:

$$u_k = \frac{\theta_{k+1}^r - (1 + \hat{a}_k)\theta_k + \hat{a}_k\theta_{k-1}}{\hat{b}_k}. \tag{11}$$

In addition, due to structural limitations, u_k must be bounded, lying within a range $u_k \in [u_{\min}, u_{\max}]$ that is determined by the robot's design.

Next, in order to develop the adaptation rules, we need to calculate the control error. The tracking error relative to the control target can be defined as

$$e_{k+1} = \theta_{k+1}^r - \theta_{k+1} = \theta_{k+1}^r - (1 + p_k)\theta_k + p_k\theta_{k-1} - q_k u_k + \hat{b}_k u_k - \hat{b}_k u_k. \tag{12}$$

When u_k is not in the saturation zone, we can substitute (11) into (12) to obtain

$$e_{k+1} = -(\tilde{a}_k + \zeta_k^0)(\theta_k - \theta_{k-1}) - \hat{b}_k^{-1}(\tilde{b}_k + \zeta_k^1)(\theta_{k+1}^r - (1 + \hat{a}_k)\theta_k + \hat{a}_k\theta_{k-1}) = -\big(\tilde{a}_k + \zeta_k^0 \ \tilde{b}_k + \zeta_k^1\big)\delta_k, \tag{13}$$

where

$$\delta_k = \begin{pmatrix} \dot{\theta}_k \\ u_k \end{pmatrix}.$$

Thus, we can express the expected control error magnitude as

$$|e_k| \leqslant \|\zeta_k\| \, \|\delta_k\| \leqslant \lambda\eta_k \, \|\delta_k\| .$$

4.4 Adaptation rules

Based on the control error, we can now derive the adaptation rules. First, the parameter adaptation rate should decrease with the error, stopping entirely once the error satisfies $|e_k| \leqslant \lambda\eta_k\|\delta_k\|$. Since we are aiming to estimate the parameter η_k, let the estimated value be $\hat{\eta}_k$. In addition, we define a coefficient c_k that controls the adaptation rate as [26]

$$c_k = \begin{cases} (1 - \lambda\hat{\eta}_{k-1}\left\|\delta_{k-1}\right\|\left|e_k\right|^{-1}), & \text{if } \left|e_k\right| \geqslant \lambda\hat{\eta}_{k-1}\left\|\delta_{k-1}\right\|, \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

Note that c_k lies within the range $[0, 1)$. Then, we can estimate η_k as

$$\hat{\eta}_k = \hat{\eta}_{k-1} + \sigma_k^{-1}(\lambda\gamma c_k\left|e_k\right|\left\|\delta_{k-1}\right\|), \tag{15}$$

where

$$\begin{aligned} &\sigma_k = 1 + \delta_{k-1}^{\mathrm{T}}\Lambda\delta_{k-1} + \gamma\lambda^2\|\delta_{k-1}\|^2, \\ &\Lambda = \begin{pmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{pmatrix}. \end{aligned} \tag{16}$$

Here, $|\cdot|$ denotes the absolute value. In addition, γ is a positive parameter to be set manually, and μ_1 and μ_2 are positive constants. In this paper, we set $\mu_1 = \mu_2 = \mu$ and $\hat{\eta}_0 = 0$. Finally, we can obtain the adaptation rule for the adaptive model as

$$\begin{pmatrix} \hat{a}_k \\ \hat{b}_k \end{pmatrix} = \begin{pmatrix} \hat{a}_{k-1} \\ \hat{b}_{k-1} \end{pmatrix} - \frac{c_k e_k}{\sigma_k}\Delta\delta_{k-1}, \tag{17}$$

where $\hat{b}_k > b_{\min}$.

Proposition 1. If the ACA is employed for the above design, the estimated variable values are bounded as confirmed in [26].

Proof. First, define $\tilde{y}_k = (\tilde{a}_k\ \tilde{b}_k)$ and $\tilde{\eta}_k = \eta_k - \hat{\eta}_k$. Next, consider the nonnegative Lyapunov function

$$V_k = \tilde{y}_k^{\mathrm{T}}\Lambda^{-1}\tilde{y}_k + \frac{1}{\gamma}\tilde{\eta}_k^2. \tag{18}$$

Then, the change in this function can be expressed as

$$\Delta V_k = V_k - V_{k-1} = P_1 + P_2, \tag{19}$$

where

$$\begin{aligned} &P_1 = \tilde{y}_k^{\mathrm{T}}\Lambda^{-1}\tilde{y}_k - \tilde{y}_{k-1}^{\mathrm{T}}\Lambda^{-1}\tilde{y}_{k-1}, \\ &P_2 = \frac{1}{\gamma}\left(\tilde{\eta}_k^2 - \tilde{\eta}_{k-1}^2\right). \end{aligned}$$

For P_2, we can obtain $\tilde{\eta}_k$ from (15) as follows:

$$\tilde{\eta}_k = \tilde{\eta}_{k-1} - \sigma_k^{-1}(\lambda\gamma c_k\left|e_k\right|\left\|\delta_{k-1}\right\|). \tag{20}$$

Thus, we can express P_2 as

$$P_2 = \frac{c_k^2 e_k^2\gamma\lambda^2}{\sigma_k^2}\|\delta_{k-1}\|^2 - \frac{2c_k\lambda}{\sigma_k}\left\|\delta_{k-1}\right\|\left|e_k\right|\tilde{\eta}_{k-1}. \tag{21}$$

In addition, since $\tilde{b}_k > b_{\min}$, Eq. (17) can be re-expressed as

$$m_k = \hat{y}_{k-1} - \frac{c_k e_k}{\sigma_k}\Delta\delta_{k-1}, \tag{22}$$

where

$$\hat{y}_k = f(m_k) = \begin{cases} (m_k^0\ m_k^1), & \text{if } m_k^1 > b_{\min}, \\ (m_k^0\ b_{\min}), & \text{if } m_k^1 < b_{\min}. \end{cases}$$

Thus, we can conclude that $\|y_{k-1} - f(m_k)\| \leqslant \|y_{k-1} - m_k\|$. As a result, $\tilde{y}_k^{\mathrm{T}} \Lambda^{-1} \tilde{y}_k \leqslant (y_{k-1} - m_k)^{\mathrm{T}} \Lambda^{-1} \cdot (y_{k-1} - m_k)$, and we can derive the following bound on P_1:

$$P_1 \leqslant \frac{2c_k e_k}{\sigma_k} \tilde{y}_{k-1}^{\mathrm{T}} \delta_{k-1} + \frac{c_k^2 e_k^2}{\sigma_k^2} \delta_{k-1}^{\mathrm{T}} \Lambda \delta_{k-1}. \tag{23}$$

In addition, we can obtain the following bound on ΔV_k by adding (21) and (23):

$$\Delta V_k \leqslant \frac{2c_k e_k}{\sigma_k} \tilde{y}_{k-1}^{\mathrm{T}} \delta_{k-1} - \frac{2c_k \lambda}{\sigma_k} \|\delta_{k-1}\| |e_k| \tilde{\eta}_{k-1} + \frac{c_k^2 e_k^2}{\sigma_k^2} \big(\delta_{k-1}^{\mathrm{T}} \Lambda \delta_{k-1} + \gamma \lambda^2 \|\delta_{k-1}\|^2\big). \tag{24}$$

Next, from (12), we can obtain

$$e_k^2 + e_k \tilde{y}_{k-1}^{\mathrm{T}} \delta_{k-1} = -\zeta_{k-1}^{\mathrm{T}} \delta_{k-1} \leqslant \lambda \eta \|\delta_{k-1}\| |e_k|.$$

Substituting this and the definition of c_k in (14) into (24) yields the final expression

$$\Delta V_k \leqslant -\frac{2c_k^2 e_k^2}{\sigma_k} < -\frac{c_k^2 e_k^2}{\sigma_k}. \tag{25}$$

This indicates that $\hat{y}_k$ and $\hat{\eta}_k$ are bounded. A detailed convergence analysis of the error e_k is presented in [26].

5 Simulations and experiments

In order to evaluate the effectiveness of the proposed method of controlling the depth of a gliding robotic dolphin, we conducted both simulations and aquatic experiments. The simulations were carried out in MATLAB/Simulink using the full-state dynamic model. The mechanical parameters used, obtained by Solidworks, are listed in [24]. The hydrodynamic parameter values were initially obtained using computational fluid dynamics (CFD) simulations, then adjusted based on practical experience [27]. The aquatic experiments were conducted in a pool of dimensions $5(L)$ m $\times$ $4(W)$ m $\times$ $1(H)$ m, and the whole process was recorded with an underwater video camera. Importantly, we used the simulation results to derive optimal control parameter values through offline optimization.

5.1 Control parameter optimization

For the ACA-based controller, we needed to find optimal values for four control parameters, namely $q_{\min}$, λ, γ, and μ. Hence, we needed to design a suitable objective function to solve this problem. The most important factor for depth control is the error (accuracy), followed by the transition time. Thus, we created an objective function with parameter constraints based on the error $|e|$ and transition time t_s, as follows:

$$\arg\min \quad \kappa_1 |e| + \kappa_2 t_s, \tag{26}$$

$$\text{s.t.} \begin{cases} \kappa_1 + \kappa_2 = 1, \\ 0 < \kappa_1, \kappa_2 < 1, \\ 0 < q_{\min}, \mu < 5, \\ 0 < \lambda, \gamma < 10. \end{cases}$$

Here, since the dimensions of the two optimization goals were different, we used min–max normalization to normalize the units by setting bounds on their values. In addition, by adjusting the κ_1 and κ_2 values, we could focus the optimization process more on the error or transition time. Since the dynamic model

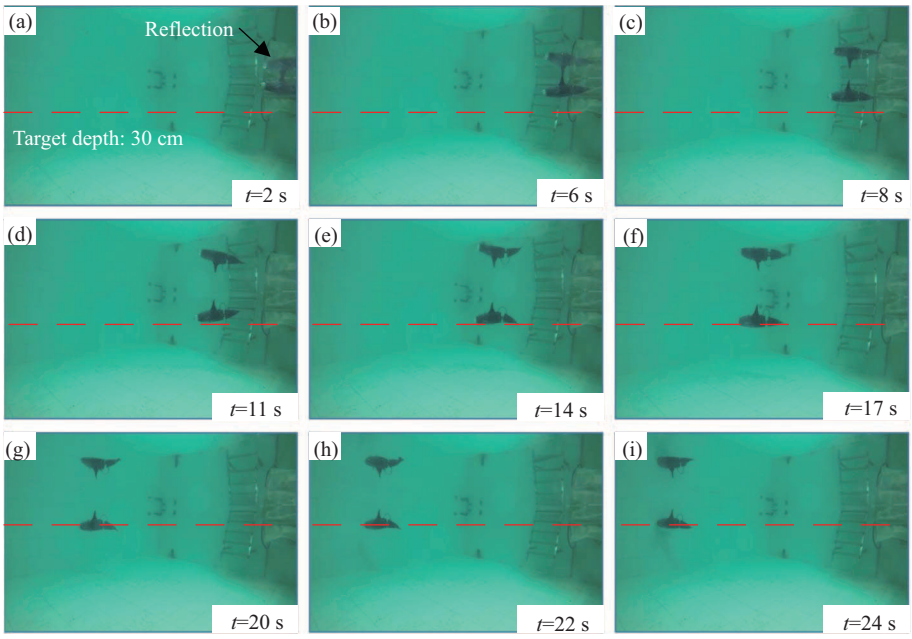

Figure 5 (Color online) Snapshot sequence for ψ-based depth control. (a) $t = 2$ s; (b) $t = 6$ s; (c) $t = 8$ s; (d) $t = 11$ s; (e) $t = 14$ s; (f) $t = 17$ s; (g) $t = 20$ s; (h) $t = 22$ s; (i) $t = 24$ s.

was nonlinear, we used MATLAB's genetic algorithm toolbox to search for an optimal set of control parameters.

By setting $\kappa_1 = 0.9$ and $\kappa_2 = 0.1$, we obtained the following optimal control parameter set: $q_{\min} = 1.5$, $\lambda = 3.51$, $\gamma = 4.19$, and $\mu = 1.59$. Then, we made some minor adjustments to this parameter set based on the results of the aquatic experiments. The final parameters used were $q_{\min} = 0.7$, $\lambda = 3.51$, $\gamma = 4.19$, and $\mu = 0.99$ for controlling ψ, and likewise for controlling β except that $q_{\min} = 1.2$. Here, we found that the control parameters calculated using our full-state dynamic model provided accurate initial values for actual operation, confirming the model's accuracy.

5.2 Results for flipper-based control

First, we used the above control parameters to test controlling the dolphin's depth by adjusting the flipper deflection angle ψ. Here, we bounded the control signal u_k to lie within the range $[-45°, 45°]$ based on mechanical constraints. Then, to generate dolphin-like motion, we set the CPG frequency to 1 Hz, the intrinsic amplitudes of the waist and caudal joints to 20° and 30°, respectively, and the phase difference between the two joints to 45°.

In this experiment, we set the target depth to 30 cm. Figure 5 shows a sequence of snapshots taken during the control process, which lasted for approximately 24 s. Figure 6 shows the real-time depth data collected by the onboard sensor. Here, we see that the experimental and simulated results are approximately consistent. In order to better analyze the control accuracy and controller performance, we calculated both the mean absolute error (MAE) and root mean square error (RMSE) based on the depth results. Once the robotic dolphin had entered a steady state, the MAE and RMSE of the control error d_e were 0.43 and 0.09 cm, respectively, demonstrating that the controller was effective. However, the transition time was somewhat long, mainly because the dolphin's forward speed was not high enough for the flippers to rapidly generate the desired pitch moments.

Figure 7 plots the flipper and pitch angles during the experiment, based on the recorded IMU data and control signal. The pitch angle has been filtered, since the dolphin's head generated significant pitch jitter (approximately ±8°) due to flapping of the waist and caudal joints, and these oscillations seriously affected the control process. To combat this, we first performed median filtering on the pitch angle based on the CPG frequency, and then used the filtered data θ for depth control. Although such filtering can

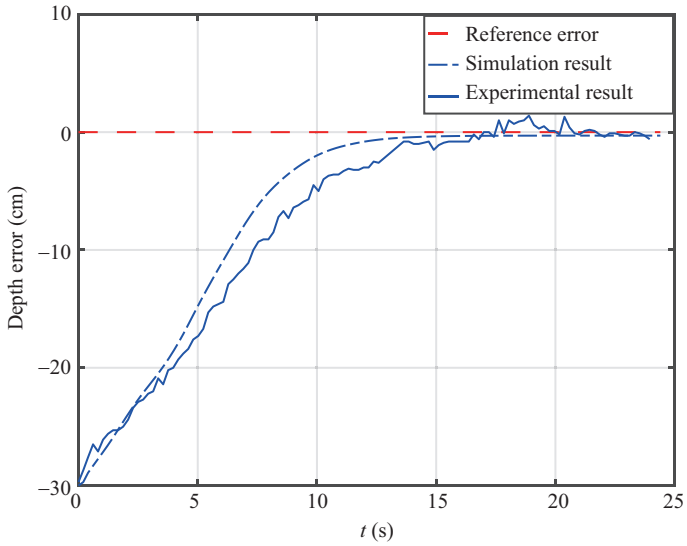

Figure 6 (Color online) Depth error for ψ-based depth control.

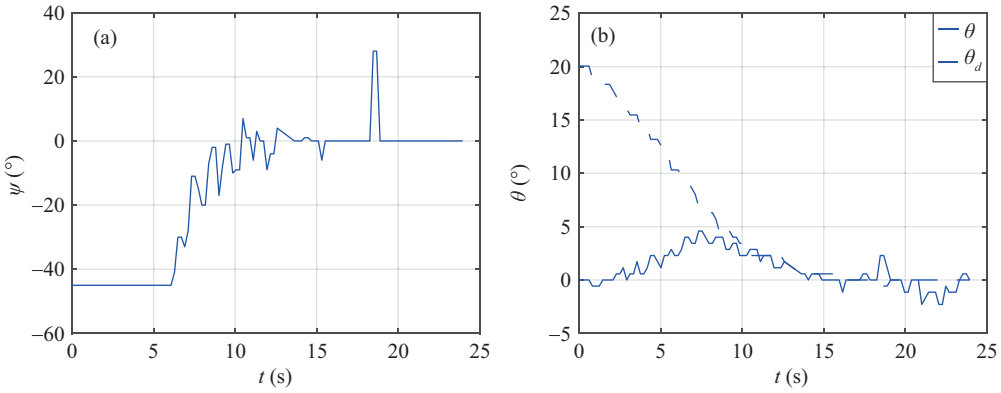

Figure 7 (Color online) Control inputs and outputs during the aquatic experiments, showing the (a) flipper angle and (b) target and actual pitch angles.

introduce a delay into the attitude signal input, this had no obvious impact on the depth control process.

In addition, as Figure 7(b) illustrates, θ_d decreased as the dolphin approached the target depth. The control input ψ was fixed at $-45°$ until $t = 6$ s, and then began to increase as the pitch angle θ approached the target θ_d. However, we should also note that θ only gradually began to track θ_d after $t = 6$ s, and the maximum pitch angle achieved was only around 5°, which also explains the insufficient pitch moment. In addition, we can see a fluctuation at around 18 s, when the actual depth became slightly offset from the target depth during the depth maintenance phase, as can be seen in Figure 6.

5.3 Results for CPG-offset-based control

The second experiment focused on controlling the depth using the CPG offset β_i. Here, we bounded β_i to lie within the range $[-18°, 18°]$. Figures 8 and 9 show a snapshot sequence and depth data recorded during the control process, respectively. Compared with ψ-based control, this approach had two advantages. First, the robot was more stable, with no fluctuations after it entered the depth maintenance phase. Second, the transition time (approximately $t_s = 10.3$ s) was significantly shorter than for ψ-based control (around $t_s = 14$ s). Again, the experimental and simulated results were in good agreement. In addition, the MAE and RMSE were 0.40 and 0.08 cm, respectively, both lower than the results for flipper-based control.

More importantly, Figure 10 shows the changes in the control variable β_i and pitch angle during the

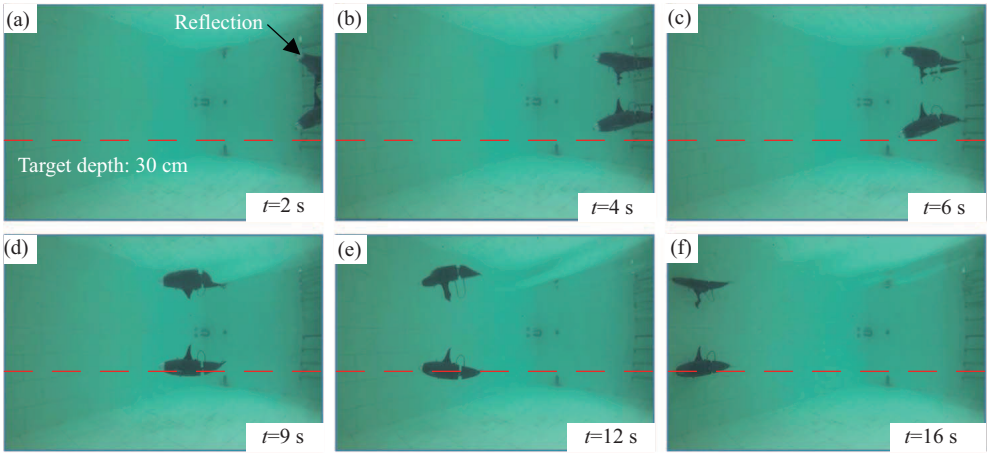

Figure 8 (Color online) Snapshot sequence for β_i-based depth control. (a) $t = 2$ s; (b) $t = 4$ s; (c) $t = 6$ s; (d) $t = 9$ s; (e) $t = 12$ s; (f) $t = 16$ s.

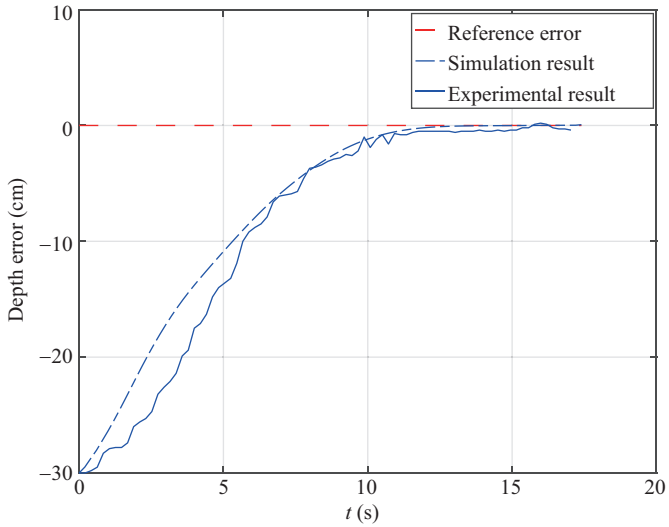

Figure 9 (Color online) Depth error for β_i-based depth control.

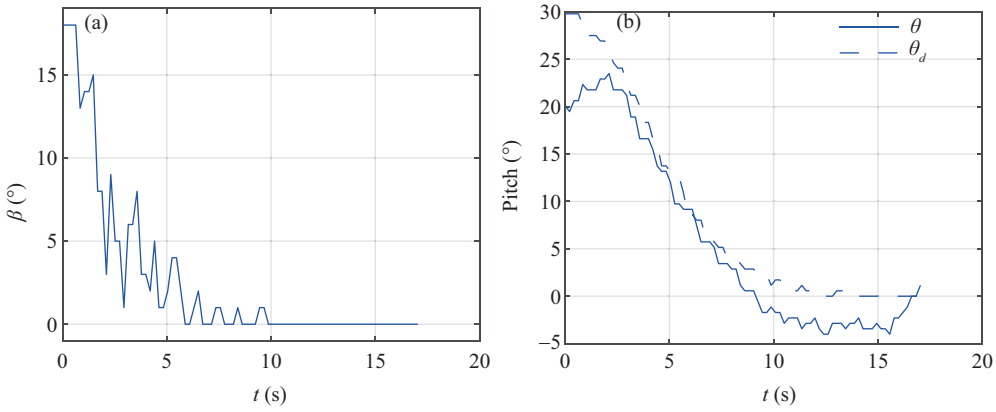

Figure 10 (Color online) Control inputs and outputs during the aquatic experiments, showing the (a) CPG offset and (b) target and actual pitch angles.

experiment. First, we should note that the actual pitch angle reached a maximum of 25°, much higher than the angle seen for flipper-based control, indicating that β_i-based control can provide more suitable pitch moments. In addition, the actual pitch angle curve more closely matched the target pitch angle

during the diving process, again confirming the controller's effectiveness. The error remained essentially small constant after the robot reached the target depth, because we included a control threshold in the control process. Specifically, the control was not applied when the depth error was below a threshold of $e_s = 0.8$ cm. It is also worth noting that the pitch angle error indicates the robot did not dive in exactly the direction indicated by the pitch angle, due to mechanical inaccuracies. In addition, compared with the results in [20], we achieved equal depth accuracy and a similarly short transition time by controlling β_i instead of ψ. However, we were able to use a lower frequency (1 Hz vs. 1.5 Hz), which could save energy.

5.4 Discussion

We have implemented two methods of controlling the depth of a gliding robotic dolphin, not only confirming the feasibility of the platform and control algorithm, but also gathering valuable engineering experience for ocean exploration. Compared with the method in [26], we were able to calculate the control parameters more accurately using our full-state dynamic model, and achieved a smaller steady-state error. In addition, our approach involves fewer control parameters than the method in [20], which not only needs to consider the SMC model's parameters, but also fuzzy rules. Our simulations and aquatic experiments have shown that controlling β_i can achieve similar even better performance than controlling ψ. In particular, controlling β_i can free the flippers from having to control the pitch moment, leaving them to focus on the more important task of adjusting the yaw moment, as the only structure that can handle this. This potentially suggests new ideas for 3D path tracking.

Nevertheless, our system does have some limitations. First, the robotic dolphin did not swim exactly along the direction given by the pitch angle, which may affect control accuracy. Improving the robot's construction could reduce this effect, and we could also consider using depth data as direct feedback information for control. Second, robots that adopt dolphin-like motions to control their depth inevitably consume large amounts of energy due to their propulsion mechanism. A novel way to address this issue would be to combine gliding and dolphin-like motions to extend the robot's operational time in practical applications.

6 Conclusion and future work

In this paper, we have presented a depth control framework based on dolphin-like motion for use with our gliding robotic dolphin, and compared the performance of two different motion control modes. First, we established a full-state dynamic model of 3D dolphin-like and gliding motions. Next, we used an LOS method to convert depth control into a pitch control problem. Then, we employed a parameterized adaptive control approach to calculate the control signal, and optimized the parameters used. Specifically, we conducted simulations using the full-state dynamic model, optimizing the control parameters by devising a suitable objective function. We have also successfully implemented our control framework in a robotic prototype. Finally, simulated and experimental results regarding the control error and transition time demonstrated that the proposed control framework is effective. More importantly, comparing the performance of the two control modes revealed that the dolphin performed better when we controlled the CPG offset rather than the flipper angle, which will contribute greatly to the goal of achieving 3D motion.

Our future work will concentrate on controlling depth with gliding motions, aiming to achieve better performance by combining this with dolphin-like motion. In addition, since the robot can successfully control its vertical motion using only its waist and caudal joints, we will aim to enable it to follow a 3D path by controlling the CPG offset and flipper angle simultaneously.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61421004, 61725305, 61633020, 61633017, 61603388) and Key Project of Frontier Science Research of Chinese Academy of Sciences

(Grant No. QYZDJ-SSW-JSC004).

References

1 Ijspeert A J. Biorobotics: using robots to emulate and investigate agile locomotion. Science, 2014, 346: 196–203
2 Wu Z X, Yu J Z, Su Z S, et al. Towards an Esox lucius inspired multimodal robotic fish. Sci China Inf Sci, 2015, 58: 052203
3 Yuan J, Yu J Z, Wu Z X, et al. Precise planar motion measurement of a swimming multi-joint robotic fish. Sci China Inf Sci, 2016, 59: 092208
4 Zhang A F, Ma S G, Li B, et al. Adaptive controller design for underwater snake robot with unmatched uncertainties. Sci China Inf Sci, 2016, 59: 052205
5 Yu J Z, Li X B, Pang L, et al. Design and attitude control of a novel robotic jellyfish capable of 3D motion. Sci China Inf Sci, 2019, 62: 194201
6 Liu J C, Wu Z X, Yu J Z, et al. Sliding mode fuzzy control-based path-following control for a dolphin robot. Sci China Inf Sci, 2018, 61: 024201
7 Nagai M. Thinking Fluid Dynamics with Dolphins. Tokyo: Ohmsha, 2002
8 Yu J Z, Su Z S, Wang M, et al. Control of yaw and pitch maneuvers of a multilink dolphin robot. IEEE Trans Robot, 2012, 28: 318–329
9 Yu J Z, Su Z S, Wu Z X, et al. An integrative control method for bio-inspired dolphin leaping: design and experiments. IEEE Trans Ind Electron, 2016, 63: 3108–3116
10 Eriksen C C, Osse T J, Light R D, et al. Seaglider: a long-range autonomous underwater vehicle for oceanographic research. IEEE J Ocean Eng, 2001, 26: 424–436
11 Sherman J, Davis R E, Owens W B, et al. The autonomous underwater glider "Spray". IEEE J Ocean Eng, 2001, 26: 437–446
12 Webb D C, Simonetti P J, Jones C P. SLOCUM: an underwater glider propelled by environmental energy. IEEE J Ocean Eng, 2001, 26: 447–452
13 Leonard N E, Paley D A, Davis R E, et al. Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey Bay. J Field Robot, 2010, 27: 718–740
14 Grasso R, Braca P, Fortunati S, et al. Dynamic underwater glider network for environmental field estimation. IEEE Trans Aerosp Electron Syst, 2016, 52: 379–395
15 Wu Z X, Yu J Z, Yuan J, et al. Mechatronic design and implementation of a novel gliding robotic dolphin. In: Proceedings of IEEE International Conference on Robotics and Biomimetics, Zhuhai, 2015. 267–272
16 Wu Z X, Yu J Z, Yuan J, et al. Towards a gliding robotic dolphin: design, modeling, and experiments. IEEE/ASME Trans Mechatron, 2019, 24: 260–270
17 Yuan J, Wu Z X, Yu J Z, et al. Sliding mode observer-based heading control for a gliding robotic dolphin. IEEE Trans Ind Electron, 2017, 64: 6815–6824
18 Shen F, Cao Z Q, Zhou C, et al. Depth control for robotic dolphin based on fuzzy PID control. Int J Offshore Polar Eng, 2013, 23: 166–171
19 Ranganathan T, Singh V, Thondiyath A. Theoretical and experimental investigations on the design of a hybrid depth controller for a standalone variable buoyancy system-vBuoy. 2018. doi: 10.1109/JOE.2018.2875576
20 Yu J Z, Liu J C, Wu Z X, et al. Depth control of a bioinspired robotic dolphin based on sliding-mode fuzzy control method. IEEE Trans Ind Electron, 2018, 65: 2429–2438
21 Yu J Z, Wang M, Tan M, et al. Three-dimensional swimming. IEEE Robot Autom Mag, 2011, 18: 47–58
22 Makrodimitris M, Aliprantis I, Papadopoulos E. Design and implementation of a low cost, pump-based, depth control of a small robotic fish. In: Proceedings of IEEE International Conference on Intelligent Robots and Systems, Chicago, 2014. 1127–1132
23 Yu J Z, Sun F H, Xu D, et al. Embedded vision-guided 3-D tracking control for robotic fish. IEEE Trans Ind Electron, 2016, 63: 355–363
24 Wang J, Wu Z X, Yang Y Q, et al. Spiraling motion of a gliding robotic dolphin based on the 3-D dynamic model. In: Proceedings of IEEE International Conference on Real-time Computing and Robotics, Kandima, 2018. 13–18
25 Ijspeert A J, Crespi A, Ryczko D, et al. From swimming to walking with a salamander robot driven by a spinal cord model. Science, 2007, 315: 1416–1420
26 Verma S, Shen D, Xu J X. Motion control of robotic fish under dynamic environmental conditions using adaptive control approach. IEEE J Ocean Eng, 2018, 43: 381–390
27 Yu J Z, Yuan J, Wu Z X, et al. Data-driven dynamic modeling for a swimming robotic fish. IEEE Trans Ind Electron, 2016, 63: 5632–5640