Trajectory Tracking Control for a Biomimetic Spherical Robot Based on ADRC

Mugen Zhou¹, Shuxiang Guo^{1,2*}, Xihuan Hou¹, He Yin¹,

Ao Li¹, Debin Xia¹, Zan Li¹, Meng Liu¹

¹Key Laboratory of Convergence Medical Engineering System and Healthcare Technology,

the Ministry of Industry and Information Technology, School of Automation, Beijing Institute of Technology

No.5, Zhongguancun South Street, Haidian District, Beijing 100081, China

²Faculty of Engineering, Kagawa University, 2217-20 Hayashi-cho, Takamatsu, Kagawa, Japan

zhoumugen@bit.edu.cn; guoshuxiang@bit.edu.cn

*Corresponding author

Abstract—In this article, a two-dimensional trajectory tracking control framework is proposed for biomimetic spherical robots (BSR) in a constrained workspace despite the lack of dynamic model parameters information and the effects of disturbances on the robot motion. Meanwhile, The research presents the general dynamics models of the robot and the thrusters allocator scheme to ensure the force generated by the propellers within the feasible range. Our trajectory tracking control framework relies on three active disturbance rejection controllers (ADRC) for the case of biomimetic spherical robots. More importantly, the controllers consider various practical operational constraints, such as continuous and smooth controller outputs, predefined velocity bounds, and thruster saturations. Finally, We assess the performance and feasibility of the proposed control framework through the simulations.

Index Terms—Biomimetic spherical robot, Trajectory tracking control, Active disturbance rejection control.

I. INTRODUCTION

In recent years, a growing number of scientists focus their attention on marine robots, including autonomous underwater vehicles (AUVs) [1], [2], unmanned surface vehicles (USVs) [3], [4], remotely operated vehicles (ROVs) [5], and biologically inspired robots [6]–[8] for commercial, military, and academic research applications [9]. These robots provide a wide range of capabilities such as underwater inspection and surveillance, ecosystem monitoring, underwater exploration [10]. Many of these applications require the marine robotic vehicles to operate accurately in a constrained workspace with disturbances of the environment and uncertainties of the model.

In particular, Accurate trajectory tracking control is essential for underwater robot applications, but designing a robust trajectory tracking is a challenging work due to uncertain model parameters, unknown disturbances, and various constraints [11]. As a result, there has been plenty of studies on this challenging topic in the past decade. The trajectory tracking problem is defined by controlling the vehicle reach and follow a desired trajectory, which is coupled with time assignments [9]. In comparison with path-following, the desired trajectory of trajectory tracking is dependent on time.

The most popular method for trajectory tracking control is sliding mode control (SMC), unlike conventional methods that are sensitive to parameter changes or rely strongly on accurate model parameters. Jiang et al. develop an adaptive robust integral sliding model controller to avoid the jump problem of the velocity by using bioinspired neurodynamics [12]. Zhou et al. integrate the backstepping method and radial basis function (RBF) neural network in the sliding mode controller to estimate the uncertainty of model parameters and the effect of external disturbances [13]. To provide the robustness and adaptation of the controller, Yan et al. design dual closed-loop integral sliding mode controllers, including a velocity loop and a position loop [14]. Without any prior knowledge of uncertainty and disturbance, sliding mode control method coupled with adaptive nonsingular integral terminal is proposed by Qiao et al. to improve the convergence speed [15]. Due to the input delay, an integrated time-delay sliding mode control strategy is designed in [16]. However, the "chattering" problem still exists, which may yield high-frequency dynamics [17].

Model predictive control (MPC) for trajectory tracking is an alternative method that combines the model dynamics and allows for minimal adjustment of controller by minimizing object function [3]. Meanwhile, MPC is an ideal tool to handle internal constraints, such as thrust saturation, velocity increment, and acceleration constraints and external constraints, including safe operating area and external disturbance [18], [19]. Hu et al. design a trajectory tracking controller using MPC to avoid obstacles [18]. Due to actuator saturation, a Lyapunov-based model predictive control (LMPC) framework is proposed by Chao et al. to improve the tracking performance and save computational resources [19]. An improved model predictive control method is proposed by Hou et al. to process pratical constraints and thrusters saturation [20]. Chao et al. develop a distributed NMPC algorithms to reduce the floating point operations [11]. A nonlinear model predictive dynamic positioning strategy to account for complex stochastic disturbances in [5]. Nevertheless, the hydrodynamic parameters identification for underwater robots is not easy to perform and the computational resources and time consumption of the MPC algorithm are high [3].

The observer has been widely applied to estimate internal model uncertainty and external environmental disturbance in robotics research field. With the help of the extended state observer (ESO), Peng et al. recover unmeasurable velocities and estimate total disturbance caused by internal uncertainty and external interference [21]. Qin et al. design a disturbance observer to deal with the effects of the external disturbances, especially thruster faults [22]. Lakhekar et al. combine the disturbance-observer-based control with fuzzy adapted S-Surface control to compensate the unknown disturbances and unmodeled dynamics [23]. Wang et al. propose a control architecture, which includes three active disturbance rejection control (ADRC) sub-controllers to allow for 3-D helical path following of an underwater biomimetic vehicle [24]. Inspired by the above motion control schemes, in particular, by ADRC [24], this paper aims to develop a trajectory tracking control scheme by using three ADRC controllers with internal uncertainties and external disturbances in a constrained workspace.

The rest of this article is organized as follows. The next section introduces the modeling of our biomimetic spherical robot [6], [25], [26] and the thrust allocation shceme. In Section III, the trajectory tracking control framework is detailed. To evaluate the robustness of the controller, we perform simulations for straight and circular trajectories in Section IV. Finally, the conclusion is provided in Section V.

II. MODELING OF THE BIOMIMETIC SPHERICAL ROBOT

A. Modeling of the Biomimetic Spherical Robot



Fig. 1. The world reference frame and the body-fixed reference frame.

To analyze the trajectory tracking clearly, the inertial reference frame $(O_E - X_E Y_E)$ and the body-fixed reference frame $(O_B - X_B Y_B)$ are described in Fig.1. The desired

trajectory is described as $[x_r(t), y_r(t), \psi_r(t)]^T$. In the world frame, states of BSR are expressed as:

$$\boldsymbol{\eta} = \left[x, y, \psi\right]^T \tag{1}$$

where (x, y) are the plane coordinates and the ψ is the yaw angle correspond to the surge orientation with respect to the positive x axis. In the body-fixed frame, motion states of the robot are represents as:

$$\mathbf{v} = \left[u, v, r\right]^T \tag{2}$$

where, u, v, r are the linear velocity in the surge, sway, and the yaw rate, respectively. And $\beta = \arctan(e_y/e_x)$ is the azimuth angles, where $e_x = x_r - x$, $e_y = y_r - y$ are the tracking error in the x-direction and y-direction, respectively.

In this paper, the trajectory tracking for biomimetic spherica robot, focuses on a two-dimensional horizontal control. The kinematic model of the BSR is described as follows:

$$\begin{cases} \mathbf{v} = \mathbf{J}(\psi)\dot{\boldsymbol{\eta}} \\ \dot{\mathbf{v}} = \mathbf{J}(\psi)(\ddot{\boldsymbol{\eta}} + \hat{\psi}\dot{\boldsymbol{\eta}}) \end{cases}$$
(3)

where $\dot{\boldsymbol{\eta}} = [\dot{x}, \dot{y}, \dot{\psi}]^T$ denotes the velocities of the BSR in x-direction, y-direction and the angular velocity around zdirection in the inertial frame, respectively; $\ddot{\boldsymbol{\eta}} = [\ddot{x}, \ddot{y}, \ddot{\psi}]^T$ represents the corresponding accelerations and angular acceleration in the inertial frame, respectively; $\dot{\mathbf{v}} = [\dot{u}, \dot{v}, \dot{r}]^T$ refers to the corresponding accelerations and angular acceleration respectively. $\boldsymbol{J}(\psi)$ is the rotation matrix depending on the yaw angle ψ :

$$\boldsymbol{J}(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0\\ -\sin(\psi) & \cos(\psi) & 0\\ 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{\hat{\psi}} = \begin{bmatrix} 0 & \dot{\psi} & 0\\ -\dot{\psi} & 0 & 0\\ 0 & 0 & 0 \end{bmatrix}$$

As the low speed of the BSR motion, the nonlinear damping is neglected since the linear damping outweigh the nonlinear damping greatly. Considering the distribution of thruster, the BSR with H configuration is unable to generate force in the sway direction in that the control of the BSR is underactuated. The dynamic model is expressed as equation (4):

$$M\dot{\mathbf{v}} + C(\mathbf{v})\mathbf{v} + D\mathbf{v} = \tau + \tau_d \tag{4}$$

With

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_u, 0, \tau_r \end{bmatrix}^T \quad \boldsymbol{\tau}_d = \begin{bmatrix} \tau_{du}, \tau_{dv}, \tau_{dr} \end{bmatrix}^T$$
$$\boldsymbol{M} = \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & 0 \\ 0 & 0 & m_{33} \end{bmatrix} \quad \boldsymbol{D} = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix}$$
$$\boldsymbol{C}(\mathbf{v}) = \begin{bmatrix} 0 & 0 & -m_{22}v \\ 0 & 0 & m_{11}u \\ m_{22}v & -m_{11}u & 0 \end{bmatrix}$$

where M is the inertial matrix including hydrodynamic additional mass. $C(\mathbf{v})$ is the coriolis and centripetal matrix.

D is the linear hydrodynamic damping matrix. τ is the input vectors, where τ_u, τ_r describe the propulsive force and force moment, respectively. $\tau_d \in \mathbb{R}^3$ represents the nonlinear external disturbances on surge, sway and yaw, which are supposed to be time-varying bounded.

B. Thrust Allocation Scheme

Considering the response and saturation characteristics of the propellers, a reasonable thrust allocation scheme is designed to make the BSR motion more stable.



Fig. 2. The propeller distribution of the BSR with H configuration.

The applied force τ_u and force moment τ_r are defined as equation (5):

$$\tau_u = -f_1 + f_2 + f_3 - f_4$$

$$\tau_r = (-f_1 + f_2 - f_3 + f_4) \cdot lsin\alpha$$

$$= (-f_1 + f_2 - f_3 + f_4) \cdot a$$
(5)

where f_1 , f_2 , f_3 and f_4 are the forces generated by the corresponding propellers, as shown in Fig.2(a). Each propeller is fixed and generates forces opposite to the direction of the water outlet. 2a = 0.055(m) is the distance between the transverse propellers, α is the angle between the line of the longitudinal thrusters and of the diagonal thrusters, l is the distance between the propeller and the geometric center, as shown in Fig.2(b).

Further, the input forces $F = [f_1, f_2, f_3, f_4]^T$ for each propeller can be written as:

$$\boldsymbol{F} = \begin{pmatrix} 1 & -\frac{1}{4} & -\frac{1}{4a} \\ 1 & \frac{1}{4} & \frac{1}{4a} \\ 1 & \frac{1}{4} & -\frac{1}{4a} \\ 1 & -\frac{1}{4} & \frac{1}{4a} \end{pmatrix} \begin{pmatrix} f_{static} \\ \tau_u \\ \tau_r \end{pmatrix}$$
(6)

where $f_{static} = 1(N)$ is a initial force, which can avoid sudden propeller start and speed jump. When every propeller has an same angular speed $\omega_{static} (\neq 0)$ and generates same force f_{static} , the total force and force moment vector $[\tau_u, \tau_r]^T$ applied to robot so the robot remains still.

It is worth pointing out that the propellers take a certain response time to reach the desired speed, so the the propellers can not start suddenly and jump in speed. In aadition, the propellers exit dead zone and saturation zone. Takeing into consideration the aforementioned constraints, we define $f_{min} = 0$ and $f_{max} = 2f_{static}$ as the lower and upper bounds of f_i , where i = 1, 2, 3, 4, respectively. Thus, we obtain the following constraints on the input force matrix:

$$f_{min} \leq f_i \leq f_{max}$$

$$0 \leq f_i \leq 2f_{static}$$

$$(7)$$

So the input vectors $[\tau_u, \tau_r]^T$ meet the *Manhattan Distance* constraint as equation (8). When the desired force or moment exceeds the feasible range, the desired force and moment vector is scaled proportionally to make it satisfy the constraint.

$$\left|\tau_{u}\right| + \left|\frac{\tau_{r}}{a}\right| \le 4f_{static} \tag{8}$$

III. TRAJECTORY TRACKING CONTROL METHODOLOGY

In this section, we give an overview of the trajectory tracking framework, which is composed of three main modules: position controller (Section III-A), yaw controller (Section III-B), and the force allocator (Section II-C), as shown in Fig.3. The input of the trajectory tracking framework is a global reference trajectory predefined and the operation constraints include predefined velocity bounds and thruster saturations. The position controller and the yaw controller all based on the ADRC controller computes the corresponding force and force moment vector that satisfies the operation constraints, respectively. The force allocator based on dynamics can map the force and moment into the forces of each propeller. Finally, the global camera provides an estimation of the robot position and yaw angle.



Fig. 3. Overview of the trajectory tracking control framework.

A. Position contronller

The position controller is composed of X position controller and Y position controller, which control x and y in the inertial reference frame.



Fig. 4. Block diagram of X position controller.

Based on the model of the BSR, the dynamic equation of position in the X-direction can be represented as follows:

$$\begin{cases} \dot{x} = v_x \\ \ddot{x} = f_x + b_x \tau_x \end{cases}$$
(9)

with

$$f_x = -\dot{\psi}\dot{y} + \frac{\cos(\psi)}{m_{11}}(m_{22}\dot{\psi}v - d_{11}u + \tau_{du}) + \frac{\sin(\psi)}{m_{22}}(m_{11}\dot{\psi}u + d_{22}v - \tau_{dv}) b_x = \frac{1}{m_{11}} \quad \tau_x = \cos(\psi)\tau_u$$
(10)

where f_x is a summation of the internal and external disturbance. b_x denotes the control gain. Specifically, Fig.4 shows that a trcking differentiator is used to generate the tracking signal x_1 and the differential signal x_2 of the x-axis coordinate value x_r of the target point on the trajectory. An ESO outputs the estimations (ie., $\zeta_{x1}, \zeta_{x2}, \zeta_{x3}$) of the systems states x, \dot{x} , and the general disturbance $f_x(\cdot) + (b_x - b_{x0})\tau_x$ based on the position value feedback. by actively compensating f_x using ζ_{x3} , the control law for equation (11) is given by:

$$\tau_x = \frac{\tau_{x0} - \zeta_{x3}}{b_{x0}} \tag{11}$$

Therefore, the equation (9) is simplified to a double integrator:

$$\ddot{x} \approx \tau_{x0}$$
 (12)

A proportional-derivative controller is designed to control it:

$$\tau_{x0} = k_{xp} e_{x1} + k_{xd} e_{x2} \tag{13}$$

where k_{xp} and k_{xd} are the proportional gain and derivative gain, respectively. $e_{x1} = x_1 - \zeta_{x1}$ and $e_{x2} = x_2 - \zeta_{x2}$ are states errors. The differential signal subjects to the acceleration limit of δ_x . More details about implement the ADRC are described in [24].

And the dynamic equation of position in the Y-direction can be represented as follows:

$$\begin{cases} \dot{y} = v_y \\ \ddot{y} = f_y + b_y \tau_y \end{cases}$$
(14)

with

$$f_{y} = \dot{\psi}\dot{x} + \frac{\sin(\psi)}{m_{11}}(m_{22}\dot{\psi}v - d_{11}u + \tau_{du}) + \frac{\cos(\psi)}{m_{22}}(-m_{11}\dot{\psi}u - d_{22}v + \tau_{dv}) b_{y} = \frac{1}{m_{11}} \quad \tau_{y} = \sin(\psi)\tau_{u}$$
(15)

Since the ADRC controller to force BSR to track y_r is

similar with tracking x_r , the detailed process is not described again. Moreover, by combining the control signal in X and Y direction, the surge control is written as:

$$\tau_u = \tau_x \cos(\psi) + \tau_y \sin(\psi) \tag{16}$$

B. Yaw controller



Fig. 5. Block diagram of yaw angle controller.

Based on the model of the BSR, the dynamic equation of yaw motion can be described as follows:

$$\begin{cases} \dot{\psi} = \omega \\ \ddot{\psi} = f_{\psi} + b_{\psi} \tau_{\psi} \end{cases}$$
(17)

with

$$f_{\psi} = -\frac{d_{33}}{m_{33}}\dot{\psi} + \frac{m_{11} - m_{22}}{m_{33}}uv + \frac{1}{m_{33}}\tau_{dr} b_{\psi} = \frac{1}{m_{33}}\tau_{\psi} = \tau_r$$
(18)

where f_{ψ} represents the general disturbance including the external disturbance and internal dynamics. It is worth noting that the desired signal of the tracking differentiator is the azimuth angles β rather than ψ_r , as shown in Fig.5. Yaw controller is analogous to the other two controller and therefore the detailed expressions are omitted here.

IV. SIMULATION AND RESULTS

The section presents two simulations of the BSR, using the developed method of trajectory tracking. The goal of the simulations is to illustrate the performance and validate the effectiveness of the controller. The dynamic model parameters and the disturbance forces applied to the BSR in the simulations are shown in Table I. The disturbance forces on

TABLE I Model Parameters

m ₁₁ =4	m ₂₂ =4	m ₃₃ =4	d_{11} =0.14	d ₂₂ =8	d ₃₃ =0.24
$\tau_{du}, \tau_{dv} = 0.1(rand - 0.5)$			$\tau_{dr} = 0.008(rand - 0.5)$		

TABLE II Controller Parameters

X	r ₀ =2	h ₀ =0.3	h=0.02	a ₁ =0.25	a ₂ =1.5
	$k_p=4$	$k_d=4$	δ=0.14	$\omega_o=8$	b ₀ =0.24
Y	$r_0=2$	h ₀ =0.3	h=0.02	a ₁ =0.25	a ₂ =1.5
	$k_p=4$	$k_d=4$	δ=0.14	$\omega_o=8$	b ₀ =0.24
Yaw	$r_0=200$	h ₀ =0.5	h=0.02	a1=0.25	a ₂ =1.5
	kp=25	$k_d=10$	δ=0.14	$\omega_o=20$	b ₀ =16.55

both surge and sway are modeled as uniform random noise with zero mean and variance of 0.1. The disturbance forces on yaw are defined as uniform random noise with zero mean and variance of 0.008. The update frequency of the controller is set to 50Hz. The parameters of position and yaw controllers are listed in Table II.

A. Straight trajectory tracking

Fig. 6. Performance of the straight trajectory tracking.

The straight trajectory is generated with a surge velocity of $v_r = 0.15$ m/s and initial yaw angle of $\theta_r = \frac{\pi}{4}$. The start point is (-2,-2) and the robot starts at point (-1,-2).

Fig. 7. The errors of BSR while tracking straight trajectory.

Fig.6 depicts the time evolution of the position for the BSR tracking the desired trajectory. The red line denotes the actual

position and the blue line the desired position. Notice that the controller tracks the straight trajectory perfectly in simulation despite the unknown disturbance. The tracking error of the BSR while tracking the straight trajectory are shown in Fig.7. $e_d = \sqrt{e_x^2 + e_y^2}$ converges to 0.087 m and the maximum e_{θ} is just 0.0016 rad. More importantly, the surge speed of the BSR converges to 0.15 m/s, which is equal to v_r .

B. Circular trajectory tracking

Fig. 8. Performance of the circular trajectory tracking.

The angular velocity ω_r of the circular trajectory is 0.1rad/s and radius R 1m. The center and the start point of the trajectory are (0,0) and (1,0), respectively. The robot starts at the origin (0,0) with initial velocity $v_0(0\text{m/s})$ and angle velocity $\omega_0(0\text{m/s})$.

Fig.8 depicts the BSR tends to the reference trajectory and reaching the reference point, finally. The red line denotes the actual trajectory and the blue line represents the reference trajectory. Fig.9 shows e_d converges to 0.059 m and e_θ tends to 0.030 rad. The velocity in surge and yaw angle converges to the speed of the reference trajectory. But because of the noise, the yaw angle rate oscillates at low frequency.

V. CONCLUSIONS

This article presents the kinematics and dynamics modeling of the biomimetic spherical robot, and the design of the trajectory tracking control framework. The thrust allocation scheme addresses the problem that the propellers can not start suddenly and jump in speed and takes into consideration the dead zone and saturation zone. More importantly, the trajectory tracking framework integrating ADRC is designed to reduce the influence of unknown disturbances and eliminates the dependence on accurate robot model parameters.

Fig. 9. The error of BSR while tracking circular trajectory.

The simulation results suggest that the ADRC controller achieve accurate trajectory tracking and ensure smoothness and continuity of controller outputs.

ACKNOWLEDGMENT

This research was supported by National Natural Science Foundation of China (61773064, 61503028), National Key Research and Development Program of China (2017YFB1304404), and National Hightech Research and Development Program (863 Program) of China (No.2015AA043202).

REFERENCES

- S. Heshmati-Alamdari, G. C. Karras, P. Marantos, and K. J. Kyriakopoulos, "A Robust Predictive Control Approach for Underwater Robotic Vehicles," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2352–2363, 2020.
- [2] S. Gu, S. Guo, and L. Zheng, "A highly stable and efficient spherical underwater robot with hybrid propulsion devices," *Autonomous Robots*, vol. 44, no. 2, 2020. doi: 10.1007/s10514-019-09895-8.
- [3] W. Wang, L. A. Mateos, S. Park, and et al, "Design, Modeling, and Nonlinear Model Predictive Tracking Control of a Novel Autonomous Surface Vehicle," in 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 6189–6196, 2018. doi: 10.1109/ICRA.2018.8460632.
- [4] W. Wang, T. Shan, P. Leoni, and et al, "Roboat II: A Novel Autonomous Surface Vessel for Urban Environments," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1740–1747, 2020. doi: 10.1109/IROS45743.2020.9340712.
- [5] Y. Cao, B. Li, Q. Li, and et al, "A Nonlinear Model Predictive Controller for Remotely Operated Underwater Vehicles With Disturbance Rejection," *IEEE Access*, vol. 8, 2020. doi: 10.1109/AC-CESS.2020.3020530.
- [6] H. Xing, S. Guo, L. Shi, and et al, "Design, modeling and experimental evaluation of a legged, multi-vectored water-jet composite driving mechanism for an amphibious spherical robot," *Microsystem Technologies*, vol. 26, no. 5, 2019. doi: 10.1007/s00542-019-04536-7.
- [7] W. Wang, X. Dai, L. Li, and et al, "Three-dimensional modeling of a fin-actuated robotic fish with multimodal swimming," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 4, pp. 1641–1652, 2018.
- [8] L. Shi, Y. Hu, S. Su, and et al, "A fuzzy pid algorithm for a novel miniature spherical robots with three-dimensional underwater motion control," *Journal of Bionic Engineering*, vol. 17, no. 5, p. 959–969, 2020.

- [9] X. Xiang, C. Yu, L. Lapierre, J. Zhang, and et al, "Survey on fuzzy-logic-based guidance and control of marine surface vehicles and underwater vehicles," *International Journal of Fuzzy Systems*, vol. 20, no. 2, pp. 572–586, 2018.
- [10] J. Guo, C. Li, and S. Guo, "Path optimization method for the spherical underwater robot in unknown environment," *Journal of Bionic Engineering*, p. 944–958, 2020. doi: 10.1007/s42235-020-0079-3.
- [11] C. Shen and Y. Shi, "Distributed implementation of nonlinear model predictive control for AUV trajectory tracking," *Automatica*, vol. 115, p. 108863, 2020. doi: 10.1016/j.automatica.2020.108863.
- [12] Y. Jiang, C. Guo, and H. Yu, "Robust trajectory tracking control for an underactuated autonomous underwater vehicle based on bioinspired neurodynamics," *International Journal of Advanced Robotic Systems*, vol. 15, no. 5, 2018. doi: 10.1177/1729881418806745.
- [13] J. Zhou, X. Zhao, T. Chen, Z. Yan, and et al, "Trajectory Tracking Control of an Underactuated AUV Based on Backstepping Sliding Mode With State Prediction," *IEEE Access*, vol. 7, 2019. doi: 10.1109/ACCESS.2019.2958360.
- [14] Z. Yan, M. Wang, and J. Xu, "Robust adaptive sliding mode control of underactuated autonomous underwater vehicles with uncertain dynamics," *Ocean Engineering*, vol. 173, pp. 802–809, 2019. doi: 10.1016/j.oceaneng.2019.01.008.
- [15] L. Qiao and W. Zhang, "Trajectory Tracking Control of AUVs via Adaptive Fast Nonsingular Integral Terminal Sliding Mode Control," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1248– 1258, 2020.
- [16] J. Zhou, X. Zhao, Z. Feng, and D. Wu, "Trajectory tracking sliding mode control for underactuated autonomous underwater vehicles with time delays," *International Journal of Advanced Robotic Systems*, vol. 17, no. 3, 2020. doi: 10.1177/1729881420916276.
- [17] S. Heshmati-Alamdari, A. Nikou, and D. V. Dimarogonas, "Robust trajectory tracking control for underactuated autonomous underwater vehicles in uncertain environments," *IEEE Transactions on Automation Science and Engineering*, 2020. doi: 10.1109/TASE.2020.3001183.
- [18] Z. Hu, D. Zhu, C. Cui, and B. Sun, "Trajectory Tracking and Replanning with Model Predictive Control of Autonomous Underwater Vehicles," *Journal of Navigation*, vol. 72, no. 2, p. 321–341, 2019.
- [19] C. Shen, Y. Shi, and B. Buckham, "Trajectory Tracking Control of an Autonomous Underwater Vehicle Using Lyapunov-Based Model Predictive Control," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5796–5805, 2018.
- [20] X. Hou, S. Guo, L. Shi, and et al, "Improved Model Predictive-Based Underwater Trajectory Tracking Control for the Biomimetic Spherical Robot under Constraints," *Applied Sciences*, vol. 10, no. 22, 2020. doi: 10.3390/app10228106.
- [21] Z. Peng and J. Wang, "Output-Feedback Path-Following Control of Autonomous Underwater Vehicles Based on an Extended State Observer and Projection Neural Networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 4, pp. 535–544, 2018.
- [22] H. Qin, Z. Wu, Y. Sun, and H. Chen, "Disturbance-Observer-Based Prescribed Performance Fault-Tolerant Trajectory Tracking Control for Ocean Bottom Flying Node," *IEEE Access*, vol. 7, 2019. doi: 10.1109/ACCESS.2019.2909733.
- [23] G. V. Lakhekar, L. M. Waghmare, and R. G. Roy, "Disturbance Observer-Based Fuzzy Adapted S-Surface Controller for Spatial Trajectory Tracking of Autonomous Underwater Vehicle," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 4, pp. 622–636, 2019.
- [24] R. Wang, S. Wang, Y. Wang, and et al, "Three-Dimensional Helical Path Following of an Underwater Biomimetic Vehicle-Manipulator System," *IEEE Journal of Oceanic Engineering*, vol. 43, no. 2, pp. 391–401, 2018.
- [25] X. Hou, S. Guo, L. Shi, and et al, "Hydrodynamic analysis-based modeling and experimental verification of a new water-jet thruster for an amphibious spherical robot," *Sensors*, vol. 19, no. 2, 2019. doi: 10.3390/s19020259.
- [26] H. Xing, S. Guo, L. Shi, and et al, "A novel small-scale turtleinspired amphibious spherical robot," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1702–1707, 2019. doi: 10.1109/IROS40897.2019.8968304.