Proceedings of The 2020 IEEE International
Conference on Real-time Computing and Robotics
September 28-29, 2020

Line-of-Sight Strategy-Based Path-Following System for a Multi-Joint Robotic Fish

Shijie Dai[1,2], Chao Zhou[3], Zhengxing Wu[1,2], Min Tan[1,2], and Junzhi Yu[1,4]

***Abstract*— This paper proposes a real-time path-following control system for a multi-joint robotic fish. The mechanical structure and dynamic model of the robotic fish for path-following are first described. Then, the framework of the path-following control algorithm is established based on the built dynamic model, including a modified line-of-sight (LOS) guidance law, an active disturbance rejection control (ADRC)-based heading controller and a proportional-integral-derivative (PID)-based speed controller. Specially, the modified LOS strategy is designed to select the tracking points and also provide the desired heading angle. Afterwards, to overcome systematic uncertainties and environmental disturbances, the ADRC method is adopted to design the heading controller. Meanwhile, the PID controller is also developed to maintain an appropriate swimming speed. Finally, simulations in both linear- and circular-path following are presented to validate the effectiveness of the proposed method.**

I. INTRODUCTION

Researches on the anatomy and morphology of real fish shed a light on the mechanism and structure design of autonomous underwater vehicles (AUVs) to make up their insufficient properties and deal with complicated mission requirments. Compared with these man-made machines, fish can exploit their undulatory or oscillatory fins to get propulsion instinctively and effectively [1]. Roboticists try to replicate not only the body shape but also the swimming locomotion of real fish. These fish-like bionic AUVs, namely robotic fish, outperform traditional AUVs in terms of propulsive efficiency, acceleration and maneuverability [2].

Generally, a completed control system of robotic fish can be established in a top-down design approach, by which the ultimate task is scheduled, planned and decomposed into subtasks [3]. Motion control, such as speed control, attitude control and depth control, is a key component of these tasks and ongoing research work has been carried out. For example, Morgansen *et al.* adopted a proportional-integral-derivative (PID) method to achieve closed-loop heading control for a two-link robotic fish and demonstrated its simplified model is sufficient for some operation conditions [4]. Verma *et al.* implemented a discrete-time terminal sliding mode control (TSMC) for speed tracking of a robotic fish, which can handle most of the unprecedented factors in modeling [5]. Castaño *et al.* offered a nonlinear model predictive control (NMPC) for a tail-actuated robotic fish to minimize the control error but the computational complexity of NMPC hampers its application on a real-time system [6].

When it comes to intelligent tasks such as autonomous cruise, obstacle avoidance, path planning and path-following, top strategies are indispensable. Path-following is a representative ability to steer the robotic fish along the desired path. Kelasidi *et al.* employed the integral line-of-sight (LOS) guidance law for the directional control of underwater snake robots against a constant disturbance and presented a formal stability analysis for the path-following controller [7]. Makrodimitris *et al.* used inverse dynamics to derive how the fish tail should move to execute desired conditions and presented a methodology for robotic fish to follow any planar trajectory [8]. Peng *et al.* converted the tracking progress into an optimization problem and illustrated the effectiveness of the proposed path-following method even with internal model uncertainty and external environment forces [9]. In previous work, the researches in path-following problem rarely focus on a multi-joint robotic fish, but only on mono-articular robotic fish or on the basis of a simplified dynamic model. What is more, some path-following strategies take too heavy calculations to be adopted for the multi-joint robot.

The objective of this paper is to offer a real-time path-following control method for a multi-joint robotic fish. First, a dynamic model is built to describe the motion of the multi-joint robotic fish. Based on this model, a practical LOS path-following system is established, including a modified LOS guidance strategy, an active disturbance rejection control (ADRC)-based heading controller and a PID-based speed controller. In particular, the optimized LOS guidance law provides proper target points in view of the coupled relationships between the heading control and the speed control. Meanwhile, the ADRC-based heading controller and the PID-based speed controller make it reliable to realize the path-following task even in the presence of disturbance. Thanks to these special design of the strategies and controllers, the overall system is very efficient and extremely lightweight to guarantee a real-time response. Simulations are finally carried out to validate the effectiveness and real-time of the proposed control method.

*This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB1310300; in part by the National Natural Science Foundation of China under Grant 61973303.

[1,2]S. Dai, Z. Wu, and M. Tan are with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China daishijie2018@ia.ac.cn; zhengxing.wu@ia.ac.cn; min.tan@ia.ac.cn

[1,4]J. Yu is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the State Key Laboratory for Turbulence and Complex System, Department of Mechanics and Engineering Science, BIC-ESAT, College of Engineering, Peking University, Beijing 100871, China junzhi.yu@ia.ac.cn

[3]C. Zhou is with Naval Research Academy, Beijing 100072, China rickzhou2010@126.com

978-1-7281-7293-4/20/$31.00 © 2020 IEEE

This paper is organized as follows. Section II presents the dynamic model of a multi-joint robotic fish. Section III discusses the LOS path-following system in detail. Simulation analysis is introduced in Section IV. Finally, the conclusions and future work are summarized in Section V.

II. DYNAMIC MODELING

Before deducing the path-following controller, the dynamic model of the robotic fish is provided first. The coordinate frames are defined in Fig. 1, each of which follows the right-hand rule. The world coordinate frame is denoted by $C_w = (o_w, x_w, y_w, z_w)$, and the robotic fish is assumed to move in the horizontal plane $o_w x_w y_w$. We simplify the trunk of the robotic fish as five discrete mechanical links connected in series. The body-fixed moving coordinate frame $C_0 = (o_0, x_0, y_0, z_0)$ is attached to the head link while the other four frames $C_i = (o_i, x_i, y_i, z_i)$ are attached to body segments in the similar way, where $i \in [1, n]$ indicates the ith body segment B_i and n is the number of joints ($n = 4$). The corresponding joint angle is notated by θ_i. Each link has a length of l_i with uniformly distributed mass so that the link center of mass (CM) is located at its midpoint.

The hydrodynamic analysis helps us comprehend interaction between the robotic fish and surrounding liquid. We use the Morrison equation [10] to model the hydrodynamic forces exerting on the body segment with respect to C_i. This (6×1) generalized force F_i can be formalized concisely by

$$F_i = -M_{ad,i}\dot{V}_i - \gamma_{ad,i} + f_{dr,i} \tag{1}$$

where $M_{ad,i}$ denotes the added inertia matrix and V_i is the (6×1) generalized velocity. $\gamma_{ad,i}$ and $f_{dr,i}$ indicate the Coriolis-centrifugal force by added mass and the drag force, respectively, both of which are velocity-related terms.

The dynamic model is constructed based on the Newton-Euler formulation, which is formalized for each segment:

$${}^{i}G_{i-1,i} = M_i\dot{V}_i + \gamma_i - F_i + {}^{i}H_{i+1}\,{}^{i+1}G_{i,i+1} \tag{2}$$

where ${}^{i}G_{i-1,i}$ represents the force exerted on B_i by B_{i-1} with respect to frame C_i. M_i is the inertia matrix and γ_i is the Coriolis-centrifugal force. ${}^{i}H_{i+1}$ is the transformation matrix, relevant to the joint angle θ_i and the link length l_i.

Enumerating these dynamic equations and executing reverse recursion from the last segment to the first segment B_0, the final dynamic equation with respect to frame C_0 can be deduced, which is coincide with that in [11] and given by

$$\begin{aligned}
M_{body}\dot{V}_0 &= {}^{0}F_{body} - {}^{0}\mathrm{E}_{body} - {}^{0}T_{body} \\
{}^{0}F_{body} &= \sum_{i=0}^{n} {}^{i}H_0{}^{T} f_{dr,i} \\
{}^{0}E_{body} &= \sum_{i=0}^{n} {}^{i}H_0{}^{T} (\gamma_i + \gamma_{ad,i}) \\
{}^{0}T_{body} &= \sum_{i=1}^{n} {}^{i}H_0{}^{T} (M_i + M_{ad,i}) \sum_{j=1}^{i} {}^{i}H_j \left(\eta_j + \ddot{\theta}_j Z\right) \\
M_{body} &= \sum_{i=0}^{n} {}^{i}H_0{}^{T} (M_i + M_{ad,i})\, {}^{i}H_0.
\end{aligned} \tag{3}$$

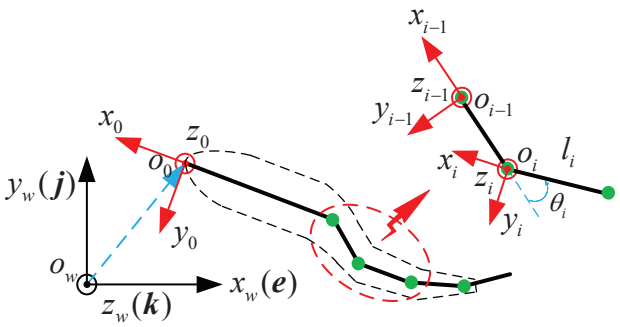

Fig. 1. Coordinate frames and notations

where $Z = [\mathbf{0}_{5\times1}, 1]^T$ is a (6×1) unit vector and η_j represents the result of ${}^{j}\dot{H}_{j-1}V_{j-1}$.

Eventually, provided with velocity V_0 and joint states, i.e., $(\theta_i, \dot{\theta}_i, \ddot{\theta}_i)$ as inputs, the joint space can be mapped to the motion state in the time domain, namely the motion state of the robotic fish can be predicted by the developed dynamic model.

Biological researches reveal that the rhythmic motion of real fish is controlled by central pattern generators (CPGs), coordinating interactions between high level control center and underlying effector organs [12]. By capturing motion information about real fish, many CPG models have been formalized for robotic fish to realize fishlike swimming [13]. Here, a Hopf oscillator-based CPG model is employed [14], which can be taken as the following forms:

$$\begin{aligned}
\dot{\xi}_i &= -\omega_i (\zeta_i - b_i) + \xi_i \left(A_i - \xi_i^2 - (\zeta_i - b_i)^2\right) + \\
&\qquad h_1 \left(\xi_{i-1}\cos\varphi_i + (\zeta_{i-1} - b_{i-1})\sin\varphi_i\right) \\
\dot{\zeta}_i &= \omega_i \xi_i + (\zeta_i - b_i)\left(A_i - \xi_i^2 - (\zeta_i - b_i)^2\right) + \\
&\qquad h_2 \left(\xi_{i+1}\sin\varphi_i + (\zeta_{i+1} - b_{i+1})\cos\varphi_i\right) \\
\theta_i &= c_i \zeta_i
\end{aligned} \tag{4}$$

where ξ_i and ζ_i are the oscillation states of the ith CPG unit. A_i and ω_i correspond to the intrinsic oscillation amplitude and frequency, respectively. b_i denotes the directional bias while φ_i is the phase difference between adjacent oscillators. Furthermore, h_1 and h_2 denote the coupling weights. Finally, the joint angles can be calculated directly by CPG through the magnification coefficient c_i. For the sake of simplicity, this paper keeps the same ω, b and c for every oscillator, and also sets a consistent φ between adjacent oscillators hereafter.

III. PATH-FOLLOWING CONTROL

Path-following problem, by definition, aims to steer the robotic fish towards and subsequently along the desired path. To realize this explicit mission requirement, a LOS path-following control scheme is presented for the robotic fish in this section.

A. LOS Path-Following System

Fig. 2 depicts a block diagram of the proposed LOS path-following system. This system consists of three main

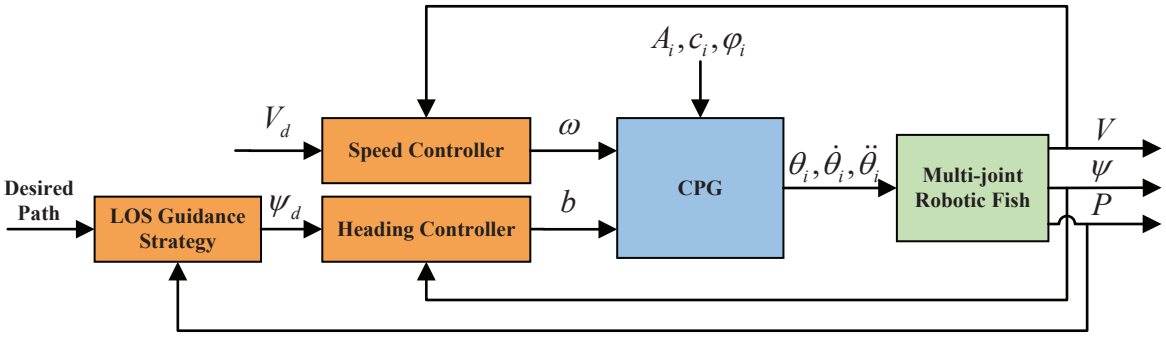

Fig. 2. Structure of the LOS path-following system

components, including an outer-loop controller, an inner-loop controller, and the robotic fish plant. The LOS guidance strategy, kernel of the outer-loop controller, expects a predefined path and the current position P of the robotic fish as input parameters to generate desired yaw angle ψ_d as an output variable. The state feedback and output feedback are given by the inner-loop ADRC and PID controller to control the heading ψ and speed V of the robot, respectively. The robotic fish plant is made up of the CPG model and the dynamic model. There is a corresponding relationship between the manipulated variables and the CPG parameters. For example, the heading ψ can be easily mapped to the directional bias b and the speed V can be mapped to the frequency ω. The CPG model with limit cycles makes it possible to keep the motion pattern of robotic fish and change the CPG parameters simultaneously to control its motion consecutively. Tracking error is utilized for measuring tracking performance in this paper, defined as a shortest distance between the current position of the robotic fish and the desired path.

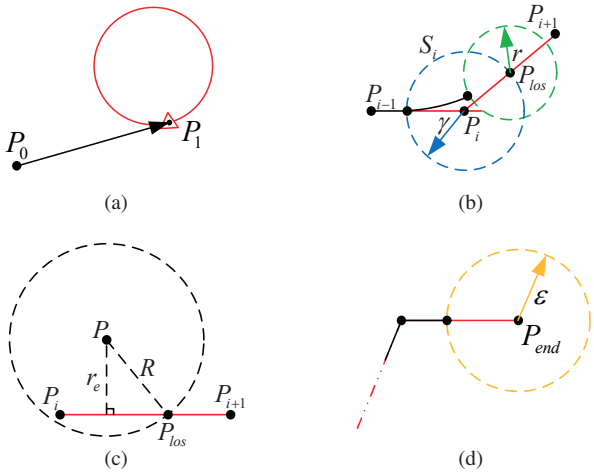

Fig. 3. Path-following states of FSM. (a) Original state. (b) Transition state. (c) Line tracking state. (d) End state.

B. Line-of-Sight Guidance Strategy

Optimized for the motion pattern of robotic fish and tightly coupled relationships between different control channels, a modified LOS guidance strategy is presented in this paper. The desired path can be decomposed into evenly distributed points along the path. With these discrete points as objectives of several target-tracking problems, the path-following progress can be finished. However, step signal is not an ideal input for a controlling system, so extra points are supposed to be interpolated. Drawing on the thought of LOS, these points are called vision points as a series of final target points, denoted by (3×1) vector $P_{los} = [x_{los}, y_{los}, z_{los}]^T$. The desired yaw angle ψ_d can be calculated by the vision points and the current position $P = [x, y, z]^T$ as follows:

$$\psi_d = \arctan \frac{y_{los} - y}{x_{los} - x} \tag{5}$$

Finite-state machine (FSM) facilitates to deal with selection and switch of so many vision points and abstracts path-following process into four states, illustrated in Fig. 3:

- Original state: Generally, the original position of the robotic fish, denoted by P_0, is at some distance away from the predefined path. The prior thing is to select the first target point on the path, determined by P_1. The closest endpoint is chosen for an open path whereas a tangent point is proper for a closed path. To guarantee a faster route for the mission, a line tracking strategy is employed from P_0 to P_1, which is particularly defined in the line tracking state.
- Transition state: The minimum turning radius imposes restrictions on the change of the yaw angle. What is more, path-following progress needs a transition state to adjust the attitude smoothly. Assume that the robotic fish is following a path from P_{i-1} to P_{i+1}. When the robot moves into the adjacent sphere of P_i, denoted by S_i (a sphere centered at P_i with a radius of γ), the current state is switched to the transition state while the intersection point of S_i and line P_iP_{i+1} is the vision point at this moment. When the robotic fish reaches the vision point, which means the distance between the robot and the vision point is less than a radius r, the current state is switched to the line tracking state.
- Line tracking state: To minimize the tracking error, an intuitive strategy is proposed to select the closest point on the following path to the current position P_i as a vision point. However, the robotic fish can not realize

a lateral movement because of a coupled relationship between the speed channel and the heading channel. That is, if the foot of a perpendicular is selected as a vision point, the tracking error may be increased paradoxically. So a margin is reserved to converge the robot to the desired path. Specify a search sphere, centered at P with a radius of R, to find a proper vision point, which is the front intersection point of the search sphere and line P_iP_{i+1}. In general, the search radius R is equal to r. Nevertheless, the search sphere is increased to $R = \sqrt{r^2 + r_e^2}$ if the following path is lost, which means a distance to the path, denoted by r_e, is greater than r. The vision point is always changing along the route to offer a continuous yaw angle as possible. Notably, if the robot moves into S_{i+1}, the current state is re-switched to the transition state.

- End state: Only if the next target point is the end point of the following path, denoted by P_{end}, the robot will not be switched to the transition state but finish the final target-tracking progress. Radius of convergence, denoted by ϵ, is determined to measure whether the robotic fish reaches the final endpoint and the path-following progress turns to termination if it reaches.

C. Heading Controller

The robotic fish is a high-order nonlinear system with internal model uncertainty and external environment disturbance. Meanwhile, limited by a compact size of the robot, its control chip can not sustain a heavy computing load. To get an ideal control effect in real time, ADRC is adopted for the heading controller in Fig. 4, which consists of the following three parts:

- Tracking Differentiator (TD): The noise amplification effect of the differentiation element makes a traditional PID controller susceptible to environment disturbances, which can be substituted by TD at a better performance. With regard to a given ψ_d, TD is defined by

$$\begin{aligned} f &= \text{fhan}\left(\psi_1 - \psi_d, \psi_2, \delta, h_0\right) \\ \psi_1 &= \psi_1 + h\psi_2 \\ \psi_2 &= \psi_2 + hf \end{aligned} \tag{6}$$

where ψ_1 is the tracking signal of ψ_d and ψ_2 is the differential of ψ_1. δ represents the time factor, h_0 is the filtering factor while h is the integration step. The function $\text{fhan}(\cdot)$ is given by

$$\begin{aligned} d &= \delta h_0^2 \\ a_0 &= h_0\psi_2 \\ y_a &= \psi_1 - \psi_d + a_0 \\ a_1 &= \sqrt{d(d + 8|y_a|)} \\ a_2 &= a_0 + \text{sign}(y_a)\left(a_1 - d\right)/2 \\ s_y &= (\text{sign}(y_a + d) - \text{sign}(y_a - d))/2 \\ s_a &= (\text{sign}(a + d) - \text{sign}(a - d))/2 \\ a &= (a_0 + y_a)\, s_y + a_2(1 - s_y) \\ f &= -\delta\left(\frac{a}{d}\right) s_a - \delta\,\text{sign}(a)(1 - s_a) \end{aligned} \tag{7}$$

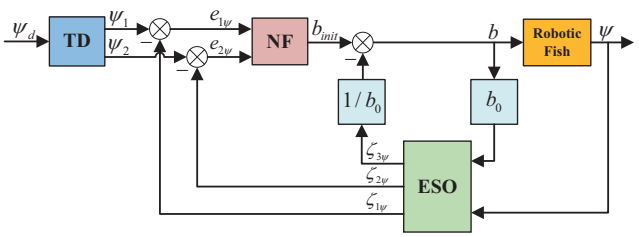

Fig. 4. Structure of the ADRC-based heading controller

where $\text{sign}(\cdot)$ represents the signum.

- Extended State Observer (ESO): ESO makes it possible to balance a trade-off between the rapidity and the overshoot through the observation and feedback of states. $\zeta_{1\psi}$, $\zeta_{2\psi}$ are estimates of ψ_1 and ψ_2, respectively. Besides, the environment disturbance is considered as an extended state in ESO, denoted by $\zeta_{3\psi}$. So ESO can be formalized as follows:

$$\begin{aligned} e &= \zeta_{1\psi} - \psi \\ \zeta_{1\psi} &= \zeta_{1\psi} + h\left(\zeta_{2\psi} - \beta_{01}e\right) \\ \zeta_{2\psi} &= \zeta_{2\psi} + h\left(\zeta_{3\psi} - \beta_{02}e + b_0 b_{init} - \zeta_{3\psi}\right) \\ \zeta_{3\psi} &= \zeta_{3\psi} + h(-\beta_{03}e) \end{aligned} \tag{8}$$

where $\beta_0 = [\beta_{01}, \beta_{02}, \beta_{03}]^T$ is the gain vector. b_{init} is the initial control variable and b_0 is a compensation factor to compensate the impact of the environment disturbance.

- Nonlinear Feedback (NF): NF improves the performance by a nonlinear combinations of state errors, consisting of a PD controller as a matter of fact, which is given by

$$b_{init} = k_{p\psi}\left(\psi_1 - \zeta_{1\psi}\right) + k_{d\psi}\left(\psi_2 - \zeta_{2\psi}\right) \tag{9}$$

where $k_{p\psi}$ and $k_{d\psi}$ are the proportional and differential gains. The directional bias b is given by

$$b = b_{init} - \zeta_{3\psi}/b_0. \tag{10}$$

D. Speed Controller

Compared with the heading controller, the speed controller plays a less important role in the path-following control system. Because of a strong coupling relation between the propulsive force and the yaw moment, the speed controller is configured to furnish an appropriate forward speed, defined as a resultant velocity V with respect to the world coordinate frame. A PID controller is adopted for the speed controller, expressed as

$$\omega = k_{pV}e_V + k_{iV}\int e_V\,dt + k_{dV}\frac{de_V}{dt} \tag{11}$$

where e_V is the velocity error that $e_V = V_d - V$.

TABLE I

PHYSICAL PARAMETERS OF THE ROBOTIC FISH

Variable	Unit	$i=0$	$i=1$	$i=2$	$i=3$	$i=4$
m_i	kg	1.528	0.159	0.159	0.171	0.091
l_i	m	0.291	0.062	0.062	0.062	0.137
c_i	m	0.18	0.044	0.044	0.045	0.037
$I_{i,z}$	kg$\cdot$ m^2($\times 10^{-4}$)	290	1.8	1.8	2.0	1.6

[1] m_i, l_i, c_i, and $I_{i,z}$ represents mass, length, the position on the x-axis in local coordinate, and the moment of inertia with respect to z-axis in local coordinate of each link, respectively.

TABLE II

PARAMETERS OF THE HOPF OSCILLATOR-BASED CPG MODEL

A_1	A_1	A_1	A_1	h_1	h_2	c	φ
8.70	19.07	25.49	40.39	4.0	5.0	6.0	70.0°

TABLE III

PARAMETERS OF THE ADRC-BASED HEADING CONTROLLER

h_0	h	δ	$k_{p\psi}$	$k_{d\psi}$	b_0	ω_0	β_{01}	β_{02}	β_{03}
0.008	0.004	300	0.019	0.038	800	20	$3\omega_0$	$3\omega_0^2$	ω_0^3

IV. SIMULATION ANALYSIS

This section presents some simulation results by MATLAB SIMULINK in order to substantiate the validity of the proposed path-following system. The simulations are carried out using the robotic fish developed in our laboratory [15]. Motivated by mimicking the locomotion of the robotic fish, a robotic fish plant is built based on the developed dynamic models and CPG models. Definitely, a portion of CPG parameters are tuned in advanced and keep constant throughout the simulations and experiments. The basic physical parameters of the employed robotic fish are tabulated in Table I while the CPG parameters are listed in Table II. The remaining parameters of the path-following system need to be adjusted according to the chief characteristically demands. Radius of the adjacent sphere is set as $\gamma = 0.3$ m on account of a minimum turning radius of the robotic fish, meanwhile, the tracking margin is smaller that $r = 0.2$ m. The convergence radius should be small enough, set at $\epsilon = 0.05$ m, to be applicable to show the robotic fish reaches the final endpoint. The controller parameters are tuned and optimized with the ADRC parameters listed in Table III. The desired resultant speed of the robotic fish is set at $V_d = 1.0$ m/s and the parameters of the PID-based speed controller are set as follows: $k_{pV} = 0.1, k_{iV} = 0.01, k_{dV} = 0.03$.

During the simulation, the initial states of the robotic fish are set to zero. To validate its stability for various tasks, the LOS path-following system is adopted to steer the robot along a combined path, including a linear path and a circular path. Specially, the linear path is emanating from the origin of the world coordinate frame and tangent to the desired circular path, which is centered at coordinate $[4, 4, 0]^T$ with a radius of 2 m. For the purpose of comparison, a traditional PID-based heading controller is also employed in the simulations. Note that both of the parameters of ADRC-based and PID-based controllers are tuned as fine as possible to compare their actual performance for the robotic fish and the PID-based parameters are shown as follows: $k_p = 0.012, k_i = 0.001, k_d = 0.014$.

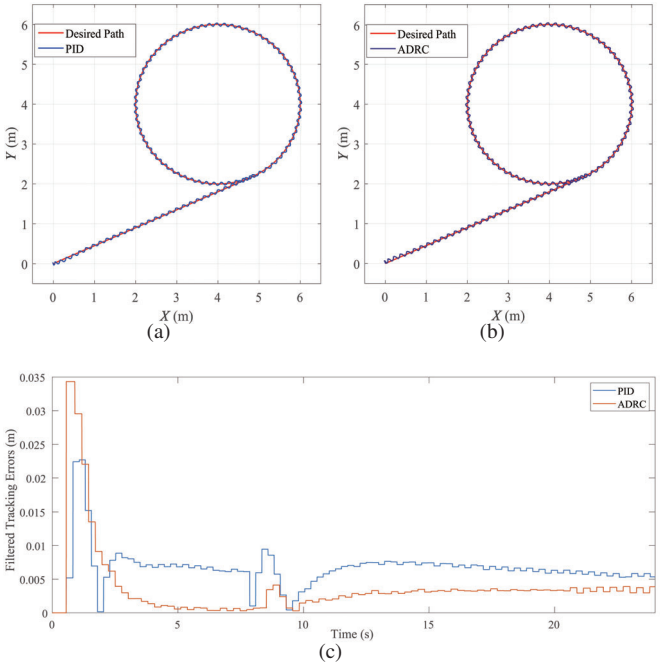

Fig. 5. Simulation results without disturbance. (a) LOS guidance strategy with the PID-based heading controller. (b) LOS guidance strategy with the ADRC-based heading controller. (c) Filtered tracking errors.

Figs. 5(a) and 5(b) show tracking curves using different heading controllers. The tracking error is introduced for a quantitative analysis. Because of the fishlike swimming, the real-time position is oscillating about the desired path even if the system possesses a nice tracking performance. Therefore, the mean filtering process is employed based on the oscillation frequency ω. Fig. 5(c) depicts the filtered tracking error. As can be seen from the simulation results, the LOS path-following system is applicable to follow the desired path with a trivial tracking error and both of the ADRC and PID controllers achieve good inner-loop control effects in an ideal environment.

Nevertheless, actual environment is full of complexity and levity due to the existence of the parametric perturbations, external disturbances and modeling errors. Therefore, subsequent simulations are implemented in the presence of uniform stochastic disturbance. These disturbances are quantified as a (6×1) generalized force on the robotic fish, taken as the following forms

$$F_{dis} = [D_{surge}, D_{sway}, 0, 0, 0, D_{yaw}]^T$$

where the surge force, the sway force and the yaw moment are considered, while the stochastic disturbance D_{surge}, D_{sway} and D_{yaw} follow a same uniform distribution pattern that $D_{surge}, D_{sway}, D_{yaw} \sim \mathcal{U}(-6, 6)$.

Fig. 6 shows the simulation results in the presence of this disturbance. It is observed that although both of these two methods can catch up and follow the desired path, the LOS path-following system with ADRC-based heading controller

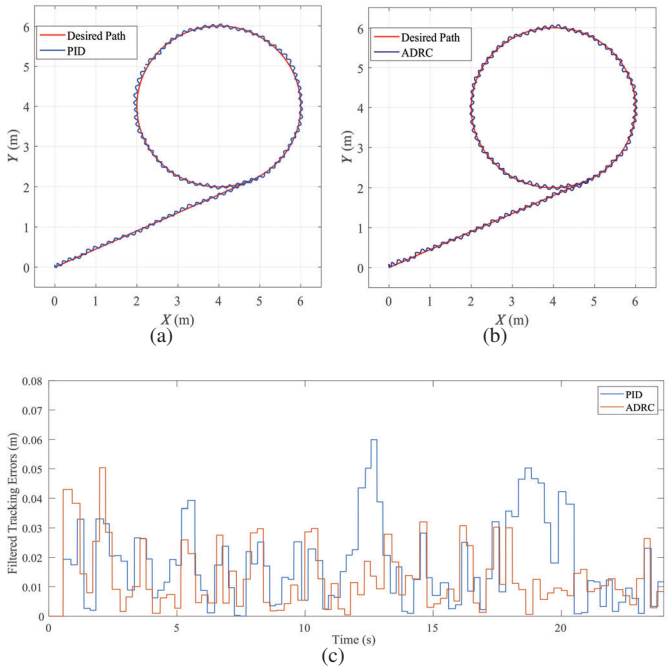

Fig. 6. Simulation results with stochastic disturbance. (a) LOS guidance strategy with the PID-based heading controller. (b) LOS guidance strategy with the ADRC-based heading controller. (c) Filtered tracking errors.

TABLE IV
CUMULATIVE TRACKING ERRORS OF ADRC AND PID CONTROLLERS

Items	Cumulative tracking errors		
	ADRC	PID	Reduction
Linear path without disturbance	0.0416	0.0533	21.95%
Circular path without disturbance	0.0551	0.1066	48.31%
Linear path with disturbance	0.1133	0.1227	7.66%
Circular path with disturbance	0.2263	0.3040	25.56%

performs significantly better. The ADRC-based inner-loop controller effectively rejects the disturbance and shows the robustness to reach a smoother convergence to the desired path.

To analyze the results numerically, the cumulative tracking error is defined as the time integral of the filtered tracking error. The cumulative errors are separately calculated for the linear path and the circular path, which are tabulated in Table IV. It can be discerned that the LOS path-following system with ADRC-based controller performs better overall for the robotic fish. With the help of the proposed LOS guidance system, the tracking errors of the linear path are relatively small so the ADRC-based heading controller may get a modest reduction compared with the PID controller, especially in the presence of external disturbance. Moreover, the ADRC-based controller is more effective for a complicated circular path, reducing the cumulative tracking error by over 25%. Therefore, the proposed LOS path-following system has an effective tracking performance by simulation and the ADRC-based heading controller can successfully attenuate the disturbance effect and reduce the cumulative tracking error to track accurately for complicated tasks.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a LOS path-following system for a multi-joint robotic fish to track a desired planar path. Based on the Newton-Euler method, a dynamic model is constructed first. Thereafter, a path-following framework including an optimized LOS guidance strategy, an ADRC-based heading controller and a PID-based speed controller is developed. In particular, the LOS guidance strategy is utilized to select continuous target points for a smooth path-following process while the inner-loop ADRC-based heading controller is implemented for different tracking tasks even with internal model uncertainty and external disturbance. Finally, extensive simulations demonstrate the performance of the proposed control strategy. In summary, the proposed LOS path-following system is effective and robust for real-time path-following tasks even in the presence of disturbance.

Future work will focus on the 3-D path-following task for a multi-joint robotic fish in an unstructured environment.

REFERENCES

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

[2] D. Scaradozzi, G. Palmieri, D. Costa, and A. Pinelli, "BCF swimming locomotion for autonomous underwater robots: A review and a novel solution to improve control and efficiency," *Ocean Eng.*, vol. 130, pp. 437–453, 2017.

[3] A. Raj, and A. Thakur, "Fish-inspired robots: Design, sensing, actuation, and autonomy - a review of research," *Bioinspir. Biomimetics*, vol. 11, no. 3, pp. 031001, 2016.

[4] K. A. Morgansen, B. I. Triplett, and D. J. Klein "Geometric methods for modeling and control of free-swimming fin-actuated underwater vehicles," *IEEE Trans. Rob.*, vol. 23, no. 6, pp. 1184–1199, 2007.

[5] S. Verma, K. Abidi, and J. X. Xu, "Terminal sliding mode control for speed tracking of a carangiform robotic fish," in *Proc. IEEE Int. Workshop Var. Struct. Syst.*, Nanjing, China, Jun. 2016, pp. 345–350.

[6] M. Castano, and X. Tan, "Model predictive control-based path-following for tail-actuated robotic fish," *J. Dyn. Syst. Meas. Control*, vol. 141, no. 7, pp. 071012, 2019.

[7] E. Kelasidi, P. Liljeback, K. Y. Pettersen, and J. T. Gravdahl, "Integral line-of-sight guidance for path following control of underwater snake robots: Theory and experiments," *IEEE Trans. Rob.*, vol. 33, no. 3, pp. 610–628, 2017.

[8] M. Makrodimitris, K. Nanos, and E. Papadopoulos, "A novel trajectory planning method for a robotic fish," in *Mediterr. Conf. Control Autom., MED*, Valletta, Malta, Jul. 2017, pp. 1119–1124.

[9] Z. Peng, J. Wang, and Q. L. Han, "Path-following control of autonomous underwater vehicles subject to velocity and input constraints via neurodynamic optimization," *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8724–8732, 2019.

[10] F. Hawary, *The Ocean Engineering Handbook*. Boca Raton, FL, USA: CRC Press, 2000.

[11] J. Yu, J. Yuan, Z. Wu, and M. Tan, "Data-driven dynamic modeling for a swimming robotic fish," *IEEE Trans. Ind. Electron.*, vol. 63, no. 9, pp. 5632–5640, 2016.

[12] A. J. Ijspeert, A. Crespi, and J. M. Cabelguen, "Simulation and robotics studies of salamander locomotion: Applying neurobiological principles to the control of locomotion in robots," *Neuroinformatics*, vol. 3, no. 3, pp. 171–196, 2005.

[13] 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.

[14] Z. Wu, J. Yu , Z. Su, M. Tan, and J. Zhang, "Kinematic comparison of forward and backward swimming and maneuvering in a self-propelled sub-carangiform robotic fish," *J. Bionic Eng.*, vol. 11, no. 2, pp. 199–212, 2014.

[15] Z. Wu, J. Yu , Z. Su, M. Tan, and Z. Li, "Towards an Esox lucius inspired multimodal robotic fish," *Sci. China Inf. Sci.*, vol. 58, no. 5, pp. 1–13, 2015.