



Article An Improved Backstepping Controller with an LESO and TDs for Robust Underwater 3D Trajectory Tracking of a Turtle-Inspired Amphibious Spherical Robot

Xihuan Hou ¹, Zan Li ², Shuxiang Guo ^{3,4,*}, Liwei Shi ³, Huiming Xing ⁵ and He Yin ³

- ¹ School of Intelligence and Information Engineer, Tangshan University, Tangshan 063000, China; diligent_hxh@126.com
- ² Cai Hong Unmanned Aerial Vehicle Co., Ltd., China Academy of Aerospace Aerodynamics, Beijing 100074, China; lizan@bit.edu.cn
- ³ Key Laboratory of Convergence Medical Engineering System and Healthcare Technology, The Ministry of Industry and Information Technology, Beijing Institute of Technology, Beijing 100081, China; shiliwei@bit.edu.cn (L.S.); yinhe@bit.edu.cn (H.Y.)
- ⁴ Department of Intelligent Mechanical Systems Engineering, Faculty of Engineering, Kagawa University, Takamatsu 761-0396, Japan
- ⁵ College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150001, China; xinghuiming@hrbeu.edu.cn
- * Correspondence: guoshuxiang@bit.edu.cn; Tel.: +86-159-1102-2732

Abstract: In this paper, a double closed-loop backstepping controller is designed for 3D trajectory tracking of a turtle-inspired amphibious spherical robot suffering from problems that include model uncertainties, environmental disturbances, and unmeasured velocity. The proposed controller scheme tackles three primary challenges: the differentiation explosion of the traditional backstepping method, unmeasured velocity, and the consideration of lumped disturbances. Beginning with an outer-loop backstepping controller, a virtual feedback variable is constructed to simplify the design of the backstepping controller. Meanwhile, to avoid the problem of differentiation explosion, tracking differentiators (TDs) are utilized to estimate the differentiation of the desired velocity in an inner-loop backstepping controller. Moreover, there are some uncertainty disturbances in the task of tracking the trajectory of a turtle-inspired amphibious spherical robot (TASR), such as the parameters of the hydrodynamic model and environmental disturbances. A linear extended state observer (LESO) is designed to estimate and compensate for the lumped disturbances. Furthermore, as the velocity states of the TASR are unmeasured, the LESO is also utilized to estimate the velocity states in surge, yaw, and heave degrees. Therefore, the TASR only needs to supply its position and orientation information for the trajectory tracking task. Note that this paper details both the design process of the proposed controller and a rigorous theoretical analysis. In addition, numerical simulations are conducted, and the results demonstrate the feasibility and superiority of the proposed method.

Keywords: turtle-inspired amphibious spherical robot; 3D trajectory tracking; backstepping control; linear extended state observer; tracking differentiators

1. Introduction

In recent years, with the shortage of land resources, human exploration of the ocean has intensified. In order to exploit and utilize marine resources more efficiently, many bio-inspired underwater robots have been developed in countries around the world. For example, Wang et al. developed a fish-shaped robot called Robcutt II [1]. Alessandro Crespi put forward a Salamandra Robotica II amphibious robot that is able to swim and walk [2]. A soft robotic fish was designed by Katzschmann [3]. Daniele Costa et al. designed a bionic autonomous underwater robot based on the fluid shape of fish bodies [4]. Our team has researched a series of amphibious spherical robots [5,6].



Citation: Hou, X.; Li, Z.; Guo, S.; Shi, L.; Xing, H.; Yin, H. An Improved Backstepping Controller with an LESO and TDs for Robust Underwater 3D Trajectory Tracking of a Turtle-Inspired Amphibious Spherical Robot. *Machines* **2022**, *10*, 450. https://doi.org/10.3390/ machines10060450

Academic Editor: Marco Ceccarelli

Received: 28 April 2022 Accepted: 31 May 2022 Published: 6 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). Generally, research concerning bionic underwater robots emphasizes control in terms of not only external appearance but also internal motion; control is a basic necessity for completing tasks. Generally, basic motion control of underwater vehicles requires tracking a predetermined underwater trajectory. At present, a large number of studies in the literature focus on the design of trajectory tracking controllers for underwater robots.

With respect to model uncertainty, which is a common problem, a controller independent of the model is the best choice; examples include the proportional integral derivative (PID) controller [7,8], the sliding-mode controller (SMC) [9,10], the neural network controller, and so on [11,12]. PID controllers perform well in tracking trajectory but lack the ability to reject disturbances. Sliding-mode controllers offer an algorithm for trajectory tracking that is very popular due not only to its model independence but also its good anti-interference performance. Elmokadem et al. proposed a terminal sliding-mode controller for AUVs that considered system uncertainty and environmental disturbances [13]. The simulation results showed that the controller was robust under bounded disturbances. Cui adopted the Lyapunov method to design an integral sliding-mode controller that successfully solved the problems involved in controlling robots with unknown velocity, unknown disturbances, and unknown hydrodynamic model parameters [10]. Although the sliding-mode controller is not sensitive to model parameters, its control quantity exhibits a high-frequency chattering phenomenon, which is not conducive to the execution of actuators. The neural network algorithm is another commonly utilized algorithm in underwater trajectory tracking. In [11], the authors proposed an efficient neural network approach with a single-layer structure for tracking control. Miao et al. designed a novel adaptive neural network tracking controller by combining dynamic surface control (DSC) and a minimal learning parameter (MLP) [12]. In addition, a radial basis function neural network was employed to account for tracking errors. Neural-network-based tracking controllers are insensitive to the dynamic model of the robot and possess strong adaptability and learning capabilities. However, due to the changeable nature of the underwater environment, the training and learning processes are difficult, which leads to poor real-time performance of the tracking algorithm.

If the kinematic model and hydrodynamic model of an underwater vehicle can be acquired, we can design a more suitable controller according to the approximate model parameters to achieve better control effectiveness.

Some algorithms based on models have been researched for trajectory tracking, such as model predictive control (MPC), backstepping control (BSC), and so on. Zhang et al. applied MPC to achieve three-dimensional trajectory tracking of an autonomous underwater vehicle (AUV). In [14], the conventional MPC algorithm was improved upon, and an event-trigger-based MPC algorithm was applied to the straight-line trajectory tracking task of an underactuated underwater vehicle (UUV). Shen et al. developed a Lyapunov-based model predictive control (LMPC) [15]. They considered practical constraints, for example, the actuator saturation. The algorithm was simulated and tested on the Saab SeaEye Falcon model AUV and achieved good results. They also proposed a nonlinear model predictive control method that realized a different type of underwater trajectory tracking for the AUV [16]. However, in order to achieve a suitable control quantity, the MPC method needs to predict many steps in advance, which results in the dimension of the weight matrix being very large. Therefore, the computational efficiency of the MPC is reduced, and real-time performance is somewhat poor.

Lyapunov-based backstepping controls represent the mainstream method for AUV tracking controls because their control law exploits 'good' nonlinearities in the system dynamics. Xu et al. adopted the backstepping method to determine virtual control quantity, then designed a controller that allowed a weight-bearing autonomous underwater vehicle (WAUV) to track its underwater trajectory [17]. In order to solve the differential explosion problem caused by using the backstepping method for the 3D path tracking of an underactuated underwater vehicle, Zhou et al. utilized the derivative output value of a biologically inspired model to replace the repeated calculation of multiple derivatives, which greatly

reduced the complexity of the controller [18]. In [19], the backstepping method and the disturbance state observer were combined to design a controller for trajectory tracking. Although backstepping-method-based controllers can be designed quickly and efficiently, they are not suitable for controlling vehicles with large dimensions; this is especially true for backstepping controllers with more than three levels. High-order derivative terms can amplify high-frequency micro-noise, which is not ideal for the controller.

From this analysis, it can be seen that each control algorithm possesses both advantages and disadvantages. In addition, the research focus of this paper, namely the turtle-inspired spherical robot (TASR), possesses characteristics such as unmeasured velocity, uncertain model parameters, and a nonlinear system; these factors present challenges for the design of a trajectory-tracking controller. Therefore, the design of a combined controller based on multiple algorithms is needed for the underwater trajectory tracking of TASRs. In this paper, the Lyapunov-based backstepping algorithm is adopted as the main controller, making full use of its good performance with respect to nonlinear systems. TDs and LESO are added to the main controller as auxiliary algorithms. ESOs are popular due to their ability to estimate uncertain information, including disturbances and state information. Xie et al. designed a reduced-order LESO to estimate disturbances that needs velocity information as input [20]. In [21], a nonlinear ESO was constructed to estimate the unmeasured velocity and disturbances. Inspired by [21], an LESO is put forward to avoid inaccurate estimates inherent in the use of nonlinear ESOs.

The primary contributions of this paper are summarized in three parts:

(1) A backstepping-based double closed-loop controller is designed. Beginning with an outer-loop backstepping controller, a virtual feedback variable is introduced to simplify the process.

(2) TDs are shown to be capable of tracking the differential signal of the desired velocity provided by the outer-loop controller, which solves the differential explosion problem.

(3) An LESO is designed to estimate the velocity states and lumped disturbances; it only requires position and orientation information. In addition, we introduce a rotating inertial coordinate to linearize the design of an ESO in terms of surge direction.

The rest of this paper is organized as follows. The modeling of a turtle-inspired amphibious spherical robot is introduced in Section 2. In Section 3, a double closed-loop backstepping controller with TDs and an LESO is designed, and the stability proof is presented. The results of the simulation are discussed in Section 4. Finally, the conclusions and directions for future work are summarized in Section 5.

2. Modeling of a Turtle-Inspired Amphibious Spherical Robot

2.1. The Mechanisms of the TASR

Based on the amphibious characteristics of sea turtles, especially the more flexible movement of sea turtles in the water, a turtle-inspired amphibious spherical robot was designed, as shown in Figure 1. As introduced in Reference [6], the robot consists of upper and lower hemispheres that are joined with a mid-plate. The upper hemisphere contains a passive water storage tank and a sealed cabin. The electrical system is placed in the sealed cabin. The lower hemisphere consists of two separate quarter spherical shells that can be opened or closed by servo motors on both sides according to the state of the TASR. Generally speaking, the two separate quarter spherical shells close when the robot maintains a hovering state but otherwise remain open. In addition, the vector propulsion mechanism is mounted on the mid-plate. Figure 2 shows the mechanical structure of the propulsion mechanism, which consists of four sets of propulsion units [6]. Each propulsion unit possesses three joints, with a water jet motor at the end of the lowest joint. Due to the limited availability of space for equipment, the TASR is only equipped with a stereo camera, an inertial measurement unit (IMU), 12 pressure sensors, and an acoustic communication module. Hence, the TASR is only able to perceive its position by use of the global locating system, the pressure sensor, and the orientation information provided by the IMU. The concrete technical specifications of the TASR are listed in Table 1.



Figure 1. Prototype of the turtle-inspired amphibious spherical robot. (**a**) The two separate quarter spherical shells are closed. (**b**) The two separate quarter spherical shells are open.



Figure 2. Mechanical structure of the vector-propulsion mechanism.

Items	Parameters
Dimension (<i>Width</i> \times <i>Length</i> \times <i>Height</i>)	$30 \text{ cm} \times 60 \text{ cm} \times 30 \text{ cm}$
Mass in air	6.5 Kg
Max thrust	3.8 N
Battery capacity	7.4 V rechargeable Ni-MH batteries (13,200 mAh)
Operation time	Average 100 min

2.2. The Circuit System of the TASR

The circuit system of the TASR is depicted in Figure 3. It contains four parts: the energy supply level, the sensor level, the decision and control level, and the executive level. The design of the circuit system involved confronting some important difficulties, including controlling multiple PWM devices, quickly processing visual information, handling decision and controller algorithms, and so on. In order to solve these problems, we adopted two control boards for the decision and control level. The TASR system contains 14 servo motors and 4 propellers in total, which require 18 PWM channels. On account of the fact that the STM32F4 series single-chip microcomputer possesses a sufficient number of PWM channels and is very commonly used, we chose it as the lower control board. The STM32F4 focuses on driving the operation of PWM devices and acquiring depth information. We utilized a Jetson series development board running a Ubuntu system as the main control board; it processes the information of the binocular camera and realizes human-computer interactions.



Figure 3. The TASR's circuit system components.

2.3. Allocation of Propeller Force

It is notable that the vector propulsion of the robot is very flexible. In order to ensure that the center of gravity and the centroid coincide, an "H"-shaped propulsion strategy was adopted for the 3D trajectory tracking task, as shown in Figure 4. The "H"-shaped propulsion strategy means that the four legs keep their hip and knee joints still, and each leg can only be adjusted by changing the angle of the ankle joint. In this situation, the robot can only generate three forces/movements: surge force τ_u , heave force τ_v , and yaw movement τ_w , which constitute the thrust vector $\boldsymbol{\tau} \in \mathbf{R}^{3\times 1}$.



Figure 4. "H"-shaped propulsion strategy.

The total force/movement τ of the robot is composed of the thrust of each propeller. $F = [F_1, F_2, F_3, F_4]$ is the thrust vector of propeller, τ is defined as:

τ

$$= BF \tag{1}$$

where $B \in \mathbf{R}^{4 \times 4}$ is the allocation matrix of thrust. When the thrusters are allocated according to "H" mode, *B* is defined as follows:

$$\boldsymbol{B} = \begin{bmatrix} -\cos\gamma_{1} & \cos\gamma_{2} & \cos\gamma_{3} & -\cos\gamma_{4} \\ 0 & 0 & 0 & 0 \\ -\sin\gamma_{1} & -\sin\gamma_{2} & -\sin\gamma_{3} & -\sin\gamma_{4} \\ \cos\gamma_{1} & -\cos\gamma_{2} & \cos\gamma_{3} & -\cos\gamma_{4} \end{bmatrix}$$
(2)

where γ_i (*i* =1,2,3,4) represents the angles of the four ankles, as plotted in Figure 5. The range of γ_i is set as $-\pi/6 \le \gamma_i \le \pi/6$.



Figure 5. Definition of γ_i .

2.4. Kinematics and Dynamics Model of the TASR

For convenience of analysis, two coordinate systems are considered: the inertial frame $O - X_I Y_I Z_I$ and the robot body frame $O - X_B Y_B Z_B$, as shown in Figure 6. The inertial coordinate system is used to reflect the global position and attitude angle of the robot. When analyzing the speed and thrust of the robot, it is convenient to use the body coordinate system. The position and attitude vector of the 6-DOF robot in the inertial frame is defined as:

$$\boldsymbol{\eta} = [x, y, z, \phi, \psi, \theta] \tag{3}$$

where *x*, *y*, and *z* represent the position of the TASR in the inertial frame, and ϕ , ψ , and θ stand for the Euler angle of roll, pitch, and yaw. For the body coordinates, the velocity vector of the TASR is written as follows:

١

$$\mathbf{v} = [u, v, w, p, q, r] \tag{4}$$

where u, v, and w are linear velocities of the TASR in x-direction, y-direction, and z-direction, respectively, denoted as the surge, sway, and heave, respectively. p, q, r are the angular velocities around x-axis, y-axis, z-axis, respectively. As the "H"-shaped propulsion strategy is adopted for trajectory tracking, the driving force can only be provided in terms of the degrees of surge, yaw, and heave. Therefore, the degrees of sway, roll, and pitch are neglected in this paper. Let $\eta = [x, y, z, \theta]$ and $\mathbf{v} = [u, w, r]$. The relationship between η and \mathbf{v} is shown as follows:

$$\dot{\boldsymbol{\eta}} = \mathcal{R}(\theta) \mathbf{v} \tag{5}$$

where

$$\mathcal{R}(\theta) = \begin{bmatrix} \cos\theta & 0 & 0\\ \sin\theta & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1 \end{bmatrix}$$
(6)

According to the Newton method, the dynamic model of the TASR is established:

$$M\dot{\mathbf{v}} + C(\mathbf{v})\mathbf{v} + D(\mathbf{v})\mathbf{v} + g(\eta) = \tau$$
(7)

where $M \in \mathbb{R}^{3\times3}$ is the inertia and added mass matrix; $C(\mathbf{v}) \in \mathbb{R}^{3\times3}$ is the Coriolis and centripetal matrix; $D(\mathbf{v}) \in \mathbb{R}^{3\times3}$ is the damping matrix, which is positive-definite; $g(\eta)$ is the restoring force and torque vector; and τ is the driving force vector.



Figure 6. Definition of the inertial frame and the robot body frame.

As the structure of the robot is symmetrical, the inertial matrix and added matrix can be expressed as Equation (9) [22]. Because the underwater motion speed of the robot is not high, i.e., it is less than 0.5 m/s, the Coriolis and centripetal matrix $C(\mathbf{v})$ can be ignored. Moreover, because the center of gravity and the buoyant center are at the same position, which is the geometric center of the BSR, the restoring force and torque vector $g(\eta)$ can be set to 0 vector ($g(\eta) = \mathbf{0}_{4\times 1}$). The simplified model is shown as follows:

$$M\dot{\mathbf{v}} + D(\mathbf{v})\mathbf{v} = \tau \tag{8}$$

where

$$\boldsymbol{M} = \begin{bmatrix} m_{11} & 0 & 0\\ 0 & m_{22} & 0\\ 0 & 0 & m_{33} \end{bmatrix} \tag{9}$$

$$\boldsymbol{D}(\mathbf{v}) = \begin{bmatrix} d_{11} & 0 & 0\\ 0 & d_{22} & 0\\ 0 & 0 & d_{33} \end{bmatrix} + \begin{bmatrix} X_u | u | & 0 & 0\\ 0 & Z_w | w | & 0\\ 0 & 0 & N_r | r | \end{bmatrix}$$
(10)

$$\boldsymbol{\tau} = [\tau_u, \tau_w, \tau_r]^T \tag{11}$$

Equations (5) and (7) can be written as:

$$\dot{\boldsymbol{\eta}} = \mathcal{R}(\boldsymbol{\theta}) \mathbf{v} \tag{12}$$

$$\dot{\mathbf{v}} = \boldsymbol{M}^{-1}\boldsymbol{\tau} - \boldsymbol{M}^{-1}\boldsymbol{D}(\mathbf{v})\mathbf{v}$$
(13)

3. Design of the Double Closed-Loop Backstepping Controller with an LESO and TDs

In order to achieve trajectory tracking for the TASR, a double closed-loop backstepping control with an LESO and TDs algorithm is proposed, as shown in Figure 7. In addition, the Lyapunov stability proof is provided. An outer-loop kinematic-model-based backstepping controller is designed to export the desired intermediate control variables according to the

reference trajectory. An inner-loop backstepping controller is based on the dynamic model of the TASR. Furthermore, an LESO is added into the inner-loop controller to solve two primary problems. First, the design of the LESO is utilized to estimate the model uncertainty and the environmental disturbances. Second, the unmeasured velocity information is provided by an LESO estimation. Furthermore, to avoid the differential explosion of the inner-loop backstepping controller, TDs are introduced to track the derivative of the desired intermediate control variables from the outer-loop controller.



Figure 7. The scheme of the double closed-loop backstepping controller with an LESO and TDs.

3.1. Outer-Loop Backstepping-Based Controller

For the sake of convenience, the dynamics mode of the TASR in Equation (13) is unfolded as the following:

$$\begin{bmatrix} \dot{u} \\ \dot{r} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} a_1 u + a_2 u^2 + a_3 \tau_u \\ b_1 r + b_2 r^2 + b_3 \tau_r \\ c_1 w + c_2 w^2 + c_3 \tau_w \end{bmatrix}$$
(14)

The reference pose of the robot is written as $\eta_{ref} = [x_{ref}, y_{ref}, z_{ref}\theta_{ref}]$. The derivative of the reference trajectory is denoted as $\dot{\eta}_{ref} = [u_{ref}, v_{ref}, w_{ref}, r_{ref}]$. Note that u_{ref} and v_{ref} represent velocity in the inertial frame. Further, the pose error in the TASR body coordinate system is defined as $\eta_{e_B} = [x_{e_B}, y_{e_B}, \theta_{e_B}, z_{e_B}]$. The subscript e_B represents the error in the frame $O - X_B Y_B Z_B$. Figure 8 depicts the pose error in the horizontal plane. The pose error in the inertial frame can be transformed into that of the body frame by Equation (15).

$$\boldsymbol{\eta}_{e_B} = \begin{bmatrix} x_{e_B} \\ y_{e_B} \\ z_{e_B} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & 0 \\ -\sin\theta & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} (\boldsymbol{\eta}_{ref} - \boldsymbol{\eta}) = \mathcal{T}(\theta)(\boldsymbol{\eta}_{ref} - \boldsymbol{\eta})$$
(15)

where $\mathcal{T}(\theta)$ is the matrix of rotation transformation, and η_e can be regarded as the error between the target trajectory and the robot. If $\lim_{t\to\infty}(|x_{e_B}| + |y_{e_B}| + |\theta_{e_B}| + |z_{e_B}|) = 0$, the TASR is considered as having tracked the reference trajectory.





After deriving the two sides of Equation (15), the following state equation of pose error is obtained:

$$\dot{\boldsymbol{\eta}}_{e_B} = \begin{bmatrix} \dot{\boldsymbol{x}}_{e_B} \\ \dot{\boldsymbol{y}}_{e_B} \\ \dot{\boldsymbol{z}}_{e_B} \\ \dot{\boldsymbol{\theta}}_{e_B} \end{bmatrix} = \begin{bmatrix} ry_{e_B} - u + u_{ref}cos\theta_{e_B} \\ rx_{e_B} + u_{ref}sin\theta_{e_B} \\ w_{ref} - w \\ r_{ref} - r \end{bmatrix}$$
(16)

It can be seen from Equation (16) that x_{e_B} , y_{e_B} , and θ_{e_B} are coupled. x_{e_B} is regulated by forward speed u in the robot coordinate system. θ_{e_B} is regulated by heading angular velocity r. In addition, y_{e_B} is regulated by x_{e_B} and θ_{e_B} together. So, we can adjust x_{e_B} and θ_{e_B} to make $\lim_{t\to\infty} |x_{e_B}| = 0$, $\lim_{t\to\infty} |y_{e_B}| = 0$, and $\lim_{t\to\infty} |z_{e_B}| = 0$.

According to the above analysis, a virtual feedback variable is constructed:

$$\overline{x}_{e_B} = x_{e_B} - k_1 \tanh(r) y_{e_B} \tag{17}$$

where k_1 is a positive constant and tanh() is a common nonlinear function. A specific expression of tanh() is expressed as follows:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$
(18)

It can be found that $|\tanh(x)| \le 1$, if and only if $x \to +\infty$ or $x \to -\infty$, $|\tanh(x)| = 1$. The derivative of $\tanh(x)$ is:

$$\frac{d(\tanh(x))}{dx} = (1 - \tanh^2(x)) \tag{19}$$

The virtual feedback variables corresponding to Equation (17) are analyzed below. The first-order derivative of $k_1 \tanh(r)$, with respect to intermediate control variable r, is continuous and bounded. If the external control makes $\lim_{t\to\infty} \overline{x}_{e_B} = 0$, then $\lim_{t\to\infty} x_{e_B} = k_1 \tanh(r)y_{e_B}$. According to $\dot{y}_{e_B} = -rx_e + u_{ref}sin\theta_{e_B}$ in Equation (16), if $\lim_{t\to\infty} \theta_{e_B} = 0$, then $\lim_{t\to\infty} y_{e_B} = \lim_{t\to\infty} (-rx_e) = \lim_{t\to\infty} (-k_1r \tanh(r)y_{e_B})$. Next, let us take part of the Lyapunov function $V_y = \frac{1}{2}y_{e_B}^2$. It can be seen that V_y is positive-definite. The derivative of the function over time t is $\dot{V}_y = y_{e_B}\dot{y}_{e_B} = -k_1r \tanh(r)y_{e_B}^2$. The derivative of V_y is positive-definite, and $\lim_{t\to\infty} y_{e_B} = 0$. This shows that y_{e_B} is an indirectly controlled quantity and is dominated by x_{e_B} and θ_{e_B} once again.

The candidate function of Lyapunov is chosen:

$$V_1 = \frac{1}{2}\overline{x}_{e_B}^2 + \frac{1}{2}y_{e_B}^2 + \frac{2}{k_3}(1 - \cos(\frac{\theta_{e_B}}{2})) + \frac{1}{2}z_{e_B}^2$$
(20)

where k_3 is a positive constant. It is obvious that Equation (20) consists of four parts, and each of them is positive-definite. When \overline{x}_e , y_{e_B} , θ_{e_B} , and z_{e_B} are equal to 0, then $V_1 = 0$. The output angle range of IMU is $\theta \in (-\pi, \pi)$, the yaw angle range of the reference trajectory is $\theta_{ref} \in (-\pi, \pi)$, and the angle error range is $\theta_e \in (-2\pi, 2\pi)$. The derivative of Equation (20) is written as follows:

$$\dot{V}_1 = \overline{x}_{e_B} \dot{\overline{x}}_{e_B} + y_e \dot{y_e} + \frac{1}{k_3} \sin((\frac{\theta_{e_B}}{2})\dot{\theta_{e_B}} + z_{e_B} \dot{z}_{e_B})$$
(21)

By introducing Equations (16) and (17) into Equation (21), we obtain:

$$\dot{V}_{1} = \overline{x}_{e_{B}}[ry_{e_{B}} - u + u_{ref}\cos\theta_{e_{B}} - k_{1}(1 - \tanh^{2}(r))\dot{(r)}y_{e_{B}} - k_{1}\tanh(r)(-rx_{e_{B}} + u_{ref}\sin\theta_{e_{B}})] + y_{e}[-rx_{e_{B}} + u_{ref}\sin\theta_{e_{B}}] + \frac{1}{k_{3}}\sin(\frac{\theta_{e_{B}}}{2})(r_{ref} - r) + z_{e_{B}}(w_{ref} - w)$$
(22)

If we assign $x_{e_B} = \overline{x}_{e_B} + k_1 \tanh(r) y_{e_B}$ into Equation (22), then we obtain

$$\dot{V}_{1} = \overline{x}_{e_{B}}[-u + u_{ref}\cos\theta_{e_{B}} - k_{1}(1 - \tanh^{2}(r))\dot{(r)}y_{e_{B}} - k_{1}\tanh(r)(-rx_{e} + u_{ref}\sin\theta_{e_{B}})] + [-k_{1}r\tanh(r)y_{e_{B}}^{2}] + \frac{1}{k_{3}}\sin(\frac{\theta_{e_{B}}}{2})[(r_{ref} - r) + 2k_{3}\cos(\frac{\theta_{e_{B}}}{2})u_{ref}y_{e_{B}}] + z_{e_{B}}(w_{ref} - w)$$
(23)

We assume that when $t \in [0, +\infty)$, u_{ref} , r_{ref} , w_{ref} , \dot{u}_{ref} , \dot{r}_{ref} , and \dot{w}_{ref} of the reference trajectory are all bounded, and u_{ref} , r_{ref} , and w_{ref} do not converge to 0 at the same time. Then, the controller law of the outer-loop backstepping controller is presented as follows:

$$u = u_d = u_{ref} \cos \theta_{e_B} - k_1 (1 - \tanh^2(r_d)) \dot{r}_d y_{e_B} - k_1 \tanh(r_d) (-r_d x_{e_B} + u_{ref} \sin \theta_{e_B})$$

$$+k_2(x_{e_B}-k_1\tanh(r_d)y_{e_B}) \tag{24a}$$

$$r = r_d = r_{ref} + 2k_3 \cos(\frac{\sigma_{e_B}}{2}) u_{ref} y_{e_B} + k_4 \sin(\frac{\sigma_{e_B}}{2})$$
(24b)

$$w = w_d = w_{ref} + k_5 z_{e_B} \tag{24c}$$

where u_d , r_d , and w_d represent the desired values of the intermediate control quantities u, r, and w, respectively.

When the three intermediate control variables reach their corresponding expected values, Equation (25) can be obtained.

$$\dot{V}_1 = -k_2 \overline{x}_{e_B}^2 - k_1 r \tanh(r) y_{e_B}^2 - \frac{k_4}{k_3} \sin^2(\frac{\theta_{e_B}}{2}) - k_5 z_{e_B}^2$$
(25)

where k_1, k_2, k_3, k_4 , and k_5 are positive scalar. If and only if $\overline{x}_{e_B}, y_{e_B}, \theta_{e_B}$, and z_{e_B} are 0, $\dot{V}_1 = 0$, then \dot{V}_1 is negative-definite. \dot{r}_d appears in u_d , and \dot{r}_d is as follows:

$$\dot{r}_{d} = \dot{r}_{ref} + 2k_{3}(\dot{y}_{e_{B}}u_{ref} + y_{e_{B}}\dot{u}_{ref})\cos(\frac{\theta_{e_{B}}}{2}) - k_{3}u_{ref}y_{e_{B}}\sin(\frac{\theta_{e_{B}}}{2})\dot{\theta}_{e_{B}} + \frac{1}{2}k_{4}\cos(\frac{\theta_{e_{B}}}{2})\dot{\theta}_{e_{B}}$$
(26)

where \dot{y}_{e_B} and $\dot{\theta}_{e_B}$ appear in \dot{r}_d and are computed according to Equations (27) and (28):

$$\dot{y}_{e_B} = -[r_{ref} + 2k_3 y_{e_B} u_{ref} \cos(\frac{\theta_{e_B}}{2}) + k_4 \sin(\frac{\theta_{e_B}}{2})] x_{e_B} + u_{ref} \sin(\theta_{e_B})$$
(27)

$$\dot{\theta}_e = -2k_3 y_e \nu_{ref} \cos(\frac{\theta_e}{2}) - k_4 \sin(\frac{\theta_e}{2}) \tag{28}$$

The stability analysis of the outer-loop backstepping controller is carried out as follows. Due to the fact that u_{ref} , r_{ref} , w_{ref} , \dot{u}_{ref} , \dot{r}_{ref} , and w_{ref} of the reference trajectory are all bounded, and the fact that every part of Equation (25) is negative-definite, then $\overline{x}_{e_B}^2$, $r \tanh(r)y_{e_B}^2$, $\sin^2(\frac{\theta_{e_B}}{2})$, and $z_{e_B}^2$ all converge to 0. Further, according to $\lim_{t\to\infty} r \tanh(r)y_{e_B}^2 = 0$, $\lim_{t\to\infty} y_{e_B} = 0$. Due to $\lim_{t\to\infty} \overline{x}_{e_B}^2 = 0$, $\lim_{t\to\infty} x_{e_B} = k_1 \tanh(r)y_{e_B}$. Because preconditions u_{ref} and r_{ref} do not converge to 0 at the same time, r does not converge to 0 in the process of $t \to \infty$, so $\lim_{t\to\infty} y_{e_B} = 0$ leads to $\lim_{t\to\infty} x_{e_B} = 0$. According to $\lim_{t\to\infty} \sin^2(\frac{\theta_{e_B}}{2}) = 0$, $\lim_{t\to\infty} \theta_{e_B} = 0$. To sum up, $\lim_{t\to\infty} (|x_{e_B}| + |y_{e_B}| + |\theta_{e_B}| + z_e) = 0$, and the TASR accomplishes the trajectory tracking task.

3.2. The Inner Loop of the Backstepping Controller

According to the previous section, the main problem is making the TASR track the desired trajectory by adjusting the intermediate control variables u, v, and r to their corresponding desired values. However, these intermediate control quantities cannot be set directly. The TASR system is a typical nonlinear second-order system model. Generally speaking, the actual control quantities τ_u , τ_r , and τ_w should be adjusted to make $u \rightarrow u_d$, $r \rightarrow r_d$, and $w \rightarrow w_d$, and then the robot can track the reference trajectory.

Errors between the expected speed and the actual speed of the robot are constructed in Equation (29).

$$\begin{bmatrix} u_e \\ r_e \\ w_e \end{bmatrix} = \begin{bmatrix} u_d - u \\ r_d - r \\ w_d - w \end{bmatrix}$$
(29)

In order to make the error converge to zero, a candidate function of Lyapunov is chosen as follows:

$$V_2 = V_1 + \frac{1}{2}u_e^2 + \frac{1}{2}r_e^2 + \frac{1}{2}w_e^2$$
(30)

If V_1 is positive-definite and the last three terms are all positive-definite, then V_2 is positive-definite. By computing the derivation of Equation (30), the following equation is obtained:

$$\dot{V}_2 = \dot{V}_1 + u_e \dot{u}_e + r_e \dot{r}_e + w_e \dot{w}_e \tag{31}$$

Equation (29) is written as:

$$\begin{bmatrix} u \\ r \\ w \end{bmatrix} = \begin{bmatrix} u_d - u_e \\ r_d - r_e \\ w_d - w_e \end{bmatrix}$$
(32)

Substituting Equations (23), (24a)–(24c), and (32) into (31), the following Equation is obtained:

$$\dot{V}_{2} = -k_{2}\overline{x}_{e_{B}}^{2} - k_{1}r\tanh(r)y_{e_{B}}^{2} - \frac{k_{4}}{k_{3}}\sin^{2}(\frac{\theta_{e_{B}}}{2}) - k_{5}z_{e_{B}}^{2} + u_{e}(\overline{x}_{e_{B}} + \dot{u}_{e}) + r_{e}(\frac{1}{k_{3}}\sin(\frac{\theta_{e_{B}}}{2}) + \dot{(r)}_{e}) + w_{e}(z_{e_{B}} + \dot{w}_{e})$$
(33)

Assuming that $t \in [0, +\infty)$, u_{ref} , r_{ref} , w_{ref} , \dot{r}_{ref} , \ddot{w}_{ref} , \ddot{u}_{ref} , \ddot{r}_{ref} , and \ddot{w}_{ref} of the reference trajectory are all continuous and bounded. So, we combine the dynamic model given by Equation (14) and obtain the following control law:

$$\tau_u = \tau_{ud} = \frac{x_{e_B} - k_1 \tanh(r) y_{e_B} + \dot{u}_d + k_6 (u_d - u) - a_1 u - a_2 u^2}{a_3}$$
(34a)

$$\tau_r = \tau_{rd} = \frac{\frac{1}{k_3} \sin(\frac{\theta_{e_B}}{2}) + \dot{r}_d + k_7(r_d - r) - b_1 r - b_2 r^2}{h_2}$$
(34b)

$$\tau_w = \tau_{wd} = \frac{z_{e_B} + \dot{w}_d + k_8(w_d - w) - c_1w - c_2w^2}{c_3}$$
(34c)

For the stability analysis, the control law provided by Equations (34a)–(34c) is introduced into the derivative of the Lyapunov candidate function, i.e., Equation (33), and the following equation is obtained:

$$\dot{V}_2 = -k_2 \overline{x}_{e_B}^2 - k_1 r \tanh(r) y_{e_B}^2 - \frac{k_4}{k_3} \sin^2(\frac{\theta_{e_B}}{2}) - k_5 z_{e_B}^2 - k_6 u_e^2 - k_7 r_e^2 - k_8 w_e^2$$
(35)

The first four terms constitute \dot{V}_1 . According to the previous section, \dot{V}_1 is negative-definite. If k_6 , k_7 , and k_8 remain positive, then the last three terms are negative-definite. To sum up, \dot{V}_2 is negative-definite, which means that $\lim_{t\to\infty} u_e = 0$, $\lim_{t\to\infty} r_e = 0$, and $\lim_{t\to\infty} w_e = 0$. In other words, when $t \to \infty$, then $u \to u_d$, $r \to r_d$, $w \to w_d$, and finally $x_{e_B} \to 0$, $y_{e_B} \to 0$, $\theta_{e_B} \to 0$, $z_{e_B} \to 0$, and the TASR completes the task of tracking the reference trajectory.

3.3. Design of an Inner-Loop Backstepping Controller with TDs

It can be found that the control law described in Equations (34a)–(34c) contains differentiation terms u_d , r_d , and w_d . As outputs of the outer-loop controller, u_d , r_d , and w_d are described in Equations (24a)–(24c), which also contain differentiation terms. If the differentiation of Equations (24a)–(24c) is taken directly as with the traditional backstepping technique, the problem of "explosion of terms" may result. In order to avoid this problem, TDs referring to the ADRC methods described in the literature [23] are adopted to track the differentiation of u_d and r_d . The TDs are designed as follows:

$$\begin{cases} \dot{u}_1 = u_2 \\ \dot{u}_2 = \text{fhan}(u_1 - u_d, u_2, R_0, h_0) \end{cases}$$
(36)

$$\begin{cases} \dot{r}_1 = r_2 \\ \dot{r}_2 = \text{fhan}(r_1 - r_d, r_2, R_1, h_1) \end{cases}$$
(37)

where u_1 is the tracking signal of u_d and u_2 is the tracking signal of the differentiation of u_d . The same goes for r_1 and r_2 . The definition of the function fhan refers to the literature [23]. The discrete forms of the above equations are presented as Equations (38) and (39):

$$\begin{cases} fh = \text{fhan}(u_1(k) - u_d(k), u_2(k), R_0, h_0) \\ u_1(k+1) = u_1(k) + hu_2(k) \\ u_2(k+1) = u_2(k) + hfh \end{cases}$$
(38)

$$\begin{cases} fh = \text{fhan}(r_1(k) - r_d(k), r_2(k), R_1, h_1) \\ r_1(k+1) = \dot{r}_1(k) + hr_2(k) \\ r_2(k+1) = \dot{r}_2(k) + hfh \end{cases}$$
(39)

h represents the sampling time. For consistency of symbols, u_2 is marked as \hat{u}_d and r_2 is marked as \hat{r}_d . Using the u_2 , r_2 produced by the TDs, the controller can be written as:

$$\tau_u = \tau_{ud} = \frac{x_{e_B} - k_1 \tanh(r) y_{e_B} + \hat{u}_d + k_6 (u_d - u) - a_1 u - a_2 u^2}{a_3}$$
(40a)

$$\tau_r = \tau_{rd} = \frac{\frac{1}{k_3}\sin(\frac{\theta_{e_B}}{2}) + \hat{r}_d + k_7(r_d - r) - b_1r - b_2r^2}{b_3}$$
(40b)

$$\tau_w = \tau_{wd} = \frac{z_{e_B} + \dot{w}_d + k_8(w_d - w) - c_1w - c_2w^2}{c_3}$$
(40c)

3.4. Design of an LESO for an Inner-Loop Backstepping Controller

We must consider that the TASR is affected by ocean currents, propeller thrust fluctuations, and other factors when it is performing operations in a lake or sea. In addition, different speeds make the hydrodynamic parameters a_i,b_i , and c_i , (i = 1,2,3) change to some extent (we assume that a_3, b_3 , and c_3 change only slightly). These can be considered lumped disturbances. The dynamic equation in the actual environment with disturbances can be written as follows:

$$\begin{bmatrix} \dot{u} \\ \dot{r} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} a_1 u + a_2 u^2 + a_3 \tau_u + \Omega_u \\ b_1 r + b_2 r^2 + b_3 \tau_r + \Omega_r \\ c_1 w + c_2 w^2 + c_3 \tau_w + \Omega_w \end{bmatrix}$$
(41)

where, Ω_u , Ω_r , and Ω_w are the total disturbances in *u*, *r*, and *w* directions from the ocean current, propeller thrust fluctuation, and so on.

However, the velocity of the TASR is unmeasured directly. It is necessary to combine the kinematic and dynamic models together to form a second-order system to design an extended state observer.

A typical nonlinear second-order system with disturbances is expressed as follows:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f(\cdot) + b\tau \\ y &= x_1 \end{aligned}$$
 (42)

where f(t, x) is the lumped disturbance, including a nonlinear uncertain term and an external environment disturbance, *b* is the model parameter, and τ is the controller input. Taking $x_1 = x_1$, $x_2 = x_2$, $x_3 = f(t, x) + \Omega$, the state-space equation is expressed as follows:

$$\begin{bmatrix} \dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ y \end{bmatrix} = \begin{bmatrix} x_2\\ x_3 + b\tau\\ h\\ x_1 \end{bmatrix}$$
(43)

where *h* is the derivative of $f(\cdot)$ and |h| is bounded. Referring to [24], the following observer can be constructed:

$$\begin{bmatrix} \dot{z}_1\\ \dot{z}_2\\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} z_2 - \beta_1 e\\ z_3 - \beta_2 e + b\tau\\ -\beta_3 e \end{bmatrix}$$
(44)

where $e = z_1 - y$, and z_1, z_2 , and z_3 are the estimates of x_1, x_2 , and x_3 . β_1, β_2 , and β_3 are gain parameters. According to the literature [25], gain parameters are adjusted based on a bandwidth ω_0 . Generally, $\beta_1 = 3\omega_0, \beta_2 = 3\omega_0^2$, and $\beta_3 = \omega_0^3$.

According to the kinematic model in Equation (6) and the dynamic model with disturbances in Equation (41), it can be seen that the models for yaw and heave are easily transformed into the typical nonlinear second-order system expressed in Equation (42). Hence, for the heave and yaw directions, the estimates of lumped disturbances are presented as z_{3w} and z_{3r} , and the estimates of velocity are marked as z_{2w} and z_{2r} , respectively. In contrast, for the surge direction, an expression as in Equation (42) cannot be directly employed. To solve this problem, a rotation inertia coordinate, as shown in Figure 9, is established.



Figure 9. Diagram of the rotation inertial coordinate.

In the frame $O_{RI} - X_{RI}Y_{RI}Z_{RI}$, the kinematic and dynamic models of the TASR with respect to surge direction are expressed in Equation (45). Note that Equation (45) is easily transformed into the typical nonlinear system expressed in Equation (42). According to Equation (44), the lumped disturbance and velocity for surge direction can be obtained and marked as z_{3u} and z_{2u} , respectively.

$$\dot{x}_{RI} = u$$

$$\dot{u} = a_1 u + a_2 u^2 + a_3 \tau_u + \Omega_u$$
(45)

Following from the above analysis, the control law of the double closed-loop backstepping controller with TDs and an LESO is concluded as follows:

$$\tau_u = \tau_{ud} = \frac{x_{e_B} - k_1 \tanh(r) y_{e_B} + \hat{u}_d + k_6 (u_d - z_{2u}) - z_{3u}}{a_3}$$
(46a)

$$\tau_r = \tau_{rd} = \frac{\frac{1}{k_3}\sin(\frac{\theta_{e_B}}{2}) + \hat{r}_d + k_7(r_d - z_{2r}) - z_{3r}}{b_3}$$
(46b)

$$\tau_w = \tau_{wd} = \frac{z_{e_B} + \dot{w}_d + k_8(w_d - z_{2w}) - z_{3w}}{c_3}$$
(46c)

4. Simulation Results

To verify the feasibility of the proposed control scheme, which we refer to as TDs-LESO-BSC below, we conduct some simulation analyses using the professional software MAT-LAB2015a on a system running Windows 7, equipped with an Intel core i7 CPU at 3.6 GHz and 8 GB of RAM. For the simulation, some parameters of the proposed controller need to be set. The model parameters are acquired by online recursive least-squares-based model identification [22]: $a_3 = 2.4$, $b_3 = 476.2$, and $c_3 = 2.8$. The gains in Equation (46a)–(46c) are selected as follows: $k_1 = 1$, $k_2 = 2$, $k_3 = 1$, $k_4 = 5$, $k_5 = 2$, $k_6 = 4$, $k_7 = 10$, and $k_8 = 2$. The parameters of the TDs in Equations (38) and (39) are set as follows: $h_0 = 0.08$, $R_0 = 10$, $h_1 = 0.06$, $R_1 = 10$, and h = 0.01. The bandwidth of the LESO in Equation (44) for the directions of surge, yaw, and heave are set as 35, 24, and 30, respectively. Meanwhile, the actuated forces in the surge and heave directions are bounded between -4 N and 4 N, and the actuated torque in the yaw direction is bounded between -0.5 Nm and 0.5 Nm. The sampling time is 0.02 s.

4.1. 3D Trajectory Tracking Performance

To verify the 3D trajectory tracking performance, the reference 3D trajectory with respect to time t is described by

$$\begin{cases} x_{ref}(t) = 2\cos(\frac{\pi}{15}t - \frac{\pi}{2}) \\ y_{ref}(t) = 2\sin(\frac{\pi}{15}t - \frac{\pi}{2}) \\ z_{ref}(t) = -0.1t \end{cases}$$
(47)

whose shape is a standard spiral line with radius of 2 m. The original state of the TASR is set as x = 0, y = -1.5, z = 0, and $\theta = \frac{\pi}{4}$. In addition, the following disturbances in surge, yaw, and heave degrees are introduced to the TASR. Note that gains in the sinusoidal signal are related to different values of the positive constant β . Because the max torque of the TASR with respect to the yaw is 0.5 Nm , the disturbance introduced for the yaw is a little smaller than that for the other two directions.

$$\begin{cases} dtb_u = \beta \sin(\frac{\pi}{75}t) \\ dtb_r = 0.1\beta \sin(\frac{\pi}{75}t) \\ dtb_w = \beta \sin(\frac{\pi}{75}t) \end{cases}$$
(48)

With $\beta = 1$, the curves of the tracking trajectory are shown in Figure 10. The proposed tracking controller successfully drives the TASR toward the desired trajectory, which verifies the stability and feasibility of the control method. The tracking errors of the TASR states are plotted in Figure 11. All errors converge almost to zero. The required control forces are shown in Figure 12. The control inputs in the surge and heave directions converge to a sinusoidal signal, which demonstrates the anti-interference ability of the TDs-LESO-BSC. On account of the fact that the disturbance in yaw is slight, the control input converges to a steady value.



Figure 10. The trajectory considered in simulation.



Figure 11. State errors of the TASR.



Figure 12. Actual control inputs.

Furthermore, the estimated performance of the LESO is evaluated in Figure 13. It can be seen that the estimates of velocity and lumped disturbances are consistent with the actual values after 10 s. Accurately observed lumped disturbances improve the anti-interference properties of the controller. Further, accurate estimates of velocity are able to solve the problem that miniature bio-underwater robots cannot perceive their velocity.



Figure 13. The estimations of velocity and lumped disturbances. (**a**) The estimated value of the velocity; (**b**) The estimated value of the lumped disturbance.

For a more intuitive perspective on the tracking accuracy of the TDs, the contrasts between the actual and estimated values are described in Figure 14. Derivations of the surge and yaw velocities can be tracked accurately. Moreover, the derivation tracking performance is evaluated by the RMS errors, where the RMS derivation tracking errors for u_d and r_d are 0.0015 m/s² and 0.00001 rad/s², respectively. Accurate tracking of the derivation for surge and yaw velocities improves the trajectory tracking effect. In the simulation, the running time of the controller is recorded. The running times with TDs and without TDs are 0.701 s and 0.924 s, respectively. The design that includes TDs improves the efficiency of the proposed controller.



Figure 14. The estimations of u_d , \dot{u}_d , r_d , and \dot{r}_d based on fhan.

4.2. Robust Test

In order to investigate the robustness of the TDs-LESO-BSC further, we simulate TASR tracking control under different conditions with respect to lumped disturbances, with values of β varying in {1.0, 3.0, 5.0}. The RMS tracking errors under different lumped disturbance conditions are computed in Table 2. It can be seen that tracking quality is insensible to enhanced disturbances. The simulation results demonstrate the robustness of the TDs-LESO-BSC. However, the insensitivity of the tracking quality only holds for limited interference. Once the control input bounds restrict the disturbance-resisting capabilities of the TASR, the tracking quality worsens, even though the disturbances can be estimated accurately.

Table 2. RMS tracking errors with different disturbances.

	eta=1.0	eta= 2.0	eta=5.0
Surge RMS e_x (m)	0.001	0.002	0.002
Sway RMS e_y (m)	0.015	0.022	0.023
Heave RMS e_z (m)	0.000	0.001	0.004
Yaw RMS e_{θ} (rad)	0.006	0.007	0.007

4.3. Comparative Simulations

To reveal the superiority of the proposed control scheme, a simulation comparing it to a traditional backstepping controller (BSC) is conducted in this paper. The parameters of the BSC are consistent with the proposed controller. With $\beta = 5.0$, the curves of the tracking trajectory are shown in Figure 15. It is found that the TASR driven by the TDs-LESO-BSC controller is able to track the desired trajectory accurately. However, on account of a serious disturbance, the trajectory based on BSC fails to converge to the desired trajectory. The tracking error of the TASR can be seen in Figure 16. The position and yaw angle errors converge to 0 when the TASR is driven by the TDs-LESO-BSC controller. However, the errors seriously fluctuate during the adjustment of the BSC controller. Further, to analyze the tracking error results quantitatively, the root mean square (RMS) error is employed to evaluate tracking accuracy. As can be seen from Table 3, the RMS errors of the TDs-LESO-



BSC are distinctly smaller than those of the BSC, which highlights the superior 3D tracking performance of TDs-LESO-BSC.

Figure 15. The comparison of different controllers for tracking a spiral trajectory with serious disturbances.



Figure 16. The contrasts of the state tracking errors in the spiral trajectory tracking case.

Table 3. The RMS errors of the TASR in the spiral trajectory case.

	TDs-LESO-BSC	BSC	
Surge RMS e_x (m)	0.002	0.037	
Surge RMS e_{y} (m)	0.023	0.061	
Surge RMS e_z (m)	0.004	0.470	
Surge RMS e_{θ} (rad)	0.007	0.009	

In order to further verify the superiority of the TDs-LESO-BSC, another comparison is investigated. In this case, the desired trajectory for a raster scan is described by Equation (49). The initial state of the robot is $[x, y, z, \theta] = [0, -1.5, -25, \pi/4]$. The parameters for the controller are the same as in the above simulation scenarios. Disturbances as defined in Equation (48) with $\beta = 5.0$ are exerted. The curves of the tracking trajectory are depicted in Figure 17. It is obvious that the TDs-LESO-BSC controller is able to resist the serious disturbances and drive the TASR to track the desired trajectory. In contrast, the trajectory based on the traditional BSC exhibits significant oscillation. Figure 18 shows the contrast of the state errors. The state errors based on TDs-LESO-BSC mostly converge to zero. The position errors based on the BSC are larger, especially in the X-axial and Z-axial

directions. For a more intuitive observation, the RMS values for the state errors are listed in Table 4. It can be seen that the RMS errors in the X, Y, and Z axial directions are smaller for TDs-ESO-BSC than for the BSC. The above analysis verifies the anti-interference ability and accurate trajectory tracking ability of the proposed controller.

$$\begin{cases} = [0.5t, 0, 0.1t - 30], & 0 < t \le 60 \text{ s} \\ [x_{ref}(t), y_{ref}(t), z_{ref}(t)] = [30, 0.5t - 30, 0.1t - 30], & 60 < t \le 80 \text{ s} \\ [x_{ref}(t), y_{ref}(t), z_{ref}(t)] = [70 - 0.5t, 10, 0.1t - 30], & 80 < t \le 140 \text{ s} \\ [x_{ref}(t), y_{ref}(t), z_{ref}(t)] = [0, 0.5t - 60, 0.1t - 30], & 140 < t \le 160 \text{ s} \\ [x_{ref}(t), y_{ref}(t), z_{ref}(t)] = [0.5t - 80, 20, 0.1t - 30], & 160 < t \le 220 \text{ s} \end{cases}$$



Figure 17. The comparison of different controllers for tracking a raster scan trajectory with serious disturbances.



Figure 18. The contrasts of the state tracking errors in the raster scan trajectory tracking case.

Table 4. The RMS errors of the TASR in the raster scan trajectory case.

	TDs-LESO-BSC	BSC	
Surge RMS e_x (m)	0.010	0.148	
Surge RMS e_{y} (m)	0.035	0.068	
Surge RMS e_z (m)	0.291	4.541	
Surge RMS e_{θ} (rad)	0.020	0.023	

4.4. Discussion

On the basis of the above comprehensive simulation analysis, the proposed double closed-loop backstepping controller with TDs and LESO is demonstrated to be a feasible and robust trajectory tracking control method for the TASR. In addition, a comparison of simulation times verifies that the TDs not only avoid the "explosion of term" problem but also improve computing efficiency. Further, the robust test and comparative simulations indicate that the proposed control scheme, enhanced by the LESO, has a satisfactory capability to resist disturbances and model uncertainties, thus ensuring accurate trajectory tracking performance. However, the algorithm cannot reject all environmental disturbances due to the limited thrust capabilities of the robot's systems.

In addition to this, there are notable advantages to the proposed control scheme. The proposed LESO guarantees that the controller is able to achieve trajectory tracking by relying only on position and orientation information in an unmeasured velocity situation. This scheme provides an alternative method of trajectory tracking for miniature bio-inspired underwater robots that cannot implement a DVL due to limited space.

5. Conclusions and Future Work

In this paper, we implemented a TASR control scheme, consisting of a backstepping controller, TDs, and an LESO, to handle 3D trajectory tracking tasks while considering the challenges presented by a nonlinear system, unmeasured velocity, and model parameter uncertainties; we also considered ambient interference. The double closed-loop backstepping method is known for its ability to process nonlinear systems. However, the defects of the backstepping method are obvious. For example, with an increase in system order, there are too many unknown parameters and the problem of differential explosion. Therefore, we introduced a virtual feedback variable into the outer-loop backstepping control that decreased the confined adjusted parameters. Furthermore, to avoid the problem of "explosion of terms", TDs were designed to track the derivatives of desired velocities. For the problems of unmeasured velocity, model parameter uncertainties, and ambient interference, an LESO was designed to reject lumped disturbances and control the TASR by relying only on position and orientation information. Theoretical analysis and simulation results verified the feasibility of the proposed control scheme.

In the future, this control scheme will be implemented in a TASR to verify the robustness of the controller in a real environment.

Author Contributions: Conceptualization, X.H.; methodology, X.H.and Z.L.; software, X.H.; validation, X.H.; formal analysis, X.H. and Z.L; investigation, H.Y.; resources, H.X.; data curation, X.H.; writing—original draft preparation, X.H.; writing—review and editing, X.H.; visualization, X.H.; supervision, S.G. and L.S.; project administration, S.G.; funding acquisition, S.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (61773064, 61503028).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Wang, R.; Wang, S.; Wang, Y.; Tang, C.; Tan, M. Three-Dimensional Helical Path Following of an Underwater Biomimetic Vehicle-Manipulator System. *IEEE J. Ocean. Eng.* 2018, 43, 391–401. [CrossRef]
- Crespi, A.; Karakasiliotis, K.; Guignard, A.; Ijspeert, A.J. Salamandra Robotica II: An Amphibious Robot to Study Salamander-Like Swimming and Walking Gaits. *IEEE Trans. Robot.* 2013, 29, 308–320. [CrossRef]
- Katzschmann, R.K.; DelPreto, J.; MacCurdy, R.; Rus, D. Exploration of underwater life with an acoustically controlled soft robotic fish. Sci. Robot. 2018, 3, eaar3449. [CrossRef] [PubMed]
- Costa, D.; Palmieri, G.; Palpacelli, M.C.; Panebianco, L.; Scaradozzi, D. Design of a Bio-Inspired Autonomous Underwater Robot. J. Intell. Robot. Syst. Theory Appl. 2018, 91, 181–192. [CrossRef]

- 5. Shi, L.; Guo, S.; Mao, S.; Yue, C.; Li, M.; Asaka, K. Development of an Amphibious Turtle-Inspired Spherical Mother Robot. *J. Bionic Eng.* **2013**, *10*, 446–455. [CrossRef]
- Xing, H.; Guo, S.; Shi, L.; Hou, H.; Su, S.; Chen, Z.; Liu, Y.; Liu, H.; Hu, Y.; Xia, D.; et al. A Novel Small-scale Turtle-inspired Amphibious Spherical Robot. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 1702–1707.
- Kim, M.J.; Baek, W.K.; Ha, K.N.; Joo, M.G. Way-point tracking for a hovering AUV by PID controller. In Proceedings of the 2015 15th International Conference on Control, Automation and Systems (ICCAS), Busan, Korea, 13–16 October 2015; pp. 744–746. [CrossRef]
- Shi, L.; Su, S.; Guo, S.; Tang, K.; Pan, S.; He, Y.; Xing, H.; Chen, Z.; Guo, P. A fuzzy PID control method for the underwater spherical robot. In Proceedings of the 2017 IEEE International Conference on Mechatronics and Automation (ICMA), Takamatsu, Japan, 6–9 August 2017; pp. 626–631. [CrossRef]
- Elmokadem, T.; Zribi, M.; Youcef-Toumi, K. Trajectory tracking sliding mode control of underactuated AUVs. *Nonlinear Dyn.* 2016, 84, 1079–1091. [CrossRef]
- Cui, R.; Chen, L.; Yang, C.; Chen, M. Extended State Observer-Based Integral Sliding Mode Control for an Underwater Robot With Unknown Disturbances and Uncertain Nonlinearities. *IEEE Trans. Ind. Electron.* 2017, 64, 6785–6795. [CrossRef]
- 11. Pan, C.Z.; Lai, X.Z.; Yang, S.X.; Wu, M. An efficient neural network approach to tracking control of an autonomous surface vehicle with unknown dynamics. *Expert Syst. Appl. Int. J.* 2013, 40, 1629–1635. [CrossRef]
- 12. Miao, B.; Li, T.; Luo, W. A DSC and MLP based robust adaptive NN tracking control for underwater vehicle. *Neurocomputing* **2013**, *111*, 184–189. [CrossRef]
- 13. Elmokadem, T.; Zribi, M.; Youcef-Toumi, K. Terminal sliding mode control for the trajectory tracking of underactuated Autonomous Underwater Vehicles. *Ocean. Eng.* 2017, 129, 613–625. [CrossRef]
- Liu, C.; Gao, J.; Zhang, G.; Xu, D. Robust event-triggered model predictive control for straight-line trajectory tracking of underactuated underwater vehicles. In Proceedings of the OCEANS 2017—Aberdeen, Aberdeen, UK, 19–22 June 2017; pp. 1–5. [CrossRef]
- 15. Shen, C.; Shi, Y.; Buckham, B. Trajectory Tracking Control of an Autonomous Underwater Vehicle Using Lyapunov-Based Model Predictive Control. *IEEE Trans. Ind. Electron.* **2018**, *65*, 5796–5805. [CrossRef]
- Shen, C.; Buckham, B.; Shi, Y. Modified C/GMRES Algorithm for Fast Nonlinear Model Predictive Tracking Control of AUVs. IEEE Trans. Control. Syst. Technol. 2017, 25, 1896–1904. [CrossRef]
- 17. Xu, W.; Li, H.; Zhang, J.; Zhu, Y.; Zhang, H. Trajectory Tracking for Underwater Rescue Salvage Based on Backstepping Control. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 1956–1961. [CrossRef]
- Zhou, J.; Ye, D.; He, D.; Xu, D. Three-dimensional trajectory tracking of an underactuated UUV by backstepping control and bio-inspired models. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 966–972. [CrossRef]
- 19. Liu, S.; Liu, Y.; Wang, N. Nonlinear disturbance observer-based backstepping finite-time sliding mode tracking control of underwater vehicles with system uncertainties and external disturbances. *Nonlinear Dyn.* **2017**, *88*, 465–476. [CrossRef]
- Xie, T.; Li, Y.; Jiang, Y.; An, L.; Wu, H. Backstepping active disturbance rejection control for trajectory tracking of underactuated autonomous underwater vehicles with position error constraint. *Int. J. Adv. Robot. Syst.* 2020, 17, 1729881420909633. [CrossRef]
- Kong, S.; Sun, J.; Qiu, C.; Wu, Z.; Yu, J. Extended State Observer-Based Controller With Model Predictive Governor for 3-D Trajectory Tracking of Underactuated Underwater Vehicles. *IEEE Trans. Ind. Inform.* 2021, 17, 6114–6124. [CrossRef]
- Hou, X.; Guo, S.; Shi, L.; Xing, H.; Yin, H.; Li, Z.; Zhou, M.; Xia, D. Improved Model Predictive-Based Underwater Trajectory Tracking Control for the Biomimetic Spherical Robot under Constraints. *Appl. Sci.* 2020, 10, 8106. [CrossRef]
- 23. Han, J. From PID to Active Disturbance Rejection Control. IEEE Trans. Ind. Electron. 2009, 56, 900–906. [CrossRef]
- 24. Chen, X.; Li, D.; Gao, Z.; Wang, C. Tuning method for second-order active disturbance rejection control. In Proceedings of the 30th Chinese Control Conference, Yantai, China, 22–24 July 2011; pp. 6322–6327.
- 25. Gao, Z. Scaling and bandwidth-parameterization based controller tuning. In Proceedings of the 2003 American Control Conference, Denver, CO, USA, 4–6 June 2003; Volume 6, pp. 4989–4996. [CrossRef]