IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 8, 2022, accepted June 16, 2022, date of publication June 21, 2022, date of current version June 24, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3185062

Curved Path Following Controller for 4W Skid-Steering Mobile Robots Using Backstepping

YANG CHEN[1,2], NAN LI[1,2], WEI ZENG[1,2], SHIQIAN ZHANG[1], AND GUIFANG MA[1]
[1]School of Physics and Mechatronics Engineering, Longyan University, Longyan 364012, China
[2]School of Mechanical Engineering and Automation, Fuzhou University, Fuzhou 350116, China

Corresponding author: Yang Chen (chenyang4117@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61503172 and Grant 61773194; in part by the Natural Science Foundation of Fujian Province, China, under Grant 2020J05197; and in part by the Educational Research Project of Fujian Educational Bureau for Young Teachers under Grant JT180531.

ABSTRACT Path following is a fundamental problem in skid-steered mobile robots (SSMR). In this study, a Lyapunov stable curved path following controller was designed to generate the steering control command for an SSMR. In contrast to the existing path following controller design methods, where the complete dynamic model of the robot is considered or not, the steering dynamic characteristics approximated by a first-order model are considered in this study. Together with the kinematic model, a steering control law for following a curved path is designed by using the backstepping technique and Lyapunov stability theory. The proposed method was verified on a real SSMR platform to realize following the straight-line, square, and circular paths. Compared with the steering control law that does not consider the steering dynamics, the proposed method can make the robot converge to the predefined path faster with a smaller error overshoot.

INDEX TERMS Path following, SSMR, Lyapunov stability, backstepping, steering dynamics.

I. INTRODUCTION

Wheeled mobile robots have been applied in many fields, such as power inspection, explosive disposal, nuclear power station inspection, fire control, and lunar exploration. A skid-steered mobile robot (SSMR) moves by generating different torques using wheel motors on both sides. These robots are known for their mechanical robustness because they do not require a steering system and only need to move on rough surfaces [1]. Many advanced technologies including path planning [2], cooperative control [3], [4], and obstacle avoidance, have been designed, tested, and applied in SSMRs in recent years. A common requirement for the realization of these technologies is that the robot can accurately follow the predefined path [2].

Many papers address path following control of SSMR in the literature. Basically, the control methods can be classified into linear and nonlinear techniques. When applying linear methods, the traditional proportional-integral-derivative (PID) control and some robust control methods are often adopted. In [5], a modified PID controller was developed to track the desired velocity produced by the desired path. Mojaev *et al.* [6] used PD control with ϕ_c as the error input to produce the rotational velocity command, where ϕ_c is the arctangent function of the orthogonal distance. However, the PID parameters should be tuned by trial and error, and they have little robustness to environmental disturbances. To make the controller more robust to external disturbances and unmodeled parts, a prescribed performance adaptive neural observer-based PID controller has been developed, in which the parametric uncertainties of the models are compensated by a radial basis function neural network [7]. In [8], an adaptive-robust tracking control law, which combined the inverse dynamics control technique and an adaptive robust PID control strategy, was developed to preserve robustness to both parametric and nonparametric uncertainties. In [9], the authors have presented the linear quadratic Gaussian (LQG) technique to track the lateral position of the robot with a simplified first-order digital model approximation. In [10], the LQR method was applied to realize path following, where the linear dynamics with respect to the desired heading error were required. It should be pointed out that the difference between the heading angle and the desired path angle should be relatively small [9], [10]. Otherwise, the linearization conditions for the controller design will not be met, which will make the system unstable.

The associate editor coordinating the review of this manuscript and approving it for publication was Min Wang.

Alternative approaches to realizing the path following are based on nonlinear techniques, which include nonlinear model predictive control (NMPC), intelligent control, and Lyapunov stability-based control methods. Yu *et al.* presented a disturbance-observer-based model predictive control for the path following problems of wheeled mobile robots with input disturbances [11]. Ribeiro *et al.* developed a constrained NMPC-based control scheme to follow visual paths by considering the aspects of stability and feasibility directly from an image plane [12]. Wang *et al.* designed a robust MPC strategy based on a kinematics model with a constraint relationship for the robot. A delayed neural network was applied to solve the corresponding optimization problem [13]. To compensate for the errors originating from disturbances or model discrepancies, the robust tube-based NMPC technique has been applied for the motion control of SSMRS with terra-mechanical constraints [14], [15]. When using the NMPC technique, the path following error can converge quickly with respect to its predictive ability, but its calculation becomes larger with an increase in the number of prediction steps.

In [16], a fuzzy behavior-based approach for navigation was proposed to realize wall-following. In [17], the authors applied the Takagi-Sugeno fuzzy logic controller to make the robot follow an adaptive curvature-based selected point. In [18], the authors presented a fuzzy logic controller to make the designed sliding surfaces of the distance and heading errors approach zero, using an expert system. The fuzzy logic technique can improves the path following performance, but the nominal stability of some controllers cannot be guaranteed, and the influence of dynamic characteristics is not considered.

The Lyapunov-stability-based path following methods are interesting. Ibrahim *et al.* designed a point-to-point tracking algorithm to track a trajectory defined by a set of waypoints [19]. Dai *et al.* applied the sliding mode control technique to derive the steering control command [20]. Xie *et al.* developed a collision-free trajectory tracking controller for a 4W SSMR with a combination of a new coupled fractional-order sliding mode control and obstacle avoidance scheme [21]. Kapitanyuk *et al.* designed a vector-field algorithm to guide a mobile robot along a general smooth planar path [22]. Chen *et al.* presented a guidance vector field (GVF) based controller such that the mobile robot could follow the desired path with an admissible error in the presence of uncertainties, including surface friction, unmodeled dynamics, and disturbances [23]. A robust sliding mode control was proposed to follow the circular path, and its stability was proved using the Lyapunov direct method [24]. In [25], the authors designed a state feedback controller for mobile robots based on super-twisting algorithms. The results were compared with state feedback and first-order sliding mode control. Moro *et al.* presented a simple path following method that can achieve asymptotic convergence to a generic 2-D curve represented by its implicit equation [26]. For other realistic nonholonomic vehicles, vector field and nested saturation techniques have been studied primarily for straight lines and circular paths [27]–[29].

Motivated by the above discussions, it can be found that the steering control laws generated by Lyapunov stability-based methods are usually simple and useful. In some studies, the dynamic model of the robot was considered in the design of some controllers, however, the model was too complex and required more accurate modeling and higher calculations. In the other studies, the path following controllers only considered the kinematic model and did not consider the dynamic model, which degraded the following performance. This study considered the steering dynamic characteristics approximated by a first-order model together with the kinematic model, and a steering control law was designed using the backstepping technique and Lyapunov stability theory. Compared with the steering control law that does not consider the steering dynamics, the proposed method has a better straight-line path following performance.

The main contributions of this study are summarized as follows. First, the error dynamics for following a class of curved paths represented by an implicit function are derived. The corresponding Lyapunov functions are constructed for deriving the commands. And then, the Barbalat's lemma is applied to guarantee the stability of the corresponding nonlinear dynamics. Finally, the Lyapunov-stability-based guidance laws are obtained. Second, in contrast to the existing path following controller design methods, where the complete dynamic model of the robot is considered or not, the steering dynamic characteristics approximated by a first-order model are considered to derive the commanded rotational velocity in this study. Third, to the best of our knowledge, this is the first time that the proposed path following controller, which is designed based on the combined kinematic/first-order dynamic model, has been presented and realized in an actual 4W skid-steering mobile robot. Compared with the steering control law that does not consider the steering dynamics, the proposed method can make the robot converge to the predefined path faster with a smaller error overshoot.

The remainder of this paper is organized as follows. In the next section, the problem description is presented. In Section III, a Lyapunov stable nonlinear controller is designed to generate the desired rotational velocity command for a class of curved paths which are represented by implicit functions. Considering the steering dynamics, the backstepping technique was applied to generate the commanded rotational velocity. The experimental results on a real 4W skid-steering mobile robot platform are presented in Section IV, and a comparison of the different steering control laws is also discussed. Finally, conclusions are presented in Section V.

II. PROBLEM DESCRIPTION

While developing a model that is useful for path following control design, some side effects, including the suspensions and tire deformation, have been neglected. The following assumptions were made [1]–[30]:

(1) The robot moves on a horizontal plane.
(2) Robot speed below 10 km/h.
(3) Perfect longitudinal traction without wheel slippage.
(4) The lateral effort on the tires is proportional to the vertical load.
(5) The robot is symmetric with respect to the vehicle sides.
(6) The center of gravity (COG) coincides with the geometric center.

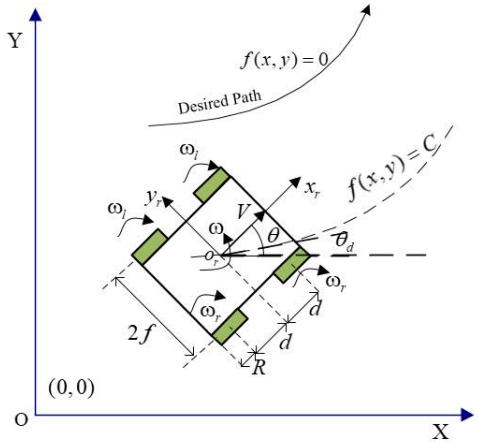

FIGURE 1. Mobile robot velocities and robot position presentation.

As shown in Fig. 1, a fixed reference frame is defined as XOY. A moving frame $x_r o_r y_r$ is attached to the robot body, where the origin is at the center of mass of the robot. An approximate kinematic model of a wheeled mobile robot was discussed in [31], where the overall translational velocity V is defined in (1) and the rotational velocity ω in (2), R is the radius of the wheel, and $2f$ is the distance between the left and right wheels. Using the commanded translational and rotational velocities, the wheel rotation speed commands can be calculated based on (1).

$$\begin{cases} V = \dfrac{\omega_r + \omega_l}{2} R \\ \omega = \dfrac{\omega_r - \omega_l}{2f} R. \end{cases} \tag{1}$$

In this study, the mobile robot was controlled by translational and rotational velocities which are calculated by the path following controller. The left and right robot wheels were controlled using a PID speed controller. The compact form of the kinematics of a mobile robot can be modelled using (2) from [1].

$$\begin{cases} \dot{x} = V\,cos\,(\theta) - l\omega\,sin\,(\theta) \\ \dot{y} = V\,sin\,(\theta) + l\omega\,cos\,(\theta) \\ \dot{\theta} = \omega. \end{cases} \tag{2}$$

where x and y are the COG of the robot in the fixed reference frame, θ is the orientation in the 2D frame, ω is the angular velocity, l is the distance between the robot's center of rotation and the COG.

In this study, it is assumed that the distance between the robot's center of rotation and the COG is so small that the item $l\omega$ in (2) can be ignored. Thus, the kinematic model used for deriving the path following command can be approximated as

$$\begin{cases} \dot{x} = Vcos(\theta) \\ \dot{y} = Vsin(\theta) \\ \dot{\theta} = \omega \end{cases} \tag{3}$$

The steering control is realized by controlling the rotation speed of the left and right wheels of the robot, and the left and right robot wheels are controlled by a well-tuned PID speed controller. Thus, the steering dynamics can be represented by a first-order system as

$$\dot{\omega} = \alpha_\omega\,(\omega_c - \omega)\,, \tag{4}$$

where ω_c is the rotational velocity command, $\alpha_\omega > 0$ is a constant, which can be derived using the system identification method.

Assumption 1: The speed V and its derivative $\dot{V}$ are both bounded, and $V_{p1} \geq V \geq V_{p2} > 0$, where V_{p1} and V_{p2} are all positive constants.

Definition 1 (Horizontal Path): Let $\mathcal{P}_r = \{(x, y)|f\,(x, y) = 0, x, y \in R\}$ be an implicit expression of a reference path, where $f\,(x, y)$ is a twice-continuously differentiable function.

Definition 2 (Level Set): A level set of function $f\,(x, y)$ is the set $\{(x, y)|f\,(x, y) = C\}$, where C is a given constant.

It can be found that the value $f\,(x, y)$ when the robot is in (x, y) can be used as the distance value. Given that the gradient of $f\,(x, y)$ is not zero on the path, the values of $f\,(x, y)$ can represent the position of the robot relative to the desired path. If $f\,(x, y) = 0$, then the robot is on the path.

Let f_x and f_y be the first-order partial derivatives of $f(x, y)$. The gradient modules of $f\,(x, y)$ is $\|f\| = \sqrt{f_x^2 + f_y^2}$. As shown in Fig. 1, the vector (f_x, f_y) represents the desired orientation along the level path. The desired heading θ_d can be expressed as

$$\theta_d = \begin{cases} tan^{-1}\left(-\dfrac{f_x}{f_y}\right), & if\ f_y \neq 0 \\ cot^{-1}\left(-\dfrac{f_y}{f_x}\right), & if\ f_x \neq 0 \end{cases} \tag{5}$$

For a given $f\,(x, y)$, the value of $\|\nabla f\|$ may be zero at some points (x, y). If the robot's initial position is at these points, the steering control law to be designed below will fail, and the robot will lose its control. In the following, the safe moving area for the robot is defined as $D_s = \{(x, y)\,|\,x, y \in R, \|\nabla f\| \geq \lambda\}$, where λ is a positive constant.

Assumption 2: f_x, f_y, f_{xx}, f_{xy}, and $f_{yy}$are bounded in any bounded domain $D \subset R^2$ [26].

The virtual distance error e_d is introduced as

$$e_d = f\,(x, y)\,. \tag{6}$$

Let the heading error e_θ defined as

$$e_\theta = \theta - \theta_d. \tag{7}$$

Form (5), the sine value of variable e_θ can be derived as $sin(e_\theta) = sin\left(\theta - tan^{-1}\left(-\frac{f_x}{f_y}\right)\right) = sin(\theta)\frac{f_y}{\|\nabla f\|} + cos(\theta)\frac{f_x}{\|\nabla f\|}$. By differentiating (6) with respect to time, we obtain:

$$\begin{aligned}\dot{e}_d &= \frac{d}{dt}f(x,y) = f_x\dot{x} + f_y\dot{y}\\ &= f_x V cos(\theta) + f_y V sin(\theta)\\ &= V\|\nabla f\| sin(e_\theta).\end{aligned} \tag{8}$$

The time derivative of (7) is

$$\dot{e}_\theta = \dot{\theta} - \dot{\theta}_d = \omega - \frac{d}{dt}tan^{-1}\left(-\frac{f_x}{f_y}\right). \tag{9}$$

The error kinematics model suitable for control purposes is summarized as

$$\begin{cases}\dot{e}_d = V\|\nabla f\| sin(e_\theta)\\ \dot{e}_\theta = \omega - \frac{d}{dt}tan^{-1}\left(-\frac{f_x}{f_y}\right)\end{cases} \tag{10}$$

Using the error kinematics model (10) and the the steering dynamics (4), the designed feedback control law ω_c should make the errors e_d and e_θ converge to zero.

III. STEERING CONTROL LAW FOR THE CURVED PATH FOLLOWING

A. STEERING CONTROL LAW WITHOUT CONSIDERING THE STEERING DYNAMICS

Define $D_1 = \{(e_d, e_\theta)||e_d| \le e_{d0}, |e_\theta| < \pi\}$ where e_{d0} is a positive constant. Let $\mathbf{x} = (e_d, e_\theta)^T$, we choose the Lyapunov function $V_1(\mathbf{x})$ as:

$$V_1(\mathbf{x}) = k_1\int_0^{e_d} f_{sat}(y)dy + (1 - cos(e_\theta)), \tag{11}$$

where k_1 is a positive constant. $f_{sat}(x)$ is the saturation function defined as

$$f_{sat}(x) = \begin{cases}x_0, & x > x_0\\ x, & -x_0 \le x \le x_0\\ -x_0, & x < x_0\end{cases} \tag{12}$$

The parameter x_0 is a given positive constant. It has the fact that

$$\frac{e_d^2}{2} = \int_0^{e_d} y dy \ge \int_0^{e_d} f_{sat}(y)\,dy \ge 0. \tag{13}$$

Moreover, $(1 - \cos(e_\theta)) \ge 0$. It can be found that $V_1(\mathbf{x}) > 0$ in $D_1 - \{(0,0)^T\}$. The time derivative of $V_1(\mathbf{x})$ is:

$$\begin{aligned}\dot{V}_1(\mathbf{x}) &= k_1\dot{e}_d f_{sat}(e_d) + \dot{e}_\theta sin(e_\theta)\\ &= \left(k_1 V\|\nabla f\| f_{sat}(e_d) + \omega - \dot{\theta}_d\right) sin(e_\theta).\end{aligned} \tag{14}$$

Choosing the following control law ω_d as

$$\omega_d = -k_1 V\|\nabla f\| f_{sat}(e_d) + \dot{\theta}_d - k_2 V^2\|\nabla f\| sin(e_\theta), \tag{15}$$

where k_2 is a given positive constant. Substituting ω in (10) and (14) with ω_d shown in (15), (10) becomes as

$$\begin{cases}\dot{e}_d = V\|\nabla f\| sin(e_\theta)\\ \dot{e}_\theta = -k_1 V\|\nabla f\| f_{sat}(e_d) - k_2 V^2\|\nabla f\| sin(e_\theta)\end{cases} \tag{16}$$

(16) can be rewritten as

$$\dot{\mathbf{x}} = g(\mathbf{x}, t), \tag{17}$$

where

$$g(\mathbf{x},t) = \begin{bmatrix} V\|\nabla f\| sin(e_\theta)\\ -k_1 V\|\nabla f\| f_{sat}(e_d) - k_2 V^2\|\nabla f\| sin(e_\theta)\end{bmatrix}. \tag{18}$$

It can be found that $g : D_1 \to R^2$ is a locally Lipschitz map from the domain $D_1 \subset R^2$ into R^2 with the **Assumption 1** and **Assumption 2** adopted. $(0,0)^T$ is the only equilibrium point for (17) in D_1. Furthermore,

$$\dot{V}_1(\mathbf{x}) = -k_2 V^2\|\nabla f\| sin^2(e_\theta). \tag{19}$$

It can be found that $\dot{V}_1(\mathbf{x})$ is negative semidefinite. Because $V, f_x, f_y, f_{xx}, f_{xy}$, and f_{yy} are bounded and $\|\nabla f\| \ge \lambda$, one can derive that $d\|\nabla f\|/dt$ is bounded [26]. It can be found from (12) that $V_1(\mathbf{x})$ is lower-bounded. Since V, $\|\nabla f\|$, and their derivatives are all bounded, by (19) $\dot{V}_1(\mathbf{x})$ is negative semidefinite and uniformly continuous in time. According to Barbalat's lemma [32], $\dot{V}_1(\mathbf{x}) \to 0$ as $t \to \infty$. Moreover, it has $sin(e_\theta) \to 0$, and $\dot{e}_d \to 0$ as in (16). Because e_d is bounded, it gets that e_d tends to a finite limit, $\bar{e}_d$ as $t \to \infty$. Because $e_\chi \to 0$ as $t \to \infty$, and $\dot{e}_\theta$ in (16) is uniformly continuous, it is derived that $\dot{e}_\theta \to 0$ as $t \to \infty$. Hence, it gets that $\lim_{t\to\infty} f_{sat}(e_d) \to 0$, and $\bar{e}_d = 0$. Therefore, the kinematic control law (15) can asymptotically drive e_d and e_χ towards zero.□

Theorem 1: Considering the kinematic error model of the robot described in (10), the control law ω_d shown in (15) can asymptotically drives e_d and e_χ towards zero.

B. STEERING CONTROL LAW CONSIDERING THE STEERING DYNAMICS

With the help of the backstepping technique, an auxiliary control input ν was introduced as

$$\dot{\omega} = \nu. \tag{20}$$

The course rate error e_ω is defined as

$$e_\omega = \omega - \omega_d. \tag{21}$$

By differentiating (21) with respect to time, it follows that

$$\dot{e}_\omega = \dot{\omega} - \dot{\omega}_d = \nu - \dot{\omega}_d. \tag{22}$$

By combining (10) and (22), we obtain:

$$\begin{cases}\dot{e}_d = V\|\nabla f\| sin(e_\theta)\\ \dot{e}_\theta = \omega - \frac{d}{dt}tan^{-1}\left(-\frac{f_x}{f_y}\right)\\ \dot{e}_\omega = \nu - \dot{\omega}_d\end{cases} \tag{23}$$

We define $D_2 = \{(e_d, e_{\theta e_\omega})||e_d| \le e_{d1}, |e_\theta| < \pi, |e_\omega| \le e_{\omega 1}\}$, where $e_{\omega 1}$ is a positive constant. Let $\mathbf{x}_1 = (e_d, e_\chi, e_\omega)^T$, and choose the candidate Lyapunov function $V_2(\mathbf{x}_1)$ as

$$V_2(\mathbf{x}_1) = k_1\int_0^{e_d} f_{sat}(y)\,dy + (1 - cos(e_\theta)) + \frac{1}{2}e_\omega^2. \tag{24}$$

It has the fact that $V_2(\mathbf{x}_1) > 0$ in $D_2 - \{(0,0,0)^T\}$. The time derivative of $V_2(\mathbf{x}_1)$ is

$$\begin{aligned}\dot{V}_2(\mathbf{x}_1) &= k_1 \dot{e}_d f_{sat}(e_d) + \dot{e}_\theta sin(e_\theta) + e_\omega \dot{e}_\omega \\ &= k_1 V_g \|\nabla d\| f_{sat}(e_d) sin(e_\theta) \\ &\quad + (\omega - \dot{\theta}_d) sin(e_\chi) + e_\omega \dot{e}_\omega \\ &= sin(e_\theta)(-\omega_d + \omega + \omega_d \\ &\quad + k_1 V_g \|\nabla d\| f_{sat}(e_d) - \dot{\theta}_d) \\ &\quad + e_\omega (\nu - \dot{\omega}_d) \\ &= sin(e_\theta)(\omega_e + \omega_d \\ &\quad + k_1 V_g \|\nabla d\| f_{sat}(e_d) - \dot{\theta}_d) \\ &\quad + e_\omega (\nu - \dot{\omega}_d). \end{aligned} \tag{25}$$

Substitute ω_d from (15), $\dot{V}_2(\mathbf{x}_1)$ can be derived as

$$\dot{V}_2(\mathbf{x}_1) = -k_2 V_g^2 \|\nabla d\| sin^2(e_\theta) + e_\omega (\nu - \dot{\omega}_d + \sin(e_\theta)). \tag{26}$$

By choosing

$$\nu = \dot{\omega}_d - sin(e_\theta) - k_\omega \alpha_\omega e_\omega, \tag{27}$$

where $k_\omega$is a positive constant. By substituting ω and ν in (23) with ω_d and ν as shown in (15) and (27), respectively, (23) becomes

$$\begin{cases} \dot{e}_d = V \|\nabla f\| sin(e_\theta) \\ \dot{e}_\theta = -k_1 V \|\nabla f\| f_{sat}(e_d) - k_2 V^2 \|\nabla f\| sin(e_\theta) \\ \dot{e}_\omega = -sin(e_\theta) - k_\omega \alpha_\omega e_\omega \end{cases} \tag{28}$$

(28) can be rewritten as

$$\dot{\mathbf{x}}_1 = h(\mathbf{x}_1, t), \tag{29}$$

where

$$h(\mathbf{x}_1, t) = \begin{bmatrix} \dot{e}_d = V \|\nabla f\| sin(e_\theta) \\ \dot{e}_\theta = -k_1 V \|\nabla f\| f_{sat}(e_d) - k_2 V^2 \|\nabla f\| sin(e_\theta) \\ \dot{e}_\omega = -sin(e_\theta) - k_\omega \alpha_\omega e_\omega \end{bmatrix}. \tag{30}$$

It can be found that $h : D_2 \to R^3$ is a locally Lipschitz map from domain $D_2 \subset R^3$ into R^3 with **Assumption 1**and **Assumption 2**. $(0,0,0)^T$ is the only equilibrium point for (29) in D_2. Furthermore,

$$\dot{V}_2(\mathbf{x}_1) = -k_2 V^2 \|\nabla f\| sin^2(e_\theta) - k_\omega \alpha_\omega e_\omega^2. \tag{31}$$

It can be found from (24) that $V_2(\mathbf{x}_1)$ is lower-bounded. Since V, $\|\nabla f\|$, and their derivative are all bounded, by (31) $\dot{V}_2(\mathbf{x}_1)$ is negative semidefinite and uniformly continuous in time. According to Barbalat's lemma [32], $\dot{V}_2(\mathbf{x}_1) \to 0$ as $t \to \infty$. Hence, $e_\theta \to 0$, $e_\omega \to 0$ as $t \to \infty$. Moreover, it has $sin(e_\theta) \to 0$, and $\dot{e}_d \to 0$ as $t \to \infty$ in (28). Because e_d is bounded, it gets that e_d tends to a finite limit, $\bar{e}_d$ as $t \to \infty$. Because $e_\chi \to 0$ as $t \to \infty$, and $\dot{e}_\theta$ in (28) is uniformly continuous, it is derived that $\dot{e}_\theta \to 0$ as $t \to \infty$. Hence, it gets that $\lim_{t\to\infty} f_{sat}(e_d) \to 0$, and $\bar{e}_d = 0$. Therefore, the auxiliary control input ν shown in (27), together with ω_d shown in (15), can asymptotically drive e_d, e_χ, and e_ω towards zero. It also has $\omega \to \omega_d$ as $t \to \infty$.

The rotational velocity command ω_c can be computed from (4) as

$$\omega_c = \frac{\nu}{\alpha_\omega} + \omega = \frac{\dot{\omega}_d - sin(e_\theta)}{\alpha_\omega} + \omega - k_\omega e_\omega. \tag{32}$$

Theorem 2: Considering the combined kinematic/dynamic error model of the robot described in (23), the rotational velocity command ω_c shown in (32) can asymptotically drives e_d, e_χ, and e_ω towards zero.

IV. EXPERIMENTS

To verify the effectiveness of the proposed method, some experiments have been studied in a 4W skid-steering mobile robot, as shown in Fig. 2. The mobile robot is equipped with two control boards based on Raspberry Pi 4B and STM32F103RCT6 respectively. The Raspberry Pi 4B based control board obtains the data from the UWB positioning module, lidar, and depth camera through USB serial ports. A driver board is used to drive the four motors. An MPU6050 IMU module is integrated into the driver control board. The two control boards communicate with each other via an I2C serial bus. The DCDC module integrated into the STM32 based control board provides a 5v@3a max power supply for the Raspberry Pi 4B based control board. The block diagram of the hardware is shown in Fig. 3.

FIGURE 2. A 4W skid-steering mobile robot for experiment.

The Raspberry Pi 4B based control board runs the Ubuntu Raspbian system and is equipped with a Kinetic Kame version ROS operating system. The main structure and control parameters of the robot are listed in Table 1.

TABLE 1. The parameters of the robot simulation.

Parameter	Value	Parameter	Value
R	40 mm	k_1	4
F	88 mm	k_2	6.5
D	75 mm	k_ω	1.5
Control Frequency	100 Hz	α_ω	3.03

The control structure is shown in Fig. 4. The proposed curved path following controller is applied to generated the rotational velocity command ω_c. Together with the forward linear velocity command V_c, the rotational speed commands of the left and right wheels of the robot (ω_l and ω_r) can be

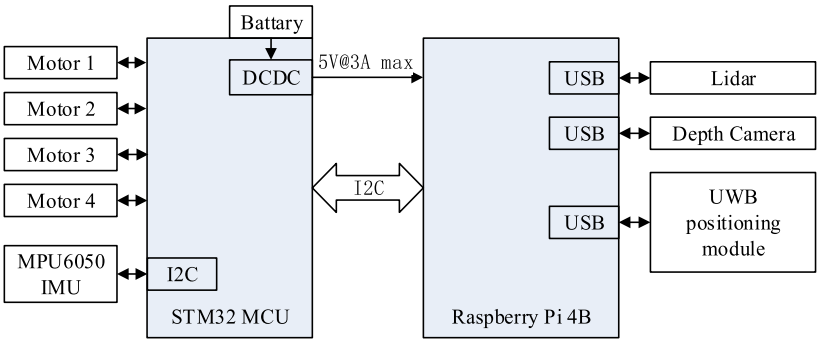

FIGURE 3. The block diagram of the hardware.

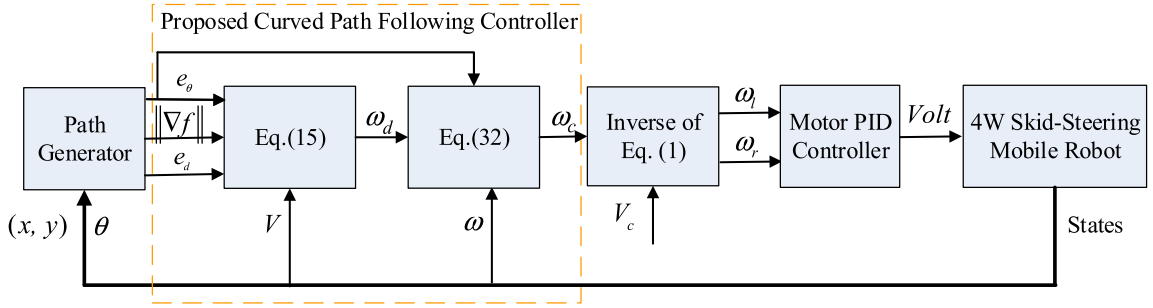

FIGURE 4. Proposed control structure.

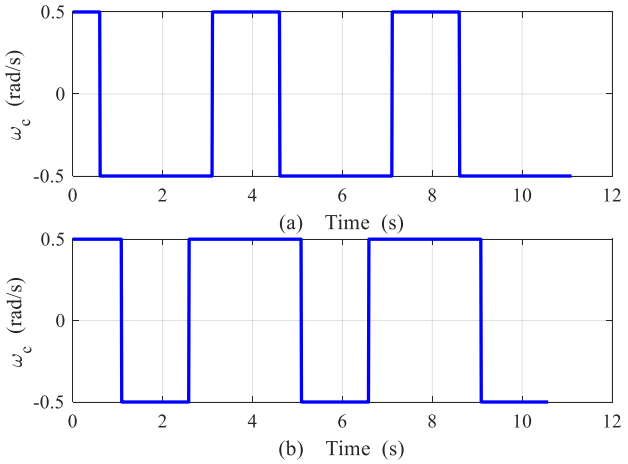

FIGURE 5. The input ω_c for identifying the closed-loop roll dynamics. (a) identification; (b) validation.

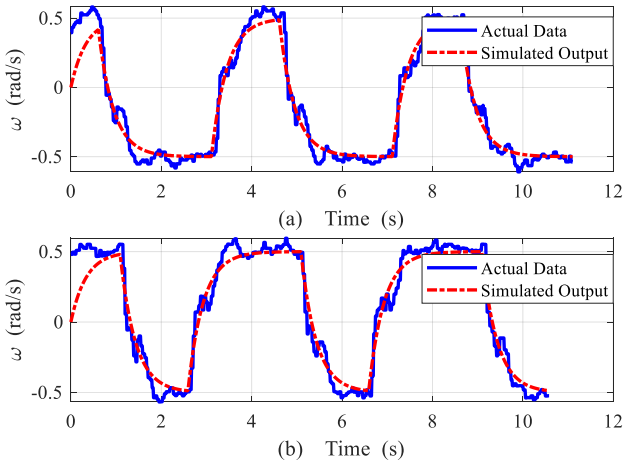

FIGURE 6. Comparison of the actual and estimated rotational velocities. (a) identification; (b) validation.

obtained by the inverse calculation of (1). Finally, the PID controllers generate the voltage commands to control the DC motors that correspond to the four wheels. In this study, the forward linear velocity command, V_c was set to 0.3m/s.

In the following, the robot uses the control laws ω_d shown in (15) and ω_c shown in (32) to realize path following respectively, which are referred to as PFC_ω_d and PFC_ω_c.

A. IDENTIFICATION OF PARAMETER α_ω

The parameter α_ω of the steering dynamics (4) is important for calculating the rotational velocity command. In this study, α_ω was derived using the parameter identification technique. The input ω_c used to identify the dynamics is shown in Fig. 5(a). The input ω_c used to validate the identified dynamics is shown in Fig. 5(b). The predict error method was applied to identify the dynamics (4) using the System Identification Toolbox in Matlab. Comparisons of the actual and rotational velocities over the identification and validation are shown in Figs. 6(a) and (b) respectively. The corresponding best-fit value (100 indicates a perfect fit, and 0 indicates a poor fit) was 82.3. The identified parameter α_ω is 3.03.

B. STRAIGHT-LINE PATH FOLLOWING

The response comparison of the two proposed methods while following the four directional paths is shown in Fig. 7. The

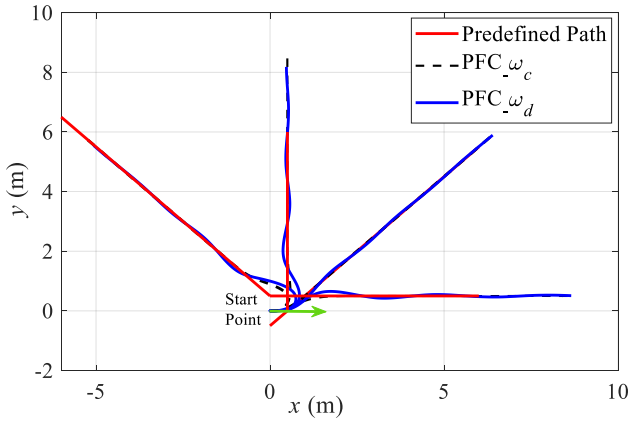

FIGURE 7. Comparison of the four directional paths following with the methods PFC_ω_d and PFC_ω_c.

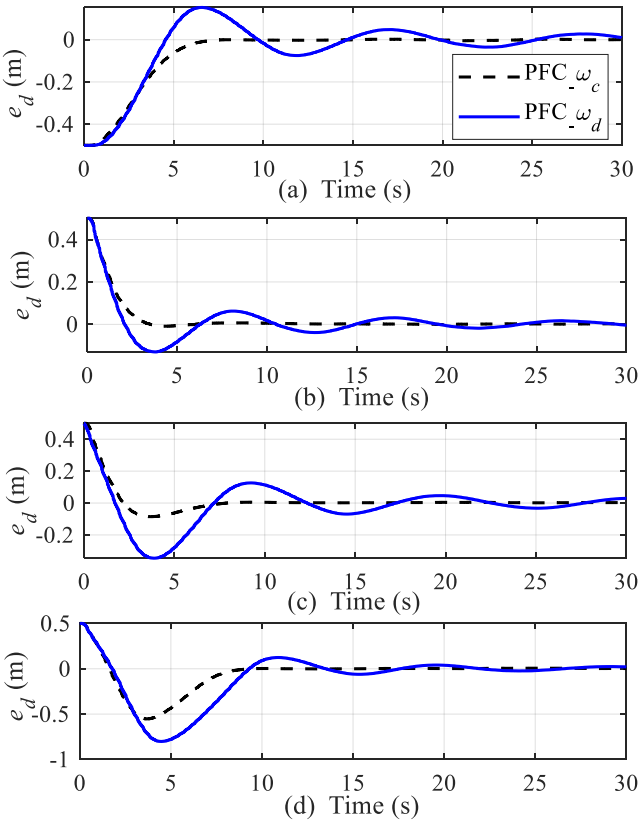

FIGURE 8. Comparison of the distance e_d for the directional path following with the methods PFC_ω_d and PFC_ω_c. (a) straight-line heading at 0^o, (b) straight-line heading at 45^o, (c) straight-line heading at 90^o, (d) straight-line heading at 135^o.

initial position and heading of the robot were set as (0, 0) and 0 deg respectively. Fig. 8 presents a comparison of the error distances e_d.

The response to directional paths can be evaluated by the rise time, convergence distance, and error overshoot. The rise time is the time the robot travels along a straight line before the absolute value of the distance $|e_d|$ the first time becomes less than 0.025 m. The convergence time is the time the robot travels along a straight line before the absolute value of the distance $|e_d|$ becomes less than 0.025 m. The error overshoot is the maximum deviation from the straight reference line.

TABLE 2. The performance of the response comparison with the two methods.

Line (^{O})	Rise time (s)		Convergence time (s)		Error overshoot (m)	
	PFC_ω_d	PFC_ω_c	PFC_ω_d	PFC_ω_c	PFC_ω_d	PFC_ω_c
0	4.41	5.92	>30	5.92	0.15	<0.01
45	1.96	2.89	17.99	2.89	0.35	<0.01
90	1.61	2.00	>30	6.34	0.35	0.085
135	1.72	1.60	20.92	8.37	0.80	0.55

The performance of the responses to the four directional paths using the three methods is presented in Table 2. When following the four directional paths at 0^o, 45^o, 90^o, and 135^o, the rise times obtained using the PFC_ω_d method are all shorter than those obtained using the PFC_ω_c method. However, the convergence times obtained using the PFC_ω_d method are longer than those obtained using the PFC_ω_c method. Moreover, the error overshoots obtained using the PFC_ω_c method are smaller than those obtained using the PFC_ω_d method. In particular, when following the directional paths at 0^o, 45^o, and 90^o, the error overshoots obtained using the PFC_ω_c method are smaller than 0.1 m. This can also be observed in Fig. 6. This indicates that the PFC_ω_c method can make the robot move towards the desired straight-line path faster than the PFC_ω_d method.

Figs. 9-12 show the rotational velocity command response when following the four directional paths at 0^o, 45^o, 90^o, and 135^o with the two proposed methods, respectively. It can be observed from the four figures that the PFC_ω_c method can make the robot track the rotational velocity command faster and more precisely. This also indicates that the PFC_ω_c method can make the robot follow a predefined straight-line path with a shorter convergence time and smaller overshoot.

C. SQUARE PATH FOLLOWING

The desired square path was defined with four waypoints, A (0, 0), B (5.5, 0), C (5.5, 5.5), and D (0, 0.5), as shown in Fig. 13(a). The initial position and heading of the robot were set as (0, 0) and 0deg respectively. A comparison of the path following with the methods PFC_ω_d and PFC_ω_c is also shown in Fig. 13(a). Fig. 13(b) shows a comparison of the distance while following the predefined square path using the two methods. While executing square path following, the predefined behavior that the robot will switch segments in advance with 0.35 m was adopted.

Table 3 lists the square path following performance comparison using the two methods PFC_ω_d and PFC_ω_c.

The error overshoots, displayed as elliptic marks I, II, and III, are listed in Table 3. It indicates that, the error overshoots of e_d by using the method PFC_$\omega_c$are 0.17 m, 0.16 m, and 0.17 m respectively. while by using the method PFC_ω_d, the corresponding error overshoots are 0.38m, 0.40m, and 0.40m respectively. This shows that the

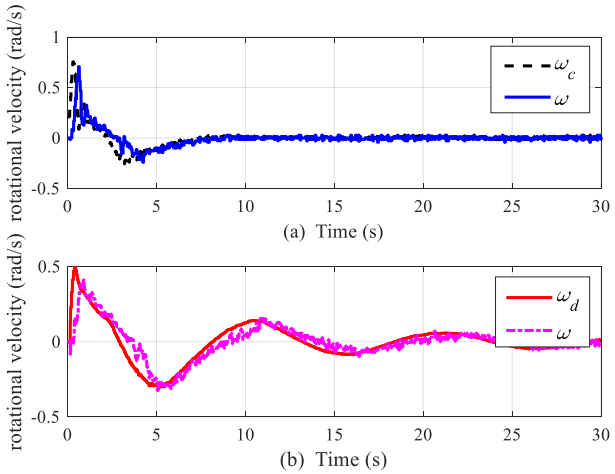

FIGURE 9. Comparison of the rotational velocity command response while following the directional path at 0^o. (a) ω_c, (b) and ω_d.

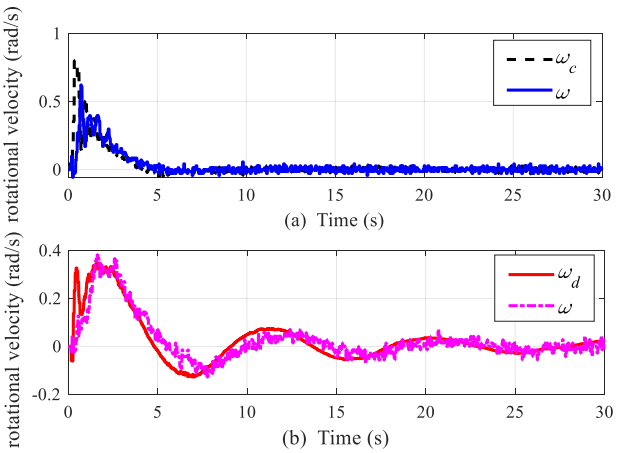

FIGURE 10. Comparison of the rotational velocity command response while following the directional path at 45^o. (a) ω_c, (b) and ω_d.

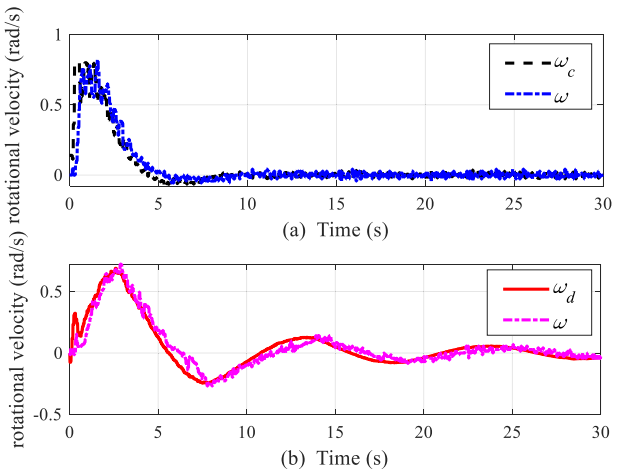

FIGURE 11. Comparison of the rotational velocity command response while following the directional path at 90^o. (a) ω_c, (b) and ω_d.

error overshoots when using the PFC_ω_c method are much smaller than those obtained using the PFC_ω_d method. The results demonstrate the effectiveness of the PFC_ω_c method.

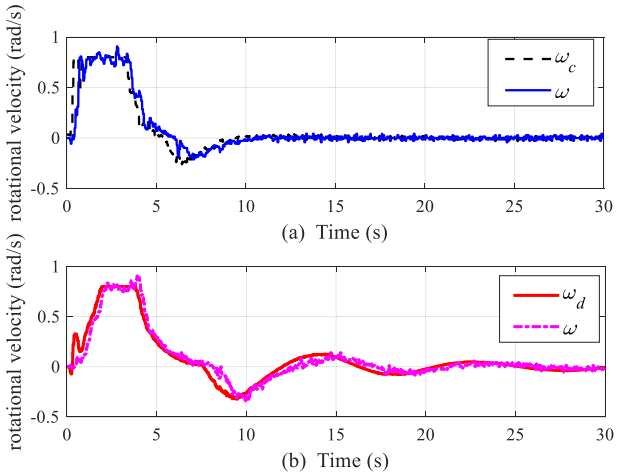

FIGURE 12. Comparison of the rotational velocity rate command response while following the directional path at 135^o. (a) ω_c, (b) and ω_d.

TABLE 3. The performance Comparison of the square path following with the two methods.

Area	Convergence time (s)		Error overshoot (m)	
	PFC_ω_d	PFC_ω_c	PFC_ω_d	PFC_ω_c
I	>20	5.53	0.38	0.17
II	>20	5.57	0.40	0.16
III	>20	5.55	0.40	0.17

TABLE 4. The performance comparison of the circular path following with the two methods.

Area	Convergence time (s)		Error overshoot (m)	
	PFC_ω_d	PFC_ω_c	PFC_ω_d	PFC_ω_c
I	6.29	2.43	0.12	0.03
II	8.04	4.20	0.10	0.03

Figs. 13(c) and (d) show the rotational velocity command response when following the square path with the two proposed methods. It can be seen from the four figures that the method PFC_ω_c can make the robot track the rotational velocity command more precisely. The above comparative analysis shows that the robot can achieve the best square path following performance with the PFC_ω_c method.

D. CIRCULAR PATH FOLLOWING

The initial position of the robot was set to (0, 0). The desired path consists of two concentric circles, which are defined as $\sqrt{(x-1)^2+(y-1)^2}-1=0$ and $\sqrt{(x-1)^2+(y-1)^2}-1.4=0$ respectively. The robot first ran towards circle A with a radius of 1m. Thirty seconds later, the robot started to run towards circle B with a radius of 1.4m.

Fig. 14(a) shows that the PFC_ω_c method allows the vehicle to reach the predefined path with a smaller overshoot than that using the PFC_ω_d method. The distance e_d presented in Fig. 14(b) further shows that there is almost no overshoot (shown as elliptic marks I, and II) when the robot runs toward

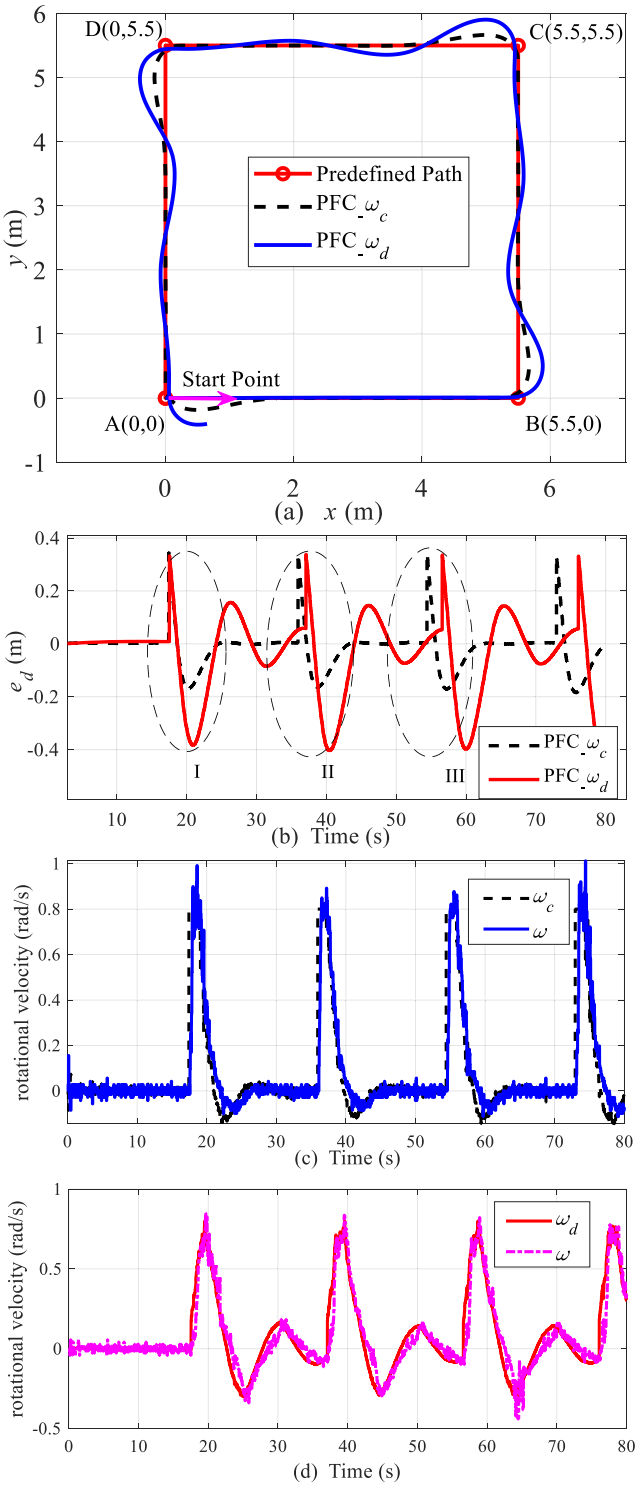

FIGURE 13. Comparison of the square path following with the methods PFC_ ω_d and PFC_ ω_c. (a) the position, (b) the distance e_d, (c) ω_c and its responding ω, (d) ω_d and its responding ω.

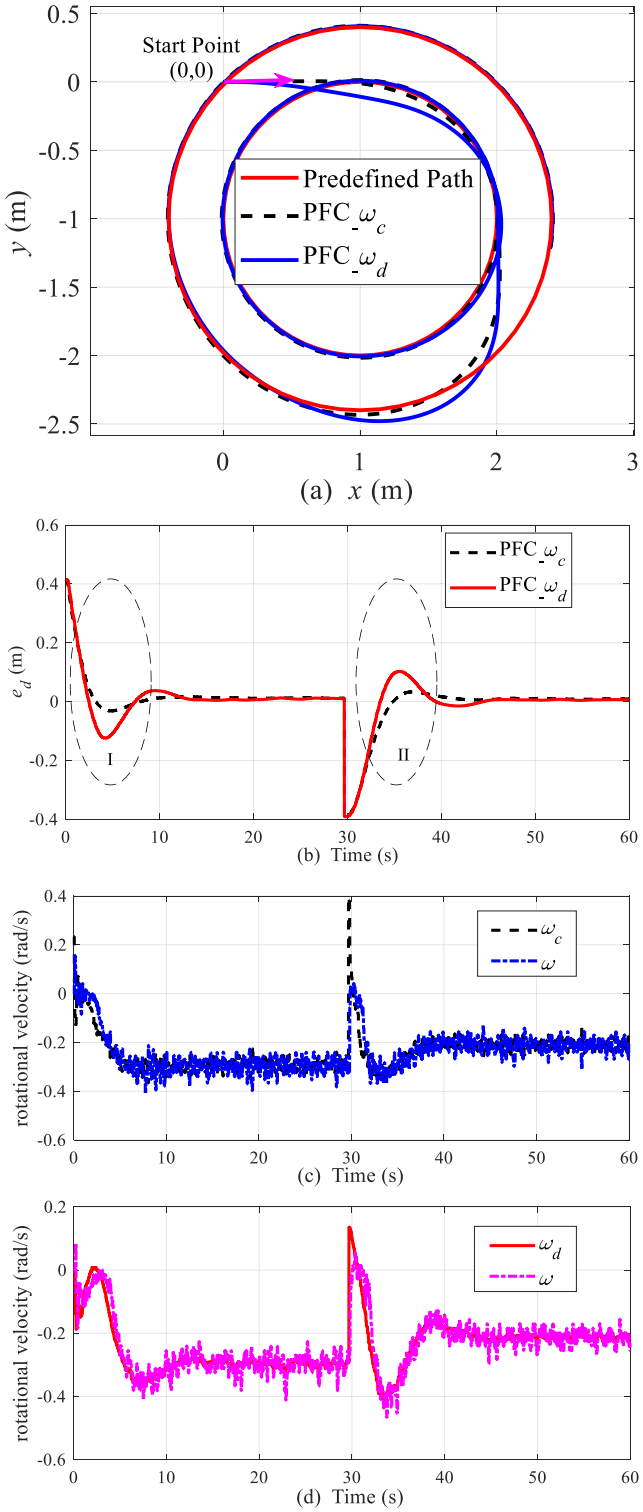

FIGURE 14. Comparison of the circular path following with the methods PFC_ω_d and PFC_ω_c . (a) the position, (b) the distance e_d, (c) ω_c and its responding ω, (d) ω_d and its responding ω.

the predefined circular path using the PFC_ω_c method. This implies that the PFC_ω_c method can make the robot move toward the desired circular path more smoothly. As listed in Table 4, the error overshoots of e_d for the first circular path following using the methods PFC_ω_c and PFC_ω_d are 0.03 m and 0.12 m respectively. When switching to fly towards a circular path with a larger radius, the error overshoots of e_d for the second circular path following using the methods PFC_ω_c and PFC_ω_d are 0.03 m and 0.10 m respectively.

Table 4 also listed the convergence time for the robot to follow the two circles A and B. it can be found that it takes only 2.43 seconds for the robot to converge to Circle A with the PFC_ω_c method. When the other method was used, the convergence times were both greater than 6 s.

The above indicates that the PFC_ω_c method will still make the robot have the best circular path following performance compared with those using the PFC_ω_d method.

V. CONCLUSION

A horizontal steering control law for the path following of a 4W skid-steering mobile robot was presented in this paper. Based on the combination of the kinematic model of the robot and its steering dynamic characteristics approximated by a first-order model, the error dynamics of the path following were derived. The backstepping technique is used to derived the rotational velocity command. The Barbalat's lemma has been applied to guaranteed the corresponding nonautonomous nonlinear systems. The experiments on a real robot have been carried out to verify the effectiveness of the proposed method.

Compared with the steering control law that does not consider the steering dynamics, the rotational velocity command designed based on the combined kinematic/ steering dynamic models can make the robot converge the predefined path faster with a smaller error overshoot.

In the future, the steering dynamics will also be considered for more precise path following. Because the dynamics have model uncertainties and control constraints, neural [7], fuzzy [33], and fuzzy-neural [34] approximation techniques will be applied to approximate the unmodeled part and disturbance. The funnel control technique [35] will also be considered to make the robot reach the set path following performance.

When the mobile robot moves on different types of ground, its steering dynamic characteristics are different. Therefore, when carrying out path following control on various types of ground, the controller design needs to consider the ground switching, the pattern-based autonomous smooth switching control technique [36] will be considered to handle this problem. Furthermore, in the actual system, the controller design based on discrete system is more suitable for on-line operation. When implementing path following control based on discrete systems, it may face the problem of state estimation caused by the coupling of control input and unknown control gains. In this case, the filter-based event-triggered adaptive fuzzy control method [37] is an option to solve such problems.

REFERENCES

[1] I. Matraji, K. Al-Wahedi, and A. Al-Durra, "Higher-order super-twisting control for trajectory tracking control of skid-steered mobile robot," *IEEE Access*, vol. 8, pp. 124712–124721, 2020.

[2] M. Dirik, A. F. Kocamaz, and O. Castillo, "Global path planning and path-following for wheeled mobile robot using a novel control structure based on a vision sensor," *Int. J. Fuzzy Syst.*, vol. 22, pp. 1880–1891, Jun. 2020.

[3] Z. Zuo, J. Song, and Q. L. Han, "Coordinated planar path-following control for multiple nonholonomic wheeled mobile robots," *IEEE Trans. Cybern.*, early access, Mar. 11, 2021, doi: 10.1109/TCYB.2021.3057335.

[4] A. Loria, J. Dasdemir, and N. Alvarez Jarquin, "Leader–follower formation and tracking control of mobile robots along straight paths," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 2, pp. 727–732, Mar. 2018.

[5] W. Yu, O. Y. Chuy, E. G. Collins, and P. Hollis, "Analysis and experimental verification for dynamic modeling of a skid-steered wheeled vehicle," *IEEE Trans. Robot.*, vol. 26, no. 2, pp. 340–353, Apr. 2010.

[6] A. Mojaev and A. Zell, "Tracking control and adaptive local navigation for nonholonomic mobile robots," in *Proc. Conf. Intell. Auton. Syst. (IAS)*, Mar. 2004, pp. 521–528.

[7] K. Shojaei, "A prescribed performance PID control of robotic cars with only posture measurements considering path curvature," *Eur. J. Control*, vol. 65, May 2022, Art. no. 100616, doi: 10.1016/j.ejcon.2022.100616.

[8] K. Shojaei, A. M. Shahri, and B. Tabibian, "Design and implementation of an inverse dynamics controller for uncertain nonholonomic robotic systems," *J. Intell. Robotic Syst.*, vol. 71, no. 1, pp. 65–83, Jul. 2013.

[9] B. Fernandez, P. J. Herrera, and J. A. Cerrada, "A simplified optimal path following controller for an agricultural skid-steering robot," *IEEE Access*, vol. 7, pp. 95932–95940, 2019.

[10] A. Ratnoo, P. B. Sujit, and M. Kothari, "Adaptive optimal path following for high wind flights," *IFAC Proc. Volumes*, vol. 44, no. 1, pp. 12985–12990, Jan. 2011.

[11] S. Yu, Y. Guo, L. Meng, T. Qu, and H. Chen, "MPC for path following problems of wheeled mobile robots," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 247–252, 2018.

[12] T. T. Ribeiro and A. G. S. Conceição, "Nonlinear model predictive visual path following control to autonomous mobile robots," *J. Intell. Robotic Syst.*, vol. 95, no. 2, pp. 731–743, Aug. 2019.

[13] D. Wang, W. Wei, Y. Yeboah, Y. Li, and Y. Gao, "A robust model predictive control strategy for trajectory tracking of omni-directional mobile robots," *J. Intell. Robotic Syst.*, vol. 98, no. 2, pp. 439–453, May 2020.

[14] A. J. Prado, M. Torres-Torriti, and F. A. Cheein, "Distributed tube-based nonlinear MPC for motion control of skid-steer robots with Terra-mechanical constraints," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 8045–8052, Oct. 2021.

[15] L. J. Prado, M. Torres-Torriti, J. Yuz, and F. A. Cheein, "Tube-based nonlinear model predictive control for autonomous skid-steer mobile robots with tire–terrain interactions," *Control Eng. Pract.*, vol. 101, Aug. 2020, Art. no. 104451, doi: 10.1016/j.conengprac.2020.104451.

[16] N. Hacene and B. Mendil, "Fuzzy behavior-based control of three wheeled omnidirectional mobile robot," *Int. J. Autom. Comput.*, vol. 16, no. 2, pp. 163–185, Apr. 2019.

[17] S. Ghaffari and M. Homaeinezhad, "Autonomous path following by fuzzy adaptive curvature-based point selection algorithm for four-wheel-steering car-like mobile robot," *Proc. Inst. Mech. Eng., C, J. Mech. Eng. Sci.*, vol. 232, no. 15, pp. 1989–1996, 2017.

[18] M. Q. Zaman and H. Wu, "Fuzzy sliding mode control based path-following of a complete wheeled mobile robot system," in *Proc. 60th Annu. Conf. Soc. Instrum. Control Eng. Jpn. (SICE)*, Sep. 2021, pp. 386–391.

[19] F. Ibrahim, A. A. Abouelsoud, A. M. R. Fath Elbab, and T. Ogata, "Path following algorithm for skid-steering mobile robot based on adaptive discontinuous posture control," *Adv. Robot.*, vol. 33, no. 9, pp. 439–453, Apr. 2019.

[20] P. Dai, J. Taghia, S. Lam, and J. Katupitiya, "Integration of sliding mode based steering control and PSO based drive force control for a 4WS4WD vehicle," *Auto. Robots*, vol. 42, no. 3, pp. 553–568, Jun. 2017.

[21] Y. Xie, X. Zhang, W. Meng, S. Zheng, L. Jiang, and J. Meng, "Coupled fractional-order sliding mode control and obstacle avoidance of a four-wheeled steerable mobile robot," *ISA Trans.*, vol. 108, pp. 282–294, Feb. 2021.

[22] Y. A. Kapitanyuk, A. V. Proskurnikov, and M. Cao, "A guiding vector-field algorithm for path-following control of nonholonomic mobile robots," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 4, pp. 1372–1385, Jul. 2018.

[23] J. Chen, C. Wu, G. Yu, D. Narang, and Y. Wang, "Path following of wheeled mobile robots using online-optimization-based guidance vector field," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 4, pp. 1737–1744, Aug. 2021, doi: 10.1109/TMECH.2021.3077911.

[24] J. Keighobadi, M. S. Sadeghi, and K. A. Fazeli, "Dynamic sliding mode controller for trajectory tracking of nonholonomic mobile robots," *IFAC Proc. Volumes*, vol. 44, no. 1, pp. 962–967, Jan. 2011.

[25] I. Salgado, D. Cruz-Ortiz, O. Camacho, and I. Chairez, "Output feedback control of a skid-steered mobile robot based on the super-twisting algorithm," *Control Eng. Pract.*, vol. 58, pp. 193–203, Jan. 2017.

[26] A. Morro, A. Sgorbissa, and R. Zaccaria, "Path following for unicycle robots with an arbitrary path curvature," *IEEE Trans. Robot.*, vol. 27, no. 5, pp. 1016–1023, Oct. 2011.
[27] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for miniature air vehicles," *IEEE Trans. Robot.*, vol. 23, no. 3, pp. 519–529, Jun. 2007.
[28] R. W. Beard, F. Ferrin, and J. Humpherys, "Fixed wing UAV path following in wind with input constraints," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 6, pp. 2103–2117, Nov. 2014.
[29] S. Zhao, X. Wang, Z. Lin, D. Zhang, and L. Shen, "Integrating vector field approach and input-to-state stability curved path following for unmanned aerial vehicles," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 8, pp. 2897–2904, Aug. 2020.
[30] L. Caracciolo, A. de Luca, and S. Iannitti, "Trajectory tracking control of a four-wheel differentially driven mobile robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1999, pp. 2632–2638.
[31] J. L. Martínez, A. Mandow, J. Morales, S. Pedraza, and A. García-Cerezo, "Approximating kinematics for tracked mobile robots," *Int. J. Robot. Res.*, vol. 24, pp. 867–878, Oct. 2005.
[32] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
[33] X. Bu, Q. Qi, and B. Jiang, "A simplified finite-time fuzzy neural controller with prescribed performance applied to waverider aircraft," *IEEE Trans. Fuzzy Syst.*, early access, Jun. 14, 2021, doi: 10.1109/TFUZZ.2021.3089031.
[34] X. Bu and Q. Qi, "Fuzzy optimal tracking control of hypersonic flight vehicles via single-network adaptive critic design," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 1, pp. 270–278, Jan. 2022.
[35] X. Bu, B. Jiang, and H. Lei, "Non-fragile quantitative prescribed performance control of waverider vehicles with actuator saturation," *IEEE Trans. Aerosp. Electron. Syst.*, early access, Feb. 25, 2022, doi: 10.1109/TAES.2022.3153429.
[36] H. Shi, M. Wang, and C. Wang, "Pattern-based autonomous smooth switching control for constrained flexible joint manipulator," *Neurocomputing*, vol. 492, pp. 162–173, Jul. 2022.
[37] L. Huang and M. Wang, "Filter-based event-triggered adaptive fuzzy control for discrete-time MIMO nonlinear systems with unknown control gains," *IEEE Trans. Fuzzy Syst.*, early access, Oct. 26, 2021, doi: 10.1109/TFUZZ.2021.3122231.

YANG CHEN received the B.S. degree in material forming and control engineering from Beihang University, Beijing, China, in 2007, and the Ph.D. degree in mechatronics engineering from the Robotics Institute, Beihang University, in 2013.

From 2013 to 2014, he was an Engineer at the Shanghai Aerospace Control Technology Research Institute. Afterwards, he joined at the School of Physics and Mechatronics Engineering, Longyan University, Longyan, China, where he is currently an Associate Professor. Since 2019, he has been a Graduate Supervisor with the School of Mechanical Engineering and Automation, Fuzhou University. His research interests include system modeling, parameter identification, intelligent control of ground mobile robots, and small UAVs.

NAN LI received the B.S. degree in vehicle engineering from Northeast Agricultural University, Harbin, China, in 2017. He is currently pursuing the M.S. degree in mechanical engineering with Fuzhou University, Fuzhou, China. He is also a master's student who is jointly trained by Fuzhou University and Longyan University.

His research interests include ROS-based unmanned system design, path following control of ground mobile robots, and small UAVs.

WEI ZENG received the B.S. degree in mechanical engineering and automation from the South China University of Technology, Guangzhou, China, in 2002, the M.S. degree in control engineering from Xiamen University, Xiamen, China, in 2008, and the Ph.D. degree in control theory and control engineering from the South China University of Technology, in 2012.

From 2012 to 2014, he was a Postdoctoral Research Fellow at the School of Mechanical and Automotive Engineering, South China University of Technology. From 2017 to 2018, he was a Visiting Scholar at the Faculty of Health Sciences, The University of Sydney, Australia. He has been with the School of Physics and Mechatronics Engineering, Longyan University, Longyan, China, since 2003, where he is currently a Professor. Since 2019, he has been a Graduate Supervisor with the School of Mechanical Engineering and Automation, Fuzhou University. His research interests include nonlinear adaptive system identification and control, intelligent robot control, dynamic pattern recognition, and their applications in gait recognition, gesture recognition, and biomedicine.

SHIQIAN ZHANG received the B.S. degree in electrical technology from the Shandong University of Science and Technology, Jinan, China, in 1998, and the M.S. degree in electrical engineering from Fuzhou University, Fuzhou, China, in 2007.

He is currently an Associate Professor with Longyan University. His research interests include the intelligent sensing detection and intelligent control of mobile robots.

GUIFANG MA received the B.S. degree in electrical engineering and automation from Fujian Agriculture and Forestry University, Fuzhou, China, in 2005, and the M.S. degree in power electronics and power transmission from Jiangsu University, in 2009.

From 2009 to 2019, she was a Lecturer at Longyan University. She is currently an Associate Professor with Longyan University. Her research interests include the intelligent control of motors and ground mobile robots.