



# Article Highly Robust Adaptive Sliding Mode Trajectory Tracking Control of Autonomous Vehicles

Fengxi Xie <sup>1,†</sup>, Guozhen Liang <sup>1,†</sup> and Ying-Ren Chien <sup>2,\*</sup>

- <sup>1</sup> Department of Electrical Engineering and Computer Science, Technische Universität Berlin, 10623 Berlin, Germany
- <sup>2</sup> Department of Electrical Engineering, National Ilan University, Yilan 260007, Taiwan
- \* Correspondence: yrchien@niu.edu.tw
- + These authors contributed equally to this work.

**Abstract:** Autonomous driving technology has not yet been widely adopted, in part due to the challenge of achieving high-accuracy trajectory tracking in complex and hazardous driving scenarios. To this end, we proposed an adaptive sliding mode controller optimized by an improved particle swarm optimization (PSO) algorithm. Based on the improved PSO, we also proposed an enhanced grey wolf optimization (GWO) algorithm to optimize the controller. Taking the expected trajectory and vehicle speed as inputs, the proposed control scheme calculates the tracking error based on an expanded vector field guidance law and obtains the control values, including the vehicle's orientation angle and velocity on the basis of sliding mode control (SMC). To improve PSO, we proposed a three-stage update function for the inertial weight and a dynamic update law for the learning rates to avoid the local optimum dilemma. For the improvement in GWO, we were inspired by PSO and added speed and memory mechanisms to the GWO algorithm. Using the improved optimization algorithm, the control performance was successfully optimized. Moreover, Lyapunov's approach is adopted to prove the stability of the proposed control schemes. Finally, the simulation shows that the proposed control scheme is able to provide more precise response, faster convergence, and better robustness in comparison with the other widely used controllers.

**Keywords:** autonomous vehicles; trajectory tracking; high robustness; vector field guidance law; sliding mode control; improved particle swarm optimization; improved grey wolf optimization

# 1. Introduction

In recent years, researchers have demonstrated that autonomous vehicles (AVs) can possibly reduce road accident rates and enhance transportation efficiency [1–3], thus drawing massive attention from both industrial and academic fields.

Trajectory tracking is one of the basic functions of AVs, which aims to follow a desired trajectory within a certain period of time and then to maintain motion stability continuously [4]. Control theory is a fundamental element in various domains [5]. Recently, different kinds of methods and controllers associated with trajectory tracking have been developed. For PID, Han et al. proposed a lateral path-following controller, which can fine-tune the control parameters by integrating a neural network [6]. In regards to multitask control (e.g., tracking precision, driving comfort, and ride stability), due to the limited capacity for handling several control objectives, a PID controller is not the ideal control scheme for the trajectory tracking of AVs [7–11]. For fuzzy logic control, El et al. designed a Takagi–Sugeno fuzzy controller, in which the stability was proved by integrating linear matrix inequalities in a Lyapunov stability analysis [12]. In addition to PID control, SMC is also widely used in improving system robustness. Compared with PID, SMC requires fewer parameters to be adjusted and has a faster response. It is considered a powerful control technique for suppressing external disturbances [13–17]. He et al. proposed a backstepping SMC controller to collision-free path tracking [18]. Guo et al. introduced an adaptive fuzzy



Citation: Xie, F.; Liang, G.; Chien, Y.-R. Highly Robust Adaptive Sliding Mode Trajectory Tracking Control of Autonomous Vehicles. *Sensors* **2023**, 23, 3454. https://doi.org/10.3390/ s23073454

Academic Editors: Mustafa Ilhan Akbas and Jun Chen

Received: 13 February 2023 Revised: 14 March 2023 Accepted: 22 March 2023 Published: 25 March 2023



**Copyright:** © 2023 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/). sliding-mode controller for the steering and brake control [19]. In order to achieve trajectory tracking of an unmanned agricultural tractor, Matveev et al. developed a nonlinear SMC control strategy, which showed a robust performance [20]. However, SMC has a significant drawback, namely chattering, and a number of methods have been proposed to effectively mitigate the chattering effect and to improve the robustness of the system [21–25].

For trajectory modeling, input parameters include vehicle states such as velocity, orientation angle, and acceleration. However, these states could only be obtained through specific equipment. To this end, researchers have designed robust estimators by integrating cameras, global navigation satellite system (GNSS), and inertial measurement unit (IMU) [26–30]. These techniques could be used in the future whole-vehicle test verification of our proposed approach.

For all of the aforementioned control schemes, to achieve good performance, the gain of their controller needs to be manually adjusted, and in different application scenarios, the gain value of the controller can be very different and people may spend a lot of time on fine-tuning the parameters. In order to overcome this difficulty and to make the control scheme more effective, an optimization algorithm can be carefully designed for adaptive control parameter tuning. Traditional optimization techniques, such as gradientbased methods, may not always work efficiently or may even fail to converge to a global optimum. In contrast, PSO and GWO are population-based optimization algorithms that mimic the behavior of a swarm or a pack of animals, respectively, aiming to search for the global optimum in a multidimensional solution domain. Firstly, designing a good cost function is an important part of the optimization algorithm [31]. Fateh et al. applied PSO to optimize robust control of robot manipulators [32]. A fuzzy controller can also be tuned by PSO [33,34]. To obtain a robust and adaptive PID controller, Elkaranshway et al. adopted PSO for parameter fine-tuning [35]. However, the inertial weight and learning rates of the PSO algorithm introduced in previous work were usually set to a constant, often leading to a local optimum dilemma. While PSO is commonly used in optimizing control schemes, there are also other algorithms that can be adopted, such as the GWO. As a meta-heuristic optimization algorithm, GWO has been shown to have better explorationexploitation trade-off, faster convergence, and the ability to handle multiple objectives when compared with PSO [36]. GWO is also less susceptible to premature convergence than PSO [37]. Furthermore, GWO has been improved with the use of crossover operators and hybridization with other algorithms, such as the sine cosine algorithm, leading to even better performance. Therefore, GWO may be an alternative for optimizing control schemes.

In this paper, we combine the vector field guidance law approach with trajectory tracking control. In recent years, the vector field approach has been well introduced to provide a better solution for many control problems [38]. By creating a vector field around the target path, we ensure that the tracking error approaches zero asymptotically, even if the vehicle undergoes external disturbances [39]. More recently, scholars have been trying to extend the vector field approach in different path-control-related problems [40–42].

However, it is noteworthy that the fundamental mission of autonomous vehicles is to ensure passengers' safety [43]. Therefore, the controller should be adaptable. Even in dangerous scenarios (driving at high speed on complex trajectories), the controller can autonomously tune its gain and minimize errors. Current studies mainly focus on path following and do not meet the adaptive requirements of autonomous vehicles. Additionally, the predefined curves in previous studies are simple and differ from practical traffic scenarios.

Given this problem, this paper implements a high-accuracy and adaptive trajectory tracking controller by integrating sliding mode control and novel evolutionary optimization algorithms. This study has three contributions: (1) A vector field guidance law has been designed and extended from a simple curve to a sine trajectory and a polynomial trajectory, which assembles the lane-change trajectory in practical traffic scenarios. (2) A sliding mode controller of velocity and angle are designed separately to ensure better performance, and improved optimization algorithms, named IPSO and PGWO, are designed. (3) We

integrate the designed vector field guidance law, optimization algorithm, and sliding mode controllers to create an adaptive control scheme and formulate the corresponding Lyapunov function to demonstrate system stability. We compare the proposed control scheme against other commonly employed methods. The simulation results prove that our design outperforms traditional methods in complex scenarios and considerably enhances the system's robustness.

Regarding the arrangement of this paper, Section 2 introduces the vehicle kinematic model. In Section 3, the structure and the theory of optimized control strategy are described. Section 4 presents and analyzes the simulation results of different controllers. Finally, Section 5 summarizes the workflow of this study and discusses possible future improvements.

## 2. Preliminaries

Referring to Figure 1, we obtain the vehicle's kinematics model:

$$\begin{aligned} \dot{x} &= v \cos(\theta) \\ \dot{y} &= v \sin(\theta) \\ \dot{\theta} &= \sigma^{-1} v \tan(\delta) \\ \dot{v} &= F \end{aligned} \tag{1}$$

where  $\sigma > 0$  represents the distance between the front axle and the rear axle of this vehicle. To better describe the trajectory of an autonomous vehicle, we use the midpoint of the rear axle as the reference point, which can also be represented as (x, y). In an inertial frame with Cartesian coordinates (X, Y), x and y are the longitudinal position and lateral position of the vehicle, respectively. v is the vehicle velocity at the point (x, y), while  $\theta \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$  is the angular orientation of the vehicle with respect to the X axis. In fact, lateral control of the vehicle is achieved by changing the steering angle of the car. Here, we derive the relationship between the orientation angle  $\theta$  and the steering angle of the front wheels by analyzing the kinematics of the vehicle. F represents the acceleration of the vehicle.



Figure 1. Vehicle model.

# 3. Controller Design

# 3.1. Control Scheme of Proposed Controller

The architecture of the IPSO-SM controller is shown in Figure 2. The desired trajectory and velocity are used as input in the model. With the help of the expanded vector field guidance method, a set of desired yaw angles will be generated from the input trajectory. The orientation and velocity controller will output the optimal yaw angle and velocity according to the current angle error and velocity error, respectively, under the optimization of IPSO. Then, after the calculation of the vehicle kinematic model, the current position of the vehicle will be the output.



Figure 2. Control scheme of the proposed controller.

## 3.2. Vector Field Guidance Law

The vector field guidance law has been expanded from the simple curve [39] to the complex curve in this paper. For a certain trajectory,  $\omega$ ,  $\omega_c$  is defined as the closest point and  $\omega_v$  is the location of the vehicle; then, the distance vector  $v_1(\omega_c)$  and the scalar distance  $v_1(\omega_c)$  are defined as

$$\begin{aligned}
\nu_1(\omega_c) &= \omega_c - \omega_v \\
\nu_1(\omega_c) &= ||\omega_c - \omega_v||
\end{aligned} \tag{2}$$

with  $v_2(\omega_c)$  as the curve's tangent vector at  $\omega_c$ .

To facilitate the calculation, two operators are defined here.

$$\kappa_1 = (2/\pi) \arctan\left(k_f \nu_1\right)$$

$$\kappa_2 = \sqrt{1 - \kappa_1^2}$$
(3)

where  $k_f \in \mathbb{R}^+$ .

The guidance vector  $\omega_v$  is designed as

$$\Theta(\boldsymbol{\omega}_{c}) = -\kappa_{1} \frac{\boldsymbol{\nu}_{1}(\boldsymbol{\omega}_{c})}{\boldsymbol{\nu}_{1}(\boldsymbol{\omega}_{c})} + \kappa_{2} \boldsymbol{\nu}_{2}(\boldsymbol{\omega}_{c})$$
(4)

To achieve accurate angle control, the desired yaw angle is calculated for each position of the vehicle in motion. Since  $\Theta$  is a series of vectors forming a trajectory,  $\Theta(1)$  and  $\Theta(2)$  are used to denote the first and second values of trajectory vector, respectively. Then, the desired yaw angle is

$$\psi_d = \arctan\left(\frac{\mathbf{\Theta}(1)}{\mathbf{\Theta}(2)}\right) \tag{5}$$

where the subscript d means a desired value, which should be tracked. The yaw angle of the vehicle is assumed to be capable of tracking the desired yaw angle  $\psi_d$ . Substituting (4) into (5) yields the derivative of position error  $p_e$ 

$$\dot{p}_e = V \sin(\psi_d - \psi_p) = V \sin\left(\psi_d - \arctan\left(\frac{\nu_2(1)}{\nu_2(2)}\right)\right).$$
(6)

where  $\psi_p$  denotes the current yaw angle. Then, to simplify the calculation, we define

$$\nu_3 = \kappa_1 \frac{\nu_1(\boldsymbol{\omega}_c)}{\nu_1(\boldsymbol{\omega}_c)} + (1 - \kappa_2)\nu_2(\boldsymbol{\omega}_c)$$
(7)

Therefore, (6) can be simplified to (8):

$$\dot{p}_e = -V\sin\left(\arctan\left(\frac{\nu_3(1)}{\nu_3(2)}\right)\right) \tag{8}$$

To prove the stability of the method, we use the position error  $p_e$  to design a Lyapunov candidate  $V_1$ :

$$V_1 = \frac{1}{2} p_e^2,$$
 (9)

where  $p_e^2$  is always > 0 when  $p_e$  is not equal to zero. Additionally, according to (8), because  $v_1(\omega_c)$  and  $v_2(\omega_c)$  are a set of orthogonal vectors, it is then  $\arctan\left(\frac{v_3(1)}{v_3(2)}\right) < \pi/2$ ; then,  $\sin\left(\psi_d - \arctan\left(\frac{v_2(1)}{v_2(2)}\right)\right) > 0$ . Thus,  $\dot{p}_e < 0$  holds. Therefore,  $\exists \sigma_1 > 0$ , the derivative of  $V_1$  can be obtained as  $\dot{V}_1 = p_e \dot{p}_e$ 

$$\begin{array}{l} -p_e p_e \\ \leq -\sigma_1 V_1. \end{array} \tag{10}$$

#### 3.3. Orientation Angle Controller Design

To achieve lateral control of the vehicle, it is necessary to minimize the yaw error. Define  $\psi$  as the current yaw angle of the vehicle, so the angle error  $e_{\psi,1}$  is

$$e_{\psi,1} = \psi - \psi_d \tag{11}$$

Similar to Section 3.2, a Lyapunov candidate is considered as (12):

$$V_2 = \frac{1}{2} e_{\psi,1}^2 \tag{12}$$

The differential of this candidate is described as (13):

$$\dot{V}_{2} = e_{\psi,1} \dot{e}_{\psi,1} 
= e_{\psi,1} (\dot{\psi} - \dot{\psi}_{d})$$
(13)

In order to bring the defined error closer to the true error, here, we introduce the nominal error. Define the nominal yaw angle error as (14), where  $k_{\psi,1} > 0$ .

$$e_{\psi,2} = \dot{\psi} - \dot{\psi}_d + k_{\psi,1} e_{\psi,1} \tag{14}$$

Combining Equations (13) and (14) yields

$$\dot{V}_2 = e_{\psi,1}e_{\psi,2} - k_{\psi,1}e_{\psi,1}^2 \tag{15}$$

Based on the principle of sliding mode control, we define the sliding mode surface  $s_1$ , which consists of angle errors

$$s_1 = k_{\psi,2} e_{\psi,1} + e_{\psi,2} \tag{16}$$

where  $k_{\psi,2}$  is the sliding mode surface parameter and  $k_{\psi,2} > 0$ .

According (11) and (14), the differential of  $e_{\psi,1}$  can be described as (17):

$$\dot{e}_{\psi,1} = \dot{\psi} - \dot{\psi}_d 
= e_{\psi,2} - k_{\psi,1} e_{\psi,1}$$
(17)

To prove the stability of the orientation angle controller, a Lyapunov candidate  $V_3$  for this controller should be designed in order to ensure that the Lyapunov function decreases along the sliding mode surface and to reduce the number of variables, and simplifying the expression of the Lyapunov function,  $V_3$  should include  $s_1$  and  $V_2$ . Therefore, it is considered as follows:

$$V_3 = V_2 + \frac{1}{2}s_1^2 \tag{18}$$

After differentiating, we substitute (16) and (17) into this and then derive (19):

$$\begin{aligned} \dot{V}_{3} &= \dot{V}_{2} + \frac{1}{2}\dot{s}_{1}^{2} \\ &= e_{\psi,1}e_{\psi,2} - k_{\psi,1}e_{\psi,1}^{2} + s_{1}\dot{s}_{1} \\ &= e_{\psi,1}e_{\psi,2} - k_{\psi,1}e_{\psi,1}^{2} + s_{1}(k_{\psi,2}\dot{e}_{\psi,1} + \dot{e}_{\psi,2}) \\ &= e_{\psi,1}e_{\psi,2} - k_{\psi,1}e_{\psi,1}^{2} + s_{1}[k_{\psi,2}(e_{\psi,2} - k_{\psi,1}e_{\psi,1}) + \ddot{\psi} - \ddot{\psi}_{d} + k_{\psi,1}\dot{e}_{\psi,1}] \end{aligned}$$
(19)

In order to speed up the convergence of the sliding surface and to make the whole process smoother and less chattering, the reaching law is designed as

$$\dot{s}_1 = -k_{\psi,3}|s_1|^{\alpha_1}\operatorname{sgn}(s_1) - \left(\frac{1}{2k_{\psi,2}} + e^{f_{\psi}(s_1)}|s_1|^{\beta_1}\right)s_1,\tag{20}$$

in which  $k_{\psi,3} > 0$ ,  $1 > \alpha_1 > 0$ ,  $1 > \beta_1 > 0$ , and  $f_{\psi}(s_1)$  is designed as

$$f_{\psi}(s_1) = \begin{cases} |s_1|(|s_1| - \Delta_{\psi}), |s_1| \ge \Delta_{\psi} \\ -\frac{1 - |s_1| / \Delta_{\psi}}{|s_1| + \Delta_{\psi}}, |s_1| < \Delta_{\psi} \end{cases}$$
(21)

where  $\Delta_{\psi}$  denotes the thickness of the sliding mode surface.

Combining the above equations, the sliding mode controller is designed as

$$u_{d} = -k_{\psi,3}|s_{1}|^{\alpha_{1}}\operatorname{sgn}(s_{1}) - \left(\frac{1}{2k_{\psi,2}} + e^{f_{\psi}(s_{1})}|s_{1}|^{\beta_{1}}\right)s_{1} - k_{\psi,2}(e_{\psi,2} - k_{\psi,1}e_{\psi,1}) + \ddot{\psi}_{d} - k_{\psi,1}\dot{e}_{\psi,1}$$
(22)

Then, the desired steering angle of the front wheels can be designed as

$$\delta_d = \arctan\left(v^{-1}\sigma u_d\right) \tag{23}$$

Then, substituting (20) into (19), the Lyapunov candidate  $V_3$  can be described as

$$\begin{split} \dot{V}_{3} &= e_{\psi,1}e_{\psi,2} - k_{\psi,1}e_{\psi,1}^{2} + s_{1} \left[ -k_{\psi,3}|s_{1}|^{\alpha_{1}}\operatorname{sgn}(s_{1}) - \left(\frac{1}{2k_{\psi,2}} + e^{f_{\psi}(s_{1})}|s_{1}|^{\beta_{1}}\right)s_{1} \right] \\ &= e_{\psi,1}e_{\psi,2} - k_{\psi,1}e_{\psi,1}^{2} - k_{\psi,3}|s_{1}|^{\alpha_{1}}|s_{1}| - \frac{1}{2k_{\psi,2}}s_{1}^{2} - e^{f_{\psi}(s_{1})}|s_{1}|^{\beta_{1}}s_{1}^{2} \\ &= e_{\psi,1}e_{\psi,2} - k_{\psi,1}e_{\psi,1}^{2} - \frac{1}{2k_{\psi,2}}\left(k_{\psi,2}e_{\psi,1} + e_{\psi,2}\right)^{2} - k_{\psi,3}|s_{1}|^{\alpha_{1}+1} - e^{f_{\psi}(s_{1})}|s_{1}|^{\beta_{1}}s_{1}^{2} \\ &= -k_{\psi,1}e_{\psi,1}^{2} - \frac{1}{2k_{\psi,2}}\left(k_{\psi,2}^{2}e_{\psi,1}^{2} + e_{\psi,2}^{2}\right) - k_{\psi,3}|s_{1}|^{\alpha_{1}+1} - e^{f_{\psi}(s_{1})}|s_{1}|^{\beta_{1}}s_{1}^{2} \end{split}$$

$$(24)$$

According to (12) and (19),  $V_2 \ge 0$  and  $V_3 \ge 0$ . Additionally,  $k_{\psi,1} > 0$ ,  $k_{\psi,2} > 0$ ,  $k_{\psi,3} > 0$ , and  $e^{f_{\psi}(s_1)}|s_1|^{\beta_1} > 0$ . Therefore, the conclusion that  $\exists \sigma_3 > 0$  can be given such that  $\dot{V}_3 \le -\sigma_3 V_3$ .

#### 3.4. Velocity Controller Design

The controller of velocity is similar to the orientation angle controller; however, in the orientation angle controller, the desired yaw angle of the vehicle is calculated by the guidance law, while the desired velocity of the vehicle here is defined artificially. Define the velocity error  $e_{v,1}$  as (25), where v is the current velocity and  $v_d$  is the desired velocity.

$$e_{v,1} = v - v_d \tag{25}$$

For the velocity error, similar to Section 3.3, consider a Lyapunov candidate:

$$V_4 = \frac{1}{2}e_{v,1}^2 \tag{26}$$

Differentiate it as

$$\dot{V}_4 = e_{v,1} \dot{e}_{v,1} 
= e_{v,1} (\dot{v} - \dot{v}_d)$$
(27)

Similar to the orientation controller, in order to improve the robustness of the controller, here, the nominal velocity error is defined as

$$e_{v,2} = \dot{v} - \dot{v}_d + k_{v,1} e_{v,1} \tag{28}$$

where  $k_{v,1} > 0$ .

Combining (27) and (28) yields (29), here, the differential of  $V_4$  can be described with the velocity error and nominal velocity error.

$$\dot{V}_4 = e_{v,1}e_{v,2} - k_{v,1}e_{v,1}^2 \tag{29}$$

Define the sliding mode surface  $s_2$ , which consists of  $e_{v,1}$  and  $e_{v,2}$ :

$$s_2 = k_{v,2} e_{v,1} + e_{v,2} \tag{30}$$

where  $k_{v,2}$  is the sliding mode surface parameter and  $k_{v,2} > 0$ . According (25) and (28), differentiate  $e_{v_1}$  as (31):

Consider a Lyapunov candidate  $V_5$ , which consists of  $V_4$  and  $s_2$ :

$$V_5 = V_4 + \frac{1}{2}s_2^2 \tag{32}$$

Combining (29), (30), and (32) yields

$$\dot{V}_{5} = \dot{V}_{4} + \frac{1}{2}\dot{s}_{2}^{2}$$

$$= e_{v,1}e_{v,2} - k_{v,1}e_{v,1}^{2} + s_{2}\dot{s}_{2}$$

$$= e_{v,1}e_{v,2} - k_{v,1}e_{v,1}^{2} + s_{2}(k_{v,2}\dot{e}_{v,1} + \dot{e}_{v,2})$$

$$= e_{v,1}e_{v,2} - k_{v,1}e_{v,1}^{2} + s_{2}[k_{v,2}(e_{v,2} - k_{v,1}e_{v,1}) + \ddot{v} - \ddot{v}_{d} + k_{v,1}\dot{e}_{v,1}]$$
(33)

In order to speed up the convergence of the sliding surface and to make the whole process smoother and less chattering, the reaching law  $\dot{s}_2$  is designed as

$$\dot{s}_2 = -k_{v,3}|s_2|^{\alpha_2}\operatorname{sgn}(s_2) - \left(\frac{1}{2k_{v,2}} + e^{f_v(s_2)}|s_2|^{\beta_2}\right)s_2 \tag{34}$$

in which  $k_{v,3} > 0$ ,  $1 > \alpha_2 > 0$ ,  $1 > \beta_2 > 0$ , and  $f_v(s_2)$  is designed as

$$f_{v}(s_{2}) = \begin{cases} |s_{2}|(|s_{2}| - \Delta_{v}), |s_{2}| \ge \Delta_{v} \\ -\frac{1 - |s_{2}| / \Delta_{v}}{|s_{2}| + \Delta_{v}}, |s_{2}| < \Delta_{v} \end{cases}$$
(35)

in which  $\Delta_v$  denotes the thickness of the sliding mode surface.

The sliding mode controller is designed as

$$F_{d} = -k_{v,3}|s_{2}|^{\alpha_{2}}\operatorname{sgn}(s_{2}) - \left(\frac{1}{2k_{v,2}} + e^{f_{v}(s_{2})}|s_{2}|^{\beta_{2}}\right)s_{2} - k_{v,2}(e_{v,2} - k_{1}e_{v,1}) + \ddot{v}_{d} - k_{v,1}\dot{e}_{1}$$
(36)

Then,

$$\dot{V}_{5} = e_{v,1}e_{v,2} - k_{v,1}e_{v,1}^{2} + s_{2}\left[-k_{v,3}|s_{2}|^{\alpha_{2}}\operatorname{sgn}(s_{2}) - \left(\frac{1}{2k_{v,2}} + e^{f_{v}(s_{2})}|s_{2}|^{\beta_{2}}\right)s_{2}\right]$$

$$= e_{v,1}e_{v,2} - k_{v,1}e_{v,1}^{2} - k_{v,3}|s_{2}|^{\alpha_{2}}|s_{2}| - \frac{1}{2k_{v,2}}s_{2}^{2} - e^{f_{v}(s_{2})}|s_{2}|^{\beta_{2}}s_{2}^{2}$$

$$= e_{v,1}e_{v,2} - k_{v,1}e_{v,1}^{2} - \frac{1}{2k_{v,2}}(k_{v,2}e_{v,1} + e_{v,2})^{2} - k_{v,3}|s_{2}|^{\alpha_{2}+1} - e^{f_{v}(s_{2})}|s_{2}|^{\beta_{2}}s_{2}^{2}$$

$$= -k_{v,1}e_{v,1}^{2} - \frac{1}{2k_{v,2}}\left(k_{v,2}^{2}e_{v,1}^{2} + e_{v,2}^{2}\right) - k_{v,3}|s_{2}|^{\alpha_{2}+1} - e^{f_{v}(s_{2})}|s_{2}|^{\beta_{2}}s_{2}^{2}$$
(37)

According to (26) and (32),  $V_4 \ge 0$ ,  $V_5 \ge 0$ . Additionally,  $k_{v,1} > 0$ ,  $k_{v,2} > 0$ ,  $k_{v,3} > 0$ , and  $e^{f_v(s_2)}|s_2|^{\beta_2} > 0$ . Therefore, the conclusion can be given that  $\exists \sigma_5 > 0$  such that  $\dot{V}_5 \le -\sigma_5 V_5$ .

#### 3.5. Stability Analysis

Sections 3.2–3.4 introduced the controllers as parts of the proposed approach. The Lyapunov candidates of each part are designed, and the stability of them are proved separately. To prove the stability of the whole proposed controller, a Lyapunov function that includes all of the Lyapunov candidates of the sub-controllers should be designed. Consider the following Lyapunov candidate:

$$V = V_1 + V_3 + V_5$$
  
=  $\frac{1}{2} \left( p_e^2 + e_{\psi,1}^2 + s_1^2 + e_{v,1}^2 + s_2^2 \right)$  (38)

where  $V_1$ ,  $V_3$ , and  $V_5$  are all quadratic, and they are added together, so there are no extra parameters in V. This means that the Lyapunov candidate V has included all of the characteristics of the sub-controller candidates, and its stability can be easily proved.

Applying (10), (24), and (37) into (39) yields

$$\dot{V} \leq -\sigma_1 V_1 - \sigma_3 V_3 - \sigma_5 V_5 
\leq -\sigma V$$
(39)

where  $\sigma = \min[\sigma_1, \sigma_3, \sigma_5]$ .

Therefore, the conclusion can be drawn that location deviations  $p_e$  and  $e_{\psi,1}$ ,  $e_{\psi,2}$ ,  $s_1$ ,  $e_{v,1}$ ,  $e_{v,2}$ , and  $s_2$  are bounded and can be eliminated to a small neighborhood around zero.

## 3.6. Optimization Algorithm

## 3.6.1. Traditional PSO and GWO Algorithm

• In traditional PSO, a set of  $n \in \mathbb{Z}^+$  particles is defined with particles in each iteration, such as a certain *i*th particle in the *k*th iteration with a velocity and a position, separately denoted by  $v_i(k)$  and  $x_i(k)$ . The updating law of the particles is designed as

$$v_i(k+1) = wv_i(k) + c_1r_1(k)[p_i(k) - x_i(k)] + c_2r_2(k)[p_g(k) - x_i(k)]$$
  

$$x_i(k+1) = x_i(k) + v_i(k)$$
(40)

in which  $p_i$  is the best position in the position history of the *i*th particle,  $p_g$  is the global best position of all the particles,  $c_1$  is the self-learning factor,  $c_2$  is the team-learning factor, and w is the inertial weight. Normally, the parameters of w,  $c_1$ , and  $c_2$  are set to constant values, while  $r_1 \in (0, 1)$  and  $r_2 \in (0, 1)$  are random numbers. Although the PSO algorithm is logically simple and fast in finding the optimal solution, it often converges directly when a local optimal solution is found while ignoring the global optimal solution. This is often referred to as the local optimum problem and indicates that there is still room for improvement.

• The optimization algorithm GWO is inspired by the hunting process of wolves. In the traditional GWO algorithm, there are N individuals in D-dimensional space at  $X = (x^1, x^2, ..., x^D)$ , which represents the grey wolf population. Define the solution with the best result as  $\alpha$  and the solutions with the second and third best result as  $\beta$  and  $\delta$ . Then, a candidate would be defined as  $\omega$ , the whole hunting process is led by the leading wolves, and the candidate  $\omega$  will follow them. First, they will track and chase their goal, and the action can be described as follows:

$$D = |C \cdot X_p(k) - X(k)|$$
  

$$X(k+1) = X_p(k) - A \cdot D$$
(41)

where *D* is the distance between the prey and grey wolves,  $X_p$  is the position of the prey, and *X* is the position of the wolves.  $A = 2a \cdot rand(0, 1) - a$  and  $C = 2 \cdot rand(0, 1)$ . *a* is one of the factors that decrease linearly from two to zero. Then, they will start to hunt, and  $\alpha$ ,  $\beta$ , and  $\delta$  will lead the grey wolf population to surround the prey, which can be described as follows:

$$D_{\alpha} = |C_1 \cdot X_{\alpha} - X|$$
  

$$D_{\beta} = |C_2 \cdot X_{\beta} - X|$$
  

$$D_{\delta} = |C_3 \cdot X_{\delta} - X|$$
(42)

where  $D_{\alpha}$ ,  $D_{\beta}$ , and  $D_{\delta}$  represent the distance between  $\alpha$ ,  $\beta$ , and  $\delta$  and the candidates.  $C_1$ ,  $C_2$ , and  $C_3$  are random vectors, and X is the position of the current wolf. Now, we define

$$X_1 = X_{\alpha} - A_1 \cdot (D_{\alpha})$$
  

$$X_2 = X_{\beta} - A_2 \cdot (D_{\beta})$$
  

$$X_3 = X_{\delta} - A_3 \cdot (D_{\delta}).$$
(43)

Then, the position of the wolf in the next step can be calculated as

$$X(k+1) = \frac{X_1 + X_2 + X_3}{3} \tag{44}$$

Therefore, the current grey wolf will move around the prey and get close to it. In Equation (41), *A* is a random factor in the range of (-2a, 2a); when |A| is greater or equal to one, the grey wolf will keep searching for a better solution, and when |A| is smaller than one, then the grey wolf will be forced to attack the prey. This method can avoid the local optimization problem.

## 3.6.2. Improvement in PSO Algorithm

#### Improvement in inertial weight

In the PSO algorithm, inertial weight is considerably important for reaching an optimal trade-off between local search and global search, determining how the movements of the previous particle will be propagated to the current particle. On one hand, low values will reduce the influences of previous particles, encouraging the current particle to search in different directions and regions. This kind of behavior can be regarded as exploitation. On the other hand, high values will keep the current particle searching in the same direction. Compared with the behaviors of low values, this process can be considered as an exploration. In order to reach an ideal compromise between the search accuracy and the search speed, we propose a three-stage inertial weight update law as (45):

$$w(k) = \begin{cases} w_{\text{start}}, & \left(0 < k \leq \frac{1}{3}k_{\max}\right) \\ (w_{\text{start}} - w_{\text{end}} + 1) \operatorname{rand}(1) + 0.3, & \left(\frac{1}{3}k_{\max} < k \leq \frac{2}{3}k_{\max}\right) \\ w_{\text{end}}, & \left(\frac{2}{3}k_{\max} < k \leq k_{\max}\right) \end{cases}$$
(45)

where  $w_{\text{start}}$  is a relatively large value,  $w_{\text{end}}$  is a relatively small value, rand(1) is a random value in the range of 0 to 1, k is the number of current iteration, and  $k_{\text{max}}$  is the maximum iteration number. As shown in (45), the particles are initialized with a relatively high inertial weight value at the early stage  $\left(0 < k \leq \frac{1}{3}k_{\text{max}}\right)$ , ensuring a fast search for the optimal value. Meanwhile, the assigned high initial values prevent the particle weight from becoming too low in the later iterations and falling into the local optimum. During the middle phase  $\left(\frac{1}{3}k_{\text{max}} < k \leq \frac{2}{3}k_{\text{max}}\right)$ , we adopt a random update approach for the inertial weight to achieve a balanced exploratory and a smooth exploitative search procedure. The core of the weight update function here consists of an assigned interval bounded by  $w_{\text{start}}$  and  $w_{\text{end}}$  and its multiplier random number  $rand(1) \in (0, 1)$ . The proposed update law can help to avoid premature convergence. In the close-out iterations  $\left(\frac{2}{3}k_{\text{max}} < k \leq k_{\text{max}}\right)$ , the particle weight values will be reduced to a relatively low value, maintaining a conservative search for the optimal solution. The proposed three-stage inertial weight update law allows for an enhanced optimization performance.

#### Improvement in learning factor

 $c_1$  can be interpreted as the impact of each particle's previous experiences, while  $c_2$  is the influence of global learning. If  $c_1 > 0$ ,  $c_2 = 0$ , all particles perform the search independently. In the circumstance of a high value  $c_1$  and a low value  $c_2$ , the particles focus mainly on searching for their best solutions, leading to divergence. In contrast, if  $c_1 = 0$ ,  $c_2 > 0$ , the particles will all reach the same solution. Combining a low value  $c_1$  and a high value  $c_2$  will result in a sub-optimal solution in the search. Inappropriate

learning factor value will lead to poor performance of the PSO algorithm. Therefore, the new learning factor is designed as (46) and (47).

$$c_1(k) = e^{\left(\frac{P_i(k) - P_g(k)}{|P_g(k) + \varepsilon|}\right)}$$
(46)

$$c_2(k) = e^{\left(-p_i(k) + p_g(k)\right)} \tag{47}$$

where  $p_i(k)$  is the historical individual optimum in the k-th iteration,  $p_g(k)$  is the population optimum in the *k*-th iteration, and  $\varepsilon$  is a small constant value. We adopt  $p_g$  as a penalty term in the update function of  $c_1(k)$  to control its value within a reasonable interval and vice versa. The exponential form is introduced to increase the sensitivity to the variations in the fitness values.

#### 3.6.3. Improvement in GWO Algorithm

Traditional grey wolf algorithms converge slowly and tend to fall into local optima. In the flow of the algorithm, the algorithm keeps selecting the three best-positioned wolves in the pack as the alpha wolf, and the pack approaches the direction of the alpha wolf. However, the wolves do not have a concept of speed when searching for prey, which means that the wolves are likely to travel too fast as they approach the prey and fail to find better prey as they go. Additionally, grey wolves do not keep track of where they have reached individually during their travels. These problems will tend to lead traditional grey wolf algorithms into local optimal problems. In the PSO algorithm, the particles update their velocity and position based on the individual optimum and the global optimum. Therefore, inspired by the PSO algorithm, in this study, the grey wolves will have memory, that is, they will calculate their own optimal solution during the hunting process and calculate the solution with the best position in the whole population, the grey wolves will travel at a certain range of speed, and the whole process can be described as follows:

$$v_{i}(k) = X_{p}(k) - A \cdot |C \cdot X_{p}(k) - X_{i}(k)|$$
  

$$X_{i}(k+1) = X_{i}(k) + v_{i}(k)$$
(48)

where  $v_i(k)$  is the velocity of the *i*th wolf,  $X(k)_g$  is the global best solution of all the wolves, and  $X(k)_{ip}$  is the personal best solution of the *i*th wolf.  $X_p(k)$  is the leading wolves' solution, which is a value calculated from the global best solution and the personal best solution of a current wolf, which is described as follows:

$$X_p(k) = \frac{X_g(k) + X_{ip}(k)}{2}$$
(49)

3.6.4. Optimization of Controller

The cost function is designed as

$$J = \int_0^\infty a |e_{\psi,1}(t)| + b|\delta_d(t)| + c|e_{v,1}(t)| + d|F_d(t)|dt$$
(50)

where a, b, c, d > 0. Additionally, the optimization objective is the control parameters including  $k_{\psi,1}, k_{\psi,2}, k_{\psi,3}, k_{v,1}, k_{v,2}, k_{v,3}, \kappa_1, \kappa_2, \alpha_1, \beta_1, \alpha_2$ , and  $\beta_2$ .

With these strategies, we use the proposed optimization algorithm to optimize the controller. The optimization process will be illustrated by the following pseudo-code (see Algorithms 1 and 2):

Algorithm 1: IPSO.
Initialize all parameters and values;
<b>foreach</b> <i>Iteration</i> k <b>do</b>
foreach Particle i do
Update $w(k)$ as (45);
Update $c_1(k)$ , $c_2(k)$ as (46), (47);
Update velocity and position of particles as (40);
Calculate fitness value as (50);
Compare the fitness value with $p_i$ ;
Update fitness value;
Set the best fitness value as $p_g$

# Algorithm 2: PGWO.

Initialize all parameters and values;
foreach Iteration k do
foreach Wolf i do
Update $X_p$ as (49);
Update velocity and position of particles as (48);
Caculate fitness value as (48);
Compare the fitness value with $X_{ip}$ ;
Update fitness value;
Set the best fitness value as $X_g$

# 4. Simulation Results and Discussion

In this section, leveraging Matlab/Simulink, the performance of the proposed controller is evaluated through three representative examples. The proposed control algorithm was also compared against two different types of controllers, including a PID controller and an SMC controller.

The parameters of this control scheme are listed in Table 1.

Parameter	Value	Units	Parameter	Value	Units
$k_{\psi,1,min}$	0	_	$\alpha_{1,min}$	0	_
$k_{\psi,1,max}$	0.005	_	$\alpha_{1,max}$	1	_
$k_{\psi,2,min}$	0	_	$\beta_{1,min}$	0	_
$k_{\psi,2,max}$	10	_	$\beta_{1,max}$	1	—
$k_{\psi,3,min}$	0	—	$\alpha_{2,min}$	0	—
$k_{\psi,3,max}$	10	—	$\alpha_{2,max}$	1	—
$k_{v,1,min}$	0	_	$\beta_{2,min}$	0	_
$k_{v,1,max}$	0.005	_	$\beta_{2,max}$	1	_
$k_{v,2,min}$	0	_	w <sub>start</sub>	0.8	_
k <sub>v,2,max</sub>	10	_	$w_{end}$	0.5	_
$k_{v,3,min}$	0	_	$k_{max}$	20	_
k <sub>v,3,max</sub>	10	_	ε	$e^{-50}$	_
а	10,000	_	b	0.1	_
С	0.1	_	d	0.001	_

Table 1. Main parameters of the controller.

# Simulation on a Complex Trajectory

The test scenario is a complicated trajectory. We set the velocity of the controlled vehicle to 20 m/s. We compared the proposed method with other widely used algorithms such as cuckoo search optimization-based Sliding mode control (CO-SM), sliding mode

control (SM), backstepping control (BP), PD control, and PID control. The trajectory tracking result is shown in Figure 3. The cost function value of different evolutionary algorithms is shown in Figure 4. Figure 5a–c demonstrates the angular orientation error and the location deviation.



Figure 3. Trajectory tracking comparison.



Figure 4. The cost function value.



orientation error, and (c) location deviation.

In Figure 3, it can be seen that the optimized controllers have better performances than the traditional controllers. The PGWO-SM controller has the smoothest and best response. In second and third place are the IPSO-SM and CO-SM controllers. In Figure 4, the cost function value of PGWO-SM is more optimal than IPSO-SM and CO-SM. After three iterations, the PGWO-SM controller has already found the optimal solution, but IPSO-SM and CO-SM can only find their optimal solution at about 20 iterations. Then, it is followed by the unoptimized controllers, in the following order: SM, BP, PID, and PD controllers. Figure 5a,b demonstrate that the optimized controller is more stable than the other controllers in terms of angular orientation. The angular orientation of the PGWO-SM and IPSO-SM controllers are smoother than the SMC and PID controllers, and the angular orientation error of PGWO-SM converges within 1.5 s, which is faster than IPSO-SM and CO-SM, and is significant faster than the other unoptimized controllers; when the vehicle is steered, the error is typically within 0.05 degrees for PGWO-SM and within 0.1 degrees for IPSO-SM and CO-SM. However, the errors for the other traditional controllers are between 0.2 and 0.6 degrees, much greater than the errors for the optimized controller. The local deviation is shown in Figure 5c. When the vehicle is steering at high speed, the PGWO-SM local deviation is usually in the range of 0–0.3 m, with significantly more minor fluctuations in amplitude and duration than the other algorithms. The simulation results show that the optimized controller is significantly more responsive and robust than the conventional SM and PID controllers, while the PGWO-SM controller performs better than the IPSO-SM and CO-SM controllers. To describe the result more clearly, the results are quantified and shown in Table 2. The mean position error and maximal error are calculated here, and this result further proves that our proposed controller is better and can improve the robustness of the system.

Controller	Mean Position Error (m)	Maximal Position Error (m)
PGWO-SM	0.0434	0.6229
CO-SM	0.0640	1.0375
IPSO-SM	0.0529	0.8137
SM	0.0942	2.2759
BP	0.1161	2.2262
PD	0.1507	2.7368
PID	0.1511	2.7215

Table 2. Comparison of mean value and maximal position error for the simulation results.

In summary, to simulate a complex and hazardous driving scenario, we designed a challenging curved trajectory and controlled the vehicle to travel at high speeds. However, the proposed controller achieves excellent trajectory tracking performance even in such scenarios by minimizing deviations between real and desired trajectories, ensuring rapid convergence even in the presence of large disturbances. Moreover, the controller produces smooth steering angle changes throughout the vehicle's journey, making it highly suitable for practical vehicle control applications in challenging road conditions.

#### 5. Conclusions

We summarized the contributions of the proposed approach as follows: In order to achieve high tracking accuracy in different trajectories, a vector-field-based adaptive sliding mode controller has been presented and described by the bicycle kinematic model. In the proposed method, the vector field guidance law was successfully expanded from simple trajectories to different curves. Additionally, separate control loops for orientation angle and velocity were proposed based on the sliding mode principle. Moreover, two optimization algorithms named IPSO and PGWO were introduced to optimize the sliding mode controller so that the controller was able to output the optimal orientation angle and velocity. We conducted a comparison with the other four algorithms, and the simulation results illustrated that the proposed controller had a faster, smoother, and more precise response compared with the traditional controllers. However, evolutionary algorithms are more demanding in terms of arithmetic power. Furthermore, in this paper, the gain in the controller is searched by the optimization algorithm offline. In the future, more efficient optimization algorithms should be developed, and an online optimization algorithm-based controller for trajectory tracking should be built. Integrated with the research of the longitudinal control approaches, we are going to apply the proposed scheme in a collision avoidance system and eventually apply it to a real car.

Author Contributions: Methodology, F.X., G.L. and Y.-R.C.; Software, F.X. and G.L.; Validation, F.X. and G.L.; Formal analysis, F.X. and G.L.; Investigation, F.X.; Data curation, G.L.; Writing—original draft, F.X. and G.L.; Writing—review & editing, Y.-R.C.; Visualization, G.L.; Project administration, Y.-R.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

**Data Availability Statement:** The data used to support the findings of this study are available from the corresponding author upon request.

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

#### References

- 1. Anderson, J.; Nidhi, K.; Stanley, K.; Sorensen, P.; Samaras, C.; Oluwatola, O. *Autonomous Vehicle Technology: A Guide for Policymakers;* Rand Corporation: Santa Monica, CA, USA, 2014.
- Khodayari, A.; Ghaffari, A.; Ameli, S.; Flahatgar, J. A historical review on lateral and longitudinal control of autonomous vehicle motions. In Proceedings of the 2010 International Conference On Mechanical And Electrical Technology, Singapore, 10–12 September 2010; pp. 421–429.
- Levinson, J.; Askel, J.; Becker, J.; Dolson, J.; Held, D.; Kammel, S.; Kolter, J.; Langer, D.; Pink, O.; Pratt, V. Others Towards fully autonomous driving: Systems and algorithms. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; pp. 163–168.
- 4. Li, D.; Du, L. Auv trajectory tracking models and control strategies: A review. J. Mar. Sci. Eng. 2021, 9, 1020. [CrossRef]
- 5. Zhang, X.; Tao, R.; Dou, Z.; Qiu, S.; Liu, J.; Liu, Y. Fast two-vector-based model-free predictive current control of permanent magnet synchronous motor. *IET Electr. Power Appl.* **2022**, *17*, 217–231. [CrossRef]
- Han, G.; Fu, W.; Wang, W.; Wu, Z. The lateral tracking control for the intelligent vehicle based on adaptive PID neural network. Sensors 2017, 17, 1244. [CrossRef] [PubMed]
- Potluri, R.; Singh, A. Path-tracking control of an autonomous 4WS4WD electric vehicle using its natural feedback loops. *IEEE Trans. Control Syst. Technol.* 2015, 23, 2053–2062. [CrossRef]
- Al-Mayyahi, A.; Wang, W.; Birch, P. Path tracking of autonomous ground vehicle based on fractional order PID controller optimized by PSO. In Proceedings of the 2015 IEEE 13th International Symposium on Applied Machine Intelligence and Informatics (SAMI), Herl'any, Slovakia, 22–24 January 2015; pp. 109–114.
- Zennir, Y.; Allou, S. Comparison of PID and fuzzy controller for path tracking control of autonomous electrical vehicles. In Proceedings of the 2018 International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM), Algiers, Algeria, 28–31 October 2018; pp. 1–6.
- Allou, S.; Zennir, Y.; Belmeguenai, A. Fuzzy logic controller for autonomous vehicle path tracking. In Proceedings of the 2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Monastir, Tunisia, 21–23 December 2017; pp. 328–333.
- Zakaria, M.A.; Zamzuri, H.; Mamat, R.; Mazlan, S.A.; Abd Rahman, M.A.; Abd Rahman, A.H. Dynamic curvature path tracking control for autonomous vehicle: Experimental results. In Proceedings of the 2014 International Conference on Connected Vehicles and Expo (ICCVE), Vienna, Austria, 3–7 November 2014; pp. 264–269.
- 12. El Hajjaji, A.; Bentalba, S. Fuzzy path tracking control for automatic steering of vehicles. *Robot. Auton. Syst.* **2003**, *43*, 203–213. [CrossRef]
- 13. Yao, Q.; Tian, Y.; Wang, Q.; Wang, S. Control Strategies on Path Tracking for Autonomous Vehicle: State of the Art and Future Challenges. *IEEE Access* 2020, *8*, 61211–161222. [CrossRef]
- 14. Guo, J.; Li, K.; Luo, Y. Coordinated control of autonomous four wheel drive electric vehicles for platooning and trajectory tracking using a hierarchical architecture. *J. Dyn. Syst. Meas. Control* **2015**, 137, 101001. [CrossRef]
- 15. Ren, Y.; Zheng, L.; Khajepour, A. Integrated model predictive and torque vectoring control for path tracking of 4-wheel-driven autonomous vehicles. *IET Intell. Transp. Syst.* **2019**, *13*, 98–107. [CrossRef]
- 16. Ren, D.; Zhang, J.; Zhang, J. Sliding mode control for vehicle following with parametric uncertainty. *Electr. Mach. Control* **2010**, *14*, 73–78.

- 17. Jiang, L.; Yang, J. Path tracking of automatic parking system based on sliding mode control. *Trans. Chin. Soc. Agric. Mach* 2019, 50, 356–364.
- 18. He, X.; Liu, Y.; Lv, C.; Ji, X.; Liu, Y. Emergency steering control of autonomous vehicle for collision avoidance and stabilisation. *Veh. Syst. Dyn.* **2019**, *57*, 1163–1187. [CrossRef]
- 19. Guo, J.; Hu, P.; Wang, R. Nonlinear Coordinated Steering and Braking Control of Vision-Based Autonomous Vehicles in Emergency Obstacle Avoidance. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3230–3240. [CrossRef]
- 20. Matveev, A.; Hoy, M.; Katupitiya, J.; Savkin, A. Nonlinear sliding mode control of an unmanned agricultural tractor in the presence of sliding and control saturation. *Robot. Auton. Syst.* **2013**, *61*, 973–987. [CrossRef]
- 21. Zhang, L.; Wang, S.; Bai, J. Fast-super-twisting sliding mode speed loop control of permanent magnet synchronous motor based on SVM-DTC. *IEICE Electron. Express* **2021**, *18*, 20200375. [CrossRef]
- 22. Nguyen, L.; Phung, M.; Ha, Q. Iterative Learning Sliding Mode Control for UAV Trajectory Tracking. *Electronics* **2021**, *10*, 2474. [CrossRef]
- 23. Zhang, L.; Bai, J.; Wu, J. Speed sensor-less control system of surface-mounted permanent magnet synchronous motor based on adaptive feedback gain supertwisting sliding mode observer. J. Sens. 2021, 2021, 8301359. [CrossRef]
- Zhang, L.; Bai, J.; Wu, J. SPMSM sliding mode control based on the new super twisting algorithm. *Complexity* 2021, 2021, 2886789. [CrossRef]
- 25. Nie, Y.; Zhang, M.; Zhang, X. Trajectory Tracking Control of Intelligent Electric Vehicles Based on the Adaptive Spiral Sliding Mode. *Appl. Sci.* **2021**, *11*, 11739. [CrossRef]
- Xia, X.; Hashemi, E.; Xiong, L.; Khajepour, A. Autonomous Vehicle Kinematics and Dynamics Synthesis for Sideslip Angle Estimation Based on Consensus Kalman Filter. *IEEE Trans. Control Syst. Technol.* 2022, 31, 179–192. [CrossRef]
- Liu, W.; Xiong, L.; Xia, X.; Lu, Y.; Gao, L.; Song, S. Vision-aided intelligent vehicle sideslip angle estimation based on a dynamic model. *IET Intell. Transp. Syst.* 2020, 14, 1183–1189. [CrossRef]
- Liu, W.; Xia, X.; Xiong, L.; Lu, Y.; Gao, L.; Yu, Z. Automated vehicle sideslip angle estimation considering signal measurement characteristic. *IEEE Sens. J.* 2021, 21, 21675–21687. [CrossRef]
- 29. Xiong, L.; Xia, X.; Lu, Y.; Liu, W.; Gao, L.; Song, S.; Yu, Z. IMU-based automated vehicle body sideslip angle and attitude estimation aided by GNSS using parallel adaptive Kalman filters. *IEEE Trans. Veh. Technol.* **2020**, *69*, 10668–10680. [CrossRef]
- 30. Xia, X.; Xiong, L.; Huang, Y.; Lu, Y.; Gao, L.; Xu, N.; Yu, Z. Estimation on IMU yaw misalignment by fusing information of automotive onboard sensors. *Mech. Syst. Signal Process.* **2022**, *162*, 107993. [CrossRef]
- Parivallal, A.; Sakthivel, R.; Wang, C. Guaranteed cost leaderless consensus for uncertain Markov jumping multi-agent systems. J. Exp. Theor. Artif. Intell. 2023, 35, 257–273. [CrossRef]
- 32. Fateh, M.; Khorashadizadeh, S. Optimal robust voltage control of electrically driven robot manipulators. *Nonlinear Dyn.* **2012**, *70*, 1445–1458. [CrossRef]
- 33. Bingül, Z.; Karahan, O. A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control. *Expert Syst. Appl.* **2011**, 38, 1017–1031. [CrossRef]
- Wong, C.; Wang, H.; Li, S. PSO-based Motion Fuzzy Controller Design for Mobile Robots. Int. J. Fuzzy Syst. 2008, 10, 284–292. [CrossRef]
- 35. Elkaranshway, H.; Bayoumi, E.; Soliman, H. PSO-based robust PID control for flexible manipulator systems. *Int. J. Model. Identif. Control.* **2011**, *14*, 1–12.
- 36. Mohamed, A. Solving large-scale global optimization problems using enhanced adaptive differential evolution algorithm. *Complex Intell. Syst.* **2017**, *3*, 205–231
- [CrossRef]
- 37. Mirjalili, S.; Lewis, A. Grey wolf optimizer. Adv. Eng. Softw. 2014, 69, 46-61. [CrossRef]
- 38. Sujit, P.; Saripalli, S.; Sousa, J. Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles. *IEEE Control Syst. Mag.* 2014, 34, 42–59. [CrossRef]
- 39. Nelson, D.; Barber, D.; McLain, T.; Beard, R. Vector field path following for miniature air vehicles. *IEEE Trans. Robot.* **2007**, 23, 519–529.
- 40. Griffiths, S. Vector field approach for curved path following for miniature aerial vehicles. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO, USA, 21–24 August 2006; p. 6467. [CrossRef]
- 41. Liang, Y.; Jia, Y.; Wang, Z.; Matsuno, F. Combined vector field approach for planar curved path following with fixed-wing UAVs. In Proceedings of the 2015 American Control Conference (ACC), Chicago, IL, USA, 1–3 July 2015; pp. 5980–5985.
- 42. Liang, Y.; Jia, Y.; Du, J.; Zhang, J. Vector field guidance for three-dimensional curved path following with fixed-wing UAVs. In Proceedings of the 2015 American Control Conference (ACC), Chicago, IL, USA, 1–3 July 2015; pp. 1187–1192.
- 43. Bishop, R. Intelligent vehicle applications worldwide. IEEE Intell. Syst. Their Appl. 2000, 15, 78-81.

**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.