Global tracking control for car-like mobile robots with zero-crossing driving velocity*

Kai Yan

Abstract— **This work proposes a smooth time-varying controller to address the trajectory tracking problems of car-like mobile robots. Currently, literature does not suggest globally asymptotically stable controllers solving this problem. Unlike the prototypical method of transforming the model into a nonholonomic chained-form system, the proposed method is designed based on the original tracking error equation, and therefore our approach does not have singularities of chained-form transformations. Opposing current methods, our control law satisfactorily addresses the cases where the vehicle's velocity passes through zero. In general, our redesigned control law has no singularities, which can satisfy the requirement that the vehicle's velocity can cross zero and at the same time have a global attraction region. The design of the controller is mainly divided into two steps. Firstly, the linear velocity and steering angle of the robot are regarded as control inputs, which are designed by making the derivative of a positive definite Lyapunov-like function semi-negative definite. In the next step, another control input is designed by the backstepping approach. Furthermore, the global convergence of the state trajectory to the reference one is strictly proved by Barbalat's Lemma. Finally, simulated and actual experiments on a car-like robot demonstrate the effectiveness of the proposed control scheme.**

I. Introduction

Mobile robots have attracted considerable attention in recent decades due to their wide application in the transportation, security, mining, and service industries. Owning to the nonholonomic constraints imposed by the rolling wheels, controlling mobile robots has been an important research topic.

The motion control problems of mobile robots are categorized by point-to-point motion [1],[2], path following [3] and trajectory tracking [4]-[8]. Specifically, the trajectory tracking problems refer to following a given trajectory (a geometric path with an associated timing law) in Cartesian space by a reference point on the robot. Various control strategies have been used to design trajectory tracking controllers for mobile robots, including conventional PID to more advanced controllers such as feedback linearization control (FLC) [4], model predictive control (MPC) [5], sliding mode control (SMC) [6], and robust control [7]. For example, in [8], the authors improve the command-tracking performance induced by the WMR slippage by combing a slippage-dependent forward feedback controller and a traditional position-error-based controller.

*This work was supported by National Natural Science Foundation of China(No.62133001)

The authors are with The Seventh Research Division, School of Automation Science and Electrical Engineering, Beihang University, Beijing, P. R. China (e-mail: yankai2016@buaa.edu.cn)

However, most control algorithms focus on unmanned air vehicles [9], unicycle-type robots [10], or tractor-trailer vehicles [11]-[13]. With the rapid development of self-driving technology, research on control schemes for car-like models is becoming more meaningful. Typically, the car-like models utilize a transformed model of chained forms. In [4], [14], the point stabilization problem of the car-like robot is solved by utilizing a transformed model of chained forms. Although the transformation is generic and elegant, it is not globally defined because singularity problem may occur in the transformation from the kinematic equation to chained system. Furthermore, the chained-form transformations also lack to guarantee the dynamic control quality invariance between the chained-system configuration space and the vehicle task space where the motion is realized [15]. For example, the overshoot may be minimal in the chained-system, but it may be significant in the original tracking error equation.

Among various tracking controllers, feedback linearization control has been a canonical method. In [16] and [17], due to the decoupling matrix being singular, static feedback fails to solve the input-output linearization. Therefore, the authors added integrators to a subset of the input channels to overcome the singularity of the decoupling matrix. However, when the car-like robot's linear velocity equals zero, the decoupling matrix is still singular. Hence, this method is invalid when the car-like vehicle stops and reverses its motion direction. The work of [18] utilize Lyapunov's direct method to design the controller, which suffers from a similar problem, as it cannot address the linear velocity passing through zero.

In [19], an indirect method is adopted by redefining the system output, selecting the representative point in front of the robot instead of the rear axle's midpoint. Nevertheless, this scheme cannot control the robot's rear wheel to track a particular trajectory precisely, and the robot can only move forward. A general feedback control framework for the car-like robots has also been proposed [15], which applies the motion controllers dedicated to the unicycle model to realize the motion task for the car-like kinematics. This method still suffers from a singularity when the linear velocity equals zero. Alternative methods also have some limitations. For example, the feedback controller for car-like trajectory tracking based on standard linear control theory is simple and effective but only locally defined.

To the best of our knowledge, a global trajectory tracking controller for the kinematic model of car-like robots with rigorous mathematical proof has not been proposed yet.

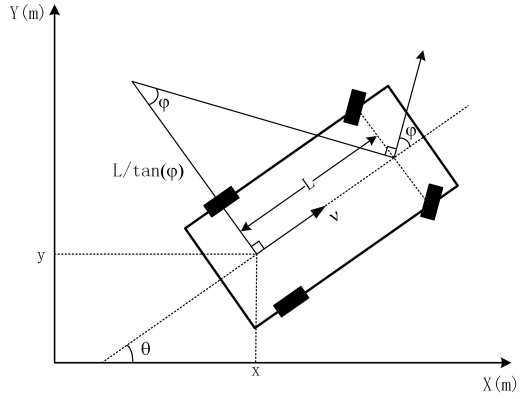


Fig. 1. The kinematic model of a car-like mobile robot.

Hence, our research fills this research gap and provides the corresponding rigorous proof. The proposed method does not utilize the transformed model of chained forms to avoid the locality of the chained transformations. Additionally, the proposed approach is not an improved variant of the dynamic feedback linearization method, which avoids the problem that the linear velocity cannot cross zero. Unlike current methods, this paper defines a positive definite function, and the proposed controller is designed such that the derivative of this positive definite function is semi-negative definite, which is a very primitive, straightforward, but efficient approach.

The main contribution of this paper is proposing for the first time a global trajectory tracking controller for the car-like vehicle, where the controller is still valid when the vehicle's velocity passes through zero. The existing controllers are either not global or require that the sign of robot velocity is invariant. Hence, compared with the existing results on nonholonomic car-like mobile robots, our controller can deal with a broader range of situations, such as tracking the reference trajectory of velocity zero-crossing or starting in arbitrary initial states. Furthermore, we provide the rigorous mathematical justifications of the global convergence based on Barbalat's Lemma [20]. In the simulations and experiments section, two other methods are discussed in comparison to the proposed method. We also conducted real robot experiments. Both simulation results and real experiments demonstrate the effectiveness and superiority of our approach.

The remainder of this paper is organized as follows. Section II formulates the tracking problem, and Section III presents the controller and mathematical proofs. Section IV provides the results of the simulation examples and the actual experiments, and Section V concludes this work and presents some future research directions.

II. Problem formulation

A. Kinematic model of the car-like robot

The kinematic mechanism of an automobile is the Ackermann model, which is often simplified to a two-wheel model called the car-like model used in this article. As illustrated in Fig. 1, the front wheel can be steered while the rear wheel orientation is fixed. Provided that the robot complies with ideal nonholonomic constraints, it affords a pure wheel rolling without slipping. In this paper, we consider the rear wheel drives the vehicle, and the kinematic equations of the car-like robot concerning the position of the rear wheels' axle center are [4]:

$$\dot{x} = v\cos\theta,\ \dot{y} = v\sin\theta,\ \dot{\theta} = \frac{v}{L}\tan\varphi,\ \dot{\varphi} = w \qquad (1)$$

where x, y and θ are the rear wheel's cartesian coordinates and the yaw angle in a fixed frame, and φ represent the steering angle as depicted in Fig.1. v and w are control inputs, and L is the distance between the front and rear wheels.

B. Description of trajectory tracking problem

In the trajectory tracking task, the robot must follow the desired trajectory, which is feasible when the kinematic model of the same car-like robot can generate it. Therefore, we design a reference trajectory generated by the same kinematic model:

$$\dot{x}_r = v_r\cos\theta_r,\ \dot{y}_r = v_r\sin\theta_r,\ \dot{\theta}_r = \frac{v_r}{L}\tan\varphi_r,\ \dot{\varphi}_r = w_r \qquad (2)$$

where $x_r, y_r, \theta_r, \varphi_r$ denote the status of the reference robot and v_r, w_r are the reference control inputs. Accordingly, if the reference model's initial state and control inputs are known, the configurations of the reference trajectory will be uniquely determined.

We define the tracking error as the coordinates of the reference robot in the body coordinate system:

$$\begin{pmatrix} x_e \\ y_e \\ \theta_e \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{pmatrix} \qquad (3)$$

Deriving (3) in terms of time, the following differential equations of tracking error are obtained:

$$\begin{cases} \dot{x}_e = -v + v_r\cos\theta_e + \dfrac{1}{L}y_e v\tan\varphi \\ \dot{y}_e = v_r\sin\theta_e - \dfrac{1}{L}x_e v\tan\varphi \\ \dot{\theta}_e = \dfrac{1}{L}v_r\tan\varphi_r - \dfrac{1}{L}v\tan\varphi \end{cases} \qquad (4)$$

The objective of the trajectory tracking task is to achieve $\lim\limits_{t\to\infty}(x_r - x) = \lim\limits_{t\to\infty}(y_r - y) = \lim\limits_{t\to\infty}(\theta_r - \theta) = 0$. As a result of the invertibility of coordinate transformations (3), the objective is equivalent to $\lim\limits_{t\to\infty} x_e = \lim\limits_{t\to\infty} y_e = \lim\limits_{t\to\infty} \theta_e = 0$.

Consequently, the trajectory tracking problem can be stated as: given the control inputs v_r, w_r of the reference model (2), for the mobile robot system (1) a feedback control law $v(\cdot)$ and $w(\cdot)$ should be designed ensuring the asymptotic vanishing of the tracking error (x_e, y_e, θ_e).

III. Main result

This part first designs the controller and then studies the stability of the entire system.

A. Controller synthesis

It can be noticed from Fig.1 that the curvature of the rear wheel trajectory is $\tan\varphi/L$. For the sake of brevity, we denote u, u_r as the rear wheel curvatures of the real and reference car-like robots, respectively, i.e.,

$$u = \frac{\tan\varphi}{L}, \quad u_r = \frac{\tan\varphi_r}{L} \tag{5}$$

Substituting (5) into (4) results in:

$$\begin{cases} \dot{x}_e = -v + v_r\cos\theta_e + y_e u v \\ \dot{y}_e = v_r\sin\theta_e - x_e u v \\ \dot{\theta}_e = u_r v_r - u v \end{cases} \tag{6}$$

In this paper, the design idea of the controller is to take u and v as control inputs to asymptotically stabilize the system (6), and then we apply the backstepping method to obtain the input variable w. In order to design the control input u and v, the author in [21] consider the following positive definite function:

$$V_1 = 0.5(x_e^2 + y_e^2) + k_1(1-\cos\theta_e) \tag{7}$$

Its derivative along (4) is:

$$\dot{V}_1 = x_e(-v + v_r\cos\theta_e) - k_1 v\sin\theta_e\left(u - \frac{y_e v_r}{k_1 v} - \frac{v_r u_r}{v}\right)$$

To make $\dot{V}_1 \le 0$, the control inputs are derived [21]:

$$\begin{aligned} v_{d1} &= v_r\cos\theta_e + k_2 x_e \\ u_{d1} &= \frac{y_e v_r}{k_1 v} + \frac{v_r u_r}{v} + v\sin\theta_e \end{aligned} \tag{8}$$

where k_1, k_2 are all positive constants. Substitute $u = u_{d1}, v = v_{d1}$ into $\dot{V}_1$:

$$\dot{V}_1 = -k_2 x_e^2 - k_1(v_{d1}\sin\theta_e)^2 \le 0 \tag{9}$$

However, the velocity v appears in the denominator of (8). Hence, the controller has a drawback when the car-like vehicle stops and reverses its motion direction. Furthermore, the attractive region is not global because V_1 is not a radially unbounded function. Therefore, the control law faces the same problems as existing control laws, as mentioned earlier. Therefore, we design another controller by considering a new positive definite function:

$$V_2 = 0.5(x_e^2 + y_e^2 + \theta_e^2) \tag{10}$$

Computing its time derivative along with (6) and rearrange the resulting terms yields:

$$\begin{aligned} \dot{V}_2 =& (v_r - v)(x_e + u\theta_e) + v_r((x_e(\cos\theta_e - 1) + \\ & y_e\sin\theta_e + \theta_e(u_r - u))) \end{aligned} \tag{11}$$

We design the control inputs such that:

$$v_{d2} = v_r + k_1(x_e + u\theta_e) \tag{12}$$

$$u_{d2} = u_r + x_e f_1 + y_e f_2 + k_2 v_r\theta_e \tag{13}$$

where k_1, k_2 are positive constants. Terms f_1 and f_2 are given by:

$$f_1 = \begin{cases} \dfrac{\cos\theta_e - 1}{\theta_e}, & \text{if } \theta_e \ne 0 \\ 0, & \text{if } \theta_e = 0 \end{cases}, \quad f_2 = \begin{cases} \dfrac{\sin\theta_e}{\theta_e}, & \text{if } \theta_e \ne 0 \\ 1, & \text{if } \theta_e = 0 \end{cases} \tag{14}$$

By substituting $u = u_{d2}, v = v_{d2}$ into (11), $\dot{V}_2$ is reduced to $\dot{V}_2 = -k_1(x_e + u_{d2}\theta_e)^2 - k_2(v_r\theta_e)^2 \le 0$. It is worth noting that f_1 and f_2 are continuous functions with respect to the state θ_e. Since θ_e is a continuous function of time, the control input v and u must be continuous with regard to time.

The next step employs the backstepping method to obtain w. Substituting $v = v_{d2}$ into (11), we can get:

$$\begin{aligned} \dot{V}_2 =& (v_r - v_{d2})(x_e + u\theta_e) + v_r((x_e(\cos\theta_e - 1) + \\ & y_e\sin\theta_e + \theta_e(u_r - u_{d2} + u_{d2} - u))) \\ =& -k_1(x_e + u_{d2}\theta_e)^2 - k_2(v_r\theta_e)^2 + v_r\theta_e z \end{aligned} \tag{15}$$

where $z = u_{d2} - u$ and its derivative is:

$$\begin{aligned} \dot{z} =& \dot{u}_{d2} - \dot{u} \\ =& \dot{u}_r + k_2\dot{v}_r\theta_e + \dot{x}_e f_1 + \dot{y}_e f_2 + \\ & \dot{\theta}_e(x_e\frac{\partial f_1}{\partial\theta_e} + y_e\frac{\partial f_2}{\partial\theta_e} + k_2 v_r) - \frac{\sec^2\varphi}{L}\dot{\varphi} \\ =& H(v_r, u_r, \dot{v}_r, \dot{u}_r, x_e, y_e, \theta_e, z) - \frac{\sec^2\varphi}{L}w \end{aligned} \tag{16}$$

where $H(v_r, u_r, \dot{v}_r, \dot{u}_r, x_e, y_e, \theta_e, z)$ equals:

$$\begin{aligned} H =& \dot{u}_r + k_2\dot{v}_r\theta_e + ((v_r + k_1(x_e + (u_r + x_e f_1 + y_e f_2 + \\ & k_2 v_r\theta_e - z)\theta_e))(y_e(u_r + x_e f_1 + y_e f_2 + k_2 v_r\theta_e - z) \\ & - 1)v_r\cos\theta_e)f_1 + (v_r\sin\theta_e - x_e(v_r + k_1(x_e + (u_r \\ & + x_e f_1 + y_e f_2 + k_2 v_r\theta_e - z)\theta_e))(u_r + x_e f_1 + y_e f_2 + \\ & k_2 v_r\theta_e - z))f_2 + (u_r v_r - (v_r + k_1(x_e + u\theta_e))(u_r + \\ & x_e f_1 + y_e f_2 + k_2 v_r\theta_e - z))\left(x_e\frac{\partial f_1}{\partial\theta_e} + y_e\frac{\partial f_2}{\partial\theta_e} + k_2 v_r\right) \end{aligned} \tag{17}$$

And $\frac{\partial f_1}{\partial\theta_e}$ and $\frac{\partial f_2}{\partial\theta_e}$ are caculated as:

$$\begin{aligned} \frac{\partial f_1}{\partial\theta_e} &= \begin{cases} \dfrac{-\theta_e\sin\theta_e - \cos\theta_e + 1}{\theta_e^2}, & \text{if } \theta_e \ne 0 \\ -\dfrac{1}{2}, & \text{if } \theta_e = 0 \end{cases} \\ \frac{\partial f_1}{\partial\theta_e} &= \begin{cases} \dfrac{\theta_e\cos\theta_e - \sin\theta_e}{\theta_e^2}, & \text{if } \theta_e \ne 0 \\ -\dfrac{1}{2}, & \text{if } \theta_e = 0 \end{cases} \end{aligned} \tag{18}$$

For the following Lyapunov function candidate:

$$V_3 = V_2 + 0.5z^2 \tag{19}$$

Substituting (15)(16) into $\dot{V}_3$, the time derivative of V_3 yields:

$$\begin{aligned} \dot{V}_3 =& \dot{V}_2 + z\dot{z} \\ =& -k_1(x_e + u_{d2}\theta_e)^2 - k_2(v_r\theta_e)^2 + \\ & z\left(v_r\theta_e + H - \frac{\sec^2\varphi}{L}w\right) \end{aligned} \tag{20}$$

In order to make V_3 negative, the input variable w_d is designed as:

$$\begin{aligned} &v_r\theta_e + H - \frac{\sec^2\varphi}{L} w_d = -k_3 z \\ \Rightarrow &w_d = L\cos^2\varphi(H + v_r\theta_e + k_3 z) \end{aligned} \tag{21}$$

where k_3 is a positive constant. With the control $v = v_{d2}, w = w_d$, (20) can be reduced to:

$$\dot{V}_3 = -k_1(x_e + u\theta_e)^2 - k_2(v_r\theta_e)^2 - k_3 z^2 \leq 0 \tag{22}$$

B. Stability analysis

Proving the entire system's stability relies on the Barbalat's Lemma (Lemma 8.2 in [20]), stating that if $\phi : R \to R$ is a uniformly continuous function on $[0, \infty)$ and supposing that $\lim_{t\to\infty} \int_0^t \phi(\tau)d\tau$ exists and is finite, then, $\phi(t) \to 0$ as $t \to \infty$.

Theorem 1 The control law $v = v_{d2}, w = w_d$ (12)(21) guarantees that the tracking error (x_e, y_e, θ_e) globally uniformly converge to zero, provided the reference trajectory satisfies the following conditions:

C1: $v_r(t), \dot{v}_r(t), w_r(t), \dot{w}_r(t)$ are bounded.

C2: $\lim_{t\to\infty} v_r(t)$ does not exist or not convergent to zero.

Proof

Since V_3 is nonincreasing and bounded below, it converges to a non-negative value, i.e. $\lim_{t\to\infty} V_3 = V_{3lim}$ exists and is finite. Therefore, x_e, y_e, θ_e are bounded. According to (6)(12)(13)(16)(20) and condition C1, $u_d, v_d, u_r, v_r, \dot{u}, \dot{v}, \dot{u}_r, \dot{v}_r, \dot{x}_e, \dot{y}_e, \dot{\theta}_e, \dot{z}, \dot{V}_3, \ddot{V}_3$ are all bounded.

The boundness of $\ddot{V}_3$ indicates that $\dot{V}_3$ is uniformly continuous and $\lim_{t\to\infty} \int_0^t \dot{V}_3 dt = \lim_{t\to\infty} V_3$ exists and is finite. By applying Barbalat's lemma, $\lim_{t\to\infty} \dot{V}_3 = \lim_{t\to\infty}(-k_1(x_e + u\theta_e)^2 - k_2(v_r\theta_e)^2 - k_3 z^2) = 0$. Then we conclude that:

$$\lim_{t\to\infty}(x_e + u\theta_e) = 0$$

$$\lim_{t\to\infty} v_r\theta_e = 0 \tag{23}$$

$$\lim_{t\to\infty} z = 0 \tag{24}$$

As v_r is bounded, we obtain:

$$\begin{aligned} &\lim_{t\to\infty} v_r(x_e + u\theta_e) = 0 \\ \Rightarrow &\lim_{t\to\infty}(v_r x_e + v_r\theta_e u_{d2} - v_r\theta_e z) = 0 \\ \Rightarrow &\lim_{t\to\infty} v_r x_e = 0 \end{aligned} \tag{25}$$

Owning to the boundedness of $d^2(v_r^2\theta_e)/dt^2$, one can infer that $d(v_r^2\theta_e)/dt$ are uniformly continuous. As a result of $\lim_{t\to\infty} v_r\theta_e = 0 \Rightarrow \lim_{t\to\infty} v_r^2\theta_e \Rightarrow \lim_{t\to\infty} \int_0^t (d(v_r^2\theta_e)/dt)dt = -v_r^2(0)\theta_e(0)$, the limit $\lim_{t\to\infty} \int_0^t (d(v_r^2\theta_e)/dt)dt$ exists and is finite. Using Barbalat's Lemma again, we obtain:

$$\begin{aligned} &\lim_{t\to\infty} \frac{d}{dt}(v_r^2\theta_e) = \lim_{t\to\infty}(2v_r\dot{v}_r\theta_e + v_r^2\dot{\theta}_e) = 0 \\ \Rightarrow &\lim_{t\to\infty} v_r^2\dot{\theta}_e = 0 \end{aligned} \tag{26}$$

Substituting $\dot{\theta}_e$ into (26) results in:

$$\begin{aligned} &\lim_{t\to\infty} v_r^2(u_r v_r - u v_{d2}) = 0 \\ \Rightarrow &\lim_{t\to\infty} v_r^2(u_r v_r - (u_{d2} - z)v_{d2}) = 0 \\ \Rightarrow &\lim_{t\to\infty} v_r^2(u_r v_r - u_{d2} v_{d2}) = 0 \\ \Rightarrow &\lim_{t\to\infty} v_r^2(u_r v_r - (u_r + x_e f_1 + y_e f_2 + k_2 L v_r\theta_e) \\ &(v_r + k_1(x_e + \theta_e u_{d2}))) = 0 \\ \Rightarrow &\lim_{t\to\infty} v_r^3 y_e f_2 = 0 \end{aligned} \tag{27}$$

The next step is critical and for clarity it is divided into two cases.

- Case 1: $\theta_e(t) = 0$. According to (14), $f_2 = 1$. Then $\lim_{t\to\infty} v_r^3 y_e f_2 = \lim_{t\to\infty} v_r^3 y_e = 0$
- Case 2: $\theta_e(t) \neq 0$. Then $f_2 = \frac{\sin\theta_e}{\theta_e} \Rightarrow \lim_{t\to\infty} v_r^3 y_e f_2 = \lim_{t\to\infty} v_r^3 y_e \frac{\sin\theta_e}{\theta_e} = \lim_{t\to\infty}(v_r^3 y_e\theta_e \frac{\sin\theta_e - \theta_e}{\theta_e^2} + v_r^3 y_e) = 0$. As $v_r, y_e, \frac{\sin\theta_e - \theta_e}{\theta_e^2}$ are bounded, according to (23), we obtain $\lim_{t_2\to\infty} v_r\theta_e \cdot v_r^2 y_e \frac{\sin\theta_e - \theta_e}{\theta_e^2} = 0$. Therefore, $\lim_{t\to\infty} v_r^3 y_e = 0$

In summary, we conclude $\lim_{t\to\infty} v_r^3 y_e = 0$. Combining (23)(24)(25), we obtain:

$$\begin{aligned} &\lim_{t\to\infty} v_r x_e \cdot v_r^2 x_e = \lim_{t\to\infty} v_r\theta_e \cdot v_r^2\theta_e = \lim_{t\to\infty} v_r^3 y_e \cdot y_e = \\ &\lim_{t\to\infty} v_r^3 z^2 = 0 \Rightarrow \lim_{t\to\infty} v_r^3(x_e^2 + y_e^2 + \theta_e^2 + z^2) = \\ &\lim_{t\to\infty} v_r^3 V_{3lim} = 0 \end{aligned} \tag{28}$$

Considering the condition C2: $\lim_{t\to\infty} v_r(t) \neq 0$ or does not exist, one can infer that:

$$\begin{aligned} &V_{3lim} = \lim_{t\to\infty}(x_e^2 + y_e^2 + \theta_e^2 + z^2) = 0 \\ \Rightarrow &\lim_{t\to\infty} x_e = \lim_{t\to\infty} y_e = \lim_{t\to\infty} \theta_e = \lim_{t\to\infty} z = 0 \end{aligned} \tag{29}$$

The previous analysis holds globally. Therefore, the tracking error is globally uniformly convergent to zero, achieving the objective of the global trajectory tracking problem.

Remark 1 Before using the backstepping approach, the design of v_{d2}, u_{d2} are not unique. For example, if the positive definite function V_2 is set as $V_2 = \sqrt{x_e^2 + y_e^2 + \epsilon^2} - \epsilon + 0.5\theta_e^2$ $(0 < \epsilon < 1)$, computing its time derivative yields $\dot{V}_2 = (v_r - v)\left(\frac{x_e}{\sqrt{x_e^2 + y_e^2 + \epsilon^2}} + u\theta_e\right) + v_r\left(\frac{x_e(\cos\theta_e - 1)}{\sqrt{x_e^2 + y_e^2 + \epsilon^2}} + \frac{y_e\sin\theta_e}{\sqrt{x_e^2 + y_e^2 + \epsilon^2}} + \theta_e(u_r - u)\right)$. Therefore, we can redesigned v_{d2}, u_{d2} as:

$$v_{d2} = v_r + \frac{k_1 x_e}{\sqrt{x_e^2 + y_e^2 + \epsilon^2}} + k_1 u\theta_e \tag{30}$$

$$u_{d2} = u_r + \frac{x_e f_1}{\sqrt{x_e^2 + y_e^2 + \epsilon^2}} + \frac{y_e f_2}{\sqrt{x_e^2 + y_e^2 + \epsilon^2}} + k_2 v_r\theta_e \tag{31}$$

The method for calculating w_d and the proof of stability are almost identical with the above process, and will not be

repeated here. Compared to (12)(13), (30)(31) adds a gain with a saturation value to y_e. The advantage is that it can speed up the convergence of y_e, and the effect can be seen in the following experiments.

IV. Simulations and Experiments

This section verifies the effectiveness of the proposed control laws by numerical simulations and experiments.

TABLE I
Time spent for MSE to be less than 0.01

	case 1	case 2
$k_1 = 1, k_2 = 1, k_3 = 1$	6.372s	41.910s
$k_1 = 3, k_2 = 3, k_3 = 3$	3.318s	17.531
$k_1 = 10, k_2 = 10, k_3 = 10$	17.551s	5.780s
$k_1 = 22, k_2 = 22, k_3 = 22$	39.286	3.132s
$k_1 = 30, k_2 = 30, k_3 = 30$	53.725	6.752s

* Case 1: the reference trajectory is $x_r(t) = 2\cos t, y_r(t) = 2\sin t$. The initial states are $x(0) = -3, y(0) = -3, \theta(0) = 0, \phi(0) = 0$.

* Case 2: the reference trajectory is $x_r(t) = 0.7\cos 0.4t, y_r(t) = 0.7\sin 0.4t$. The initial states are $x(0) = -2, y(0) = -2, \theta(0) = 0, \phi(0) = 0$.

A. Simulation results

The numerical tests are conducted on a car-like robot with $L = 0.15m$. First, in order to analyze the influence of the controller parameters on the convergence speed, we calculate the mean-square-error (MSE) in the form of $(x_e^2 + y_e^2 + \theta_e^2)^{\frac{1}{2}}$. Then, for different parameters k_1, k_2, k_3, we obtain the time spent for the MSE to be less than 0.01 and the corresponding results are reported in Table I.

In case 1, the reference trajectory is a circle with a $2m$ radius. When the control parameters are $k_1 = 3, k_2 = 3, k_3 = 3$, the convergence speed is the fastest. In case 2, the circular reference trajectory has a radius of $0.7m$, which converges fastest for $k_1 = 22, k_2 = 22, k_3 = 22$. We find that the smaller the curvature radius of the reference trajectory, the larger the optimal controller parameters. Then, in the following simulations and experiments, we utilize the appropriate parameters according to the data in the table.

To verify the effectiveness and superiority of the proposed approach, the faulty controller (8) mentioned in this paper and the dynamic feedback linearization methon in [4] will be discussed in comparison to our proposed method. Then, all competitor methods will track the same reference trajectory under the same initial conditions. To intuitively compare the control input of all competitor methods, the control input w of the first method (8) mentioned in this paper is also obtained through the backstepping method(without presenting the expression in this paper). For the following trials, we simulate the following three scenes:

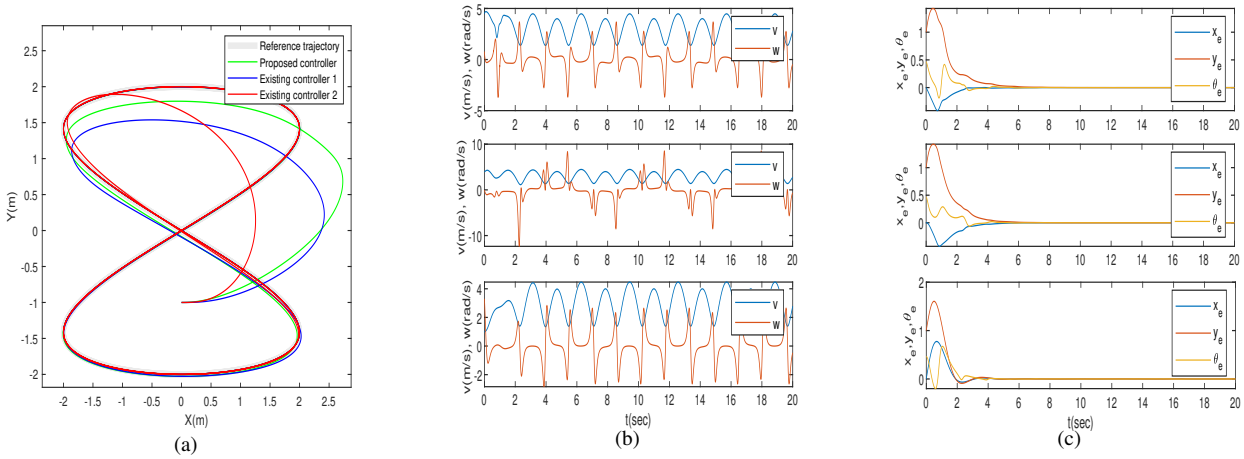
- Scene 1: Eight-shaped trajectory: $x_r(t) = 2\sin 2t, y_t(t) = 2\sin t$. The corresponding reference inputs are $v_r(t) = \sqrt{16\cos^2 2t + 4\cos^2 t}, w_r = Lv_r((32\cos 2t\cos t - 8\cos 2t\cos t)v_r^2 - 3(16\sin 2t\cos t - 8\sin t\cos 2t)(-32\cos 2t\sin 2t - 4\cos t\sin t))/(v_r^6 + L^2(16\sin 2t\cos t - 8\sin t\cos 2t)^2)$. The initial states of the real robot are $x(0) = 0, y(0) = -1, \theta(0) = 0, \phi(0) = 0$.
- Scene 2: Eight-shaped trajectory: $x_r(t) = 2\sin 2t, y_t(t) = 2\sin t$. The corresponding reference inputs are $v_r(t) = \sqrt{16\cos^2 2t + 4\cos^2 t}, w_r = Lv_r((32\cos 2t\cos t - 8\cos 2t\cos t)v_r^2 - 3(16\sin 2t\cos t - 8\sin t\cos 2t)(-32\cos 2t\sin 2t - 4\cos t\sin t))/(v_r^6 + L^2(16\sin 2t\cos t - 8\sin t\cos 2t)^2)$. The initial states of the real robot are $x(0) = 0, y(0) = -1, \theta(0) = -\pi, \phi(0) = 0$.
- Scene 3: Reciprocating straight-line trajectory: $x_r(t) = 2\sin t, y_t(t) = 0$. The corresponding reference inputs are $v_r(t) = 2\cos t, w_t(t) = 0$. The initial states of the real robot are $x(0) = 0, y(0) = -1, \theta(0) = 0, \phi(0) = 0$.

For convenience of description, we refer to the control law (8) in this paper as existing controller 1 and dynamic feedback linearization methon in [4] as existing controller 2. The expression of reference inputs v_r, u_r can be derive from (2), and details about existing controller 2 and the v_r, u_r calculation method can be found in [4]. The simulation results considering the robot trajectory and control inputs are illustrated in Fig.(2)-(4).

As depicted in Fig.2, the three controllers can successfully track the reference trajectory in Scene 1. In Scene 2, the initial state $\theta(0)$ changes from 0 to $-\pi$ compared to Scene 1. In this scene, existing controller 1 fails to track the reference trajectory, and from Fig.3(b), it is evident that the simulation program for existing controller 1 stops running around the 5th second, and a substantial value occurs for the control input w. Because the linear velocity v of existing controller 1 passes through zero, resulting in the divergence of the control input w.

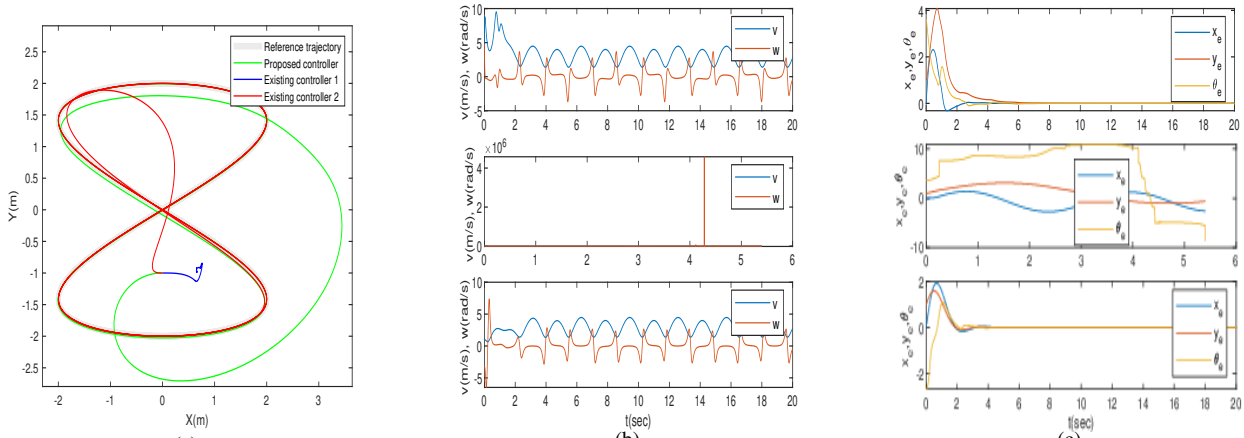
The reference trajectory of Scene 3 is a reciprocating straight-line trajectory, and the reference linear velocity is zero at both ends of the reference trajectory. Fig.3(b) reveals that existing controller 1 and 2 ended prematurely by 20 seconds due to the divergence of w. Therefore, when the linear velocity of the reference trajectory crosses zero, existing controller 1 and 2 are not appealing. By comparing Scene 1 and Scene 2, we find that even if the linear velocity of the reference trajectory will not cross zero, the linear velocity of the robot may also cross zero under some initial state, which makes existing controller 1 and 2 unusable.

In conclusion, our proposed controller performs well in all three simulation scenes. The tracking errors converge to zero quickly, and the resulting control inputs are continuous. In fact, for any feasible trajectory that satisfies conditions C1 and C2, the car-like robot can converge to the desired trajectory.



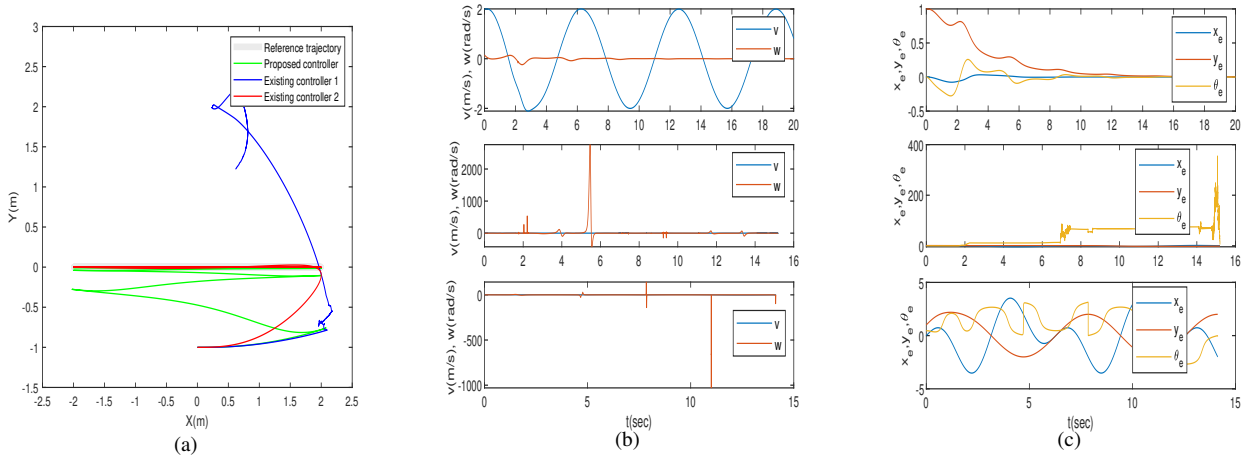
(a) (b) (c)

Fig. 2. Simulation results of Scene 1, (a): Robot trajectories under the three control laws. (b)(c): From top to bottom are the control inputs and errors of the proposed controller of (12)(21), existing controller 1 and existing controller 2, respectively. Existing controller 1 and existing controller 2 refer to the controller (8) and the dynamic feedback linearization method in [4], respectively.



(a) (b) (c)

Fig. 3. Simulation results of Scene 2, (a): Robot trajectories under the three control laws. (b)(c): From top to bottom are the control inputs and errors of the proposed controller of (12)(21), existing controller 1 and existing controller 2, respectively. Existing controller 1 and existing controller 2 refer to the controller (8) and the dynamic feedback linearization method in [4], respectively.



(a) (b) (c)

Fig. 4. Simulation results of Scene 3, (a): Robot trajectories under the three control laws. (b)(c): From top to bottom are the control inputs and errors of the proposed controller of (12)(21), existing controller 1 and existing controller 2, respectively. Existing controller 1 and existing controller 2 refer to the controller (8) and the dynamic feedback linearization method in [4], respectively.

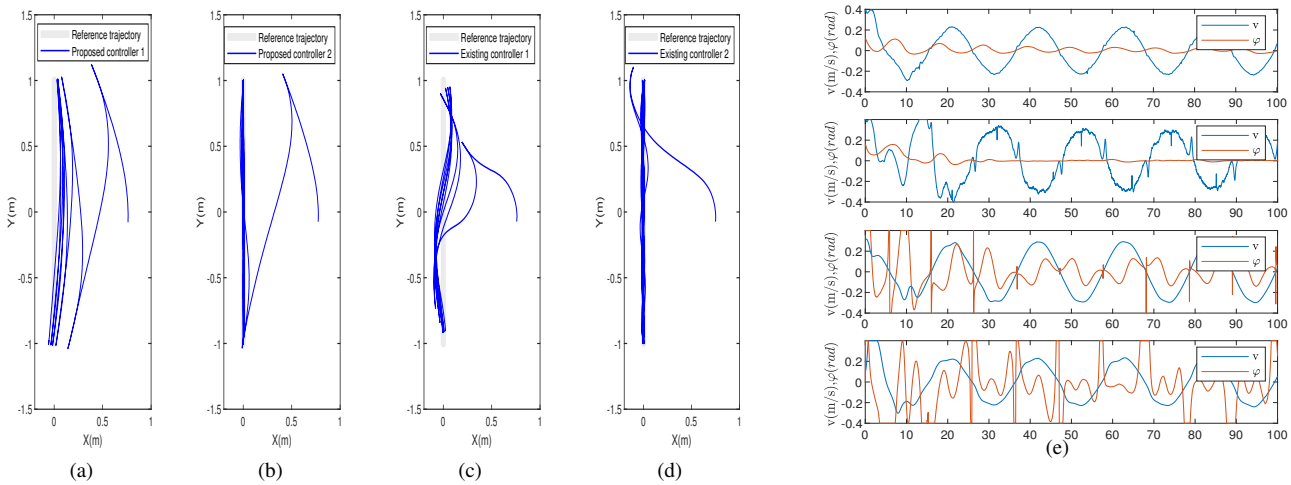


Fig. 5. Results of the real experiment: (a)(b)(c)(d) Robot trajectory under four controller. (e) From top to bottom are the control inputs of the proposed controller 1 of (12)(13), the proposed controller 2 of (30)(31), existing controller 1 and existing controller 2, respectively.

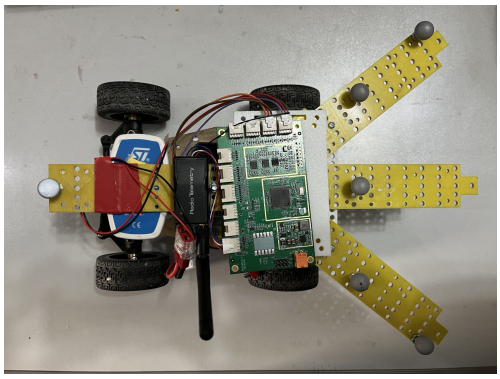


Fig. 6. The car-like robot used in the experiments.

B. Real experiments

The actual robot (see Fig.6) used for testing the control algorithms is a low-cost car-like robot built by WHEELTEC. Two drive motors attached to the robot's rear axle provide the translational speed, and a servo motor controls the steering angle. The motor control board equipped with the robot contains a PI controller, used to manage the motor in a low-level control layer. The translational speed and steering angle information are sent to the motor control board to control the robot. Therefore, we equip the robot with a main control board STM32F407VET6 to perform the proposed control law and send the control input to the motor control board through the serial port peripherals. The motor control board can generate pulse-width modulation (PWM) signals with different duty cycles according to the control inputs.

To obtain the robot's states, we employ the NaturalPoint OptiTrack indoor positioning system along with four near-infrared (IR) cameras and six IR reflectors. The position(x, y) and orientation θ are obtained from the personal computer (PC) running the OptiTrack software and are sent to the main control board of the robot through the wireless serial port module.

The real robot's parameter $L = 0.15m$ is the same as the simulation's. In order to compare the performance of the proposed method and other methods when the linear velocity crosses zero in the real experiments, we verified the simulation scenario of Fig.4 in the real experiment. Due to the limited lab space, the reference trajectory is set to $x_r(t) = 0, y_r(t) = \sin 0.3t$, and the corresponding reference inputs are $v_r(t) = 0.3\cos 0.3t, w_t(t) = 0$. The parameters of the proposed controllers are set to $k_1 = 2, k_2 = 5$. Since the steering angle of the real robot is driven directly by the servo, the control inputs (12)(13)(denoted as proposed controller 1) and (30)(31)(denoted as proposed controller 2) are used in the real experiment. Experiment results for the reciprocating straight reference trajectory are reported in Fig.5.

Fig.5 display the robot trajectory and control inputs under four controllers. Due to actuator saturation limitations, the range of the steering angle φ is limited to $[-0.4, 0.4]$. By comparing Fig.5(a)(b)(c)(d), it can be found that the proposed controller 1 (Fig.5(a)) has a poor convergence speed. According to Table I, when the radius of the reference trajectory is small, we can increase the value of k_1, k_2 to speed up the convergence. However, in the actual experiment, the experimental effect is not good with a larger k_1 due to the discontinuity of the controller. In comparison, the proposed controller 2 ($\epsilon = 0.1$ in **Remark 1** (30)(31)) does not have this problem. Under the same values of k_1, k_2, the convergence speed of the proposed controller 2 is significantly faster because we accelerate the convergence of error y_e in the proposed controller 2.

Compared with the simulation result in Fig.4, the real robot under existing controller 1 and 2 runs normally and no program errors occur. Because the time interval between two control commands is 0.05s, the generated linear velocity in the real experiment is not continuous. That is, the generated linear velocity is almost impossible to be exactly zero at a certain moment. Therefore, there is no program error in real

experiments.

However, in real experiments, existing controller 1 and 2 have other problems. Fig.5(e) shows the resulting control inputs of four controllers. We can find that the resulting steering angle φ of existing controller 1 and 2 reaches the maximum value or changes rapidly when the linear velocity v crosses zero, which may cause damage to the robot's mechanical structure. However, the steering angle φ produced by two proposed controllers have a small range of variation. From the expression (8) of existing controller 1, it can be seen that when the linear velocity is close to zero, $u = u_{d1}$ is very large. Therefore, the steering angle φ calculated by (5) is close to $\pm\pi/2$. Because of the steering angle saturation limitations, the range of φ is limited to $[-0.4, 0.4]$. In existing controller 2, a similar situation occurs. The linear velocity v appears in the denominator of the controller. In contrast, our proposed controllers do not have this problem, that is, no singularity occurs.

On the other hand, when the linear velocity v is close to zero, the steering angle φ generated by existing controller 1 and 2 is close to $\pm\pi/2$. Therefore, the control command can not be accurately executed because of the saturation limitations of the steering angle. However, the proposed controllers perform well without any problems. As long as the reference steering angle of the reference trajectory is also within the range $[-0.4, 0.4]$, the proposed control algorithm can perform the trajectory tracking task accurately.

V. Conclusion and Future Research

This paper designs a global asymptotic trajectory tracking controller for the kinematic model of a car-like mobile robot and provides rigorous mathematical proof of its global convergence. Existing controllers are either not global or require that the sign of the robot velocity is constant. Hence, this work fills the research gap on global trajectory tracking for car-like robots. Compared with the existing results on nonholonomic car-like mobile robots, our controller effectively deals with a broader range of situations, such as tracking the reference trajectory of velocity zero-crossing or starting in arbitrary initial states. In the simulations and real experiments, we challenge our scheme against two other methods verifying the effectiveness and superiority of our approach.

Of course, there still remain problems to be solved. For example, in real experiments, when the radius of the reference trajectory is small, there is a problem of slow convergence, which needs to be solved in future work. Furthermore, this paper only considers the model of car-like mobile robot. How to generalize the control algorithm to other type of mobile robot, like tractor-trailer robots and ships, is as well as challenging. In future work, the proposed trajectory tracking controller may be applied to path following problems and point stabilization problems of the car-like mobile robot.

References

[1] P. Panahandeh, K. Alipour, B. Tarvirdizadeh, and A. Hadi, "A kinematic lyapunov-based controller to posture stabilization of wheeled mobile robots," *Mechanical Systems and Signal Processing*, vol. 134, p. 106319, 2019.

[2] B. L. Ma and S. K. Tso, "Robust discontinuous exponential regulation of dynamic nonholonomic wheeled mobile robots with parameter uncertainties," *International Journal of Robust and Nonlinear Control*, vol. 18, no. 9, pp. 960–974, 2008.

[3] W. Yao, H. G. de Marina, B. Lin, and M. Cao, "Singularity-free guiding vector field for robot navigation," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1206–1221, 2021.

[4] A. De Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot motion planning and control*. Springer, pp. 171–253, 1998.

[5] B. Lindqvist, S. S. Mansouri, A.-a. Agha-mohammadi, and G. Nikolakopoulos, "Nonlinear mpc for collision avoidance and control of uavs with dynamic obstacles," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6001–6008, 2020.

[6] J. Mukherjee, S. Mukherjee, and I. N. Kar, "Sliding mode control of planar snake robot with uncertainty using virtual holonomic constraints," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1077–1084, 2017.

[7] Y. Jia, "Robust control with decoupling performance for steering and traction of 4ws vehicles under velocity-varying motion," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 3, pp. 554–569, 2000.

[8] W. Li, J. Guo, L. Ding, J. Wang, and H. Gao, "Slippage-dependent teleoperation of wheeled mobile robots on soft terrains," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4962–4969, 2021.

[9] S. Park, J. Deyst, and J. P. How, "Performance and lyapunov stability of a nonlinear path following guidance method," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 6, pp. 1718–1728, 2007.

[10] D. Soetanto, L. Lapierre, and A. Pascoal, "Adaptive, non-singular path-following control of dynamic wheeled robots," in *42nd IEEE International Conference on Decision and Control* , vol. 2, 2003, pp. 1765–1770.

[11] N. T. Binh, N. A. Tung, D. P. Nam, and N. H. Quang, "An adaptive backstepping trajectory tracking control of a tractor trailer wheeled mobile robot," *International Journal of Control, Automation and Systems*, vol. 17, no. 2, pp. 465–473, Feb 2019.

[12] P. Kassaeiyan, B. Tarvirdizadeh, and K. Alipour, "Control of tractor-trailer wheeled robots considering self-collision effect and actuator saturation limitations," *Mechanical Systems and Signal Processing*, vol. 127, pp. 388–411, 2019.

[13] P. Kassaeiyan, K. Alipour, and B. Tarvirdizadeh, "A full-state trajectory tracking controller for tractor-trailer wheeled mobile robots," *Mechanism and Machine Theory*, vol. 150, p. 103872, 2020.

[14] B. Ma and S. Tso, "Unified controller for both trajectory tracking and point regulation of second-order nonholonomic chained systems," *Robotics and Autonomous Systems*, vol. 56, no. 4, pp. 317–323, 2008.

[15] M. Michałek and K. Kozłowski, "Feedback control framework for car-like robots using the unicycle controllers," *Robotica*, vol. 30, no. 4, pp. 517–535, 2012.

[16] T. Taniguchi, L. Eciolaza, and M. Sugeno, "Tracking control for a nonholonomic car-like robot using dynamic feedback linearization based on piecewise bilinear models," in *IEEE International Conference on Fuzzy Systems* , 2014, pp. 2465–2471.

[17] R. Rashad, "Dynamic trajectory tracking of a car-like robot," Ph.D. dissertation, Institute of Computer Science Department VII: Robotics and Telematics, Julius-Maximilians-Universität of Würzburg, Germany, 2012.

[18] Z. Cao, Y. Zhao, and Y. Fu, "Trajectory tracking control approach of a car-like mobile robot," *ACTA ELECTONICA SINICA*, vol. 40, no. 4, pp. 632–635, 2012.

[19] R. Applonie and Y. Jin, "A novel steering control for car-like robots based on lyapunov stability," in *American Control Conference*, 2019, pp. 2396–2401.

[20] H. K. Khalil, "Nonlinear systems, third edition," *Prentice Hall*, vol. 115, pp. 322–323, 2002.

[21] H. M. Wu, "Nonlinear trajectory-tracking control of a car-like mobile robot in the presence of input saturations and a pulse disturbance," in *58th Annual Conference of the Society of Instrument and Control Engineers of Japan*, 2019, pp. 1498–1502.