




Article

Trajectory Following Control of an Unmanned Vehicle for Marine Environment Sensing

Tegen Eyasu Derbew ¹, Nak Yong Ko ^{1,*} and Sung Hyun You ²

¹ Interdisciplinary Program in IT-Bio Convergence Systems, Department of Electronic Engineering, Chosun University, Gwangju 61452, Republic of Korea; eyasuderbew2008@gmail.com

² Department of Electronic Engineering, Chosun University, Gwangju 61452, Republic of Korea; you@chosun.ac.kr

* Correspondence: nyko@chosun.ac.kr; Tel.: +82-62-230-7108

Abstract: An autonomous surface vehicle is indispensable for sensing of marine environments owing to its challenging and dynamic conditions. To accomplish this task, the vehicle has to navigate through a desired trajectory. However, due to the complexity and dynamic nature of a marine environment affected by factors such as ocean currents, waves, and wind, a robust controller is of paramount importance for maintaining the vehicle along the desired trajectory by minimizing the trajectory error. To this end, in this study, we propose a robust discrete-time super-twisting second-order sliding mode controller (DSTA). Besides, this control method effectively suppresses the chattering effect. To start with, the vehicle's model is discretized using an integral approximation with nonlinear terms including environmental disturbances treated as perturbation terms. Then, the perturbation is estimated using a time delay estimator (TDE), which further enhances the robustness of the proposed method and allows us to choose smaller controller gains. Moreover, we employ a genetic algorithm (GA) to tune the controller gains based on a quadratic cost function that considers the tracking error and control energy. The stability of the proposed sliding mode controller (SMC) is rigorously demonstrated using a Lyapunov approach. The controller is implemented using the Simulink[®] software. Finally, a conventional discrete-time SMC based on the reaching law (DSMR) and a heuristically tuned DSTA controller are used as benchmarks to compare the tracking accuracy and chattering attenuation capability of the proposed GA based DSTA (GA-DSTA). Simulation results are presented both with or without external disturbances. The simulation results demonstrate that the proposed controller drives the vehicle along the desired trajectory successfully and outperforms the other two controllers.

Keywords: sensing; marine; vehicle; sliding mode control; trajectory following; estimation; Lyapunov stability; genetic algorithm; chattering



Citation: Derbew, T.E.; Ko, N.Y.; You, S.H. Trajectory Following Control of an Unmanned Vehicle for Marine Environment Sensing. *Sensors* **2024**, *24*, 1262. <https://doi.org/10.3390/s24041262>

Academic Editors: Charalampos P. Bechlioulis and George C. Karras

Received: 17 January 2024

Revised: 11 February 2024

Accepted: 13 February 2024

Published: 16 February 2024



Copyright: © 2024 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/>).

1. Introduction

Marine environment sensing, monitoring, and management are essential to protect the ocean. In this regard, autonomous surface vehicles (ASVs) or unmanned surface vehicles (USVs) (hereafter, ASVs or USVs can be used interchangeably) play a crucial role in sensing and monitoring the ocean, providing essential information [1–4]. The ocean covers two-thirds of the Earth's surface and has a substantial impact on weather patterns, marine ecosystems, coastal regions, and the dynamics of climate change [1]. It is a source of energy, food, and materials; moreover, it is used for transportation and recreational purposes [5]. Marine ecosystem degradation, ocean temperature rise, plastic waste contamination, and pollution, etc are posing a threat to mankind [1]. To address this problem, surface vehicles equipped with sensors can be applied to collect data. For autonomous, efficient, and accurate sensing of marine environments, following a desired sensing trajectory is essential. Moreover, autonomous trajectory following of USVs has been utilized

in various application areas, including military operations, transportation, surveillance, and offshore infrastructure [6–9]. This paper focuses on the trajectory following control of USVs.

Different types of USVs are used depending on the application area; however, all USVs share some basic components, such as the hull, data collection equipment, communication systems, guidance, navigation and control (GNC), and propulsion and power systems [10]. The GNC system plays a significant role in a vehicle's capability of accomplishing a given mission. One of the main challenges to develop USVs is the lack of a reliable and automated GNC that can operate in sophisticated and hazardous operating conditions, and overcome the possible failures in sensors, actuators, and communication [10,11]. The general block diagram of GNC is shown in Figure 1. In the navigation system, different onboard sensors, such as global positioning system (GPS), inertial measurement unit (IMU), compass, sonars, and cameras can be incorporated to measure the states of the vehicle, that is, position, orientation, velocity, and acceleration, as well as to gather data regarding the environment for scientific research. Additionally, state estimator or observer can be used to estimate the states of the vehicle if there are no adequate sensors or to estimate the unmeasurable states of the system.

The guidance system generates a continuous and smooth trajectory to the control system based on the information provided by the navigation system, desired mission, and environmental conditions. In this study, we focus on developing the control system as part of the GNC, assuming that the navigation system provides the necessary states, which are the vehicle's position, orientation, velocity, and acceleration. The control system operates to drive the vehicle to follow the command velocity provided by the guidance module, utilizing information provided by the navigation systems and minimizing the position trajectory error [10]. Therefore, the tracking error in the research indicates the difference between the controlled velocity and the commanded velocity. The tracking error in the position is considered by the guidance system.

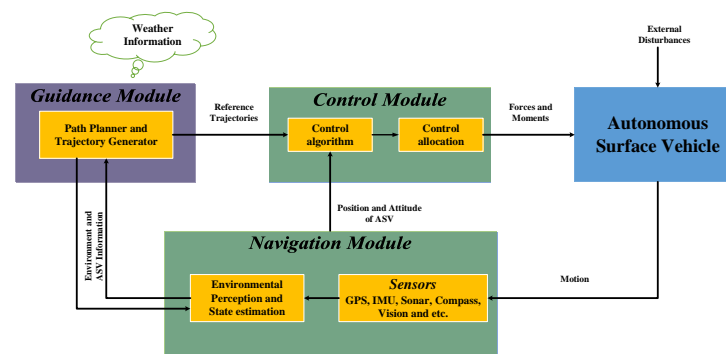


Figure 1. Guidance, navigation and control (GNC) block diagram.

Trajectory following for surface vehicles is challenging, owing to the complexity and nonlinearity of vehicle models and various environmental disturbances such as wind, waves, and currents [12]. To this end, researchers have employed various control strategies. One of the most widely used controllers is the proportional, integral, and derivative (PID) controller [13–16], but it lacks robustness and is highly dependent on the accuracy of the system dynamics [17]. Model predictive control (MPC) was applied in [18,19], which is sufficient for handling system constraints; but it is computationally intensive when the prediction horizon increases, making practical implementation difficult [12]. Other control methods that have been applied include linear quadratic regulator (LQR) [20–22], feedback linearization [21,23], backstepping control [24,25], fuzzy logic and its variants [26], neural networks [27], adaptive control [28] and other well-known control algorithm commonly used for trajectory-following applications is sliding mode controller [24].

Sliding mode control (SMC) is a robust control strategy that ensures insensitivity to matched uncertainties in line with the control input. This allows the straightforward

implementation of a robust controller, provided the uncertainties are bounded [29,30]. The fundamental principle of the SMC is to select a sliding surface within the state space in which the system has the desired characteristics. Subsequently, a controller can be designed to force the system trajectories from any initial condition within the state space to the sliding surface within finite time. Additionally, the trajectories can be maintained on the surface afterwards. In essence, SMC comprises two parts: equivalent control responsible for bringing the state trajectories from any initial condition to the sliding manifold, and discontinuous control that keeps the trajectories on the surface after they are brought to the manifold. Thus, the system is insensitive to model uncertainty and bounded exogenous disturbances [29].

However, SMC does not preserve the theoretical sliding motion formulation; rather, it switches to an infinitely high frequency for the sliding manifold. This high-frequency switching is not practically feasible due to the limitations of the actuator bandwidth. Hence, an ideal sliding mode cannot be achieved. Additionally, high-frequency switching of the control signal results in a phenomenon known as the chattering effect. Chattering can result in wear and tear of mechanical actuators as well as high energy consumption. The possible reasons for chattering, according to [30,31], are: (i) even though high-frequency switching is achievable, the existence of “parasitic dynamics”, that is, unmodeled actuator and sensor dynamics, in series with the system could result in a small oscillation around the sliding manifold, and (ii) non-ideal sliding motion causes a high-frequency oscillation.

Continuous-time second-order SMCs have been proposed to alleviate the chattering problem [32,33]. However, when continuous-time SMCs are discretized using a sample and held for digital implementation, they may lose their robustness, leading to instability. To address this limitation of continuous-time sliding mode controllers, researchers introduced discrete-time sliding mode controllers (DSMC). Moreover, implementing a DSMC is more straightforward owing to the advancement and widespread availability of computers and microcontrollers. However, the robustness of DSMC is undermined owing to the finite sampling time. In DSMC, the state trajectories converge to a bounded region near the sliding surface called the ultimate band. They exhibit a zigzag motion, termed quasi-sliding mode (QSM). Furthermore, unlike continuous-time SMC, the width of the ultimate band of a DSMC can be determined, which indicates the robustness of the controller. As noted in [34], given the desired ultimate band, the desired controller parameters can be determined; however, the limitations of the controller must be considered. Hence, by manipulating this band, the insensitivity of DSMC to uncertainties can be varied. To increase the robustness and mitigate the chattering effect of conventional DSMC, a second-order DSMC was proposed in [35,36]; however, the controller gains are not optimal.

Although there are a number of research works on DSMC, particularly based on the reaching law [37], its application to USVs has received less attention. This could be because of the difficulty in finding a discretized model of the vehicle [38]. In Ref. [39], a cascaded conventional DSMC and PI-based gain-scheduling controller were used for a straight line following ASV. Different numerical tests were conducted with and without ocean current disturbances. Although the results were satisfactory, the rudder experienced significant chattering.

Therefore, in this study, we propose a discrete-time super-twisting sliding mode controller [35], which is a robust second-order SMC, to address chattering. To the best of our knowledge, DSTA has not been applied to trajectory following for USVs. Additionally, owing to the complexity of obtaining an exact discrete-time model of the vehicle, we assumed certain hydrodynamic parameters and external disturbances, such as ocean currents, waves, and wind, as unknown perturbation. Hence, we applied a time delay estimator (TDE) to estimate the unknown system dynamics.

The TDE estimates the unknown system dynamics based on the past states of the system and control input. It was first introduced by [40] to estimate disturbances in robotic

manipulators, and was later applied to underwater control [41,42]. Then, based on this estimation, the controller cancels the uncertainties.

In many previous studies, the effectiveness of the time delay estimator has been experimentally verified in the presence of model uncertainties and external disturbances [36,41,42]. The purpose of the time delay estimator is to assess perturbation by utilizing data from a sufficiently short time ago, aiming to enhance control performance in terms of robustness and control energy. As will be detailed by using equations, the time delay estimator operates under the assumption that the perturbation does not change drastically within one sampling period, even though it dynamically evolves. In our research, the sampling period is 0.01 s, and the derivative of the perturbation remains fairly constant for this duration. This assumption is reflected in the TDE formulation, which can be regarded as the first-order approximation of the Taylor series expansion of the disturbance.

If the time delay estimator is not utilized, higher bounds on uncertainties should be assumed for robust control, leading to an increased control signal. A higher control signal, in turn, indicates higher control energy. Moreover, if the perturbation is not estimated, the controller becomes computationally intensive due to the need to adapt a large number of uncertain parameters. This restricts the real-time use of the control algorithm due to the limited onboard computational resources [43]. These limitations can be addressed using a time delay estimator, which provides information about perturbation to improve the robustness and reduce the control signal [36,41].

Furthermore, the TDE minimizes the time delay, which affects the performance of the sliding mode controller [31]. Finally, we apply a genetic algorithm (GA) to determine the optimal gains of the proposed controller. A quadratic cost function, which is the sum of the tracking error and control energy, was selected. The block diagram of the proposed method is shown in Figure 2. The main contributions of this study are summarized as follows.

1. A robust DSTA has been applied for trajectory following of an ASV. Moreover, a TDE algorithm has been used to estimate the unknown dynamics of the system, which in turn, improves the robustness of the proposed controller.
2. A GA is used to tune the controller parameters based on a quadratic objective function, which is the sum of the tracking error and control energy.
3. A linear matrix inequality (LMI) based Lyapunov approach was employed to validate the stability of the closed-loop system.
4. Finally, simulation results are presented for two scenarios, with and without the presence of exogenous disturbance. Moreover, the performance of the proposed GA based DSTA is compared with that of DSTA and a reaching law based DSMC.

The remainder of this paper is structured as follows. Section 2 describes the mathematical model of the vehicle from which the discretized model is derived using the integral approximation method. Section 3 describes the design of the proposed DSTA controller and the stability of the closed-loop system. Tuning of the parameters of the proposed controller using the GA is presented in Section 4. In Section 5, numerical results and discussions are presented, and finally, Section 6 presents the conclusions.

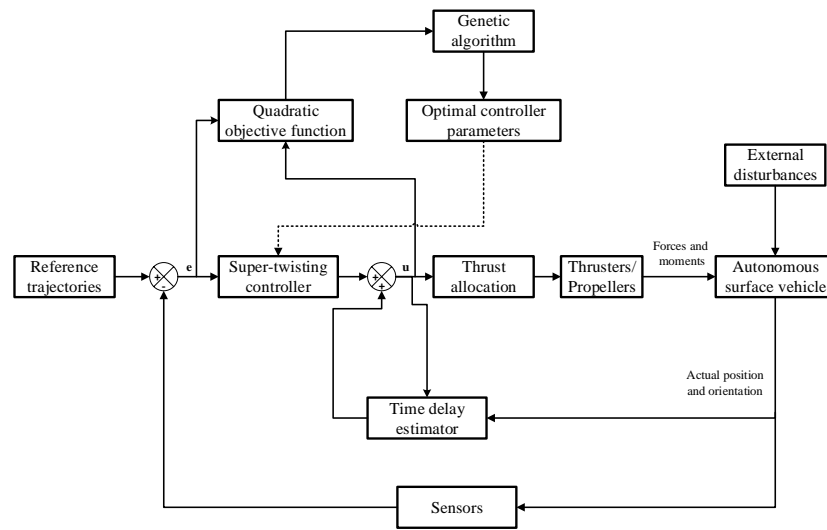


Figure 2. Block diagram of the proposed control method.

2. Mathematical Model

2.1. Continuous-Time Vehicle Model

Generally, the motion of ASVs is described using two coordinate frames of reference [11]: the earth-fixed or inertial frame denoted by $\{O_n\} = \{x_n, y_n, z_n\}$, and the body-fixed frame represented by $\{O_b\} = \{x_b, y_b, z_b\}$, as shown in Figure 3. Here, only the planar motion of the vehicle was considered. It was assumed that the vehicle had a homogeneous mass distribution and xy-plane symmetry.

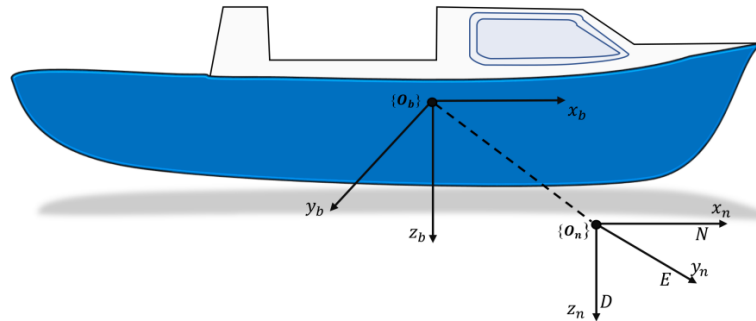


Figure 3. Illustration of body-fixed and earth-fixed reference frames of a surface vehicle.

The motion of the vehicle in the inertial frame of reference is expressed using position and orientation vector $\eta = [x, y, \psi]^T$, whereas its body-fixed velocity is represented by the velocity vector $v = [u, v, r]^T$. Here, $x(m)$ and $y(m)$ are the positions along x_n and y_n , respectively; $\psi(rad)$ is the yaw angle. Additionally, $u(m/s)$, $v(m/s)$ are the surge and sway velocities along x_b and y_b , respectively, $r(rad/s)$ is the yaw rate about z_b . Therefore, the kinematic model of the vehicle is given by:

$$\dot{\eta} = J(\psi)v, \quad (1)$$

where $J(\psi) \in \mathbb{R}^{3 \times 3}$ is the transformation matrix between the body-fixed frame and earth-fixed frame, which is:

$$J(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2)$$

and its dynamic model is given by:

$$M\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v} = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{ext}}, \quad (3)$$

where $\boldsymbol{M} = (\boldsymbol{M}_{\text{RB}} + \boldsymbol{M}_{\text{A}}) \in \mathbb{R}^{3 \times 3}$ denotes the sum of the rigid body mass and hydrodynamic added mass; $\boldsymbol{C}(\boldsymbol{v}) = (\boldsymbol{C}_{\text{RB}} + \boldsymbol{C}_{\text{A}}) \in \mathbb{R}^{3 \times 3}$ is a Coriolis-centripetal matrix due to rigid body and added mass; $\boldsymbol{D}(\boldsymbol{v}) = (\boldsymbol{D}_{\text{L}} + \boldsymbol{D}_{\text{NL}}) \in \mathbb{R}^{3 \times 3}$ is the sum of linear and nonlinear damping matrices given as follows:

$$\boldsymbol{M} = \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & m_{23} \\ 0 & m_{32} & m_{33} \end{bmatrix}, \quad (4a)$$

$$\boldsymbol{C}(\boldsymbol{v}) = \begin{bmatrix} 0 & 0 & c_{13} \\ 0 & 0 & c_{23} \\ c_{31} & c_{32} & 0 \end{bmatrix}, \quad (4b)$$

$$\boldsymbol{D}(\boldsymbol{v}) = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & d_{23} \\ 0 & d_{32} & d_{33} \end{bmatrix}, \quad (4c)$$

where $m_{11} = m - X_{\dot{u}}$, $m_{22} = m - Y_{\dot{v}}$, $m_{23} = mx_g - Y_{\dot{r}}$, $m_{32} = mx_g - N_{\dot{v}}$, $m_{33} = I_z - N_{\dot{r}}$; $c_{13} = -c_{31} = Y_{\dot{v}v}$, $c_{32} = -c_{23} = X_{\dot{u}}$; $d_{11} = -X_u - X_{|u|u}$, $d_{22} = -Y_v - Y_{|v|v}$, $d_{23} = -Y_r - Y_{|r|r}$, $d_{31} = -N_v - N_{|v|v}$, $d_{33} = -N_r - N_{|r|r}$. Here, m is the mass of the vehicle, I_z is the moment of inertia about z_b , and x_g is the center of gravity. $X_{\dot{u}}$, $Y_{\dot{v}}$, $Y_{\dot{r}}$, $N_{\dot{v}}$, and $N_{\dot{r}}$ are the hydrodynamic added-mass coefficients. X_u , Y_v , Y_r , N_v , and N_r are constant linear damping coefficients; and, $X_{|u|u}$, $Y_{|v|v}$, $Y_{|r|r}$, $N_{|v|v}$, and $N_{|r|r}$ are constant nonlinear damping coefficients. We assume fore-aft symmetry such that $Y_{\dot{r}} \approx N_{\dot{v}}$. Additionally, for low-speed applications, $N_v = Y_r$. $\boldsymbol{\tau} = [X, Y, N]^T \in \mathbb{R}^3$ denotes the forces and moments generated by the port thruster, starboard thruster, and bow thruster, respectively. $\boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^3$ denote forces and moments due to external disturbances [44]. The response of the thrusters is assumed to be faster than the dynamics of the vehicle; therefore, the dynamics of the thrusters is neglected.

2.2. Discrete-Time Vehicle Model

A discrete-time controller can be obtained either by discretizing a continuous-time controller designed using a continuous-time system model or by directly designing a discrete-time controller from the discretized system model. In this study, we followed the latter design approach.

Assumption 1. Assume that only the rigid body mass (R_{RB}) and linear damping (D_{L}) of the vehicle are known. The other parameters are unknown, and will be estimated using the TDE. This assumption also helps to decouple the sway and yaw dynamics.

Thus, based on Assumption 1 the dynamic model of the vehicle given by Equation (1) can be rewritten as follows:

$$\boldsymbol{M}_{\text{RB}}\dot{\boldsymbol{v}}(t) + \boldsymbol{D}_{\text{L}}(\boldsymbol{v}(t))\boldsymbol{v}(t) + \boldsymbol{f}(t) = \boldsymbol{\tau}(t), \quad (5)$$

where $\boldsymbol{f}(t) = \boldsymbol{M}_{\text{A}}\dot{\boldsymbol{v}}(t) + \boldsymbol{C}(\boldsymbol{v}(t))\boldsymbol{v}(t) + \boldsymbol{D}_{\text{NL}}(\boldsymbol{v}(t))\boldsymbol{v}(t) - \boldsymbol{\tau}_{\text{ext}}(t)$ is unknown or perturbation term to be estimated.

Assumption 2. The state variables and control input are assumed to be constant during the sampling time period T , that is, $\boldsymbol{v}(t)$ and $\boldsymbol{\tau}(t)$ remain constant [45].

Using Assumption 2, we compute a discrete-time model of the vehicle. By integrating Equation (5) over the sampling interval kT into $(k + 1)T$, we obtain:

$$\int_{kT}^{(k+1)T} \mathbf{M}_{\text{RB}} \dot{\mathbf{v}}(t) dt + \int_{kT}^{(k+1)T} \mathbf{D}_{\text{L}}(\mathbf{v}(t))\mathbf{v}(t) dt + \int_{kT}^{(k+1)T} \mathbf{f}(t) dt = \int_{kT}^{(k+1)T} \boldsymbol{\tau}(t) dt,$$

$$\mathbf{M}_{\text{RB}}(\mathbf{v}((k + 1)T) - \mathbf{v}(kT)) + T\mathbf{D}_{\text{L}}\mathbf{v}(kT) + \int_{kT}^{(k+1)T} \mathbf{f}(t) dt = T\boldsymbol{\tau}(kT). \quad (6)$$

In Equation (6), $\mathbf{f}(t)$ contains the terms that cannot be integrated. After rearranging Equation (6), the following discretized expression is obtained:

$$\mathbf{v}(k + 1) = \mathbf{A}\mathbf{v}(k) + \mathbf{B}\boldsymbol{\tau}(k) + \boldsymbol{\Gamma}(k), \quad (7)$$

where $\mathbf{v}(k) = [u(k), v(k), r(k)]^T$; $\boldsymbol{\tau}(k) = [X_k, Y_k, N_k]^T$; the system matrix $\mathbf{A} = \mathbf{I} - T\mathbf{M}_{\text{RB}}\mathbf{D}_{\text{L}}$; an input matrix $\mathbf{B} = T\mathbf{M}_{\text{RB}}^{-1}$; $\boldsymbol{\Gamma}(k) = \mathbf{M}_{\text{RB}}^{-1} \int_{kT}^{(k+1)T} \mathbf{f}(t) dt$. $\boldsymbol{\Gamma}(k)$ denotes, as explained in Assumption 1, hydrodynamics and parameter uncertainties and the external disturbance. \mathbf{I} denotes the identity matrix of appropriate dimension.

Assumption 3. The perturbation term $\boldsymbol{\Gamma}(k) \in \mathbb{R}^3$ satisfies the Lipschitz continuity condition

$$\|\boldsymbol{\Gamma}(k) - \boldsymbol{\Gamma}(k - 1)\| \leq T\mathbf{L}_{\gamma},$$

where $\mathbf{L}_{\gamma} > 0$ is a Lipschitz constant and T is the sampling time [46].

3. Controller Design and Stability Analysis

In this section, the proposed DSTA controller design and stability analysis using the Lyapunov approach are discussed.

3.1. Controller Design

The control problem involves designing a robust controller such that the vehicle follows a reference or desired trajectory while addressing the tracking error and chattering effect. In this study, the desired trajectories were defined in a body-fixed reference frame, that is, a body-fixed velocity trajectory following.

To design a SMC, the first step is to select the sliding surface to which the system trajectories are forced to converge. Hence, let the desired trajectories be $\mathbf{v}_d(k) = [u_d(k), v_d(k), r_d(k)]^T$. The desired trajectories are generated by the guidance system. Also, let the actual trajectories be $\mathbf{v}(k) = [u(k), v(k), r(k)]^T$ provided by the navigation system. Thus, the trajectory error is the difference between the desired and the actual trajectories, that is, $\mathbf{e}(k) = \mathbf{v}_d(k) - \mathbf{v}(k)$. The sliding surface was selected as follows:

$$\mathbf{s}(k) = \mathbf{K}_p \mathbf{e}(k), \quad (8)$$

where $\mathbf{s}(k) \in \mathbb{R}^3$; $\mathbf{K}_p \in \mathbb{R}^{3 \times 3}$ is positive scalar diagonal matrix.

For the trajectories to converge to the sliding surface, the sliding function must satisfy $\mathbf{s}(k + 1) = \mathbf{s}(k) = 0$ after some time instant k ; however, this does not occur in the real world, instead, the system trajectories are confined to a band near the sliding manifold, that is, QSM. For convergence of the system trajectories, the sliding function must satisfy $\|\mathbf{s}(k + 1)\| < \|\mathbf{s}(k)\|$. We compute $\mathbf{s}(k + 1)$ from Equation (8) as follows:

$$\begin{aligned} \mathbf{s}(k + 1) &= \mathbf{K}_p \mathbf{e}(k + 1) \\ &= \mathbf{K}_p \{\mathbf{v}_d(k + 1) - \mathbf{v}(k + 1)\} \end{aligned} \quad (9)$$

Now, by substituting Equation (7) into Equation (9), we obtain

$$s(k+1) = \mathbf{K}_p \{v_d(k+1) - \mathbf{A}v(k) - \mathbf{B}\tau(k) - \Gamma(k)\}. \quad (10)$$

The DSTA is given by [35]

$$\begin{aligned} s(k+1) &= \mathbf{K}_1 s(k) - T\Omega_1 |s(k)|^{1/2} \text{sgn}(s(k)) + T\zeta(k) \\ \zeta(k+1) &= \mathbf{K}_2 \zeta(k) - T\Omega_2 \text{sgn}(s(k)), \end{aligned} \quad (11)$$

where $\mathbf{K}_i \in \mathbb{R}^{3 \times 3}$, $0 < (\text{diagonal elements of } \mathbf{K}_i) < 1$, and $\Omega_i \in \mathbb{R}^{3 \times 3}$ for $i = 1, 2$ are diagonal matrices to be designed to ensure the convergence of DSTA. T is the sampling time period, and $\text{sgn}(s(k))$ is a signum function defined as

$$\text{sgn}(\gamma) = \begin{cases} 0, & \text{for } \gamma = 0 \\ \frac{\gamma}{|\gamma|}, & \text{for } \gamma \neq 0. \end{cases}$$

where $\gamma \in \mathbb{R}$. By equating Equation (10) with Equation (11), the control law becomes:

$$\begin{aligned} \tau(k) &= (\mathbf{K}_p \mathbf{B})^{-1} \left\{ \mathbf{K}_p v_d(k+1) - \mathbf{K}_p \mathbf{A}v(k) - \mathbf{K}_p \Gamma(k) + \right. \\ &\quad \left. - \mathbf{K}_1 s(k) + T\Omega_1 |s(k)|^{1/2} \text{sgn}(s(k)) - T\zeta(k) \right\}, \end{aligned} \quad (12)$$

where $(\mathbf{K}_p \mathbf{B})^{-1}$ must exist. The control law given by Equation (12) cannot be realized, because the perturbation term $\Gamma(k)$ is unknown. From Assumption 3, the perturbation term is bounded. Hence, we employ the TDE method to estimate the perturbation term such that $\hat{\Gamma}(k)$ is used instead of $\Gamma(k)$. The control law can be rewritten as follows:

$$\begin{aligned} \tau(k) &= (\mathbf{K}_p \mathbf{B})^{-1} \left\{ \mathbf{K}_p v_d(k+1) - \mathbf{K}_p \mathbf{A}v(k) - \mathbf{K}_p \hat{\Gamma}(k) - \mathbf{K}_1 s(k) \right. \\ &\quad \left. + T\Omega_1 |s(k)|^{1/2} \text{sgn}(s(k)) - T\zeta(k) \right\}. \end{aligned} \quad (13)$$

The perturbation term is estimated based on the values of the past state variables of the system and past control inputs, that is, the past dynamics of the system. Mathematically, this can be expressed as $\Gamma(k) = \hat{\Gamma}(k) = \Gamma(k-1)$. The estimation of the perturbation term $\hat{\Gamma}(k)$ can be obtained from Equation (7) as follows:

$$\hat{\Gamma}(k) = \Gamma(k-1) = v(k) - \mathbf{A}v(k-1) - \mathbf{B}\tau(k-1). \quad (14)$$

The estimation in Equation (14) is based on the assumption that the perturbation term does not vary significantly during the sampling time. However, if the perturbation term changes rapidly during sampling time, the estimation error may be large [45]. Hence, the estimation can be further improved if the delayed perturbation term $\Gamma(k-1)$ and differential of $\Gamma(k-1)$, which is $\Delta\Gamma(k-1)$, are used to estimate the perturbation term. Now, we get:

$$\begin{aligned} \hat{\Gamma}(k) &= \Gamma(k-1) + \Delta\Gamma(k-1) \\ &= 2\Gamma(k-1) - \Gamma(k-2). \end{aligned} \quad (15)$$

After substituting Equation (14) into Equation (15), the estimated perturbation term becomes

$$\begin{aligned} \hat{\Gamma}(k) &= 2v(k) - v(k-1) - 2\mathbf{A}v(k-1) + \mathbf{A}v(k-2) \\ &\quad - 2\mathbf{B}\tau(k-1) + \mathbf{B}\tau(k-2). \end{aligned} \quad (16)$$

Therefore, by substituting the perturbation term Equation (16) into Equation (13), we obtain the following control law:

$$\begin{aligned} \tau(k) = & (\mathbf{K}_p \mathbf{B})^{-1} \left\{ \mathbf{K}_p v_d(k+1) - \mathbf{K}_p \mathbf{A} v(k) - 2\mathbf{K}_p v(k) + \mathbf{K}_p v(k-1) + 2\mathbf{K}_p \mathbf{A} v(k-1) \right. \\ & \left. - \mathbf{K}_p \mathbf{A} v(k-2) - \mathbf{K}_1 s(k) + T\mathbf{\Omega}_1 |s(k)|^{1/2} \text{sgn}(s(k)) - T\zeta(k) \right\} + 2\tau(k-1) \\ & - \tau(k-2). \end{aligned} \tag{17}$$

The control law given by Equation (17) can be implemented easily because the perturbation term is known. The delayed state and control input of the system can be retrieved from computer memory.

3.2. Stability Analysis

The stability of a closed-loop system is provided as follows [35,36,47]:

Theorem 1. Consider the dynamics of the surface vehicle given by Equation (2). If there exists a positive definite solution $\mathbf{L} = \mathbf{L}^T > 0$ to an LMI $\mathbf{\Psi}^T[\mathbf{L} + \mathbf{L}\mathbf{\Lambda}_1\mathbf{L} + \mathbf{L}\mathbf{\Lambda}_2\mathbf{L}]\mathbf{\Psi} - (1-r)\mathbf{L} \leq -\mathbf{\Psi}$, the conditions $\mathbf{\Omega}_{1i} < 0$, and $\mathbf{\Omega}_{2i} > 0$ are met. Therefore, the proposed DSTA controller given by Equation (17) ensures asymptotic convergence of the system trajectories to a ball B_r (that is, $B_r = \{s : \|s\|^2 < r_b\}$) centered at the origin. The ball has a radius of convergence.

$$r_b = \frac{\sigma}{1-r},$$

where $0 < r < 1$ is the ball radius, s is the sliding surface, and

$$\begin{aligned} \sigma &= \frac{\bar{q}_1^2}{4\zeta_1} + \zeta_2 \epsilon_i^2 + \bar{q}_2, \\ \bar{q}_1 &= q_1 + q_2 = T^2 \mathbf{\Omega}_{1i}^2 f_{11} + 2T^3 f_{12} \mathbf{\Omega}_{1i} \mathbf{\Omega}_{2i}, \\ \bar{q}_2 &= q_2 + q_3 = T^2 \mathbf{\Omega}_{1i}^2 f_{11} + T^4 f_{22} \mathbf{\Omega}_{2i}, \\ \zeta_1 &= \lambda_{\min}(\mathbf{\Psi}), \\ \zeta_2 &= \lambda_{\max}(\mathbf{H}). \end{aligned}$$

Proof of Theorem 1. The closed-loop system dynamics can be obtained by substituting Equation (17) into Equation (10), and we obtain

$$\begin{aligned} s(k+1) &= \mathbf{K}_1 s(k) - T\mathbf{\Omega}_1 |s(k)|^{1/2} \text{sgn}(s(k)) + T\zeta(k) + \Delta\mathbf{\Gamma}(k) \\ \zeta(k+1) &= \mathbf{K}_2 \zeta(k) - T\mathbf{\Omega}_2 \text{sgn}(s(k)), \end{aligned} \tag{18}$$

where $\Delta\mathbf{\Gamma}(k) = \hat{\mathbf{\Gamma}}(k) - \mathbf{\Gamma}(k)$ denotes the estimation error. As shown in Equation (18), the estimation error affects the closed-loop dynamics of the system. In the following section, we verify the robustness of the proposed controller by assuming that the perturbation term satisfies the Lipschitz criterion, as explained in Assumption 3. Now, we rearrange Equation (18) using change of variable. Let $\rho(k) = T\zeta(k) + \Delta\mathbf{\Gamma}(k)$. From this, we obtain

$$\begin{aligned} s(k+1) &= \mathbf{K}_1 s(k) - T\mathbf{\Omega}_1 |s(k)|^{1/2} \text{sgn}(s(k)) + \rho(k) \\ \rho(k+1) &= \mathbf{K}_2 \rho(k) - T^2 \mathbf{\Omega}_2 \text{sgn}(s(k)) + \epsilon(k), \end{aligned} \tag{19}$$

where $\epsilon(k) = \Delta\mathbf{\Gamma}(k+1) - \mathbf{K}_2 \Delta\mathbf{\Gamma}(k)$. Without loss of generality, we can rewrite Equation (19) for $i = 1, 2, 3$ as:

$$\begin{aligned} s_i(k+1) &= K_{1i} s_i(k) - T\mathbf{\Omega}_{1i} |s_i(k)|^{1/2} \text{sgn}(s_i(k)) + \rho_i(k) \\ \rho_i(k+1) &= K_{2i} \rho_i - T^2 \mathbf{\Omega}_{2i} \text{sgn}(s_i(k)) + \epsilon_i(k). \end{aligned} \tag{20}$$

Equation (20) can be expressed using a compact matrix notation as follows:

$$\chi_i(k+1) = \mathbf{\Psi}(k)\chi_i(k) + \boldsymbol{\varphi}(k)\text{sgn}(s_i(k)) + \boldsymbol{\zeta}(k), \quad (21)$$

where $\chi_i(k) = [s_i(k), \rho_i(k)]^T$, and $\mathbf{\Psi}(k)$, $\boldsymbol{\varphi}(k)$, and $\boldsymbol{\zeta}(k)$ are:

$$\mathbf{\Psi}(k) = \begin{bmatrix} K_{1i} & 1 \\ 0 & K_{2i} \end{bmatrix}, \quad (22a)$$

$$\boldsymbol{\varphi}(k) = \begin{bmatrix} -T\Omega_{1i}\text{sgn}|s_i(k)|^{1/2} \\ -T^2\Omega_{2i} \end{bmatrix}, \quad (22b)$$

$$\boldsymbol{\zeta}(k) = \begin{bmatrix} 0 \\ \epsilon_i(k) \end{bmatrix}. \quad (22c)$$

We select a positive definite Lyapunov function $V(k)$ that satisfies $\lambda_{\min}(\mathbf{L})\|\chi_i(k)\|^2 \leq \|\chi_i(k)\|_{\mathbf{L}} \leq \lambda_{\max}(\mathbf{L})\|\chi_i(k)\|^2$. Here, λ_{\min} and λ_{\max} are the minimum and maximum eigenvalues of \mathbf{L} , respectively.

$$V_i(k) = \chi_i^T(k)\mathbf{L}\chi_i(k). \quad (23)$$

The difference in the Lyapunov function is:

$$\Delta V_i = V_i(k+1) - V_i(k). \quad (24)$$

By substituting Equation (21) into Equation (24), we obtain

$$\begin{aligned} \Delta V_i &= V_i(k+1) - V_i(k) \\ &= \chi_i^T(k+1)\mathbf{L}\chi_i(k+1) - \chi_i^T(k)\mathbf{L}\chi_i(k), \end{aligned} \quad (25)$$

Substituting Equation (24) into Equation (25), we obtain

$$\begin{aligned} \Delta V_i(k) &= \chi_i^T(k)\mathbf{\Psi}^T\mathbf{L}\mathbf{\Psi}\chi_i(k) + \chi_i^T(k)\mathbf{\Psi}^T\mathbf{L}\boldsymbol{\varphi}\text{sgn}(s_i(k)) + \chi_i^T(k)\mathbf{\Psi}^T\mathbf{L}\boldsymbol{\zeta} + \boldsymbol{\varphi}^T\mathbf{L}\mathbf{\Psi}\chi_i(k)\text{sgn}(s_i(k)) \\ &+ \boldsymbol{\varphi}^T\mathbf{L}\boldsymbol{\varphi} + \boldsymbol{\varphi}^T\mathbf{L}\boldsymbol{\zeta}\text{sgn}(s_i(k)) + \boldsymbol{\zeta}^T\mathbf{L}\mathbf{\Psi}\chi_i(k) + \boldsymbol{\zeta}^T\mathbf{L}\boldsymbol{\varphi}\text{sgn}(s_i(k)) + \boldsymbol{\zeta}^T\mathbf{L}\boldsymbol{\zeta} - \chi_i^T(k)\mathbf{L}\chi_i(k). \end{aligned} \quad (26)$$

After collecting these terms, Equation (26) becomes

$$\begin{aligned} \Delta V_i(k) &= \chi_i^T(k)\left\{\mathbf{\Psi}^T\mathbf{L}\mathbf{\Psi} - \mathbf{L}\right\}\chi_i(k) + \boldsymbol{\varphi}^T\mathbf{L}\boldsymbol{\varphi} + \boldsymbol{\zeta}^T\mathbf{L}\boldsymbol{\zeta} + 2\chi_i^T(k)\mathbf{\Psi}^T\mathbf{L}\boldsymbol{\varphi}(k)\text{sgn}(s_i(k)) \\ &+ 2\chi_i^T(k)\mathbf{\Psi}^T\mathbf{L}\boldsymbol{\zeta} + 2\boldsymbol{\varphi}^T\mathbf{L}\boldsymbol{\zeta}\text{sgn}(s_i(k)). \end{aligned} \quad (27)$$

Using the Λ -matrix inequality $\bar{\mathbf{A}}^T\bar{\mathbf{B}} + \bar{\mathbf{B}}^T\bar{\mathbf{A}} \leq \bar{\mathbf{A}}^T\Lambda\bar{\mathbf{A}} + \bar{\mathbf{B}}^T\Lambda^{-1}\bar{\mathbf{B}}$ [48], some of the expressions in Equation (27) are simplified as follows:

1. $2\chi_i^T(k)\mathbf{\Psi}^T\mathbf{L}\boldsymbol{\varphi}\text{sgn}(s_i(k)) \leq \chi_i^T(k)\left\{\mathbf{\Psi}^T\mathbf{L}\Lambda_1\mathbf{L}\mathbf{\Psi}\right\}\chi_i(k) + \boldsymbol{\varphi}^T\Lambda_1^{-1}\boldsymbol{\varphi}$.
2. $\chi_i^T(k)\mathbf{\Psi}^T\mathbf{L}\boldsymbol{\zeta} \leq \chi_i^T(k)\left\{\mathbf{\Psi}^T\mathbf{L}\Lambda_2\mathbf{L}\mathbf{\Psi}\right\}\chi_i(k) + \boldsymbol{\zeta}^T\Lambda_2^{-1}\boldsymbol{\zeta}$.
3. $\boldsymbol{\zeta}^T\mathbf{L}\boldsymbol{\zeta}\text{sgn}(s_i(k)) \leq \boldsymbol{\varphi}^T\mathbf{L}\Lambda_3\mathbf{L}\boldsymbol{\varphi} + \boldsymbol{\zeta}^T\Lambda_3^{-1}\boldsymbol{\zeta}$.

Hence, Equation (27) can be rewritten as

$$\begin{aligned} \Delta V_i(k) &\leq \chi_i^T(k)\left\{\mathbf{\Psi}^T\mathbf{L}\mathbf{\Psi} + \mathbf{\Psi}^T\mathbf{L}\Lambda_1\mathbf{L}\mathbf{\Psi} + \mathbf{\Psi}^T\mathbf{L}\Lambda_2\mathbf{L}\mathbf{\Psi} - \mathbf{L}\right\}\chi_i(k) + \boldsymbol{\varphi}^T\left\{\mathbf{L} + \Lambda_1^{-1} + \mathbf{L}\Lambda_3\mathbf{L}\right\}\boldsymbol{\varphi} \\ &+ \boldsymbol{\zeta}^T\left\{\mathbf{L} + \Lambda_2^{-1} + \Lambda_3^{-1}\right\}\boldsymbol{\zeta}. \end{aligned} \quad (28)$$

By adding $rV_i(k)$ to both sides of Equation (28), we obtain

$$\begin{aligned} \Delta V_i(k) \leq & \chi_i^T(k) \left\{ \Psi^T \left\{ L + L\Lambda_1 L + L\Lambda_2 L - (1-r)L \right\} \Psi \right\} \chi_i(k) + \varphi^T \left\{ L + \Lambda_1^{-1} \right. \\ & \left. + L\Lambda_3 L \right\} \varphi + \xi^T \left\{ L + \Lambda_2^{-1} + \Lambda_3^{-1} \right\} \xi - rV_i(k), \end{aligned} \quad (29)$$

where $0 < r < 1$. Using LMI, if there exists a negative definite matrix Φ satisfying the following inequality:

$$\Psi^T \left\{ L + L\Lambda_1 L + L\Lambda_2 L(1-r)L \right\} \Psi \leq -\Phi,$$

$\Delta V_i(k+1)$ can be rewritten as

$$\Delta V_i(k) \leq \chi_i^T(k) \Phi \chi_i(k) + \varphi^T \left\{ L + \Lambda_1^{-1} + L\Lambda_3 L \right\} \varphi + \xi^T \left\{ L + \Lambda_2^{-1} + \Lambda_3^{-1} \right\} \xi - rV_i(k). \quad (30)$$

$$\Delta V_i(k) \leq -\chi_i^T(k) \Phi \chi_i(k) + \varphi^T F \varphi + \xi^T H \xi - rV_i(k), \quad (31)$$

where $F = L + \Lambda_1^{-1} + L\Lambda_3 L$, and $H = L + \Lambda_2^{-1} + \Lambda_3^{-1}$.

$$\Delta V_i(k) \leq -\chi_i^T(k) \Phi \chi_i(k) + \varphi^T F \varphi + \xi^T H \xi - rV_i(k), \quad (32)$$

However, $\varphi^T F \varphi$ can be simplified as follows [36]:

$$\varphi^T F \varphi = \bar{q}_1 |s_i(k)| + \bar{q}_2. \quad (33)$$

The simplification of Equation (33) is presented in Appendix A. For the positive definite matrices Ψ and H in Equation (31), we can apply the two-norm definition: $\lambda_{\min}(\Psi) \|\chi(k)\|^2 \leq \|\chi_i(k)\|_{\Psi} \leq \lambda_{\max}(\Psi) \|\chi_i(k)\|^2$, and $\lambda_{\min}(H) \|\xi(k)\|^2 \leq \|\xi(k)\|_H \leq \lambda_{\max}(H) \|\xi(k)\|^2$. Here, λ_{\min} and λ_{\max} are the minimum and maximum eigenvalues of the matrices, respectively. Thus, Equation (32) becomes

$$\begin{aligned} \Delta V_i(k) \leq & -\chi_i^T(k) \Phi \chi_i(k) + \varphi^T F \varphi + \xi^T H \xi - rV_i(k) \leq -\lambda_{\min}(\Psi) \|\chi_i(k)\|^2 + \bar{q}_1 |s_i(k)| + \bar{q}_2 \\ & + \lambda_{\max}(H) \|\xi(k)\|^2 - rV_i(k) \leq -\lambda_{\min}(\Psi) \|\chi_i(k)\|^2 + \bar{q}_1 |s_i(k)| + \bar{q}_2 \\ & + \lambda_{\max}(H) \epsilon_1^2 - rV_i(k). \end{aligned} \quad (34)$$

Let $\zeta_1 = \lambda_{\min}(\Psi) > 0$, and $\zeta_2 = \lambda_{\max}(H)$. $|s_i(k)| \leq \|\chi_i(k)\|$, Equation (34) becomes

$$\begin{aligned} \Delta V_i(k) \leq & -\zeta_1 \|\chi_i(k)\|^2 + \bar{q}_1 |s_i(k)| + \bar{q}_2 \\ & + \zeta_2 \epsilon_1^2 - rV_i(k) \\ \leq & -\zeta_1 \|\chi_i(k)\|^2 + \bar{q}_1 \|\chi_i(k)\| + \zeta_2 \epsilon_1^2 \\ & + \bar{q}_2 - rV_i(k). \end{aligned} \quad (35)$$

The expression $\zeta_1 \|\chi_i(k)\|^2 + \bar{q}_1 \|\chi_i(k)\|$ in Equation (35) is simplified using the completing squares method as follows:

$$\begin{aligned} -\zeta_1 \|\chi_i(k)\|^2 + \bar{q}_1 \|\chi_i(k)\| &= -\zeta_1 \left\{ \|\chi_i(k)\|^2 - \frac{\bar{q}_1}{\zeta_1} \|\chi_i(k)\| \right\} \\ &= -\zeta_1 \left\{ \|\chi_i(k)\| - \frac{\bar{q}_1}{2\zeta_1} \right\}^2 + \frac{\bar{q}_1^2}{4\zeta_1}. \end{aligned} \quad (36)$$

By substituting Equation (36) into Equation (35), we obtain

$$\begin{aligned}
 \Delta V_i(k) &\leq -\zeta_1 \|\chi_i(k)\|^2 + \bar{q}_1 \|\chi_i(k)\| + \zeta_2 \epsilon_1^2 + \bar{q}_2 - rV_i(k) \\
 &\leq -\zeta_1 \left\{ \|\chi_i(k)\| - \frac{\bar{q}_1}{2\zeta_1} \right\}^2 + \frac{\bar{q}_1^2}{4\zeta_1} + \zeta_2 \epsilon_1^2 + \bar{q}_2 - rV_i(k) \\
 &\leq \frac{\bar{q}_1^2}{4\zeta_1} + \zeta_2 \epsilon_1^2 + \bar{q}_2 - rV_i \\
 &= \sigma - rV_i(k),
 \end{aligned} \tag{37}$$

where $\sigma = \frac{\bar{q}_1^2}{4\zeta_1} + \zeta_2 \epsilon_1^2 + \bar{q}_2$.

$$\begin{aligned}
 \Delta V_i(k) &\leq \sigma - rV_i \\
 V_i(k+1) - V_i(k) &\leq \sigma - rV_i(k) \\
 V_i(k+1) &\leq (1-r)V_i(k) + \sigma.
 \end{aligned} \tag{38}$$

Equation (38) is a linear, first-order, ordinary difference equation; thus, we can compute the solution by induction from the first few iterations given as follows:

$$V_i(1) \leq \sigma + (1-r)V_i(0), \tag{39a}$$

$$V_i(2) \leq \sigma + \sigma(1-r) + (1-r)^2 V_i(0), \tag{39b}$$

$$V_i(3) \leq \sigma + \sigma(1-r) + \sigma(1-r)^2 + \sigma(1-r)^3 V_i(0). \tag{39c}$$

Therefore, from the results in Equations (39a)–(39c), we can generalize the k th iteration as follows:

$$\begin{aligned}
 V_i(k) &\leq \sigma + \sigma(1-r)V_i(0) \\
 &\leq \sigma + \sigma(1-r) + \sigma(1-r)^2 + \dots + \sigma(1-r)^{k-1} + (1-r)^k V_i(0) \\
 &\leq (1-r)^k V_i(0) + \sum_{j=0}^{k-1} \sigma(1-r)^j.
 \end{aligned} \tag{40}$$

As k approaches infinity, $V_i(k)$ becomes:

$$\lim_{k \rightarrow \infty} V_i(k) \leq \frac{\sigma}{1-r}. \tag{41}$$

Hence, the radius of the ball is

$$r_b \leq \frac{\sigma}{1-r}.$$

This concludes the proof of Theorem 1. \square

4. Controller Gain Optimization Using Genetic Algorithm

In this section, the computation of the optimal gains of the proposed controller using GA is discussed. Setting controller gains, particularly for complex nonlinear systems, is difficult and requires experience. The system response can be made faster with higher controller gains; however, this leads to higher control signals, which in turn increase the energy consumption and actuator saturation. Thus, to address these problems, we applied a GA to tune the controller gains.

To determine optimal gains, an objective or cost function must first be established. In optimal control, the choice of cost function depends on the performance measures that the system must satisfy. These performance measures can be the error, time, and control energy, or a combination of them [49]. The use of the sum of tracking error and control energy,

or a combination of other performance metrics as a cost function, is a common approach and has been applied in real-world applications. Nevertheless, there is always a trade-off between the selections of the performance criteria. After selecting the cost function, a suitable optimization algorithm is chosen to find the optimal parameters of the controller based on the cost function.

In this study, GA was selected because it is convenient for discrete-time optimization [50]. GA is an evolutionary algorithm based on Darwin's theory of natural selection. Generally, it uses three operators: selection, crossover, and mutation. The operation of the GA commences by setting an initial random population comprising possible candidate solutions, and the fitness of the individuals is evaluated using the cost function. The selection operator then selects the fittest individuals, from which parents are selected for reproduction. Offsprings are produced by crossing the chromosomes of the parents. To increase the diversity of the next generation, a mutation operator is applied that prevents stagnation, that is, the similarity of the solution after several generations. This operation continues until the stopping criteria are satisfied [50,51]. The GA operation is summarized in the flowchart in Figure 4.

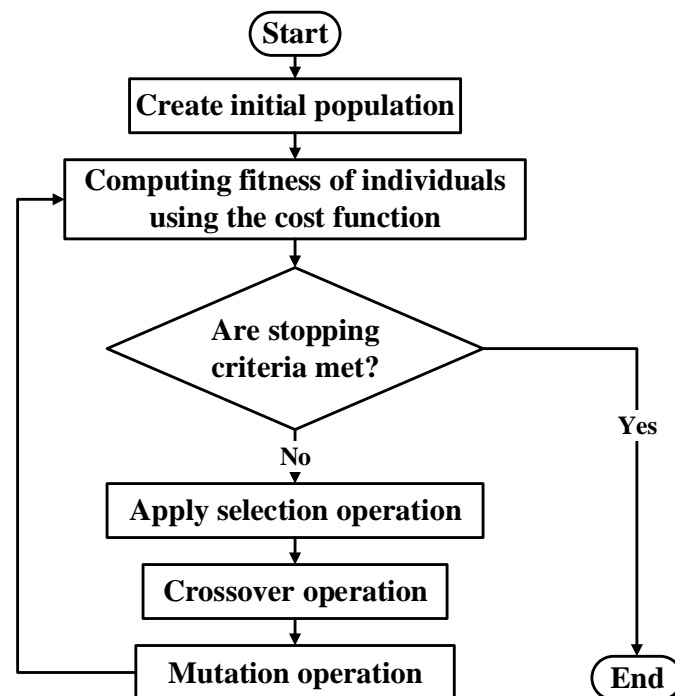


Figure 4. Flowchart of the Genetic algorithm for gain optimization.

The proposed method avoids real-time use of GA. If employed in real-time, the convergence time of GA significantly impacts control performance. Problems may arise if GA fails to converge within the desired time period corresponding to the control frequency. To address this, we employed GA to tune the controller parameters offline before real-time operation. This tuning is specifically for the nominal case, without external disturbances, and is done prior to the real-time operation of the controller.

Since ASVs can stay on the water surface for long periods of time, they should use energy efficiently apart from following the desired trajectory by minimizing the tracking error. Thus, we employ a quadratic cost function J_c , which is the sum of the tracking error and control energy, defined as follows:

$$J_c = \sum_{i=0}^{k-1} \left(\|e(i)\|_Q^2 + \|u(i)\|_R^2 \right), \quad (42)$$

where k is the number of time steps and $e(i)$ and $u(i)$ are the vectors of the tracking error and control signal at the i^{th} time instant, respectively. $Q \in \mathbb{R}^{3 \times 3}$ is a positive semi-definite matrix and $R \in \mathbb{R}^{3 \times 3}$ is a positive definite matrix. The cost function, as defined in Equation (42), incorporates weighting matrices Q and R for the tracking error index and control energy index, respectively. The values of these matrices depend on the size of the tracking error and the control energy, as mentioned in [52]. They are selected to strike a balance between tracking performance and control energy consumption. It should be noted the cost function is formulated as an unconstrained optimization problem.

The cost function given by Equation (42) is coded using MATLAB. The error and control signals are logged into the MATLAB workspace using the Simulink[®] model implementation of the vehicle. The cost function is computed from these signals. Then, the cost function and GA options were passed to the built-in MATLAB GA. Finally, the algorithm tunes the controller’s gains until the stopping criteria are reached, which in our case are the maximum number of generations and stall generations.

Figure 5 illustrates our analysis of GA convergence, showing plots of fitness, stopping criteria, and the average distance between individuals. The GA operates with a population size of 250, and its stopping criteria include a maximum of 15 generations and a stall generation of 7, as outlined in Table 1. Other GA settings are detailed in the same table. The fitness graph in Figure 5 demonstrates GA convergence to the ‘best’ solution in nearly 6 generations, terminating based on the maximum number of generations criterion. Moreover, the graph depicting the average distance between individuals gradually decreases as the generation advances, indicating the convergence of GA towards a proper solution.

Table 1. Parameter setting of genetic algorithm.

Parameter	Value
Population size	250
Maximum generation	15
stall generation	7
Crossover	Two point cross
Selection function	Roulette
Elite count	2
Crossover fraction	0.8
Mutation rate	0.01
Mutation function	Uniform
Migration direction	Forward

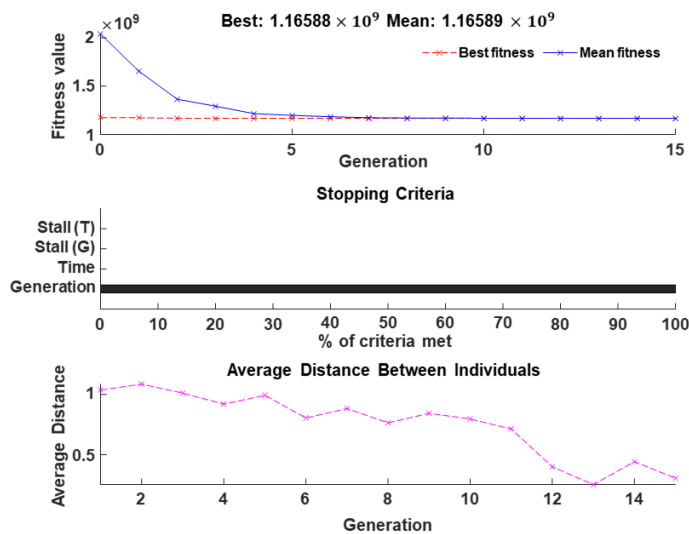


Figure 5. Illustration of convergence of genetic algorithm using fitness, stopping criteria, and average distance between individuals.

5. Simulation Results and Discussion

This section presents a verification of the performance of the proposed controller through numerical simulations. The simulation comprises two scenarios: (i) the performances of the proposed GA-based DSTA (hereafter, GA-DSTA) and DSTA controllers with TDE are compared. Additionally, to evaluate the performance of these controllers and chattering reduction, we used the commonly used conventional discrete-time SMC based on reaching law (DSMR) implemented in [45]. (ii) Since ASVs could be deployed in areas where the disturbance due to ocean current, wind, and ocean waves is significant, the robustness of the proposed controller will be tested in the presence of an external disturbance.

The reference trajectories used were body-fixed velocity trajectories. The physical parameters of the vehicle are obtained from the research vessel used in [44]. The vehicle has an overall length of 2.53 m and a 0.565 m beam, and is a scaled-down (1:17.1) model of a large research vessel. The vehicle parameters are listed in Table 2.

Table 2. Physical parameters of the ASV.

Parameter	Value	Unit	Parameter	Value	Unit
$X_{\dot{u}}$	−3.5	kg · m	$X_{ u u}$	−6.822	kg · s/m
$Y_{\dot{v}}$	−41.55	kg · m	$Y_{ v v}$	−71	kg · s/m
$N_{\dot{r}} = Y_{\dot{r}}$	−14	kg · m	$Y_{ r r}$	−35.52	kg/s
$N_{\dot{r}}$	−28	kg · m ²	$N_{ v v}$	20.6	kg/s
$X_{\dot{u}}$	−4.27	kg/s	$N_{ r r}$	−11.52	kg · m ² · s
$Y_{\dot{v}}$	−25.63	kg/s	m	125.37	kg
$Y_{\dot{r}} = N_{\dot{v}}$	−19.44	kg · m/s	x_g	0.232	m
$N_{\dot{r}}$	−32.65	kg · m ² · s	I_z	1.5	kg · m

kg: kilogram, s: second, and m: meter

5.1. Scenario 1: Nominal Case

In this case, the performance of the proposed controller without external disturbances is addressed. A piecewise body-fixed reference trajectory given in Equations (43)–(45) for surge, sway, and yaw was used to test the effectiveness of the proposed controller. Discontinuous trajectories are used to validate the controller's performance in worst-case scenarios, even though they may not be used in real-world scenarios. To assess the dynamic performance of the controller, i.e., how the controller reacts to abrupt changes in trajectory, abrupt changes have been considered. These trajectories were discretized using the sample-and-hold method. We assume that the maximum surge speed, sway speed, and yaw rate of the vehicle are 15 m/s, 2 m/s, and 1 rad/s, respectively. The initial speed of the vehicle is set to $[u_0, v_0, r_0]^T = [0 \text{ m/s}, 0 \text{ m/s}, 0 \text{ rad/s}]^T$. A sampling time has to be selected based on the update rate or frequency of the sensors and actuators. Thus, in this study, a sampling time of $T = 0.01 \text{ s}$ is chosen.

$$u_d = \begin{cases} 0.1t^2, & \text{for } 0 \leq t < 10 \\ -10 \sin(0.1t) + 5 \cos(0.1\pi t), & \text{for } 10 \leq t < 20 \\ 0.01t^2 - 2 \sin(0.05\pi t), & \text{for } 20 \leq t < 30 \\ 100e^{-\frac{t}{8}} \sin(0.8t), & \text{for } 30 \leq t < 40 \end{cases} \quad (43)$$

$$v_d = \begin{cases} 0, & \text{for } 0 \leq t < 10 \\ 2 \sin(0.05\pi t), & \text{for } 10 \leq t < 20 \\ -1.5, & \text{for } 20 \leq t < 30 \\ 0.8 \sin(0.5t), & \text{for } 30 \leq t < 40 \end{cases} \quad (44)$$

$$r_d = \begin{cases} \frac{\pi}{4}, & \text{for } 0 \leq t < 10 \\ 0.4, & \text{for } 10 \leq t < 20 \\ e^{-\frac{t}{8}} \sin(0.8t), & \text{for } 20 \leq t < 30 \\ \frac{\pi}{3}, & \text{for } 30 \leq t < 40 \end{cases} \quad (45)$$

The weighting matrices of the objective function are heuristically selected as $\mathbf{Q} = \text{diag}(1.5, 1.5, 1.5)$ and $\mathbf{R} = \text{diag}(0.01, 0.01, 0.01)$. The parameters of both the DSTA and the DSMR were also heuristically tuned. To ensure a fair comparison, we used a fairly similar values as well as to prevent higher control signals which causes control signal saturation and chattering effect. The parameters of the proposed GA-DSTA, DSTA, and DSMR controllers are listed in Table 3.

Table 3. GA-DSTA, DSTA, and DSMR controllers' gain setting.

Controller	Parameter	Value
GA-DSTA	K_p	diag(0.0023, 0.1006, 0.3481)
	K_1	diag(0.0081, 0.0145, 0.0037)
	K_2	diag(0.8750, 0.8628, 0.3212)
	Ω_1	diag(0.0009, 0.0008, 0.0006)
	Ω_2	diag(0.0009, 0.0003, 0.0008)
DSTA	K_p	diag(0.4, 0.4, 0.4)
	K_1	diag(0.5, 0.5, 0.5)
	K_2	diag(0.3, 0.3, 0.3)
	Ω_1	diag(0.02, 0.02, 0.02)
	Ω_2	diag(0.02, 0.02, 0.01)
DSMR	K_p	diag(0.8, 0.8, 0.8)
	q	diag(20, 20, 20)
	ϵ	diag(0.02, 0.02, 0.02)

To qualitatively evaluate the performance of the proposed controller, we used two performance indices: (i) the root mean square error (RMSE) was used to compute the tracking error of the three controllers for the surge, sway, and yaw rate; (ii) The Frobenius norm was used to obtain the total control energy expenditure of the three control algorithms. The RMSE and Frobenius norm are given by Equation (46) and Equation (47), respectively.

$$\star_{\text{RMSE}} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\star_d(k) - \star(k))^2}, \quad (46)$$

$$\|\mathbf{U}\|_F = \sqrt{\sum_{i=1}^M \sum_{j=1}^N |u_{ij}|^2}, \quad (47)$$

where \star represents one of the three degrees of freedom, N is the number of time steps, and M denotes the number of degrees of freedom (that is, three in this case). \mathbf{U} is a control signal matrix whose dimensions are the number of degrees of freedom by the number of time steps, and u_{ij} denotes each element of \mathbf{U} .

The simulation results of the three controllers for the nominal case of the surge, sway, and yaw rates are shown in Figures 6–8. As can be observed from the simulation results, the three controllers drive the vehicle along the desired trajectory successfully; however, the proposed GA-DSTA controller outperforms the other two controllers both in terms of overshoot and tracking error minimization.

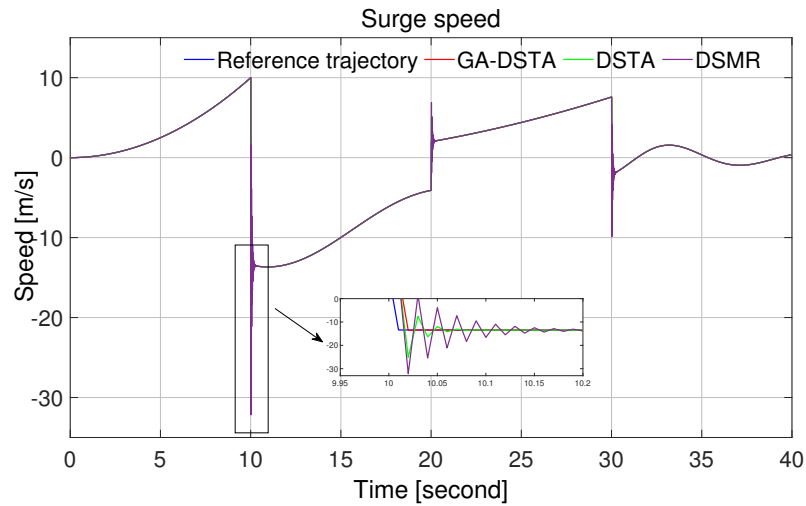


Figure 6. Surge speed trajectory following comparison of the GA-DSTA, DSTA, and DSMR controllers for Scenario 1.

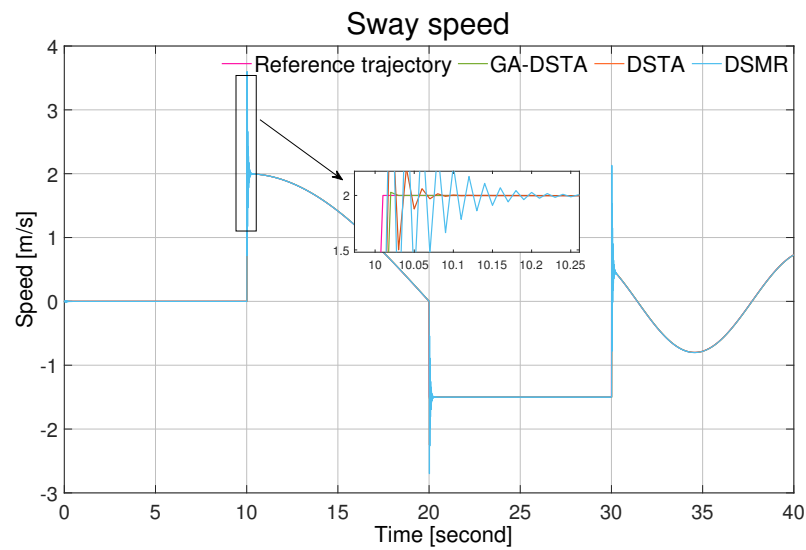


Figure 7. Sway speed trajectory following comparison of the GA-DSTA, DSTA, and DSMR controllers for Scenario 1.

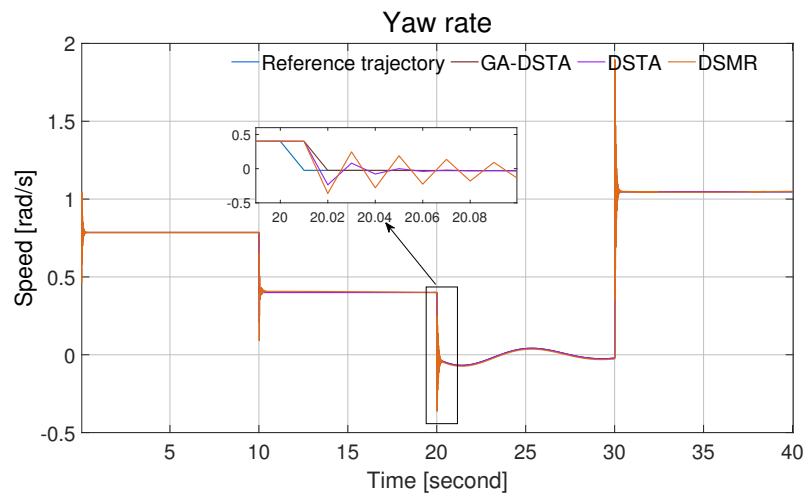


Figure 8. Yaw rate trajectory following comparison of the GA-DSTA, DSTA, and DSMR controllers for Scenario 1.

The RMSEs of the surge, sway, and yaw rate of the three controllers are listed in Table 4. It can be seen from this qualitative analysis that the proposed GA-DSTA controller yields a 13.5% and 40.01% reduction in RMSE compared to the DSTA and DSMR controllers, respectively, for the surge direction. Similarly, the percentage of reduction in RMSE was observed for the sway direction, except for the yaw rate of DSMR, which was approximately 42.56%.

Table 4. RMSEs of GA-DSTA, DSTA, and DSMR controllers for surge, sway, and yaw rate for Scenario 1.

GA-DSTA			DSTA			DSMR		
u_{RMSE}	v_{RMSE}	r_{RMSE}	u_{RMSE}	v_{RMSE}	r_{RMSE}	u_{RMSE}	v_{RMSE}	r_{RMSE}
0.4123	0.0508	0.0228	0.4761	0.0586	0.0264	0.6873	0.0845	0.0397

For the sake of brevity, the RMSEs of the three controllers are shown in the bar graph in Figure 9. As shown in the bar chart, the RMSE for the surge speed is higher than the RMSE for the sway speed and yaw rate for all the three control methods. This shows that when the speed increases, the tracking error increases as well.

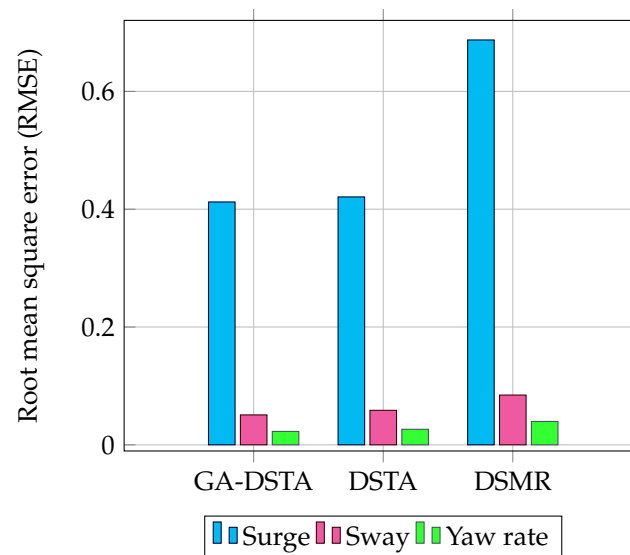


Figure 9. Comparison of RMSEs of GA-DSTA, DSTA, and DSMR controllers for surge, sway, and yaw rate for Scenario 1.

Additionally, a step change was included in the desired trajectories to test the dynamic performance of the proposed controller. The proposed controller reacts quickly to the change and tracks the desired trajectory effectively, although there is an insignificant overshoot, as shown in Figures 6–8. The conventional DSMR controller has a large overshoot compared to those of GA-DSTA and DSTA; however, it reacts immediately and settles almost within less than 0.2 s for the surge, sway, and yaw rate. This is due to the TDE algorithm, which estimates the unknown system dynamics so that the controller reacts quickly.

The expenditure of control energy of the proposed GA-DSTA controller including those for DSTA and DSMR are given in Table 5. The energy consumption of the proposed controller was lower than those of the other two controllers. It was found that there was an 41.78% reduction in the energy expenditure of the GA-DSTA controller compared with that of the DSTA controller. However, the energy consumption of the GA-DSTA controller was reduced to 66.4% of that of the DSMR controller. This is clearly shown in the bar graph in Figure 10.

Table 5. The total energy expenditure of GA-DSTA, DSTA, and DSMR controllers for surge, sway, and yaw rate for Scenario 1. RPM is revolutions per minute.

GA-DSTA	DSTA	DSMR
$\ U\ _F$ (RPM)	$\ U\ _F$ (RPM)	$\ U\ _F$ (RPM)
3.4419×10^5	5.9119×10^5	1.0244×10^6

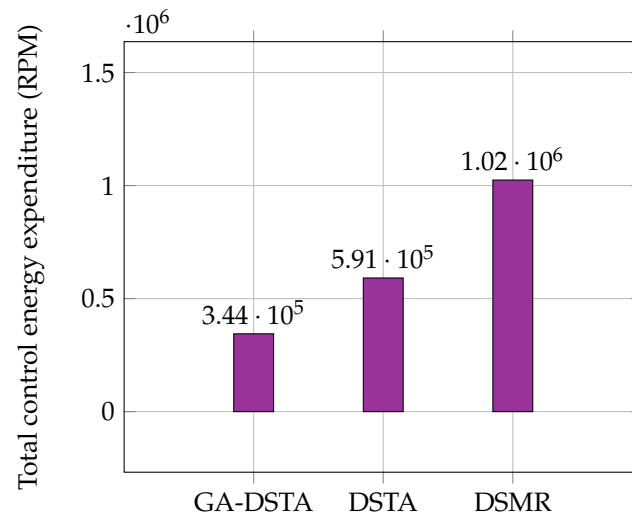


Figure 10. Comparison of the total energy expenditure of the GA–DSTA, DSTA, and DSMR controllers for Scenario 1. RPM is revolutions per minute.

The total energy consumption of the proposed method and the other two controllers are listed in Table 5. As can be seen from this energy-consumption table, the proposed method has the lowest energy consumption. The energy consumption is related to the chattering. If the chattering is significant, then the energy consumption is high as well. The GA-DSTA controller significantly minimizes the chattering compared to both DSTA and DSMR controllers, resulting in the lowest energy consumption. Additionally, the chattering by the DSTA controller is lower than that of the conventional DSMR controller, leading to lower energy consumption as well. Despite the small difference in tracking errors across the three controllers, the notable improvement in chattering by the GA-DSTA and DSTA controllers contributes to the overall reduction in energy consumption. The chattering effect in the control signals for Scenario 1 is shown in Figure 11.

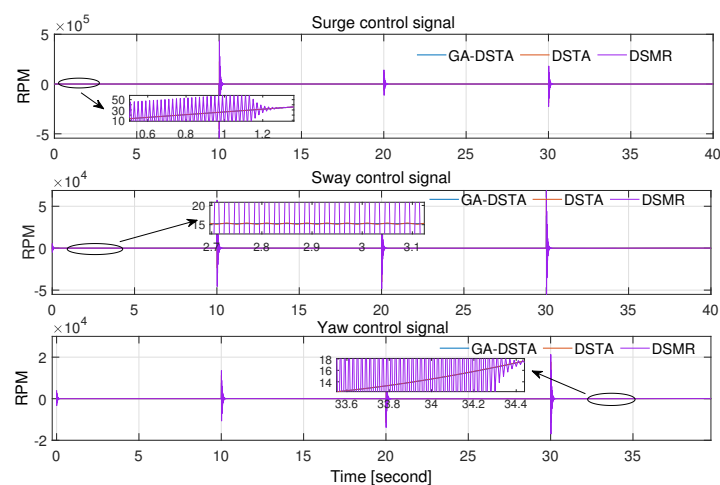


Figure 11. Control signals for surge, sway, and yaw rate of GA-DSTA, DSTA, and DSMR controllers for Scenario 1. RPM is revolutions per minute.

5.2. Scenario 2: External Disturbance

In the real world, the motion of the ASV is affected by external disturbances such as currents, wind, ocean waves, model uncertainties, and parametric variations. Thus, before deploying a control algorithm, its robustness against external disturbances must be tested. In this subsection, we present a simulation in the presence of external disturbances. Note that the external disturbances are assumed to be bounded and satisfy the Lipschitz condition given in Assumption 3. The external disturbances for the surge, sway, and yaw rates are given by Equations (48)–(50).

$$u_{\text{dis}} = \begin{cases} 0, & \text{for } 0 \leq t < 10 \\ 1.5 \sin(2\pi t) \cos(\pi t), & \text{for } 10 \leq t < 20 \\ 1.5 \text{sawtooth}(2\pi t), & \text{for } 20 \leq t < 30 \\ 0, & \text{for } 30 \leq t < 40 \end{cases} \quad (48)$$

$$v_{\text{dis}} = \begin{cases} 0, & \text{for } 0 \leq t < 10 \\ 0.5 \text{sawtooth}(0.5\pi t), & \text{for } 10 \leq t < 20 \\ 0.5 \sin(0.2\pi t) \cos(0.2\pi t), & \text{for } 20 \leq t < 30 \\ 0, & \text{for } 30 \leq t < 40 \end{cases} \quad (49)$$

$$r_{\text{dis}} = \begin{cases} 0.2 \sin(0.5\pi t) \cos(0.4t) \\ -0.1 \sin(0.2t), & \text{for } 0 \leq t < 10 \\ 0, & \text{for } 10 \leq t < 20 \\ 0.2 \text{sawtooth}(0.5\pi t), & \text{for } 20 \leq t < 30 \\ 0, & \text{for } 30 \leq t < 40 \end{cases} \quad (50)$$

where sawtooth is a MATLAB built-in sawtooth function. For the surge and sway directions, disturbances were added to the reference trajectory from $t = 10$ to $t = 30$, and disturbances for the yaw rate were added from $t = 0$ to $t = 10$ and $t = 20$ to $t = 30$. The maximum magnitudes of the disturbances for the surge u_{dis} , sway v_{dis} , and yaw rate r_{dis} were ± 1.5 m/s, ± 0.5 m/s, and ± 0.2 rad/s, respectively. The sampling time was the same as that in the nominal case.

The simulation results for the second scenario are shown in Figures 12–14 for the surge, sway, and yaw rates, respectively. Even in the presence of external disturbances, the proposed method, including the other two controllers, successfully followed the desired trajectory.

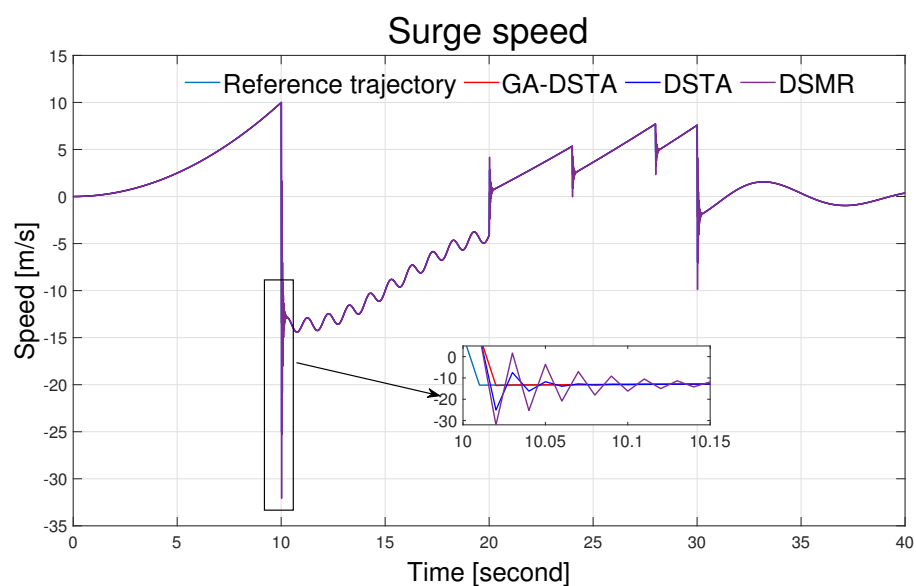


Figure 12. Surge speed trajectory following comparison of the GA-DSTA, DSTA, and DSMR controllers for Scenario 2.

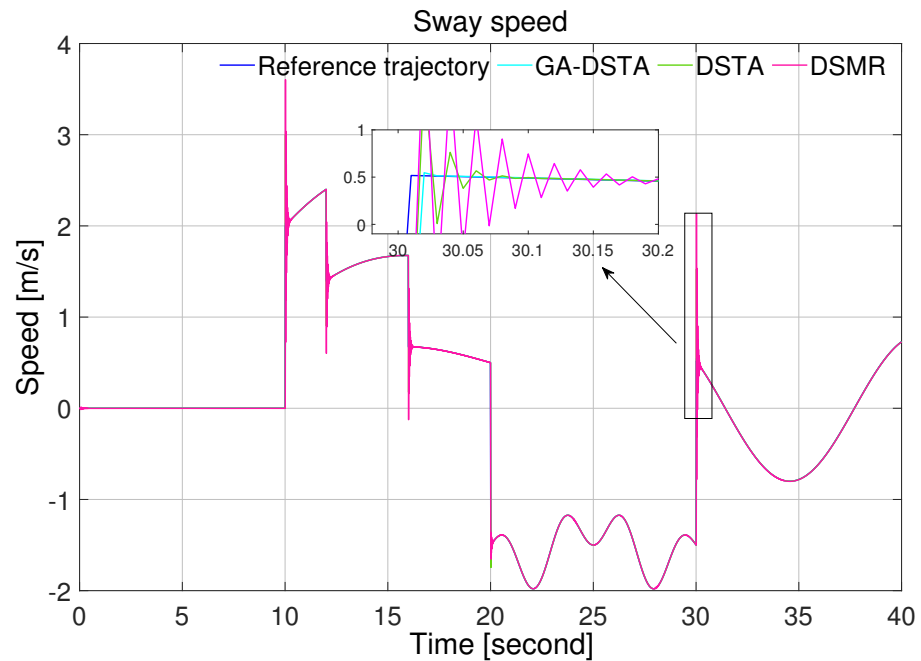


Figure 13. Sway speed trajectory following comparison of the GA-DSTA, DSTA, and DSMR controllers for Scenario 2.

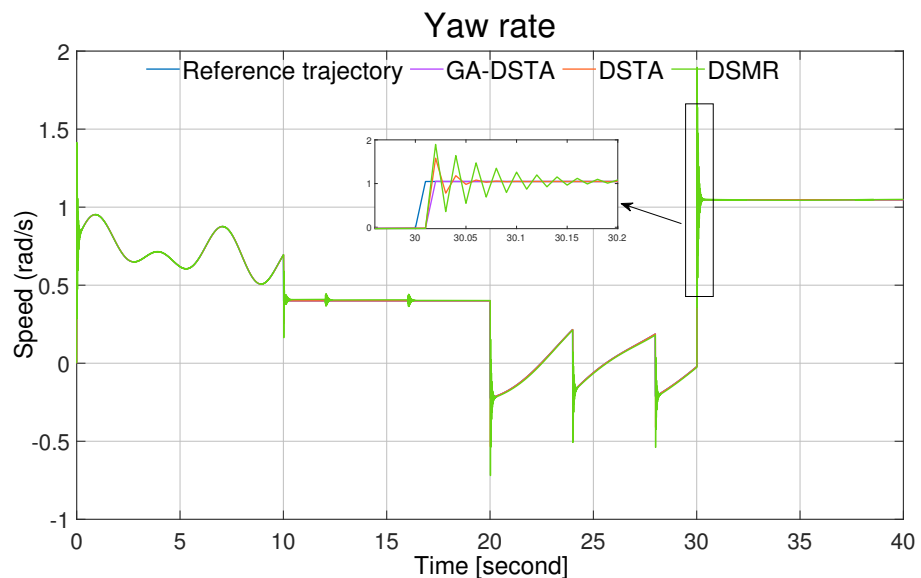
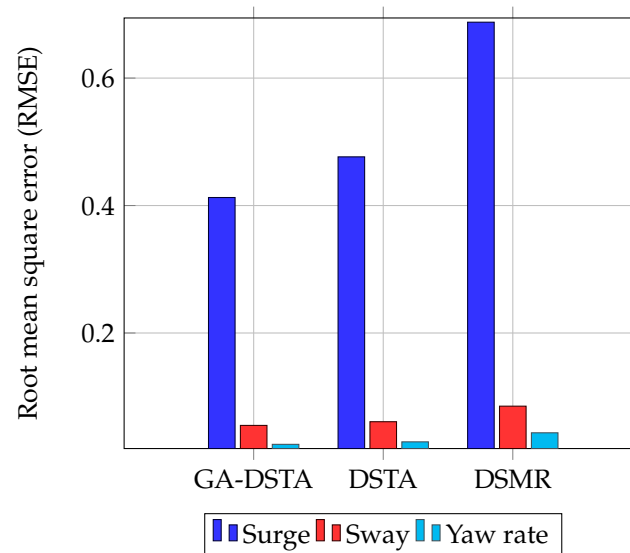


Figure 14. Yaw rate trajectory following comparison of the GA-DSTA, DSTA, and DSMR controllers for Scenario 2.

The RMSEs for the second scenario are given in Table 6. The percentages of RMSEs reduction of the proposed controller compared to the DSTA controller for the surge, sway direction, and yaw rate are approximately 13.39%, 9.046%, and 13.36%, respectively, and 40%, 35.44%, and 41.7% as compared to the DSMR, respectively. From this comparison, it can be inferred that the proposed controller outperforms the other two. The RMSEs are depicted in the bar chart in Figure 15.

Table 6. RMSEs of the GA-DSTA, DSTA, and DSMR controllers for the surge, sway, and yaw rate for Scenario 2.

GA-DSTA			DSTA			DSMR		
u_{RMSE}	v_{RMSE}	r_{RMSE}	u_{RMSE}	v_{RMSE}	r_{RMSE}	u_{RMSE}	v_{RMSE}	r_{RMSE}
0.4126	0.0550	0.0253	0.4764	0.0608	0.0292	0.6877	0.0852	0.0434

**Figure 15.** Comparison of the RMSEs of the GA-DSTA, DSTA, and DSMR controllers for the surge, sway, and yaw rate Scenario 2. RPM is revolutions per minute.

The robustness of the proposed controller to external disturbances relies on the gains of the controllers Ω_1 and Ω_2 . When the values of these gains are small, the chattering effect decreases, and the robustness of the controller degrades. The GA tunes these gains to smaller values based on a given objective function. This slightly could degrade the robustness of the proposed controller. However, the presence of the TDE algorithm prevents further performance degradation by estimating unknown system dynamics and external disturbances.

Regarding the control energy expenditure, the proposed controller utilized lower energy than the DSTA and DSMR controllers as shown in Table 7. There is a 41.6% reduction in the energy consumption of GA-DSTA compared to that in DSTA, and 66.3% compared to that in DSMR. The bar chart in Figure 16 shows the total energy consumed by the three controllers. The proposed GA-DSTA and DSTA controllers have less chattering compared to DSMR controller as shown in Figure 17.

Table 7. The total energy expenditure of the GA-DSTA, DSTA, and DSMR controllers for the surge, sway, and yaw rate for Scenario 2. RPM is revolutions per minute.

GA-DSTA	DSTA	DSMR
$\ U\ _F$ (RPM)	$\ U\ _F$ (RPM)	$\ U\ _F$ (RPM)
3.4524×10^5	5.9207×10^5	1.0250×10^6

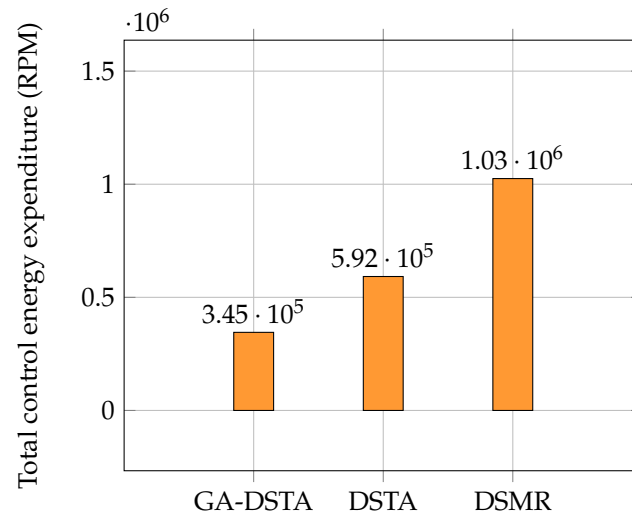


Figure 16. Comparison of total energy expenditure of the GA-DSTA, DSTA, and DSMR controllers in revolution for Scenario 2. RPM is revolutions per minute.

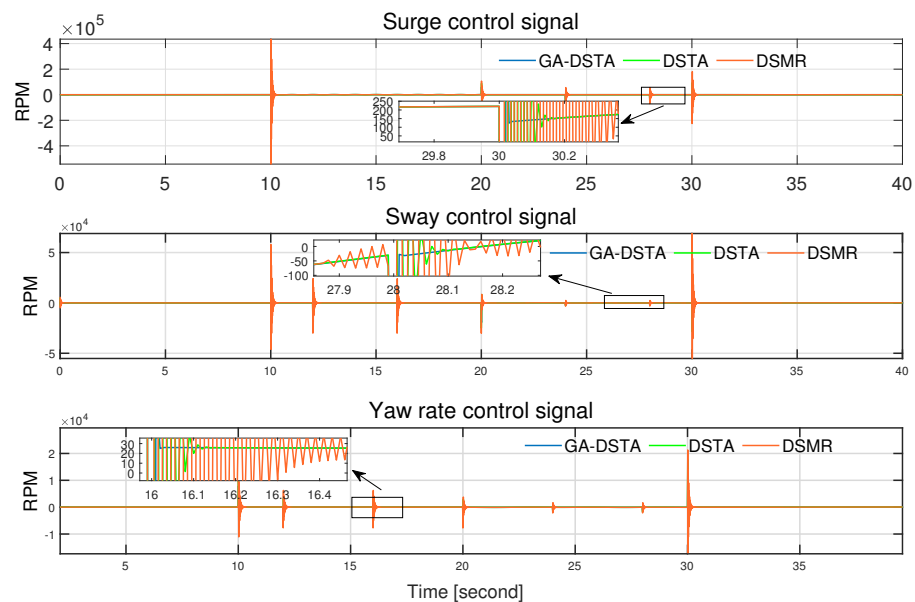


Figure 17. Control signals for surge, sway, and yaw rate of GA-DSTA, DSTA, and DSMR controllers for Scenario 2. RPM is revolutions per minute.

6. Conclusions

In this study, a GA-DSTA controller was proposed for trajectory following of an ASV. Besides, a TDE algorithm was incorporated to estimate the unknown dynamics of the system. The stability of the closed-loop system was analyzed using a Lyapunov approach. The controller gains were tuned using GA based on a quadratic performance index, the sum of the tracking error and the control energy. The performance of the proposed controller was validated with and without the presence of external disturbances, and its performance was also compared with both DSTA and DSMR. The results showed the proposed GA-DSTA controller outperformed, in terms of both the tracking error and the chattering phenomenon reduction, the other controllers. However, though GA improved the tracking capability of the proposed method by suppressing the chattering effect, its robustness slightly decreased when external disturbances were added. However, the TDE algorithms improved the robustness of the controller.

The proposed controller can be applied to the trajectory following of an ASV which can be used to replace time-consuming, expensive, and hazardous sensing tasks which

were previously performed by manned marine vehicles. In the future, the practical implementation of the proposed controller will be done.

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

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2019R111A3A01057691).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

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

Appendix A. Proof of Equation (33)

$$\begin{aligned} \varphi^T F \varphi &= \begin{bmatrix} -T\Omega_{1i}|s_i|^{\frac{1}{2}} \\ -T^2\Omega_{2i}\text{sgn}(s_i(k)) \end{bmatrix}^T \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} \begin{bmatrix} -T\Omega_{1i}|s_i|^{\frac{1}{2}} \\ -T^2\Omega_{2i}\text{sgn}(s_i(k)) \end{bmatrix} \\ &= T^2\Omega_{1i}^2 f_{11}|s_i(k)| + 2T^3 f_{12}\Omega_{1i}\Omega_{2i}|s_i(k)|^{\frac{1}{2}} + T^4 f_{22}\Omega_{2i}^2 \\ &= (q_1|s_i(k)| + q_2)|s_i(k)|^{\frac{1}{2}} + q_3 \\ &\leq q_1|s_i(k)| + q_2|s_i(k)|^{\frac{1}{2}} + q_2 + q_3 \\ &\leq (q_1 + q_2)|s_i(k)| + q_2 + q_3 \\ &= \bar{q}_1|s_i(k)| + \bar{q}_2, \end{aligned}$$

where $q_1 = T^2\Omega_{1i}^2 f_{11}$, $q_2 = 2T^3 f_{12}\Omega_{1i}\Omega_{2i}$, $q_3 = T^4 f_{22}\Omega_{2i}^2$, $\bar{q}_1 = q_1 + q_2$, and $\bar{q}_2 = q_2 + q_3$, F is a positive definite matrix. And also, $\Omega_{1i} > 0$ and $\Omega_{2i} > 0$.

References

1. Yuan, S.; Li, Y.; Bao, F.; Xu, H.; Yang, Y.; Yan, Q.; Zhong, S.; Yin, H.; Xu, J.; Huang, Z.; et al. Marine environmental monitoring with unmanned vehicle platforms: Present applications and future prospects. *Sci. Total Environ.* **2023**, *858*, 159741. [\[CrossRef\]](#)
2. Li, Y.; Tian, L.; Li, W.; Li, J.; Wei, A.; Li, S.; Tong, R. Design and Experiments of a Water Color Remote Sensing-Oriented Unmanned Surface Vehicle. *Sensors* **2020**, *20*, 2183. [\[CrossRef\]](#)
3. Yuh, J.; Marani, G.; Blidberg, D.R. Applications of marine robotic vehicles. *Intell. Serv. Robot.* **2011**, *4*, 221–231. [\[CrossRef\]](#)
4. Hung, S.-S.; Ou, G.-R.; Wu, J.R.; Hsu, C. Autonomous Water Quality Monitoring and Water Surface Cleaning for Unmanned Surface Vehicle. *Sensors* **2021**, *21*, 1102.
5. Visbeck, M. Ocean science research is key for a sustainable future. *Nat. Commun.* **2018**, *9*, 690. [\[CrossRef\]](#)
6. Rian, J.; Pang, S.; Sun, H.B.; Pang, Y.J. Development and missions of unmanned surface vehicle. *J. Mar. Sci. Appl.* **2010**, *9*, 451–457.
7. Caccia, M.; Bibuli, M.; Bono, R.; Bruzzone, G.; Bruzzone, G.; Spirandelli, E. Unmanned Surface Vehicle for Coastal and Protected Waters Applications: The Charlie Project. *Mar. Soc. Tech. J.* **2007**, *41*, 62–71. [\[CrossRef\]](#)
8. Murphy, R.R.; Steimle, E.; Griffin, C.; Cullins, C.; Hall, M.; Pratt, K. Cooperative use of unmanned sea surface and micro aerial vehicles at Hurricane Wilma. *J. Field Robot.* **2008**, *25*, 164–180. [\[CrossRef\]](#)
9. Kiencke, U.; Nielsen, L.; Sutton, R.; Schilling, K.; Papageorgiou, M.; Asama, H. The impact of automatic control on recent developments in transportation and vehicle systems. *Annu. Rev. Control* **2006**, *30*, 81–89. [\[CrossRef\]](#)
10. Liu, Z.; Zhang, Y.; Yu, X.; Yuan, C. Unmanned surface vehicles: An overview of developments and challenges. *Annu. Rev. Control* **2016**, *41*, 71–93. [\[CrossRef\]](#)
11. Fossen, T.I. *Marine Control Systems—Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*; Marine Cybernetics: Trondheim, Norway, 2002; pp. 143–169.
12. Zheng, H.; Negenborn, R.R.; Lodewijks, G. Trajectory tracking of autonomous vessels using model predictive control. In Proceedings of the 19th World Congress, Cape Town, South Africa, 24–29 August 2014.

13. Dewil, R.P.; Asfihani, T.; Nurhadi, H. Design control of autonomous surface vehicle position using proportional integral derivative method. In Proceedings of the International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA), Surabaya, Indonesia, 8–9 December 2021.
14. Meng, Y.; Ye, H.; Yang, X. Trajectory-tracking control of an unmanned surface vehicle based on characteristic modelling approach: Implementation and field testing. *IET Cyber Syst. Robot.* **2023**, *5*, 451–457. [[CrossRef](#)]
15. Sonnenburg, C.R.; Woolsey, C.A. Modeling, identification, and control of an unmanned surface vehicle. *J. Field Robot.* **2013**, *30*, 371–398. [[CrossRef](#)]
16. Liu, L.; Wang, D.; Peng, Z. ESO-Based Line-of-Sight Guidance Law for Path Following of Underactuated Marine Surface Vehicles with Exact Sideslip Compensation. *IEEE J. Ocean Eng.* **2017**, *42*, 477–487. [[CrossRef](#)]
17. Su, S.; Lee, W.B. Limitations and countermeasures of PID controllers. *Ind. Eng. Chem. Res.* **1996**, *35*, 2596–2610.
18. Wang, W.; Mateos, L.A.; Park, S.; Leoni, P.; Gheneti, B.; Duarte, F.; Ratti, C.; Rus, D. Design, modeling, and nonlinear model predictive tracking control of a novel autonomous surface vehicle. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 September 2018; pp. 6189–6196.
19. Guerreiro, B.J.; Silvestre, C.; Cunha, R.; Pascoal, A. Trajectory Tracking Nonlinear Model Predictive Control for Autonomous Surface Craft. *IEEE Trans. Control Syst. Technol.* **2014**, *22*, 2160–2175. [[CrossRef](#)]
20. Halvorsen, H. Dynamic Positioning for Unmanned Surface Vehicles. Master's Thesis, NTUT University, Department Engineering Cybernetics, Trondheim, Norway, 2008.
21. Halvorsen, H.S.; Øveraaas, H.; Landstad, O.; Smines, V.; Fossen, T.I.; Johansen, T.A. Wave motion compensation in dynamic positioning of small autonomous vessels. *J. Mar. Sci. Technol.* **2020**, *26*, 693–712. [[CrossRef](#)]
22. Athans, M. The role and use of the stochastic linear-quadratic-Gaussian problem in control system design. *IEEE Trans. Autom. Control* **1971**, *16*, 529–552. [[CrossRef](#)]
23. Rosario, R.V.C.; Cunha, J.P.V.S.; Garcia-Rosa, P.B. Stabilizing Control of an Unmanned Surface Vehicle Pushing a Floating Load. *Int. J. Control Automat. Syst.* **2020**, *18*, 3194–3203. [[CrossRef](#)]
24. Zaopeng, D.; Wan, L.; Li, Y.; Liu, T.; Zhang, G. Trajectory tracking control of underactuated USV based on modified backstepping approach. *Int. J. Naval Archit. Ocean Eng.* **2015**, *7*, 817–832.
25. Liao, Y.L.; Su, Y.M.; Cao, J. Trajectory planning and tracking control for underactuated unmanned surface vessels. *J. Central South Univ.* **2014**, *21*, 540–549. [[CrossRef](#)]
26. Xiang, X.; Yu, C.; Lapierre, L.; Zhang, J.; Zhang, Q. Survey on fuzzy-logic-based guidance and control of marine surface vehicles and underwater vehicles. *Int. J. Fuzzy Syst.* **2018**, *20*, 572–586. [[CrossRef](#)]
27. Chu, Z.; Zhu, D.; Yang, S.X. Observer-Based Adaptive Neural Network Trajectory Tracking Control for Remotely Operated Vehicle. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 1633–1645. [[CrossRef](#)] [[PubMed](#)]
28. Zhao, Y.; Mu, D.; Wang, G.; Fan, Y. Trajectory Tracking Control for Unmanned Surface Vehicle Subject to Unmeasurable Disturbance and Input Saturation. *IEEE Access* **2020**, *8*, 191278–191285. [[CrossRef](#)]
29. Samantaray, J.; Chakrabarty, S. Digital implementation of sliding mode controllers with DC-DC buck converter system. In Proceedings of the 15th International Workshop on Variable Structure System (VSS), Graz, Austria, 9–11 July 2018; pp. 255–260.
30. Haninger, K.; Hedrick, K. Discrete-time implementations of sliding mode control. In Proceedings of the American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; pp. 6519–6524.
31. Young, K.D.; Utkin, V.I.; Ozguner, U. A control engineer's guide to sliding mode control. *IEEE Trans. Control Syst. Tech.* **1999**, *7*, 328–342. [[CrossRef](#)]
32. Bartolini, G.; Ferrara, A.; Usai, E. Chattering avoidance by second-order sliding mode control. *IEEE Trans. Autom. Control* **1998**, *43*, 241–246. [[CrossRef](#)]
33. Bartolini, G.; Ferrara, A.; Giacomini, L.; Usai, A. Properties of a combined adaptive/second-order sliding mode control algorithm for some classes of uncertain nonlinear systems. *IEEE Trans. Autom. Control* **2000**, *45*, 1334–1341. [[CrossRef](#)]
34. Chakrabarty, S.; Bandyopadhyay, B. Discrete time sliding mode control with minimal quasi sliding mode band. In Proceedings of the 12th International Workshop on Variable Structure System, Mumbai, India, 12–14 January 2012; pp. 404–409.
35. Iván, S.; Chairez, I.; Bandyopadhyay, B.; Fridman, L.; Camacho, O. Discrete-time non-linear state observer based on a super twisting-like algorithm. *IET Control Theory Appl.* **2014**, *8*, 803–812.
36. Kali, Y.; Ayala, M.; Rodas, J.; Saad, M.; Doval-Gandoy, J.; Grego, R.; Benjelloun, K. Time Delay Estimation Based Discrete-Time Super-Twisting Current Control for a Six-Phase Induction Motor. *IEEE Trans. Power Electron.* **2020**, *35*, 12570–12580. [[CrossRef](#)]
37. Leśniewski, P. Discrete time reaching law based sliding mode control: A survey. In Proceedings of the 22nd International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 10–12 October 2018; pp. 734–739.
38. Kumar, R.P.; Dasgupta, A.; Kumar, C.S. Robust Tracking Control of Underwater Vehicles using Time-Delay Control in Discrete-Time Domain. In Proceedings of the OCEANS 2006—Asia Pacific, Singapore, 16–19 May 2006; pp. 1–5.
39. Bibuli, M.; Bruzzone, G.; Caccia, M.; Ippolito, G.; Longhi, S.; Orlando, G.; Pelusi, G.M. Discrete-time sliding mode control for guidance of an unmanned surface vehicle. In Proceedings of the 9th IFAC Conference on Manoeuvring and Control of Marine Craft, Arenzano, Italy, 19–21 September 2012; pp. 441–446.
40. Morgan, R.; Ozguner, U. A decentralized variable structure control algorithm for robotic manipulators. *IEEE J. Robot. Automat.* **1985**, *1*, 57–65. [[CrossRef](#)]

41. Kumar, R.P.; Kumar, C.S.; Sen, D.; Dasgupta, A. Discrete time–delay control of an autonomous underwater vehicle: Theory and experimental results. *Ocean Eng.* **2009**, *39*, 74–81. [[CrossRef](#)]
42. Kim, J.; Joe, H.; Yu, S.-c.; Lee, J.S.; Kim, M. Time–Delay Controller Design for Position Control of Autonomous Underwater Vehicle Under Disturbances. *IEEE Trans. Ind. Electron.* **2016**, *63*, 1052–1061. [[CrossRef](#)]
43. Kumar, R.P.; Dasgupta, A.; Kumar, C.S. A new tracking controller design for underwater vehicles using quadratic stabilization. *ASME J. Dyn. Sys. Meas. Control* **2008**, *130*, 24502. [[CrossRef](#)]
44. Kunal, T.; Krishnankutty, P. Dynamic positioning of an oceanographic research vessel using fuzzy logic controller in different sea states. *Mar. Syst. Ocean Technol.* **2021**, *16*, 221–236.
45. Zhang, S.; Yu, J.; Zhang, A. Discrete–time quasi–sliding mode control of underwater vehicles. In Proceedings of the 8th World Congress on Intelligent Control and Automation, Jinan, China, 7–9 July 2010; pp. 6686–6690.
46. Khalil, H. *Nonlinear Systems*, 3rd ed.; Prentice–Hall: Upper Saddle River, NJ, USA, 2002; pp. 85–94.
47. Salgado, S.K.; Chairez, I.; Bandyopadhyay, B.; Fridman, L. Super–twisting–like algorithm in discrete time nonlinear systems. In Proceedings of the 2011 International Conference on Advanced Mechatronic Systems, Zhengzhou, China, 11–13 August 2011; pp. 497–502.
48. Poznyak, A. *Advanced Mathematical Tools for Control Engineers, Deterministic Systems*; Elsevier: New York, NY, USA, 2010; Volume 1.
49. Kirk, D.E. *Optimal Control Theory: An Introduction*; Dover Publications, Inc.: Mineola, NY, USA, 2004.
50. Voratas, K. Comparison of three evolutionary algorithms: GA, PSO, and DE. *Ind. Eng. Manag. Syst.* **2012**, *11*, 215–223.
51. Immanuel, S.D.; Chakraborty, U.K. Genetic Algorithm: An Approach on Optimization. In Proceedings of the 2019 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 17–19 July 2019; pp. 701–708.
52. Tsai, J.S.-H.; Li, J.S.; Leang-San Shieh. Discretized quadratic optimal control for continuous-time two-dimensional systems. *IEEE Trans. Circ. Syst. I Fundam. Theory Appl.* **2002**, *49*, 116–125. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.