



Backstepping control of gliding robotic fish for pitch and 3D trajectory tracking[☆]

Demetris Coleman^{a,*}, Maria Castaño^b, Xiaobo Tan^a

^a Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA

^b Johns Hopkins Applied Physics Lab, Laurel, MD, 20723, USA

ARTICLE INFO

Keywords:

Robotics
Backstepping
Nonlinear control
Gliding robotic fish
Underwater glider

ABSTRACT

Underwater gliders are known for their energy-efficiency and long-duration operations, with demonstrated applications in ocean exploration, fish tracking, and environmental sampling. Many applications such as exploring a large area of underwater ruins would benefit from accurate trajectory tracking. Trajectory tracking is particularly challenging for underwater gliders due to their under-actuated, highly nonlinear dynamics. Taking gliding robotic fish as an example, a backstepping-based controller is proposed to track the desired pitch angle and reference position in the 3D space. In particular, under-actuation is addressed by exploiting the coupled dynamics and introducing a modified error term that combines pitch and horizontal position tracking errors. Two-time-scale analysis of singularly perturbed systems is used to establish the convergence of all tracking errors to a neighborhood around zero. The effectiveness of the proposed control scheme is demonstrated via simulation and experimental results, and its advantages are shown via comparison with a PID controller and a baseline backstepping controller that does not use the modified error. This paper is accompanied by a video available at: <https://youtu.be/D8Vej3weeGc>.

1. Introduction

The concept of underwater gliders was introduced by Henry Stommel just over a decade before the turn of the century (Stommel, 1989). Underwater gliders are known for their high energy-efficiency and exceptionally long operation time. They use variable buoyancy, hydrofoils, and a shifting center of gravity to realize horizontal travel. They are typically operated to achieve steady-state motion patterns, including sawtooth-like rectilinear gliding, and spirals induced by controlling the vehicle's roll angle or by deflecting control surfaces. The success of the early gliders such as SLOCUM (Stommel, 1989), Spray (Sherman, Davis, Owens, & Valdes, 2001), and Seaglider (Sliwka, Clement, & Probst, 2012) has inspired development of other underwater vehicles that exploit gliding (Yuan, Wu, Yu, & Tan, 2017; Zhang, 2014). One example is the gliding robotic fish (Ennasr, 2020; Zhang, 2014), which achieves both high energy-efficiency and high maneuverability by combining the gliding mechanism with the tail-actuated swimming of robotic fish (Wang & Tan, 2013). It has demonstrated promise in environmental sensing and fish tracking applications (Ennasr et al., 2020; Zhang, Ennasr, Litchman, & Tan, 2016).

Early work in control of gliders saw the use of PID controllers for their simplicity (Sherman et al., 2001; Sliwka et al., 2012). More advanced and model-based control methodologies have been proposed

in the past two decades. For example, Leonard and Graver used a linear quadratic regulator on linearized dynamics to control the magnitude of velocity on a steady-state glide path (Graver & Leonard, 2001; Leonard & Graver, 2001). Isa and Arshad analyzed the use of a neural network as a model predictive controller and a gain-tuner algorithm to control the pitch angle and linear velocities based on a linearized glider model (Isa & Arshad, 2013). Wang et al. used model predictive control for depth regulation along with a PID controller for maintaining heading (Wang, Wu, Tan, & Yu, 2019). Neural network-based control was used to implement a self-tuning PID controller to track the velocity along a single axis in the inertial frame (Dong, Guo, Lin, Li, & Wang, 2012). Nag, Patel, and Akbar (2013) compared fuzzy logic control against PID for pitch and depth tracking. Mahmoudian and Woosely developed an efficient path planning strategy that concatenates equilibrium turning and gliding motions and then implemented the strategy using PID controllers to reach a specified center of gravity and center of buoyancy (Mahmoudian & Woolsey, 2008). Zhang, Tan, and Khalil (2012) used nonlinear passivity-based control to stabilize the glide path of a glider in the sagittal plane with a whale-like tail. Sliding mode control has also been explored because of its robustness to disturbances. Castaño and Tan proposed a sliding mode controller for the simultaneous stabilization of pitch and yaw (Castaño & Tan,

[☆] This research was supported in part by National Science Foundation (IIS 1715714, IIS 1848945) and the Rose Graduate Fellowship Fund in Water Research.

* Corresponding author.

E-mail addresses: colem404@msu.edu (D. Coleman), castanom@msu.edu (M. Castaño), xbtan@egr.msu.edu (X. Tan).

2015). Yang and Ma used sliding mode control to track trajectories of the pitch angle and ballast mass (Yang & Ma, 2010). Mat-Noh et al. used a linearized glider model to compare an Integral Super Twisting Sliding Mode controller with several other sliding mode variants for stabilizing a gliding path between 30 and 45 degrees (Mat-Noh, Arshad, Mohd-Mokhtar, & Khan, 2017). In Ullah, Ovinis, Baharom, Javaid, and Izhar (2015), several different control strategies for underwater gliders are compared. While extensive work has been done in glider control, the focus of these approaches has been mainly on stabilization based on linearized models, or single-input–single-output control of heading, pitch, velocity, or depth.

Trajectory tracking is a fundamental and valuable ability for robots exploring complex environments. In particular, it enables improved performance for applications in oceanography, marine science, water quality monitoring, and surveillance, and has direct relevance to various sampling and target-tracking applications in the underwater environment. Extensive work has been done on trajectory tracking and path following for propeller-driven underwater vehicles (Aguiar & Hespanha, 2003; Do & Pan, 2009; Do, Pan, & Jiang, 2004; Karkoub, Wu, & Hwang, 2017; Rezaadegan, Shojaei, Sheikholeslam, & Chatraei, 2015; Wang, Wang, Wei, & Zhang, 2019; Xu, Tang, Han, & Xie, 2019; Zheng, Ruan, & Zhu, 2019). For underwater gliders, however, position control or trajectory tracking in the 3D space is scarce. One of the very few examples considering the full dynamic model of a gliding system is Cao, Cao, Zeng, and Lian (2016), where the authors proposed an adaptive backstepping controller for tracking the velocity magnitude, yaw angle and pitch angle of an underwater glider. However, position tracking was not addressed despite that it is often times more valuable when, for example, operating in cluttered underwater environments.

The difficulty in controlling the 3D position of gliding-type robots lies in the fact that only the pitch moment and vertical motion are actively controlled during gliding motion. Like multirotor drones, the robot must orient itself to achieve planar motion. Unlike multirotor drones, this does not produce an orientation-dependent thrust vector. Instead, planar motion is achieved through the lift forces applied on the wings and control surfaces from the surrounding water. The magnitude of the lift force is heavily dependent on pitch angle and the vertical velocity. In addition, a gliding robot usually cannot produce a yaw moment without a non-zero velocity and non-zero pitch angle.

In this work, a backstepping-based trajectory-tracking controller is proposed for gliding-based underwater robots. The model of a gliding robotic fish is taken as the focus. Three control inputs are used to simultaneously track the pitch angle, due to its strong influence over the planar movement, and the 3D position. To facilitate the control design, the tracking errors are expressed in a cylindrical coordinate system with its origin coinciding with that of the robot's body-fixed frame. Furthermore, an error function modifying the pitch tracking error with a term dependent on the horizontal position error is introduced to address the problem of under-actuation. An intuition and a rigorous proof are then given to explain why the proposed controller is able to achieve tracking of all four components of the reference trajectories (pitch and 3D position). Finally, simulation studies and experimental results are presented to show the efficacy of the proposed control design.

Preliminary versions of some of this work were reported in the conference papers (Coleman, Castaño, Ennasr, & Tan, 2019) (which only deals with tracking of the pitch and the position in the sagittal plane) and Coleman and Tan (2020) (which deals with tracking of the pitch and the 3D position). Both Coleman et al. (2019) and Coleman and Tan (2020) only contained simulation results. This work improves upon Coleman and Tan (2020) in several significant ways. First, the controller is redesigned to allow for more freedom in the choice of controller gains. Second, it rigorously demonstrates how the proposed scheme achieves convergence of the position and pitch angle tracking errors using two-time-scale analysis of singularly perturbed systems. Third, it compares the proposed approach via extensive simulation to

a PID controller and a baseline backstepping controller not using the modified error. Fourth, an experimental platform is developed and a procedure for estimating crucial parameters present in the model is presented. Fifth, a model-based observer is implemented to estimate the body-fixed velocities that are otherwise not directly accessible from onboard sensors. Finally, the proposed approach is implemented in a miniature gliding robotic fish and the advantage of the proposed approach over a well tuned PID controller and a baseline backstepping controller is shown.

The rest of this paper is organized as follows. Section 2 describes the system model and the problem formulation. Section 3 provides an overview of the control design process and the analysis of the closed-loop system, followed by simulation results in Section 4. Experimental results, including the experimental setup and other implementation details, are given in Section 5. Finally, concluding remarks are provided in Section 6.

2. System modeling and problem formulation

2.1. Gliding robotic fish model

The robot has two relevant reference frames shown in Fig. 1. The first is the inertial frame, represented by A_{xyz} . The origin A is a fixed point in space with an axis A_z along the direction of gravity and axes A_x and A_y defined in the horizontal plane. The body-fixed frame is denoted by O_{x_b, y_b, z_b} , with the origin O at the geometric center of the glider body as opposed to the center of gravity, which would change with the position r_{p1} of an internal movable mass (to be further explained below). It has axis O_{x_b} along the body longitudinal axis pointing towards the robot's front, axis O_{z_b} perpendicular to the O_{x_b} axis in the sagittal plane of the robot pointing towards the bottom of the robot, and axis O_{y_b} formed according to the right-hand orthonormal principle with respect to O_{x_b} and O_{z_b} . The glider is modeled as a 6 degree-of-freedom rigid body with an internal movable mass, a water tank, and a servo-actuated tail that has its own axis of rotation parallel to the robot's O_{z_b} axis, at an offset along the O_{x_b} axis. While the tail can be used for both propulsion and steering, this work only focuses on its steering capability in this work. The motion of the internal movable mass, \bar{m} , is restricted to the O_{x_b} axis by a linear actuator and its position r_{p1} has significant influence over the robot's pitch angle via its effect on the center of gravity. In Fig. 1, r_{p1} is measured along the arrow pointing from the origin O to the point mass \bar{m} . Lastly, a linear actuator-driven syringe pump controls the negative net buoyancy m_0 , which is given as the sum of (1) the mass m_s representing the portion of the robot's mass distribution (including the water in the tank) that has its own center of mass coinciding with the geometric center O thus not contributing to the moment, (2) the internal movable mass \bar{m} , and (3) the non-uniformly distributed mass m_w represented as a stationary point mass that is offset from O and accounting for the moment effect caused by the discrepancy between the robot's center of mass and its center of geometry O , minus the mass m of the water displaced by the robot. This can be expressed as $m_0 = m_s + \bar{m} + m_w - m$, where $m_0 < 0$ causes the robot to float and $m_0 > 0$ causes the robot to sink. The robot essentially controls m_0 by changing the amount of water in the tank (thus varying m_s). In summary, the control inputs include the negative net buoyancy m_0 , the position r_{p1} of the movable mass from the geometric center of the robot, and the tail angle δ .

The state vector consists of the position $b_i = [x, y, z]^T$ of the robot, the orientation with respect to the inertial frame, and the body-fixed linear velocities $v_b = [v_1, v_2, v_3]^T$ and body-fixed angular velocities $\omega_b = [\omega_1, \omega_2, \omega_3]^T$. With the orientation represented by Euler angles (roll, pitch, and yaw), $\Psi = [\phi, \theta, \psi]^T$, the state vector can be written as
$$X = [x, y, z, \phi, \theta, \psi, v_1, v_2, v_3, \omega_1, \omega_2, \omega_3]^T. \quad (1)$$

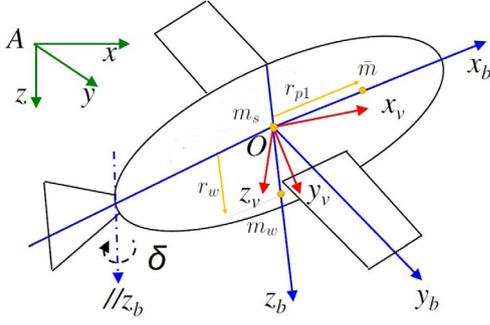


Fig. 1. Illustration of robot reference frames and mass distribution (Zhang, 2014).

The dynamic equations are (Zhang, 2014)

$$\begin{cases} \dot{b}_i = Rv_b \\ \dot{\Psi} = S_\omega \omega_b \\ \dot{v}_b = M^{-1}((Mv_b) \times \omega_b + m_0 g R^T k + F_{ext}) \\ \dot{\omega}_b = J^{-1}(-J\omega_b + (J\omega_b) \times \omega_b + (Mv_b) \times v_b + T_{ext} \\ + m_w g r_w \times (R^T k) + \bar{m} g r_p \times (R^T k)) \end{cases} \quad (2)$$

where R is a 3×3 rotation matrix parameterized by the Euler angles $\Psi = [\phi, \theta, \psi]^T$ following the ZYX convention. It is given as

$$R = \begin{pmatrix} c_\theta c_\psi & c_\psi s_\theta s_\phi - c_\phi s_\psi & s_\phi s_\psi + c_\phi c_\psi s_\theta \\ c_\theta s_\psi & c_\phi c_\psi + s_\theta s_\phi s_\psi & c_\phi s_\theta s_\psi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{pmatrix},$$

where c_q and s_q with $q = \phi, \theta, \psi$ represent sine and cosine of the variable in the subscript. S_ω , written as

$$S_\omega = \begin{pmatrix} 1 & \tan(\theta) \sin(\phi) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{pmatrix}$$

is a 3×3 matrix that relates the body-fixed angular velocities to Euler angle rates. $M = \text{diag}\{m_1, m_2, m_3\}$ is the total mass matrix incorporating the added-mass effect from the surrounding fluid, g is Earth's gravitational constant, $k = [0, 0, 1]^T$, and $F_{ext} = R_{bv} [-D, F_s, -L]^T$ is the hydrodynamic force vector. $J = \text{diag}\{J_1, J_2, J_3\}$ is the total inertia matrix, $T_{ext} = R_{bv} [M_1, M_2, M_3]^T$ is the hydrodynamic moment vector, $r_p = [r_{p1}, 0, 0]^T$, and $r_w = [0, 0, r_{w3}]^T$ is the position of the center of gravity of the non-uniformly distributed mass m_w . Note that M and J are assumed to be diagonal considering the simple and symmetric geometry of the gliding robotic fish. Also, note that in this work the non-uniformly distributed mass m_w is assumed to be located along the Oz_b axis, which is justified by the design consideration that the robot shall have zero pitch when the movable mass is at its neutral position. R_{bv} is a 3×3 rotation matrix parameterized by the angle of attack $\alpha = \arctan \frac{v_3}{v_1}$ and the side-slip angle $\beta = \arcsin \frac{v_2}{\sqrt{v_1^2 + v_2^2 + v_3^2}}$ that maps the hydrodynamic forces and moments from the velocity reference frame to the body-fixed frame. It is given by Zhang (2014)

$$R_{bv} = \begin{pmatrix} \cos(\alpha) \cos(\beta) & -\cos(\alpha) \sin(\beta) & -\sin(\alpha) \\ \sin(\beta) & \cos(\beta) & 0 \\ \sin(\alpha) \cos(\beta) & -\sin(\alpha) \sin(\beta) & \cos(\alpha) \end{pmatrix} \quad (3)$$

The hydrodynamic forces and torques, including lift L , drag D , side force F_s , roll moment M_1 , pitch moment M_2 , and yaw moment M_3 ,

are given as (Zhang, 2014)

$$\begin{cases} D = \frac{1}{2} \rho V^2 S (C_{D0} + C_D^\alpha \alpha^2 + C_D^\delta \delta^2) \\ F_s = \frac{1}{2} \rho V^2 S (C_{F_s}^\beta \beta + C_{F_s}^\delta \delta) \\ L = \frac{1}{2} \rho V^2 S (C_{L0} + C_L^\alpha \alpha) \\ M_1 = \frac{1}{2} \rho V^2 S (C_{M_R}^\beta \beta + K_{q1} \omega_1) \\ M_2 = \frac{1}{2} \rho V^2 S (C_{M_0} + C_{M_P}^\alpha \alpha + K_{q2} \omega_2) \\ M_3 = \frac{1}{2} \rho V^2 S (C_{M_Y}^\beta \beta + K_{q3} \omega_3 + C_{M_Y}^\delta \delta) \end{cases} \quad (4)$$

where the parameters associated with K_q and C notations are hydrodynamic constants, ρ is the fluid density, S is the characteristic surface area of the robot, and V is the magnitude of v_b .

For convenience, the linear and angular velocity dynamics can be abstracted as

$$\begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \\ \dot{v}_3 \\ \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{bmatrix} = \begin{bmatrix} f_{v11} + a_{v1} r_{31} u_1 + f_{v12} u_3 + f_{v13} u_3^2 \\ f_{v21} + a_{v2} r_{32} u_1 + f_{v22} u_3 + f_{v23} u_3^2 \\ f_{v31} + a_{v3} r_{33} u_1 + f_{v32} u_3 + f_{v33} u_3^2 \\ f_{\omega11} + f_{\omega12} u_3 \\ f_{\omega21} + a_{\omega2} r_{33} u_2 \\ f_{\omega31} + a_{\omega2} r_{32} u_2 + f_{\omega32} u_3 \end{bmatrix} \quad (5)$$

where $u_1 = m_0$, $u_2 = r_{p1}$, and $u_3 = \delta$ are the controls, a_{vi} ($i = 1, 2, 3$) and $a_{\omega j}$ are constants, r_{3j} ($j = 1, 2, 3$) are corresponding elements of R , and f_{vij} ($i, j = 1, 2, 3$) and $f_{\omega ij}$ ($i, j = 1, 2, 3$, when present) are the corresponding nonlinear functions of the state vector.

2.2. Problem formulation

The problem of trajectory tracking involves controlling a robot to follow a time-dependent path. In this work, the aim is to have the robot pose $P = [x, y, z, \theta]^T$, consisting of the 3D position and the pitch angle θ , follow a trajectory in the inertial coordinate system. The desired path is given by $P_d(t) = [x_d(t), y_d(t), z_d(t), \theta_d(t)]^T$. $\dot{P}_d(t)$ and $\ddot{P}_d(t)$ are assumed to be bounded and sufficiently smooth with $|\theta_d| < \frac{\pi}{2}$. It is also assumed that $P_d(t)$ is dynamically feasible, which means that it is achievable given the constraints of the robot dynamics and control inputs. To solve this tracking problem, the inertial frame error $e_P(t) = [e_x, e_y, e_z, e_\theta]^T$ is defined as

$$e_P(t) = \begin{bmatrix} x_d - x \\ y_d - y \\ z_d - z \\ \theta_d - \theta \end{bmatrix} \quad (6)$$

and regulated each error to zero. The error vector has four variables to be regulated, while the system has only three control inputs. This is handled by writing the error vector in a form that reduces the number of errors that need to be regulated to 0. According to Fig. 2, the Cartesian errors can be rewritten in the cylindrical coordinate system. This can be done by representing the position error vector (e_x, e_y) in the plane by its magnitude $e_\rho = \sqrt{e_x^2 + e_y^2}$ and angle $\eta = \arctan(e_y, e_x)$ suitably defined to give the correct quadrant. The vector is expressed in the inertial frame A , but attached to the origin of the robot's body-fixed frame $O_{x_b y_b z_b}$. The cylindrical representation of the error vector becomes $e_P^c(t) = [e_\rho, e_\psi, e_z, e_\theta]^T$ where $e_\psi = \eta - \psi$, denotes the difference between the direction of the planar tracking error vector, η , and the yaw angle ψ . When $e_\psi = 0$, the robot will point in the direction of fastest reduction of the planar tracking error. Regulating e_ρ , e_z , and e_θ to zero is equivalent to regulating e_P to zero.

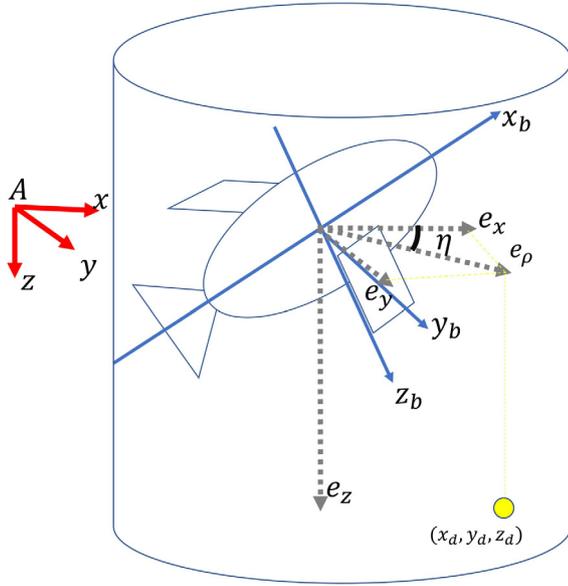


Fig. 2. Illustration of the robot error frame. A is the inertial frame and the point (x_d, y_d, z_d) is the desired position for the robot. The position error vector (e_x, e_y, e_z) is the difference between the desired position and the center of the robot. The axes x_b , y_b , and z_b represent the body-fixed coordinate frame. θ is the pitch angle of the robot.

3. Backstepping-based control design

3.1. Overview of control design

To handle the under-actuated nature of the robot, inspiration is taken from Pettersen and Lefeber's work (Pettersen & Lefeber, 2001). These authors used insight from how a ship helmsman steers a boat to minimize lateral position error, to design a controller for underactuated ships; in particular, minimizing the heading error could be temporarily sacrificed to minimize the position error. Following a similar logic, this work takes advantage of the natural motion of the gliding robotic fish to minimize the planar position error while temporarily sacrificing pitch tracking. To do this, an error function

$$\xi = e_\theta - c f_{\xi_1}(\theta_g) f_{\xi_2}(e_\rho \cos(e_\psi)) \quad (7)$$

is introduced, where $\theta_g = \theta - \alpha$ is the glide angle, α is the angle of attack defined in Section 2, f_{ξ_1} and f_{ξ_2} are bounded odd, increasing functions satisfying $f_{\xi_1}(0) = 0$, $f_{\xi_2}(0) = 0$, and $c > 0$ is a constant satisfying $c \bar{b}_1 \bar{b}_2 < \pi/2$, where \bar{b}_1 and \bar{b}_2 are the upper bounds of $|f_{\xi_1}|$ and $|f_{\xi_2}|$, respectively. These conditions are satisfied by the choice

$$\xi = e_\theta - c \tanh(\theta_g) \tanh(e_\rho \cos(e_\psi)) \quad (8)$$

with $c < \pi/2$. Eq. (8) is adopted in the simulation and experiments in this work. With the error function ξ , the modified tracking error vector $e_{p_a} = [e_z, e_\psi, \xi]^T$, is defined and will be used in the backstepping control design. The derivative of the error vector e_{p_a} can be expressed in terms of the state variables. In particular, with $e_x = e_\rho \cos \eta$ and $e_y = e_\rho \sin \eta$, one can derive

$$\begin{cases} \dot{e}_\rho = \cos(\eta)\dot{e}_x + \sin(\eta)\dot{e}_y \\ \dot{\eta} = \frac{1}{e_\rho}(\cos(\eta)\dot{e}_y - \sin(\eta)\dot{e}_x) \end{cases} \quad (9)$$

which will be useful in computing $\dot{\xi}$ and $\dot{\psi}_e$ later.

With the above formulation, trajectory tracking becomes a stabilization problem with respect to the error vector. The control objective is now to drive the modified error vector e_{p_a} to the origin. Later it is discussed how the convergence of e_{p_a} to zero implies the convergence of all elements of the original tracking error vector e_p to a neighborhood

of zero. To drive e_{p_a} to zero, the variables $\zeta_1 = \dot{e}_z$, $\zeta_2 = \dot{e}_\psi$, and $\zeta_3 = \dot{\xi}$ are defined. The physical control inputs appear in the derivatives of ζ_i . They need to be chosen to render $(e_z, e_\psi, \xi, \eta_1, \eta_2, \eta_3)$ convergent to zero. This can be done by making $\dot{\zeta}_1 = -k_1\zeta_1 - k_z e_z$, $\dot{\zeta}_2 = -k_2\zeta_2 - k_\psi e_\psi$, and $\dot{\zeta}_3 = -k_3\zeta_3 - k_\xi \xi$, where $k_z, k_\psi, k_\xi, k_1, k_2$, and k_3 are positive constants to be chosen. The system can be rewritten in a block diagonal form:

$$\begin{bmatrix} \dot{e}_z \\ \dot{\zeta}_1 \\ \dot{e}_\psi \\ \dot{\zeta}_2 \\ \dot{\xi} \\ \dot{\zeta}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -k_z & -k_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -k_\psi & -k_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -k_\xi & k_3 \end{bmatrix} \begin{bmatrix} e_z \\ \zeta_1 \\ e_\psi \\ \zeta_2 \\ \xi \\ \zeta_3 \end{bmatrix} \quad (10)$$

It can be easily shown that the eigenvalues of the linear system (10)

are pairs of the form $-\frac{k_i}{2} \pm \frac{\sqrt{k_i^2 - 4k_a}}{2}$ for $i = 1, 2, 3$ and $a = z, \psi, \xi$. These all have negative real parts as long as the gains $k_1, k_2, k_3, k_z, k_\psi, k_\xi$ are positive, implying that the state $(e_z, e_\psi, \xi, \zeta_1, \zeta_2, \zeta_3)$ is asymptotically stable.

The above analysis enables one to choose inputs u_1, u_2 , and u_3 to ensure the convergence of $(e_z, e_\psi, \xi, \zeta_1, \zeta_2, \zeta_3)$ to zero. The equations $\dot{\zeta}_i$ can be rewritten as

$$\begin{aligned} \dot{\zeta}_1 &= f_{11}u_1 + f_{12}u_2 + f_{13}u_3 + f_{14}u_3^2 + f_{15} = -k_1\zeta_1 - k_z e_z \\ \dot{\zeta}_2 &= f_{21}u_1 + f_{22}u_2 + f_{23}u_3 + f_{24}u_3^2 + f_{25} = -k_2\zeta_2 - k_\psi e_\psi \\ \dot{\zeta}_3 &= f_{31}u_1 + f_{32}u_2 + f_{33}u_3 + f_{34}u_3^2 + f_{35} = -k_3\zeta_3 - k_\xi \xi \end{aligned}$$

where $f_{ij} = \frac{\partial \dot{\zeta}_i}{\partial u_j}$ for $i = 1, 2, 3, j = 1, \dots, 4$ and $f_{i5} = \dot{\zeta}_i - \sum_{j=1}^4 (u_j f_{ij})$ with $u_4 = u_3^2$. These equations give a means to solve for the inputs such that the desired values of ζ_i are achieved. The equations can be written in a matrix form to solve for the control inputs as follows (where f_{12}, f_{31} , and f_{34} are zero):

$$\begin{bmatrix} f_{11} & 0 & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ 0 & f_{32} & f_{33} & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_3^2 \end{bmatrix} = \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \end{bmatrix} \quad (11)$$

where $\Gamma_1 = -f_{15} - k_1\zeta_1 - k_z e_z$, $\Gamma_2 = -f_{25} - k_2\zeta_2 - k_\psi e_\psi$, and $\Gamma_3 = -f_{35} - k_3\zeta_3 - k_\xi \xi$. An approach to solving (11) for the control is further discussed in Section 4.

3.2. Analysis of the closed-loop system

Under the control law (11), one can guarantee that ξ, e_ψ , and e_z all approach zero. Note that the original tracking goal is for all errors of e_p in Eq. (6) to approach zero. The error system after implementing the control law is given by

$$\begin{bmatrix} \dot{e}_z \\ \dot{e}_\psi \\ \dot{\xi} \\ \dot{e}_\rho \\ \dot{\zeta}_1 \\ \dot{\zeta}_2 \\ \dot{\zeta}_3 \end{bmatrix} = \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \cos(\eta)\dot{e}_x + \sin(\eta)\dot{e}_y \\ -k_1\zeta_1 - k_z e_z \\ -k_2\zeta_2 - k_\psi e_\psi \\ -k_3\zeta_3 - k_\xi \xi \end{bmatrix} \quad (12)$$

with $e_\theta = \xi + c f_{\xi_1}(\theta_g) f_{\xi_2}(e_\rho \cos(e_\psi))$.

An intuition for why $e_{p_a} = 0$ implies that e_p approaches 0 can be gleaned from a geometric perspective. Similar to flight kinematics, glider kinematics can be expressed in terms of the velocity magnitude and a glide angle θ_g (Graver & Leonard, 2001; Leonard & Graver, 2001; Zhang, 2014). The glide angle is defined as $\theta_g = \theta - \alpha$ and can be approximated by $\frac{\Delta z}{\Delta d}$ for a constant glide angle, where Δd and Δz are the horizontal and vertical distances traveled in a given amount of time, respectively. Using this approximation as a basis, Fig. 3 gives an intuitive understanding of what the controller is designed to accomplish. It illustrates idealized scenarios where a robot modifies its glide angle to minimize the horizontal tracking error. For instance,

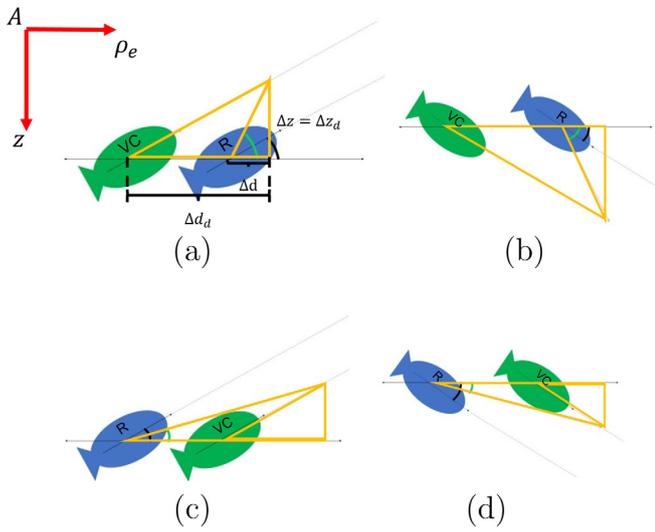


Fig. 3. Illustration of the desired behavior for the robot (R), when tracking trajectory is given by a virtual copy (VC) gliding in a plane for four different cases. Black angle marker represents $\theta = \theta_d$ and green angle marker represents $\theta = \theta_d + c f_{\xi_1}(\theta_g) f_{\xi_2}(e_\rho \cos(e_\psi))$. Δz and Δd represent the vertical travel and horizontal travel, respectively, of the robot when $\theta = \theta_d + c f_{\xi_1}(\theta_g) f_{\xi_2}(e_\rho \cos(e_\psi))$, while Δz_d and Δd_d represent the vertical travel and horizontal travel, respectively, of the virtual copy.

Fig. 3(a) shows the gliding robotic fish ahead of the desired position and at the desired pitch angle. If it maintains this pitch and depth rate, the distance to the desired position will remain constant. If it increases the pitch while maintaining the depth rate, the distance to the desired position decreases. In practice, α is often small compared to θ , making $\theta_g \approx \theta$. This means that, in essence, perturbing the pitch angle effectively changes the glide path, which slows or speeds up horizontal travel (for a given vertical travel speed), thus enabling the robot to catch up to the desired trajectory in the planar position.

Now, multi-time-scale analysis of singularly perturbed systems is used to prove this rigorously and show how the other error states converge to a neighborhood of the origin. For the ease of presentation, it is assumed $\phi = 0$ (in practice the roll ϕ is close to zero). Multi-time-scale analysis of singularly perturbed systems (Esteban, Gordillo, & Aracil, 2013; Khalil, 2002; Kokotovic, Khali, & O'reilly, 1999) is used as a tool for model reduction through a small parameter ϵ . It separates the model of a system into multiple time scales, allowing for analysis of the complete system to be broken down into analysis of reduced models and their interconnections. A brief overview of two-time-scale analysis is given in Appendix. Following the time-scale analysis framework, the system in Eq. (12) can be rewritten as a two-time-scale system as elaborated next.

A natural separation of the time-scales is between $X_1 = e_\rho$ (and e_θ) and the states controlled with the backstepping design in Section 3.1, $X_2 = [e_z, e_\psi, \xi, \zeta_1, \zeta_2, \zeta_3]^T$. For the analysis below, e_θ is not included separately due to its algebraic relationship with ξ , e_ψ , and e_ρ .

One can show that as the gains $k_1, k_2, k_3, k_z, k_\psi, k_\xi$ in (12) get larger, the dynamics for X_2 gets faster. To see this, consider gains of the following relationships (inspired by the form of the eigenvalues from the subsystems in (10)): $k_z = c_1 k_1^2$, $k_\psi = c_2 k_2^2$, and $k_\xi = c_3 k_3^2$, for some $c_1 > 0$, $c_2 > 0$, $c_3 > 0$. To scale the gains, let $k_i = \frac{k_i^0}{\epsilon}$, for a given nominal value $k_i^0 > 0$, $i = 1, 2, 3$. The dynamics (12) can then be represented in the following two-time-scale system:

$$\dot{X}_1 = f(t, X_1, X_2) \triangleq \cos(\eta) \dot{e}_x + \sin(\eta) \dot{e}_y \quad (13)$$

$$\epsilon \dot{X}_2 = g(X_2, \epsilon) \triangleq \begin{bmatrix} \epsilon \zeta_1 \\ \epsilon \zeta_2 \\ \epsilon \zeta_3 \\ -k_1^0 \zeta_1 - \frac{c_1 (k_1^0)^2}{\epsilon} e_z \\ -k_2^0 \zeta_2 - \frac{c_2 (k_2^0)^2}{\epsilon} e_\psi \\ -k_3^0 \zeta_3 - \frac{c_3 (k_3^0)^2}{\epsilon} \xi \end{bmatrix} \quad (14)$$

It can be shown that the eigenvalues of the dynamics for X_2 are $\frac{(-1 \pm \sqrt{1-4c_i})k_i^0}{2\epsilon}$ for $i = 1, 2, 3$. Therefore, the rate of dynamics of X_2 will scale with $1/\epsilon$.

Theorem 3.1. Consider the system (13) and (14). With the control law (11), it can be shown that there exists an ϵ^* such that the system (12) is uniformly ultimately bounded within a neighborhood around the origin for all $\epsilon < \epsilon^*$, and $e_{p_d}(t) \rightarrow 0$ implies $e_p(t)$ will converge to a bounded region around the origin.

Sketch of Proof. The proof follows Theorem T.1 in Appendix to prove the claims. In particular, Assumptions A.1–A.5 are shown to be satisfied for the two-time-scale-system (13) and (14). Assumption A.1 requires that the origin ($X_1 = 0, X_2 = 0$) is an isolated equilibrium point, and there exist a function $X_2 = h(t, X_1)$ such that $g(h(t, X_1), 0) = 0$, and a class κ function κ_ρ such that $\|h(t, X_1)\| \leq \kappa_\rho(\|X_1\|)$. Assumption A.2 entails finding a Lyapunov function for the reduced system $\dot{X}_1 = f(t, X_1, h(t, X_1))$. To verify A.1 and A.2, note that for any $X_1, X_2 = h(t, X_1) = [0, 0, 0, 0, 0, 0]^T$ is a unique root of $g(X_2, \epsilon) = 0$. The dynamic equation of X_1 is then analyzed under the constraints $\zeta_1 = \zeta_2 = \zeta_3 = 0$, $e_z = 0$, $e_\psi = 0$, and $\xi = 0$, as shown next.

First the identity $v_b = R_{b_v}[V, 0, 0]^T$ is used to express the body-fixed linear velocities in terms of the velocity magnitude V , angle of attack α , and side-slip angle β . Using this identity, the derivatives of the position can be expressed as $\dot{b}_i = R R_{b_v}[V, 0, 0]^T$, resulting in

$$\dot{b}_i = \begin{pmatrix} V \cos(\beta) \cos(\psi) \cos(\theta - \alpha) - V \sin(\beta) \sin(\psi) \\ V \cos(\beta) \sin(\psi) \cos(\theta - \alpha) + V \sin(\beta) \cos(\psi) \\ -V \sin(\theta - \alpha) \cos(\beta) \end{pmatrix} \quad (15)$$

Per the assumption on the dynamic feasibility of the reference trajectory $P_d(t)$, one can derive the desired pitch angle θ_d , yaw angle ψ_d , angle of attack α_d , side slip angle β_d , and velocity magnitude V_d for a reference robot from $P_d(t)$, with $|\theta_d - \alpha_d| < \pi/2$. In particular, one could choose ψ_d to be consistent with the projection of the desired velocity on the horizontal plane, making $\beta_d = 0$. Expressing \dot{e}_x and \dot{e}_y using Eq. (15) and plugging them into \dot{e}_ρ results in

$$\dot{e}_\rho = \cos(\eta - \psi_d) V_d \cos(\theta_d - \alpha_d) - \cos(e_\psi) V \cos(\beta) \cos(\theta - \alpha) - \sin(e_\psi) V \sin(\beta) \quad (16)$$

From Eq. (15), $\dot{e}_z = -V_d \sin(\theta_d - \alpha_d) - (-V \sin(\theta - \alpha) \cos(\beta))$, implying $V = \frac{\zeta_1 + V_d \sin(\theta_d - \alpha_d)}{\sin(\theta - \alpha) \cos(\beta)}$. Substituting this equation for V into \dot{e}_ρ yields

$$\begin{aligned} \dot{e}_\rho &= \cos(\eta - \psi_d) V_d \cos(\theta_d - \alpha_d) \\ &\quad - \cos(e_\psi) \frac{\zeta_1 + V_d \sin(\theta_d - \alpha_d)}{\sin(\theta - \alpha)} \cos(\theta - \alpha) \\ &\quad - \sin(e_\psi) \frac{\zeta_1 + V_d \sin(\theta_d - \alpha_d)}{\sin(\theta - \alpha)} \tan(\beta) \end{aligned}$$

The dynamics of e_ρ for the reduced system, with $e_\psi = 0$ and $\zeta_1 = 0$, can now be written as

$$\dot{e}_\rho = \cos(\psi - \psi_d) V_d \cos(\theta_d - \alpha_d) - \frac{V_d \sin(\theta_d - \alpha_d)}{\sin(\theta - \alpha)} \cos(\theta - \alpha)$$

which can be expressed as

$$\begin{aligned} \dot{e}_\rho &= -V_d \left(\frac{\sin(e_\theta - \alpha_e)}{\sin(\theta - \alpha)} \right) \\ &\quad + V_d (\cos(\psi - \psi_d) - 1) \cos(\theta_d - \alpha_d) \end{aligned} \quad (17)$$

with $\alpha_e = \alpha_d - \alpha$. The second term in Eq. (17) is bounded between $-2V_d$ and 0. One can add and subtract $V_d \sin e_\theta$ to the numerator of the first term in Eq. (17). Then, using the constraint $e_\theta = c f_{\xi_1}(\theta_g) f_{\xi_2}(e_\rho)$ when $\xi = 0$, it can be shown that

$$\dot{e}_\rho = -V_d \left(\frac{\sin(c f_{\xi_1}(\theta_g) f_{\xi_2}(e_\rho))}{\sin(\theta_g)} \right) + \eta_0(t) + V_d(\cos(\psi - \psi_d) - 1) \cos(\theta_d - \alpha_d) \quad (18)$$

where $\eta_0(t) = -V_d \frac{2 \cos(\frac{2e_\theta - \alpha_e}{2}) \sin(\frac{\alpha_e}{2})}{\sin(\theta_g)}$ is considered a perturbation. The case where the perturbation $\eta_0 = 0$ (the case $\eta_0 \neq 0$ will be dealt with afterwards) can now be analyzed. It can be seen that the sign of the first term in Eq. (18) is negative for a nonzero pitch angle unless $e_\rho = 0$, in which case the term becomes 0 since f_{ξ_1} and f_{ξ_2} are bounded, odd functions, and $|c f_{\xi_1}(\cdot) f_{\xi_2}(\cdot)| < \frac{\pi}{2}$. In the case $\theta_g = 0$, L'Hopital's rule can be used to show that $\dot{e}_\rho \leq -V_d c f'_{\xi_1}(\theta_g) f_{\xi_2}(e_\rho) \leq 0$, with the last equality holding true only when $e_\rho = 0$, since the derivative of f_{ξ_1} is positive. So the nominal reduced system for X_1 can be shown to satisfy Assumption A.2 with the Lyapunov candidate $V_{ss} = \frac{1}{2} e_\rho^2$ and $\frac{\partial V_{ss}}{\partial X_1} f(t, X_1, h(t, X_1)) \leq -\gamma_0 a_1^2(\|X_1\|)$ for a positive constant γ_0 , where $a_1(\cdot)$ is a class κ function. The existence of the Lyapunov function also implies that $e_\rho = 0$ is a stable equilibrium point of f satisfying Assumption A.1.

Assumption A.3 requires the existence of a Lyapunov function for the perturbation $X_2 - h(t, X_1)$. Because $h(t, X_1)$ is a zero vector, this is satisfied by the Lyapunov function $V_A = \frac{1}{2} X_2^T X_2$ and analysis from the backstepping design in Section 3.1.

Assumption A.4 requires the growth of the difference between the system model $f(t, X_1, X_2)$ and the reduced system model $f(t, X_1, h(t, X_1))$ to be bounded. Using Eq. (16), this can be satisfied with $\frac{\partial V_{ss}}{\partial X_1} [f(t, X_1, X_2) - f(t, X_1, h(t, X_1))] = -e_\rho (\cos(e_\psi) - 1) V \cos(\theta - \alpha) \cos(\beta) - e_\rho \sin(e_\psi) V \sin(\beta) \leq \gamma_1 a_1(\|X_1\|) \|X_2\|$, for a sufficiently large constant γ_1 .

Assumption A.5 has to do with bounding the growth of the perturbation $g(X_2, e) - g(h(t, X_1), 0)$ and $\frac{\partial V_A}{\partial t} + \frac{\partial V_A}{\partial X_1} f(t, X_1, X_2)$. Since $\frac{\partial V_A}{\partial X_1} f(t, X_1, X_2) = 0$ and V_A is independent of t , the second part of Assumption A.5 is satisfied. The first part can be satisfied using $\frac{\partial V_A}{\partial X_2} [g(X_2, e) - g(h(t, X_1), 0)] = X_2^T e \dot{X}_2 \leq \epsilon \gamma_3 \|X_2\|^2$ for any non-negative constant γ_3 since $X_2^T \dot{X}_2$ is rendered negative definite by the controller (11).

Since conditions A.1–A.5 are satisfied, a Lyapunov function for the system (13), and (14) can be constructed as

$$v(X_1, X_2) = (1 - d)V_{ss}(X_1) + dV_A(X_2)$$

where $0 < d < 1$ and there exists an ϵ^* such that for all $\epsilon \in (0, \epsilon^*]$, the equilibrium $X_1 = 0, X_2 = 0$ is uniformly asymptotically stable.

To fulfill Assumptions A.1 and A.2 the previous analysis relied on η_0 being 0. To handle the case $\eta_0 \neq 0$, one can use the theory for nonvanishing perturbations presented in Khalil (2002) (Lemma 9.3). It can then be concluded that, when $\eta_0 \neq 0$, the system (13) is ultimately bounded in a neighborhood around the origin. In particular, consider again the candidate Lyapunov function $V_{ss}(X_1) = \frac{1}{2} e_\rho^2$, but for the perturbed system. The derivative $\dot{V}_{ss}(X_1)$ along the trajectory of $\dot{X}_1 = f(t, X_1, h(t, X_1))$ now satisfies

$$\begin{aligned} \dot{V}_{ss}(X_1) &\leq -a_1^2(\|X_1\|) + \left\| \frac{\partial V_{ss}}{\partial X_1} \right\| \|\eta_0\| \\ &= -a_1^2(\|X_1\|) + \|X_1\| \|\eta_0\| \end{aligned}$$

Suppose η_0 satisfies the bound $\|\eta_0\| \leq \Delta$ for all $t \geq t_0, X_1 \in D = \{X_1 \in \mathbb{R}^1 \mid \|X_1\| < r\}$ for some $r > 0$. It can then be shown that $\dot{V}_{ss} < 0$ whenever $\|X_1\| > a_1^{-1}(\sqrt{r}\Delta)$. In other words, $\|X_1\|$ is ultimately bounded by $a_1^{-1}(\sqrt{r}\Delta)$. ■

4. Simulations

The backstepping controller proposed in this paper is compared against two baseline controllers operating on the errors e_z, e_ψ , and e_θ to show its effectiveness. The first is a PID controller and the second is

Table 1
Miniglider model parameters.

Param	Value	Param	Value
m_{0_e}	0.446	r_{p1_e}	0.445
m_0	-0.051 kg	r_{p1}	-0.1 m
m_1	15.011 kg	S	0.013 m ²
m_2	6.077 kg	C_D^α	39.50 rad ⁻²
m_3	8.291 kg	$C_{M_p}^\alpha$	0.279 m/rad
J_1	0.801 kg-m ²	$C_{L_l}^\alpha$	24.66 rad ⁻¹
J_2	0.076 kg-m ²	$C_{F_l}^\beta$	-4.650 rad ⁻¹
J_3	1.60 kg-m ²	$C_{F_l}^\delta$	-3.529 rad ⁻¹
\bar{m}	0.287 kg	C_{L0}	0.588
g	9.82 m/s ²	C_{D0}	1.985
C_{MR}^β	0.631 m/rad	K_{q1}	-11.97 m-s/rad
m_{w3}	0.819 kg	K_{q2}	-14.96 m-s/rad
r_{w3}	0.011 m	K_{q3}	-12.1 m-s/rad
C_{MY}^β	14.0 m/rad	C_{MY}^δ	-0.210 m/rad
ρ	992.2 kg/m ³	C_D^δ	4.694 rad ⁻²
C_{M_0}	0.321 m		

another backstepping controller (which uses e_θ instead of the modified error ξ). Note that the baseline backstepping design is equivalent to the proposed design with $c = 0$, so the analysis from Section 3.1 applies and guarantees the convergence of the three aforementioned errors.

4.1. Simulation setup

The simulation is carried out using MathWorks Simulink and the model parameters (see Table 1) used for simulation are based on those estimated for the physical system to be described in Section 5. Actuation is limited to the range of $[-27.9, 22.5]$ g for m_0 , $[-1.92, 1.92]$ radians for δ , and $[-55.5, 44.5]$ mm for r_{p1} . Limits are also placed on the actuation rates for m_0, r_{p1} , and δ with $|\dot{m}_0| < 1.72 \frac{\text{g}}{\text{s}}, |\dot{r}_{p1}| < 12.5 \frac{\text{mm}}{\text{s}}$, and $|\dot{\delta}| < 1.05 \frac{\text{rad}}{\text{s}}$.

To solve for the control (u_1, u_2, u_3) for the proposed method, Eq. (11) can be algebraically manipulated to produce a quadratic equation for u_3 , with u_1 and u_2 expressed as quadratic and linear, respectively, in u_3 . In simulation and experiments in this work, control computation is further simplified by making mild assumptions of $\phi = 0$ and $\dot{\eta} = 0$, both of which are reasonable given that ϕ and $\dot{\eta}$ are close to zero under typical operating conditions of a gliding robotic fish. These assumptions are not a necessity, but do simplify computing the controller by removing the quadratic term when solving for the tail angle δ . Otherwise, a quadratic equation needs to be solved, which can possibly give two solutions that one would then need to choose between. In addition, the angle of attack α may not be available for measurement in practice. But given $|\theta| \gg \alpha$ during typical gliding operation, the glide angle $\theta_g = \theta - \alpha$ can be approximated by the pitch angle θ , as adopted in simulation and experiments in this work. With these assumptions, f_{21}, f_{22}, f_{24} , and f_{33} in (11) vanish, resulting in the following expressions for the control:

$$\begin{cases} u_3 = \frac{F_2}{f_{23}} \\ u_1 = \frac{1}{f_{11}} (F_1 - f_{13}u_3 - f_{14}(u_3)^2) \\ u_2 = \frac{F_3}{f_{32}} \end{cases} \quad (19)$$

In simulation, the tracking error e_ψ is redefined as $\psi - \frac{1}{2}(\eta^+ + \eta^-)$ if $e_\rho < \epsilon_0$ for some small $\epsilon_0 > 0$ and $\psi - \eta$ otherwise, where $\eta^+ = \arctan(e_y^+, e_x^+)$ and $\eta^- = \arctan(e_y^-, e_x^-)$ with $e_x^\pm = e_x + l \cos(\psi \pm \frac{\pi}{2})$ and $e_y^\pm = e_y + l \sin(\psi \pm \frac{\pi}{2})$, for some small $l > 0$. This allows the tracking error to be defined at the point of singularity when $e_\rho = 0$. ϵ and l are taken to be 0.05 and 0.15, respectively, in simulation.

The PID control consists of a set of three controllers. The error e_ψ is used to calculate δ with gains $k_p = 1, k_i = 0.001$ and $k_d = 1$. The error e_θ is used to calculate r_{p1} with gains $k_p = 1, k_i = 0$ and $k_d = 2$. The error e_z is used to calculate m_0 with gains $k_p = 1, k_i = 0$ and $k_d = 10$.

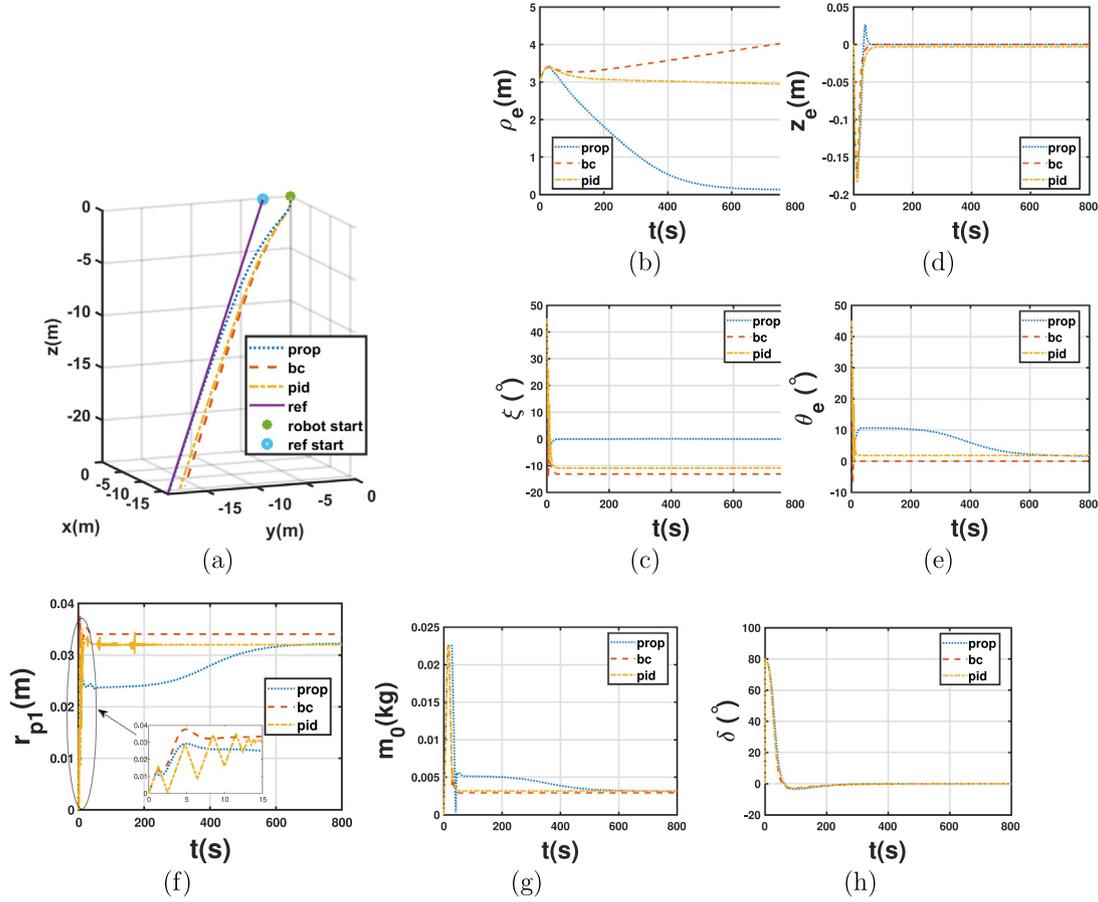


Fig. 4. Simulation results with a linear gliding reference trajectory. The legends “prop”, “bc” and “pid” indicate results from the proposed backstepping controller, the baseline backstepping controller, and the PID controller, respectively. (a): Reference and controlled trajectories in the 3D space; (b)–(e): the trajectories of tracking errors (e_ρ , e_z , e_θ , ξ); (f)–(h): the trajectories of the control inputs (r_{p1} , m_0 , δ).

The gains for the PID controller were chosen using the Matlab PID gain tuner and then manually tuned to refine performance on one of the reference trajectories. The gains for both backstepping controllers are $k_z = 1$, $k_\xi = 1$, $k_\psi = 1$, $k_1 = 10$, $k_2 = 1$, and $k_3 = 2$. These satisfy the conditions from the design in Section 3.1. For the proposed controller, c is chosen as $\frac{\pi}{9}$ for ξ . The parameters for all three controllers are kept the same over all trajectories.

Two reference trajectories are used, including a linear gliding pattern with a constant pitch angle and a constant depth rate and a sawtooth-like gliding pattern. The desired trajectories are parameterized as time-parameterized vector paths $[x_d(t), y_d(t), z_d(t), \theta_d(t)]$. It is worth noting that the results only show the actual values of the control inputs (after rate and magnitude saturation) as opposed to the values computed by the controllers. However, they coincide except for during small segments of the trajectories.

4.2. Simulation results

Table 2 shows a summary of root-mean-squared tracking errors for both reference trajectories. Fig. 4 shows the simulation results for tracking the first reference parameterized as

$$P_d(t) = \begin{bmatrix} 3 + 0.02t \\ 1.5 + 0.02t \\ 0.03t \\ -\frac{\pi}{4} \end{bmatrix}$$

In both this section and next section the units for the first three components of P_d are m and the units for the last component of P_d is radians. This trajectory particularly highlights the effect of the

Table 2

Summary of RMS tracking errors (in all cases angles in radians) for simulations.

Linear glide reference trajectory			
	Proposed	Baseline BS	PID
e_z	0.0510	0.0450	0.0630
e_ρ	2.0920	3.4910	3.1110
e_θ	0.2500	0.2130	0.2210
ξ	0.2130	0.3010	0.2720
Sawtooth-like reference trajectory			
	Proposed	Baseline BS	PID
e_z	0.5280	0.4350	0.3690
e_ρ	0.9690	1.3870	1.3640
e_θ	0.0920	0.0190	0.0180
ξ	0.0260	0.1230	0.1210

proposed controller. Here, all three controllers have similar responses in δ which orients the robot so that it faces the desired horizontal plane position. For r_{p1} , the PID controller activates the rate constraint and is much more aggressive in the initial transient than the backstepping controllers, while the proposed controller is slightly more aggressive in the control of m_0 . The PID controller and the baseline backstepping controller quickly track the pitch angle and the depth, but the resulting trajectories never converge to the desired horizontal position. The PID controller has a slight error in the pitch tracking allowing it to slowly decrease e_ρ and the baseline backstepping controller actually results in e_ρ increasing over time. On the other hand, the proposed backstepping controller, quickly tracks the depth with a bit of overshoot. Instead of tracking e_θ , it tracks ξ , which temporarily sacrifices perfect pitch

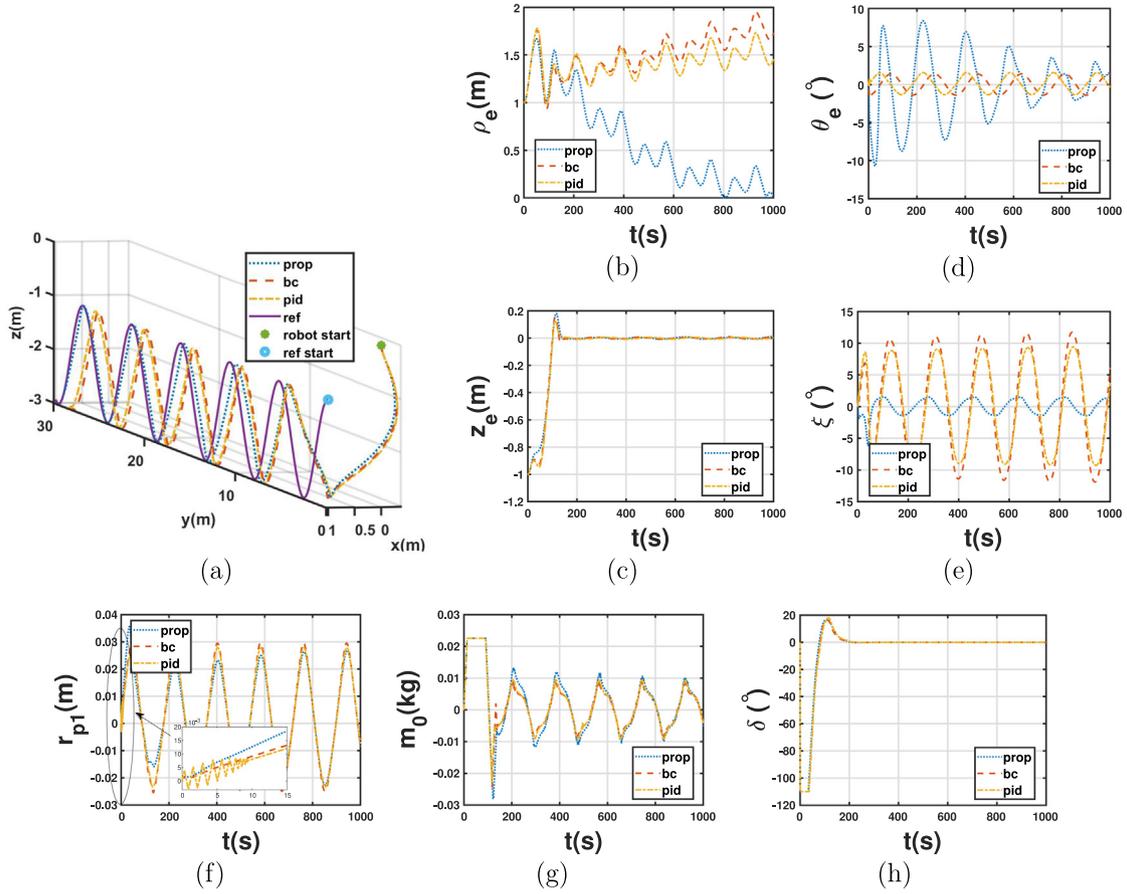


Fig. 5. Simulation results with a sawtooth-like reference trajectory constrained to a vertical plane. The legends “prop”, “bc” and “pid” indicate results from the proposed backstepping controller, the baseline backstepping controller, and the PID controller respectively. (a): Reference and controlled trajectories in the 3D space; (b)–(e): the trajectories of tracking errors ($e_\rho, e_z, e_\theta, \xi$); (f)–(h): the trajectories of the control inputs (r_{p1}, m_0, δ).

tracking in order to achieve the desired planar position. The maximum deviation of ξ is dependent on c and the current value of θ . The difference in orientation increases the value of m_0 required to achieve the depth rate and results in e_ρ decreasing much more rapidly than for the PID controller and the baseline backstepping controller. As e_ρ converges to 0, e_θ converges to 0 and m_0 and r_{p1} converge to values similar to their counterparts from the PID control and baseline backstepping control.

Fig. 5 shows the simulation results for the case of the sawtooth-like reference trajectory parameterized as

$$P_d(t) = \begin{bmatrix} 0.03t \\ 1 \\ 2 - \cos(\frac{\pi}{90}t) \\ -\frac{2\pi}{9} \sin(\frac{\pi}{90}t) \end{bmatrix}$$

It can be seen that, under the proposed controller, the 3D position tracking error (as reflected by e_ρ and e_z) converges to a small neighborhood of zero, and the (oscillating) pitch tracking error shows a consistently decreasing amplitude. On the other hand, while the oscillating e_θ values under the PID controller and the baseline backstepping controller gain smaller amplitudes quicker than the proposed controller, e_ρ values under both show an increasing trend and never converge towards zero.

5. Experiments

5.1. Experimental setup

Experiments are carried out with a miniaturized gliding robotic fish, Miniglider, in a 4.6 m long, 3.1 m wide, and 1.2 m deep indoor



Fig. 6. Miniglider robot operating in a large indoor tank during experiment. Onboard sensors are used to estimate body-fixed velocities, while AprilTags and onboard camera are used to localize the robot.

tank equipped with 42 laser-cut 15 cm by 15 cm AprilTags (Krogius, Haggemiller, & Olson, 2019; Olson, 2011). The experimental setup is pictured in Fig. 6. The AprilTags were made from Trotech $\frac{1}{16}$ inch Laserable Plastics and used to localize the Miniglider robot by an onboard camera.

A breakdown of the internal structure and actuation system is shown in Fig. 7. The robot actuation system consists of two linear actuators (Actuonix P16-P and L16-P, both with a stroke of 10 cm) with position feedback, a 60 ml syringe, a sliding mass, and a waterproof servo (Hitec HS-646WP). The servo controls the tail angle δ , one linear actuator controls the position $r_{p1} = (\mu - r_{p1c})r_{p1s}$ of the slide mass, and the other linear actuator along with the syringe controls the net buoyancy $m_0 = (\mu - m_{0c})m_{0s}$ where $\mu \in [0, 1]$ is the normalized linear actuator position for the corresponding case. The subscripts c and s represent the actuator positions corresponding to the base setting ($r_{p1} = 0, m_0 = 0$), and scaling factors, respectively. The body of the Miniglider is constructed from

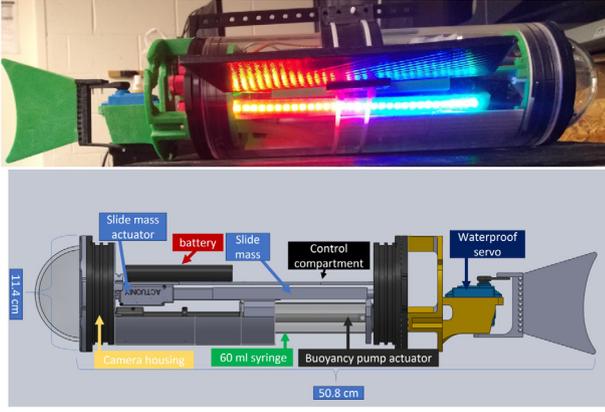


Fig. 7. Pictures of actual (top) and SolidWorks design (bottom) revealing internal mechanical structure.

a BlueRobotics 4-inch series enclosure. It also features a 3D-printed wing mount, a set of wings, a servo mount, and a tail, all with a water-resistant coating.

The power electronics consists of a Tenergy 9.6 V NiMH battery directly driving the linear actuators through an STMicroelectronics L298 motor driver and a custom PCB with 5 V and 7 V switching regulator circuits. The control electronics consists of a Raspberry Pi 4 as the main computer, an RF Xbee module for communication, and a custom Raspberry Pi shield containing the motor driver, a Microchip EMC1701 battery monitor, a DsPic30F6014 A microcontroller used to control the actuators and read analog sensors, and interfaces for various sensors. Onboard sensors include an LSM9DS1 IMU, a BlueRobotics Bar30 pressure sensor, and a Raspberry Pi Camera V1. In addition to the orientation and depth measurements from the onboard sensors, the position and the orientation of the Miniglider are also estimated based on the AprilTags.

5.2. Parameter estimation

The model parameters of the Miniglider are estimated through a combination of computational fluid dynamics (CFD) simulations and particle swarm optimization (PSO) (Poli, Kennedy, & Blackwell, 2007). SolidWorks 2020 is used to create a model of the Miniglider and the ANSYS 2020 Fluid Flow (Fluent) work flow is used to create a mesh file for simulations of a static Miniglider in a water tunnel. From the hydrodynamic forces and moments computed in the simulation, the hydrodynamic parameters of C_D , C_{F_s} , C_L , C_{M_1} , C_{M_2} , and C_{M_3} can be identified based on (see (4))

$$\begin{cases} C_D = C_{D0} + C_D^\alpha \alpha^2 + C_D^\delta \delta^2 \\ C_{F_s} = C_{F_s}^\beta \beta + C_{F_s}^\delta \delta \\ C_L = C_{L0} + C_L^\alpha \alpha \\ C_{M_1} = C_{M_R}^\beta \beta + K_{q1} \omega_1 \\ C_{M_2} = C_{M_0} + C_{M_P}^\alpha \alpha + K_{q2} \omega_2 \\ C_{M_3} = C_{M_Y}^\beta \beta + K_{q3} \omega_3 + C_{M_Y}^\delta \delta. \end{cases} \quad (20)$$

In particular, in the CFD simulation, the variables α and β can be changed by setting the direction of the water velocity relative to the Miniglider mesh and δ can be changed by changing the Miniglider mesh to have a specified tail angle. By setting two of the three variables (α , β , and δ) to zero, and varying the third, polynomial curve fitting is used to estimate the model parameters in Eq. (20) except K_{q1} , K_{q2} , and K_{q3} .

PSO is then used with the entire parameter vector taken as the state space of the particles in the PSO algorithm. 100 particles are used with parameter vectors randomly generated from a uniform distribution

across a bounded search space. One of these particles has a subset of the parameters replaced with measurable parameters (such as \bar{m}) and CFD hydrodynamic parameter estimates. The particles are optimized by minimizing the sum of weighted errors between the partial state (position, orientation, angular velocity) data measured from sensors onboard the robot during open-loop trajectories (typical gliding-like and spiral-like trajectories) and their simulated values based on the parameter vectors from the particle swarm. Table 1 contains the best performing estimate of the parameter vector.

5.3. Controller implementation

The proposed controller, PID controller, and backstepping controller are all implemented on the Miniglider's Raspberry Pi 4 for control experiments. The controllers and a model-based observer are implemented in Python 3 along with AprilTag-based localization. AprilTag based-localization uses known AprilTag poses (position and orientation) and the camera-relative AprilTag measurements to produce a pose estimate of the robot in the world frame for each detected AprilTag. A Kalman filter is then used to fuse AprilTag positions, AprilTag yaw angles, and depth from the pressure sensor. It outputs the 3D position, yaw angle, and the derivatives of the aforementioned state variables. When measurements are available, the Kalman filter reports an estimation variance of 5 to 6 cm for the x and y coordinates and a variance of 1 to 3 cm for the z coordinate. When AprilTag measurements are unavailable for an extended period of time, the model-based observer (discussed below) is used to propagate the planar position and the IMU is used to propagate the yaw angle.

The model-based observer is used to produce estimates of the linear body-fixed velocities \hat{v}_b and is driven by the depth estimation error $z - \hat{z}$. It is based on Eq. (2) and formulated as

$$\begin{aligned} \dot{\hat{v}}_b = & M^{-1} (M \hat{v}_b \times \omega_b + m_0 g R^T k + F_{ext}(\hat{v}_b)) \\ & + K_1 R^T \begin{bmatrix} 0 \\ 0 \\ z - \hat{z} \end{bmatrix} \end{aligned}$$

with $\dot{\hat{b}}_i = R \hat{v}_b + K_2 [0, 0, z - \hat{z}]^T$. \hat{b}_i is used to generate \hat{z} for the driving error $z - \hat{z}$ and maintain a planar position estimate when no AprilTags are available for measurement. K_1 and K_2 are diagonal gain matrices. As input, the velocity observer takes the current estimate of the body-fixed linear velocities v_b , the rotation matrix $R(\phi, \theta, \psi)$ (ϕ and θ obtained from the IMU, ψ calculated from the IMU and AprilTag Kalman filter), the body-fixed angular velocity ω_b from the IMU, the control inputs m_0 and δ , the current estimate \hat{z} of the depth, and the depth z from the pressure sensor. The AprilTag position estimate is not used in the model-based observer due to the occasional large localization error and relatively long periods with no available measurements.

The controller uses the same sources as the observer for depth, body-fixed angular velocity, and orientation. It also takes the estimated body-fixed linear velocities \hat{v}_b from the model-based observer. The position feedback is obtained from a combination of the AprilTag Kalman filter and model-based observer as previously explained. The position calculated onboard and pitch from the IMU are used as the ground-truth.

All controllers were operated at roughly 10 Hz. The PID gains were tuned experimentally, and were chosen as $k_p = 0.08$, $k_i = 0$, $k_d = 0.1$ for depth control, $k_p = 0.08$, $k_i = 0.05$, $k_d = 0.0375$ for pitch control, and $k_p = 1$, $k_i = 0.001$, $k_d = 1$ for yaw control. For both the baseline backstepping controller and the proposed controller, the gains are chosen as $k_z = 0.08$, $k_\psi = 1$, $k_\xi = 4$, $k_1 = 0.9$, $k_2 = 0.1$, and $k_3 = 4$ with $c = \frac{\pi}{9}$ for ξ .

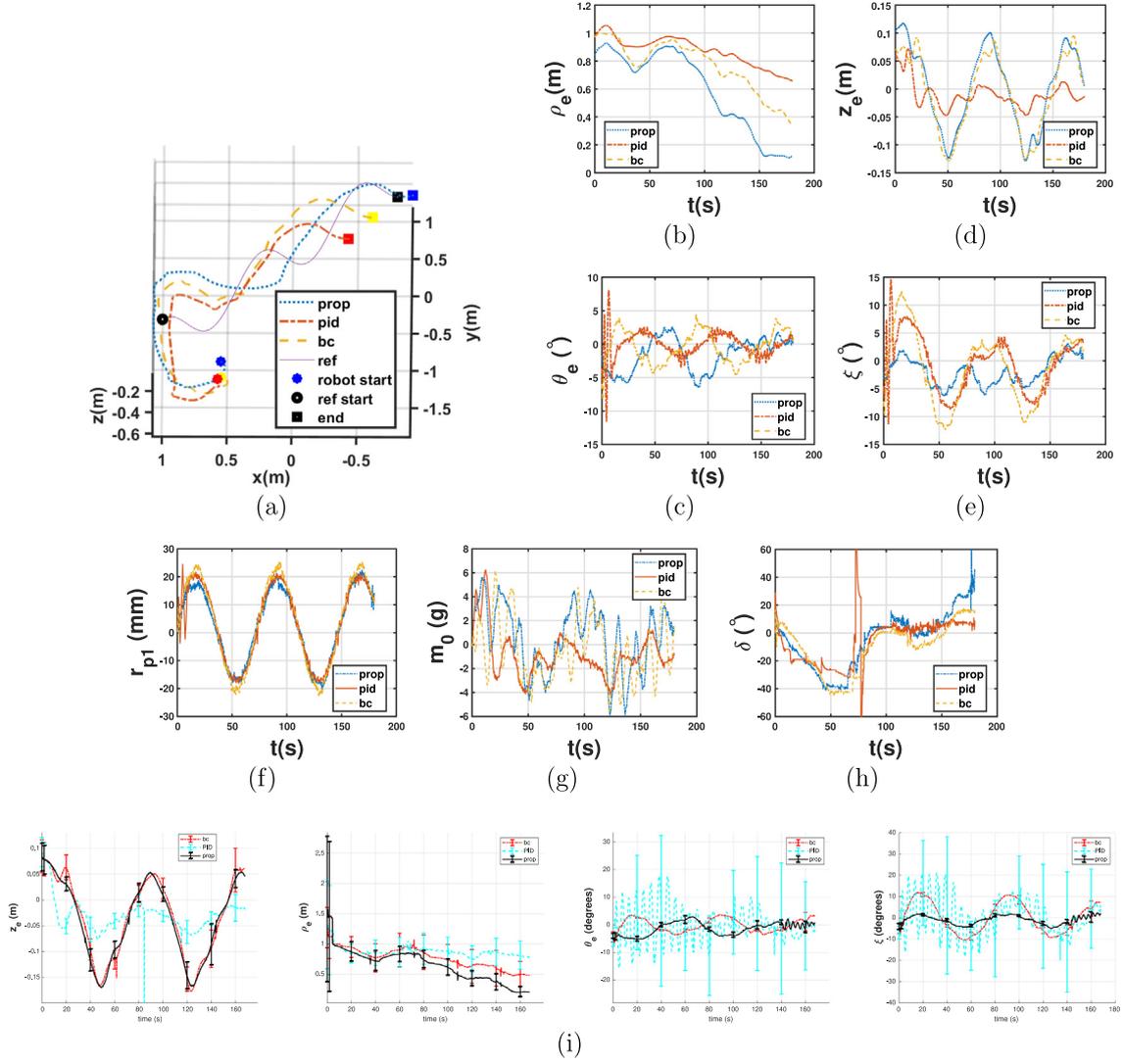


Fig. 8. Experimental results for reference trajectory in Eq. (21). The legends “bc” and “pid” indicate results from the proposed backstepping controller and the PID controller, respectively. (a): Reference and controlled trajectories in the 3D space; (b)–(e): the trajectories of tracking errors ($e_\rho, e_z, e_\theta, \xi$); (f)–(h): the control command (cmd) calculated by the two controllers as well as the trajectories of the achieved control inputs (r_{p1}, m_0, δ). (i): statistics of subset of the errors.

5.4. Experimental results

The controllers are tested on two reference trajectories; the first parameterized as

$$P_d(t) = \begin{bmatrix} -1 + 0.012t \\ -1 + 0.01t \\ 0.35 - 0.2 \cos(\frac{2\pi}{75}t) \\ -\frac{7\pi}{36} \sin(\frac{2\pi}{75}t) \end{bmatrix} \quad (21)$$

and the second parameterized as

$$P_d(t) = \begin{bmatrix} -1.5 \cos(\frac{\pi}{270}t + \frac{\pi}{4}) \\ -1.2 \cos(\frac{\pi}{270}t + \frac{\pi}{4}) \sin(\frac{\pi}{270}t + \frac{\pi}{4}) \\ 0.35 - 0.15 \cos(\frac{2\pi}{75}t) \\ -\frac{7\pi}{36} \sin(\frac{2\pi}{75}t) \end{bmatrix} \quad (22)$$

For each trajectory, ten trials are run for each controller Fig. 8 and Fig. 9 show the results of a single trial and statistics across all trials for the first and second trajectory, respectively. Both figures show the path and the reference in 3D, the post-processed errors (e_z, e_ρ, e_θ , and ξ), and the actual achieved values of the control inputs for a single trajectory. The control commands coincide with the actual control values during

most of the respective experiments. In addition, the means of the errors across the 10 trials are shown with error bars depicting standard deviation. A summary of root-mean-squared tracking errors is shown in Table 3. Of the 10 trials for the PID control scheme, the pitch control induced large oscillations 6 times for the first trajectory and 3 times (trials that immediately induced oscillations were discarded) for the second trajectory causing degraded performance in the overall tracking.

Statistically, the PID controller is more precise in tracking the depth, while the baseline backstepping controller and the proposed controller offer very similar depth tracking performance to one another. This is also reflected in the single trials as well. The control m_0 is slightly biased indicating some model inaccuracy which may affect the model-based controllers more than the PID controller, allowing it to perform better at tracking depth. The single trials indicate that all three controllers are able to provide good tracking of their respective reference pitch angles; however, the statistical results show that the PID can induce large oscillations quite often. For many of the trials, the control command for the PID controller was initially more aggressive for r_{p1} . The actuator for r_{p1} was unable to keep up with the commanded value, which was likely the cause of oscillations in pitch tracking. An example of this can be seen in Fig. 10. Both the single trials and the statistical results indicate that the propose controller is superior in tracking the

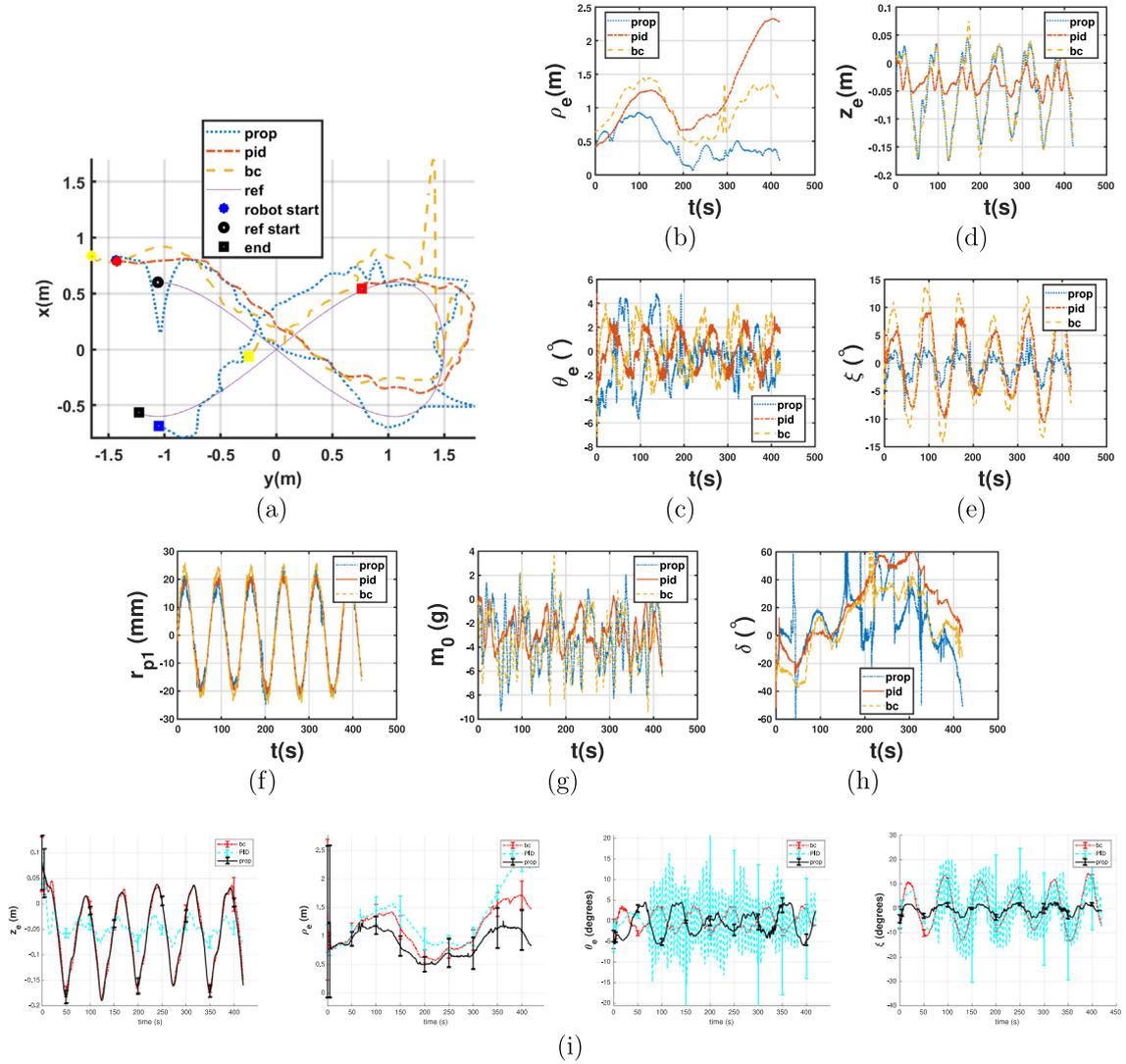


Fig. 9. Experimental results for reference trajectory in Eq. (22). The legends “bc” and “pid” indicate results from the proposed backstepping controller and the PID controller, respectively. (a): Reference and controlled trajectories in the 3D space; (b)–(e): the trajectories of tracking errors ($e_\rho, e_z, e_\theta, \xi$); (f)–(h): the control command (cmd) calculated by the two controllers as well as the trajectories of the achieved control inputs (r_{p1}, m_0, δ). (i): statistics of subset of the errors.

horizontal position. The baseline backstepping controller provides a particularly good comparison for the proposed controller because the depth profiles are almost identical suggesting the difference in planar tracking error is largely due to the pitch tracking control scheme.

The results of the velocity observer are shown for a single trial in Fig. 11. The body-fixed velocity estimates are rotated by the orientation to compute the inertial frame velocities. These match well for the depth velocity and the general trend matches for the x and y velocities.

The experimental results show some discrepancies from the simulation results in terms of performance for the controllers. One reason is that the reference trajectories in the simulation are different from those in the experiments. This allows the use of more versatile trajectories in simulation (that are not easily implementable in the tank due to the limited tank size), to better reveal the performance of different controllers. In addition, in simulation the parameters used in the simulated model are the same as the ones used in the controller design (i.e., assuming perfect model knowledge), and the measurements are assumed to be noise-free. Of course, such ideal assumptions do not hold in experiments, which accounts for some discrepancies between simulation and experimental results. Despite these differences, it is noted that the general observations of performance and behavior of each controller are largely consistent between simulation and experiments.

Table 3
Summary of RMS tracking errors (in all cases angles in radians) for experiments.

Reference trajectory in Eq. (21)			
	Proposed	Baseline BS	PID
e_z	0.070	0.070	0.039
e_ρ	0.643	0.768	0.873
e_θ	0.0412	0.03787	0.2631
ξ	0.0349	0.1119	0.3359
Reference trajectory in Eq. (22)			
	Proposed	Baseline BS	PID
e_z	0.071	0.068	0.050
e_ρ	0.884	1.115	1.297
e_θ	0.0496	0.0332	0.1181
ξ	0.0295	0.1237	0.1933

6. Conclusion and future work

In this work, a novel backstepping-based controller is presented for a gliding robotic fish. It is able to track a reference trajectory for 3D position and pitch angle using only three actuation inputs. The introduction of a hybrid error function, combining the pitch tracking

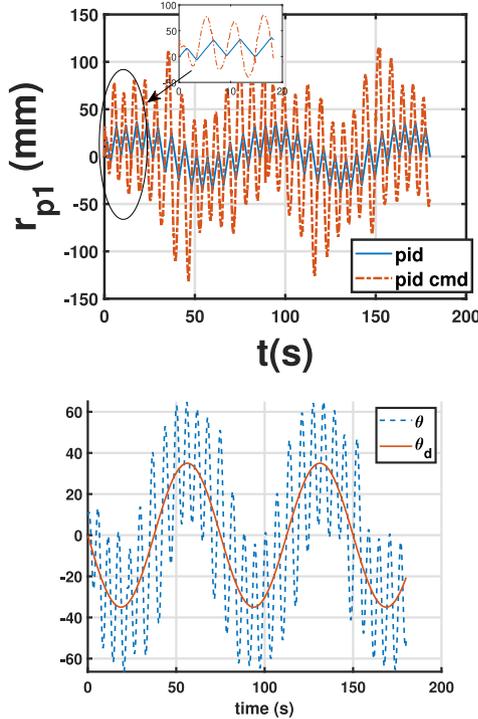


Fig. 10. Example of r_{p1} and θ when PID controller induces oscillations.

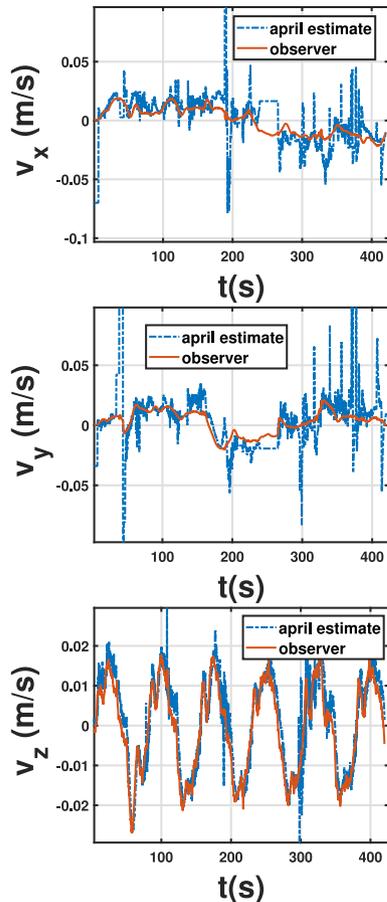


Fig. 11. The body-fixed velocity estimates from the observer converted into the inertial frame and compared with the inertial velocities estimated from the AprilTag measurements.

error with the planar position tracking error, was key to enabling successful tracking. This novel error, the depth tracking error, and the difference between the yaw and the target-point direction (in the horizontal plane) were all shown to be regulated to zero under the proposed backstepping control law. Through time-scale analysis, it was further shown that, with a vanishing hybrid error function, both the pitch error and the magnitude of the planar tracking error converge to a region around zero at a slower time-scale. The proposed controller was then evaluated with both simulation and experiments using a mini-glider robot. These results supported the efficacy of the proposed approach in tracking the 3D position and the pitch angle. Its advantages were further demonstrated via the comparison with two alternative schemes, a PID controller and a baseline backstepping controller not using the hybrid error function.

Future directions for this work include considering the dynamics of the actuators in the controller synthesis, which is expected to more naturally accommodate the actuator constraints. The model should also be expanded to capture the effect of the tail under rapid movement. This may enable design of a controller capable of taking advantage of the energy-efficient gliding-like motion from the buoyancy control as well as rapid maneuvers enable by the tail. Also, since the system parameters can change over time due to changing of parts (such as wings of different designs), it will be of interest to examine adaptive backstepping control schemes. It is also of interest to study output feedback or partial state feedback control theory for the tracking problem. In particular, while a preliminary observer design showed promise in the experimental implementation, establishing a systematic observer framework for underwater gliders remains an open problem. Lastly, the model used in this work assumes an ideal environment, but in field conditions the robot may be exposed to dynamic disturbances from waves and water currents. In that case online model estimation and controller robustification methods are of interest in ensuring robust tracking performance.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Two-time-scale analysis of singularly perturbed systems

A standard model for a two-time-scale system is given by

$$\begin{bmatrix} \dot{x} \\ \epsilon \dot{z} \end{bmatrix} = \begin{bmatrix} f(t, x, z, \epsilon) \\ g(t, x, z, \epsilon) \end{bmatrix}, \quad \begin{matrix} x \in R^n \\ z \in R^m \end{matrix} \quad (A.1)$$

Time-scale analysis requires five conditions to be satisfied for all $(t, x, z, \epsilon) \in [t_0, \infty) \times B_x \times B_z \times [0, \epsilon_1]$ so that a composite Lyapunov function can be constructed to establish the asymptotic stability of the system (A.1), where $B_x \subset R^n$ and $B_z \subset R^m$ are closed sets. This is done by studying the properties of the reduced system $\dot{x} = f(t, x, z, \epsilon)$ evolving on a manifold given by $z = h(t, x)$, the boundary-layer system $\frac{dz}{d\tau} = g(t, x, z, \epsilon)$ with $\tau = \frac{t}{\epsilon}$ and x treated as a fixed parameter, and their interconnections. The assumptions are given as follows (Khalil, 2002; Kokotovic et al., 1999).

Assumption A.1. There exists an isolated equilibrium point at the origin $(x = 0, z = 0)$ for the system (A.1) such that

$$f(t, 0, 0, \epsilon) = g(t, 0, 0, \epsilon) = 0.$$

In addition, for a given $x, z = h(t, x)$ is a unique root of $g(t, x, z, 0)$, such that $g(x, h(t, x), 0) = 0$ and there exists a class κ function ρ such that $\|h(x)\| \leq \rho(\|x\|)$.

Assumption A.2. $x = 0$ is an asymptotically stable equilibrium for the reduced-order system; namely, there exists some Lyapunov function candidate $V(t, x)$ such that

$$0 < q_1(\|x\|) \leq V(t, x) \leq q_2(\|x\|)$$

for some class κ functions q_1 and q_2 , and the following holds:

$$\frac{\partial V(t, x)}{\partial t} + \frac{\partial V(t, x)}{\partial x} f(t, x, h(t, x), 0) \leq -\gamma_1 \psi_1^2(x)$$

where γ_1 is a positive scalar and $\psi_1(x)$ is a continuous scalar function of x that vanishes only when x is 0.

Assumption A.3. There exists a Lyapunov function candidate $W(t, x, z)$ satisfying

$$0 < q_3(\|z - h(t, x)\|) \leq W(t, x, z) \leq q_4(\|z - h(t, x)\|)$$

$$W(t, x, z) > 0, \forall z \neq h(t, x), W(t, x, h(t, x)) = 0$$

for some class κ functions $q_3(\cdot)$ and $q_4(\cdot)$ and

$$\frac{\partial W(t, x, z)}{\partial z} g(t, x, z, 0) \leq -\gamma_2 \psi_2^2(z - h(t, x)), \gamma_2 > 0$$

where γ_2 is a positive constant, $\psi_2(\cdot) > 0$ is a scalar function that vanishes only when its argument is 0, and x is treated as a fixed parameter.

Assumption A.4.

$$\begin{aligned} \frac{\partial V(t, x)}{\partial x} [f(t, x, z, \epsilon) - f(t, x, h(t, x), 0)] \\ \leq \beta_1 \psi_1(x) \psi_2(z - h(t, x)) + \epsilon \alpha_1 \psi_1^2(x) \end{aligned}$$

for some non-negative constants α_1 and β_1 .

Assumption A.5.

$$\begin{aligned} \frac{\partial W(t, x, z)}{\partial z} [g(t, x, z, \epsilon) - g(t, x, h(t, x), 0)] \leq \\ \epsilon \alpha_2 \psi_2^2(z - h(t, x)) + \beta_2 \psi_1(x) \psi_2(z - h(t, x)) \\ \frac{\partial W}{\partial t} + \frac{\partial W}{\partial x} f(t, x, z, \epsilon) \leq \gamma_3 \psi_2^2(z - h(t, x)) \\ + \alpha_3 \psi_1(x) \psi_2(z - h(t, x)) \end{aligned}$$

for non-negative constants α_2 , β_2 , γ_3 , and α_3 .

Conditions A.1–A.3 guarantee asymptotic stability of the reduced and boundary-layer systems. The fourth and fifth conditions handle the interconnection between the reduced model and the boundary-layer system by looking at the derivatives of Lyapunov candidate functions $V(t, x)$ and $W(t, x, z)$, taking $g(t, x, z, \epsilon) - g(t, x, h(t, x), 0)$, $f(t, x, z, \epsilon) - f(t, x, h(t, x), 0)$, and $z - h(t, x)$ as perturbations, and imposing conditions on the growth of those perturbations.

Theorem T.1. Consider the singularly perturbed system (A.1) that satisfies Assumptions A.1–A.5. Then there exists an $\epsilon^* > 0$ such that the equilibrium of (A.1) is asymptotically stable for all $\epsilon < \epsilon^*$. Furthermore, a candidate composite Lyapunov function for the system can be constructed from the weighted sum

$$V_c(t, x, z) = (1 - d)V(t, x) + dW(t, x, z) \quad (\text{A.2})$$

where $0 < d < 1$.

References

- Aguiar, A. P., & Hespanha, J. P. (2003). Position tracking of underactuated vehicles. In *Proceedings of the 2003 American control conference, 2003, vol. 3* (pp. 1988–1993). IEEE.
- Cao, J., Cao, J., Zeng, Z., & Lian, L. (2016). Nonlinear multiple-input-multiple-output adaptive backstepping control of underwater glider systems. *International Journal of Advanced Robotic Systems*, 13(6), Article 1729881416669484.

- Castaño, M. L., & Tan, X. (2015). Simultaneous stabilization of pitch and yaw of a gliding robotic fish using sliding mode control. In *Dynamic systems and control conference, vol. 57243*. American Society of Mechanical Engineers, V001T04A004.
- Coleman, D., Castaño, M., Ennasr, O., & Tan, X. (2019). Backstepping-based trajectory tracking for underwater gliders. In *Dynamic systems and control conference, vol. 59162*. American Society of Mechanical Engineers, V003T19A005.
- Coleman, D., & Tan, X. (2020). Backstepping control of gliding robotic fish for trajectory tracking in 3D space. In *2020 American control conference* (pp. 3730–3736). IEEE.
- Do, K. D., & Pan, J. (2009). *Control of ships and underwater vehicles: Design for underactuated and nonlinear marine systems* (pp. 109–124). Springer Science & Business Media.
- Do, K. D., Pan, J., & Jiang, Z.-P. (2004). Robust and adaptive path following for underactuated autonomous underwater vehicles. *Ocean Engineering*, 31(16), 1967–1997.
- Dong, E., Guo, S., Lin, X., Li, X., & Wang, Y. (2012). A neural network-based self-tuning PID controller of an autonomous underwater vehicle. In *The proceedings of the international conference on mechatronics and automation* (pp. 5–8).
- Ennasr, O. N. (2020). *Gliding robotic fish: Design, collaborative estimation, and application to underwater sensing* (Ph.D. thesis, ProQuest Dissertations and Theses), (p. 133). Michigan State University.
- Ennasr, O., Holbrook, C., Hondorp, D. W., Krueger, C. C., Coleman, D., Solanki, P., et al. (2020). Characterization of acoustic detection efficiency using a gliding robotic fish as a mobile receiver platform. *Animal Biotelemetry*, 8(1), 1–13.
- Esteban, S., Gordillo, F., & Aracil, J. (2013). Three-time scale singular perturbation control and stability analysis for an autonomous helicopter on a platform. *International Journal of Robust and Nonlinear Control*, 23(12), 1360–1392.
- Graver, J. G., & Leonard, N. E. (2001). Underwater glider dynamics and control. In *12th international symposium on unmanned untethered submersible technology* (pp. 1742–1710).
- Isa, K., & Arshad, M. R. (2013). Neural network control of buoyancy-driven autonomous underwater glider. In *Recent advances in robotics and automation* (pp. 15–35). Springer.
- Karkoub, M., Wu, H.-M., & Hwang, C.-L. (2017). Nonlinear trajectory-tracking control of an autonomous underwater vehicle. *Ocean Engineering*, 145, 188–198.
- Khalil, H. K. (2002). *Nonlinear systems* (3rd ed.). (p. 221). Prentice Hall, Upper Saddle River, NJ.
- Kokotovic, P., Khali, H. K., & O'reilly, J. (1999). *Singular perturbation methods in control: Analysis and design, vol. 25*. SIAM.
- Krogius, M., Haggemiller, A., & Olson, E. (2019). Flexible layouts for fiducial tags. In *2019 IEEE/RSJ international conference on intelligent robots and systems* (pp. 1898–1903). IEEE.
- Leonard, N. E., & Graver, J. G. (2001). Model-based feedback control of autonomous underwater gliders. *IEEE Journal of Oceanic Engineering*, 26(4), 633–645.
- Mahmoudian, N., & Woolsey, C. (2008). Underwater glider motion control. In *Decision and control, 2008. CDC 2008. 47th IEEE conference on* (pp. 552–557). IEEE.
- Mat-Noh, M., Arshad, M., Mohd-Mokhtar, R., & Khan, Q. (2017). Control of an autonomous underwater glider using integral super-twisting sliding mode control (ISTSMLC). In *Underwater system technology: Theory and applications (USYS), 2017 IEEE 7th international conference on* (pp. 1–6). IEEE.
- Nag, A., Patel, S. S., & Akbar, S. (2013). Fuzzy logic based depth control of an autonomous underwater vehicle. In *Automation, computing, communication, control and compressed sensing (IMAC4s), 2013 international multi-conference on* (pp. 117–123). IEEE.
- Olson, E. (2011). AprilTag: A robust and flexible visual fiducial system. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 3400–3407). IEEE.
- Petersen, K. Y., & Lefeber, E. (2001). Way-point tracking control of ships. 1, In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228)* (pp. 940–945). IEEE.
- Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. *Swarm Intelligence*, 1(1), 33–57.
- Rezazadegan, F., Shojaei, K., Sheikholeslam, F., & Chatraei, A. (2015). A novel approach to 6-DOF adaptive trajectory tracking control of an AUV in the presence of parameter uncertainties. *Ocean Engineering*, 107, 246–258.
- Sherman, J., Davis, R. E., Owens, W., & Valdes, J. (2001). The autonomous underwater glider “spray”. *IEEE Journal of Oceanic Engineering*, 26(4), 437–446.
- Sliwka, J., Clement, B., & Probst, I. (2012). Sea glider guidance around a circle using distance measurements to a drifting acoustic source. In *Intelligent robots and systems (IROS), 2012 IEEE/RSJ international conference on* (pp. 94–99). IEEE.
- Stommel, H. (1989). The slocum mission. *Oceanography*, 2(1), 22–25.
- Ullah, B., Ovinis, M., Baharom, M. B., Javaid, M., & Izhar, S. (2015). Underwater gliders control strategies: A review. In *Control conference (ASCC), 2015 10th Asian* (pp. 1–6). IEEE.
- Wang, J., & Tan, X. (2013). A dynamic model for tail-actuated robotic fish with drag coefficient adaptation. *Mechatronics*, 23(6), 659–668.
- Wang, J., Wang, C., Wei, Y., & Zhang, C. (2019). Command filter based adaptive neural trajectory tracking control of an underactuated underwater vehicle in three-dimensional space. *Ocean Engineering*, 180, 175–186.

- Wang, J., Wu, Z., Tan, M., & Yu, J. (2019). Model predictive control-based depth control in gliding motion of a gliding robotic dolphin. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(9), 5466–5477.
- Xu, R., Tang, G., Han, L., & Xie, D. (2019). Trajectory tracking control for a CMG-based underwater vehicle with input saturation in 3D space. *Ocean Engineering*, 173, 587–598.
- Yang, H., & Ma, J. (2010). Sliding mode tracking control of an autonomous underwater glider. In *2010 International conference on computer application and system modeling*, vol. 4 (pp. V4–555). IEEE.
- Yuan, J., Wu, Z., Yu, J., & Tan, M. (2017). Sliding mode observer-based heading control for a gliding robotic dolphin. *IEEE Transactions on Industrial Electronics*, 64(8), 6815–6824.
- Zhang, F. (2014). *Modeling, design and control of gliding robotic fish* (Ph.D. thesis), Michigan State University. Electrical Engineering.
- Zhang, F., Ennasr, O., Litchman, E., & Tan, X. (2016). Autonomous sampling of water columns using gliding robotic fish: Algorithms and harmful-algae-sampling experiments. *IEEE Systems Journal*, 10(3), 1271–1281.
- Zhang, F., Tan, X., & Khalil, H. K. (2012). Passivity-based controller design for stabilization of underwater gliders. In *American control conference* (pp. 5408–5413). IEEE.
- Zheng, Z., Ruan, L., & Zhu, M. (2019). Output-constrained tracking control of an underactuated autonomous underwater vehicle with uncertainties. *Ocean Engineering*, 175, 241–250.