INTERNATIONAL JOURNAL OF CIRCUITS, SYSTEMS AND SIGNAL PROCESSING
DOI: 10.46300/9106.2020.14.27 Volume 14, 2020

A Back-stepping Control based on Bounded Function for Four-wheel Drive Omni-directional Mobile Robots

Jianping Chen and Jianbin Wang*
School of Computer Science and Software,
Zhaoqing University,
Zhaoqing 526061, Guangdong, China

***Abstract*—As the simply structure and flexible design, back-stepping technique has been widely applied in robot trajectory tracking control. However, there is velocity jumping problem in conventional back-stepping tracking control for four-wheel drive omni-directional mobile robots. In this paper, an improved back-stepping controller based on a bounded function is proposed. To improve control performance, a smooth and bounded tracking velocity, arising from the function, is used to instead of the jumping velocity. Simulation results of tracking different paths and comparison with the conventional back-stepping technique show that the approach is effective, and the system has a good performance with smooth outputs.**

***Keywords*—four-wheel drive, omni-directional mobile robot, velocity jump, sigmoid function, trajectory tracking.**

I. INTRODUCTION

THE Omni-directional mobile robot can move in any direction without changing any position and pose because it has the character of omni-directional mobility [1,2]. The special motion mobility makes the omni-directional mobile robots appropriate when they have to move in tight areas, avoid obstacles, and find the way to the next location, and the omni-directional mobile robots are widely applied to the human production and life practice in recent years [3]. Different robots are designed for different tasks [4]. An omni-directional power-assist-modular mobile robot for total nursing service system has been proposed. To control the mobile robotics system, dual-offset active wheel casters with a differential gear mechanism are designed and applied into the mobile robot [5]. An omni-directional mobile robot basing on the ROS (Robot Operating System) is proposed. The control structure of the robot is very simple, because the robot is built with an embedded personal computer (PC) and an Arduino MEGA 2560 controller board, which serve as the center calculation and control unit [6]. Different from the common human body tracking or object tracking with auxiliary markers, an indoor automatic omni-directional service robot using a common camera is presented in paper [7]. To achieve intelligent interaction and self-tracking, according the different target using different strategies, the robot can track independently. In paper [8], an omni-directional wheeled mobile robot for catching a flying ball using multi-camera vision systems include an active stereo vision system and a static vision system is designed. To track the flying ball and guide the omni-directional mobile wheeled robot to catch it, the dynamic model of a flying ball and Kalman filter are used to mitigate measurement noise, estimate the position and velocity of the flying ball, and predict its future trajectory. To transport materials flexibly and smoothly in a tight plant environment, an omni-directional mobile robot based on four Mecanum wheels is designed [9]. The mechanical system of the mobile robot is made up of three separable layers so as to simplify its combination and reorganization. Each modularized wheel was installed on a vertical suspension mechanism, which ensures the moving stability and keeps the distances of four wheels invariable. In lecture [10], a material conveying mobile robot with omni-directional mobility is designed. The robot uses a four-wheel driven chassis and four Mecanum wheels, and it is designed with a damping suspension mechanism. To control the robot, a Siemens S7-1217C PLC with extended modules is used as the lower-layer controller, while a laptop computer is used as the upper-layer controller. To diminish the vibrations due to the discontinuous contact points in the omni-directional mobile robot, which is driven by universal wheels or Mecanum wheels, a ball wheel mechanism with superior features including slip measurement, free-wheel modus and attrition sensing is presented [11]. To improve the intelligence of the logistics system, based on printing enterprise layout, an omni-directional mobile robot with zero torque is designed by the three Swedish wheels structure [12]. Because that moving a double-parked car by pushing is very hard and dangerous especially for the old and the weak, a developed

This work was supported by the Natural Science Foundation of Guangdong Province (No. 2016A030313015 and No. 2016A030313846), the Engineering Technology Development Center Project of Universities of Guangdong Province (2019GCZX011), the Industry University Cooperation Education Project of Ministry of Education of China (No. 201702029053), the Foundation of Higher Education Academy of Guangdong Province (No. GDJ2016027), the Natural Science Features Innovative Projects of Education Department of Guangdong Province (No. 2016KQNCX175), the Science and Technology Innovation Guidance Project of Zhaoqing City (No. 201624030902), and the Science Foundation for The Youth Scholars of Zhaoqing University (No. 201623).

omni-directional mobile robot is designed for moving a double-parked car easily and safely. The robot moves a double-parked car by rotating a wheel of a double-parked car. It has two specially designed rollers to rotate a wheel of a double-parked car and is designed so that the height of the robot is very low to be able to enter beneath a double-parked car [13]. To improve the manpower efficiency and health benefits of bed pushing during patient transfer in the hospital, a novel robotic-assisted omni-directional hospital bed transporter is introduced. And it shows reduced physical demands, less manpower required for patient transport and reduced back muscle activities through real experiment [14].

Omni-directional mobile robots are very attractive because they have a very good mobility. To move with precision in narrowly constrained environments, the accurate estimation of the position and good motion control are very important [15]. Some researches focus on the path planning, localization, navigation and simultaneous localization and mapping (SLAM) of the robot [16]. In paper [17] the authors describe the path planning of a three-wheel omni-directional mobile robot (OMR) with the harmonic potential function using a model predictive control (MPC). They use MPC in the study to rely on the ability to naturally state and control constraints in practical problems. To enable robots to autonomously move and skirt obstacles in the field, the robots must be designed so that they can exercise self-localization and path planning functions based on the results of sensor signal processing or image analysis. The authors combine the Kinect depth sensor with the omni-directional camera to construct the vision system of FIRA RoboSot and use the devices to enable the robots to position themselves and to detect their distance from obstacles [18].To deal with the shadowing effect of GPS, an indoor positioning system based on several ultrasonic sensors and the ZigBee network is presented in [19]. Mobile robots with traditional drives are not suitable for narrow, crowded indoor environment with lots of obstacles, because they have the coupled translation and rotation motions. And the translation error will affect the heading estimations of the mobile robot and the accuracy of the FastSLAM algorithm will be reduced over the time. Then an improved FastSLAM algorithm based on Omni-directional wheeled mobile robot is proposed in paper [20]. Lidars are commonly employed in mobile robot SLAM applications due to their high stability and high measuring accuracy. But the 2D Lidar, only the planar points can be detected in once scan, is not suitable for omni-directional mobile robots because they have 3-DOF planar motion capabilities. Paper [21] presents a novel design of a 2.5D Lidar device, in which 2D Lidar is mounted onto a 1-DOF liner stage that moves along the vertical direction. As a result, the 2.5D Lidar device can also perform vertical scanning within the motion range of the liner stage. Consequently, a robust SLAM algorithm based on the 2.5D Lidar device is proposed for the omni-directional mobile robot.

Some researches on the motion control of omni-directional mobile robotics have been extensively studied. A developed controller for the three-wheeled omni-directional mobile robot for the operation on a flat terrain is presented. In the paper, the controller was designed by using layered architecture, and each layer, which is named respectively with world frame, robot frame and motor frame, can be independently controlled without affecting others [22]. The omni-directional mobile robotics usually has complex mechanical structures, there is the problem of external disturbances and system parameter uncertainties in the motion control. In order to solve this problem, a passivity-based active disturbance rejection control method is proposed [23]. In which the system disturbances are estimated by the extended state observer, and the disturbance compensations are added in a controller to eliminate the influence of external disturbances and parameter uncertainties in the system. For the similar problem, some control strategies are proposed by using intelligent control technology or other control theories, just like machine learning, fuzzy control, neural network, sliding mode control, optimal control and so on. Adaptive extension intelligent control technology is used to the motion control of one kind of three wheeled omni-directional mobile robots on uneven pavement, which makes the robot have the ability of high precision, fast reaction rate, self-organizing and self-learning[24].An adaptive fuzzy cerebellar model articulation controller (AFCMAC) for solving the tracking control problem for an omni-directional mobile robot is presented, where the fuzzy logic and CMAC are combined [25].A developed machine learning method based on Gaussian Process (GP) is proposed and applied to an omni-directional mobile robotic platform. Because the GP offers great precision in learning, and the data arrives frequently and abundantly, the control in robots has a good performance [26]. Paper [27] presents the Quasi-Newton method with Broyden-Fletcher-Grodfarb-Shanno (BFGS) for online training the RBFNN. After being trained, the RBFNN is applied to control omni-directional mobile robot based on sliding mode controller. Based on the accurate dynamic model including actuator dynamics and the Coriolis force of a three-wheeled omni-directional mobile robot, a generalized minimum-energy point-to-point trajectory planning algorithm is studied, which is obtained using Pontryagin's minimum principle [28]. These control methods mentioned in the researches above are almost proposed based on the dynamic model, which is usually complex, not suitable for real application.

Different from above, some controllers are designed based on the kinematic model, which is relatively simply. Considering that the four-wheeled omni-directional mobile robot driven by four electric motors is a typical redundant system, based on its kinematic model, a minimum-energy trajectory tracking algorithm is proposed in paper [29]. Because the controller can be designed simply and with stable performance, the back-stepping technique has been widely used in the trajectory tracking control. A backstepping-based controller is proposed with proven global stability by selecting a Lyapunov function and introducing a virtual control input for the built robot model [30]. For the trajectory tracking control of an omni-directional wheeled robot for lower limbs rehabilitative training, the control problem and interference rejection are translated into L2 control design problem, and a tracking controller is presented by considering the back-stepping strategy [31]. And in [32], a tracking controller for omni-directional automated guided vehicle using back-stepping and model reference adaptive control is presented. However, the achievements in these literatures cannot deal with the velocity jumping. If the initial tracking errors were bigger or the trajectory was discrete, there would be velocity jumping in the tracking controller. It means that the acceleration or driving torque of the robot must be big enough, or even unlimited, which is impossible in practical application.

In this paper, an adaptive back-stepping tracking control based on a bounded function (BTCBF) is presented for a four-wheel omni-directional mobile robot (FOMR). Firstly, based on its kinematic model, the trajectory tracking error model is built by analyzing the motion relationship in two coordinate systems (one is the robot own coordinate frame, and the other is the world coordinate frame). After that a trajectory tracking controller is designed by the Lyapunov's stableness theory. And then a bounded function with limited smooth output is designed and used into the controller to solve the velocity jumping. Finally, an adaptive law is introduced to improve the response speed and tracking accuracy of the control system. For different trajectories, compared with the common back-stepping control, the effectiveness of the controller is demonstrated through simulation. The results show that the control system with the BTCBF has better tracking performance with smooth control variables.

II. Model of FOMR

A. Kinematic Model of FOMR

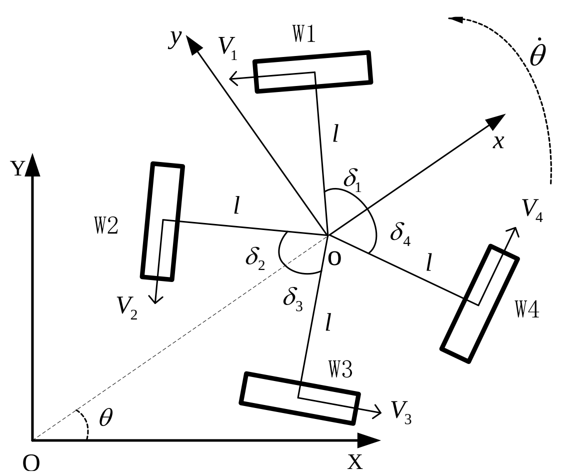


Fig. 1. Wheel placement of mobile robot

Four-wheel robots are one of the models of robots, which are used in many domains. They are omni-directional with four wheels that have the ability of moving to any direction at any time (they are holonomic mobile robots, in other words). Figure 1 shows the schematic of a four-wheel robot, the angles and directions of the four wheels. Where XOY is the world coordinate frame for robot, xoy is the robot own coordinate frame, denotes the moving direction of robot, $W_i\ (i=1,2,3,4)$ denotes every wheel, δ_i denotes the angle between wheel and x axis respectively, V_i denotes the velocity of each wheel, its positive direction is anticlockwise, l is the distance between the center of robot-body and that of each wheel. Then we have the motion function as formula (1).

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{pmatrix} -\sin(\theta+\delta_1) & \cos(\theta+\delta_1) & l \\ \sin(\theta-\delta_2) & -\cos(\theta-\delta_2) & l \\ \sin(\theta+\delta_3) & -\cos(\theta+\delta_3) & l \\ -\sin(\theta-\delta_4) & \cos(\theta-\delta_4) & l \end{pmatrix} \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{pmatrix} \quad (1)$$

According to the motion relationship in Figure 2, supposing that the pose (position and orientation) of robot in its own coordinate frame and the world coordinate frame is separately expressed as $[x\ \ y\ \ \varphi]^T$ and $[X\ \ Y\ \ \psi]^T$, the velocity vector in the own coordinate frame is $[u\ \ v\ \ \omega]^T = [\dot{x}\ \ \dot{y}\ \ \dot{\varphi}]^T$.

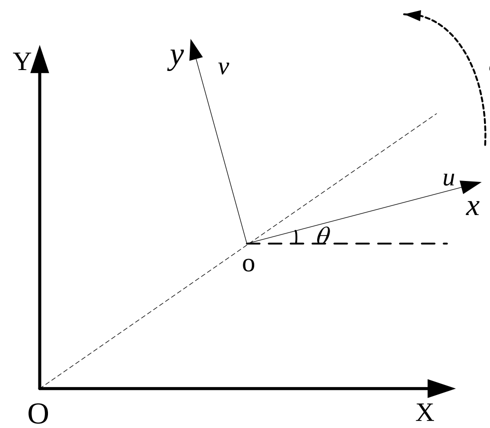


Fig. 2. The motion relationship of mobile robot

Then the kinematical equation of mobile robot can be constructed as formula (2).

$$\begin{cases} \dot{X} = u\cos\theta - v\sin\theta, \\ \dot{Y} = u\sin\theta + v\cos\theta, \\ \dot{\psi} = \omega. \end{cases} \quad (2)$$

Where $\theta = \psi - \varphi$.

B. Trajectoty Tracking Error Model of FOMR

Considering the reference posture in the world coordinate frame and robot own coordinate frame is $[X_r\ Y_r\ \psi_r]^T$ and $[x_r\ \ y_r\ \ \varphi_r]^T$, and then the reference velocity is $[\dot{X}_r\ \ \dot{Y}_r\ \ \dot{\psi}_r]^T$ and $[u_r\ v_r\ \omega_r]^T$.

Defining the errors between reference pose and real pose in the world coordinate frame and robot own coordinate frame are $\boldsymbol{\varepsilon}^T = [e_X\ \ e_Y\ \ e_\psi]^T$ and $\mathbf{e}^T = [e_x\ e_y\ \ e_\varphi]^T$, then we have the relationship between them is as formula (3).

$$\begin{bmatrix} e_x \\ e_y \\ e_\varphi \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_X \\ e_Y \\ e_\psi \end{bmatrix}. \quad (3)$$

In formula (3), $\boldsymbol{\varepsilon}^T = [e_X \;\; e_Y \;\; e_\psi]^T = [X_r - X \;\; Y_r - Y \;\; \psi_r - \psi]^T$, and $\mathbf{e}^T = [e_x \; e_y \;\; e_\varphi]^T = [x_r - x \;\; y_r - y \;\; \varphi_r - \varphi]^T$.

Introducing formula (3) into formula (2), we have the model of kinematical error is as formula (4).

$$\begin{cases} \dot{e}_x = -u + v_p \cos\varphi + \omega e_y \\ \dot{e}_y = -v + v_p \sin\varphi - \omega e_x \\ \dot{e}_\varphi = -\omega + \omega_r \end{cases} . \tag{4}$$

In formula (4), $v_p = \sqrt{\dot{X}_r^{\,2} + \dot{Y}_r^{\,2}} = \sqrt{u_r^{\,2} + v_r^{\,2}}$.

From formula (3), we obtain that the trajectory tracking errors $\mathbf{e}^T(t)$ in robot own coordinate frame, $\lim_{t\to\infty}\left\|\mathbf{e}^T(t)\right\| = 0$, meanwhile, the errors $\boldsymbol{\varepsilon}^T(t)$ in the world coordinate frame, $\lim_{t\to\infty}\left\|\boldsymbol{\varepsilon}^T(t)\right\| = 0$.

The trajectory tracking problem can be described as follows, for the robot control system described as formula (2), it means that the tracking error $\boldsymbol{\varepsilon}^T(t)$ in closed control loop can be globally asymptotically stabilized to zero with the suitable undetermined control law $\mathbf{U} = [u \;\; v \;\; \omega]^T$.

III. Design of BTCBF

A. Design of Trajectory Tracking Controller

Choosing the following Lyapunov candidate function $V_1 = \frac{1}{2}(e_x^{\,2} + e_y^{\,2} + e_\varphi^{\,2})$, considering formula (4), the time derivative is given as formula (5).

$$\begin{aligned} \dot{V}_1 &= e_x\dot{e}_x + e_y\dot{e}_y + e_\theta\dot{e}_\theta = e_x(-u + v_p\cos\varphi + \omega e_y) \\ &\quad + e_y(-v + v_p\sin\varphi - \omega e_x) + e_\varphi(\omega_r - \omega) \\ &= e_x(-u + v_p\cos\varphi) + e_y(-v + v_p\sin\varphi) \\ &\quad + e_\varphi(\omega_r - \omega) \\ &= e_x(-u + \dot{X}_\mathrm{r}\cos\theta + \dot{Y}_\mathrm{r}\sin\theta) \\ &\quad + e_y(-v - \dot{X}_\mathrm{r}\sin\theta + \dot{Y}_\mathrm{r}\cos\theta) + e_\varphi(\omega_r - \omega) \end{aligned} . \tag{5}$$

By the theory of Lyapunov stability, we can obtain the trajectory tracking control law as follows.

$$\mathbf{U} = \begin{bmatrix} u \\ v \\ \omega \end{bmatrix} = \begin{bmatrix} k_1 e_x + \dot{X}_r\cos\theta + \dot{Y}_r\sin\theta \\ k_2 e_y - \dot{X}_r\sin\theta + \dot{Y}_r\cos\theta \\ k_3 e_\varphi + \omega_r \end{bmatrix}. \tag{6}$$

Where k_1, k_2, k_3 are the positive constants.

B. Design of Bounded Function

Sigmoid function is the bounded function of S type in Biology, it can map the real number field to bounded domains [0, 1], and it has been applied in many fields, such as neural network, information science, population studies, economical research and so on. A typical simple model is described as formula (7), and its curve is shown in Figure 3.

$$S(x) = 1/(1 + e^{-x}). \tag{7}$$

Based on formula (7), a bounded function with smooth output is designed as,

$$E(t) = \mathrm{sign}(e(t))\frac{Be(t)}{A + e(t)}(1 - \mathrm{e}^{-\mathrm{sign}(e(t))(A + e(t))t}) . \tag{8}$$

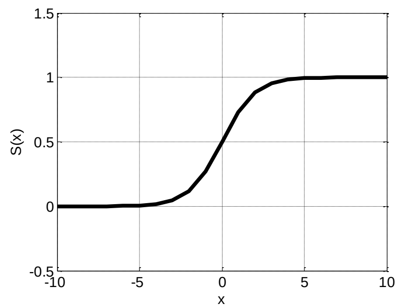


Fig. 3. Curve of sigmoid function

Where $E(t)$ is the output, $e(t)$ is the input of the function, A and B are non-negative constants, $\mathrm{sign}(*)$ is the sign function. When $e(t) > 0$, $\mathrm{sign}(e(t)) > 0$.

When $e(t) \to +\infty$, further we have,

$$\lim_{e(t)\to+\infty} E(t) = \lim_{e(t)\to+\infty}\frac{Be(t)}{A + e(t)} = B \tag{9}$$

In the same way, when $e(t) \to -\infty$, we have,

$$\lim_{e(t)\to-\infty} E(t) = \lim_{e(t)\to-\infty} -\frac{Be(t)}{A + e(t)} = -B \tag{10}$$

Even though the input signals $e(t) \to \infty$, the output $E(t)$ of the function described by formula (8) is still limited in the domain of $[-B, B]$.

At initial time $t = 0$, whatever the tracking errors $e(t)$ are, thinking about formula (8), we have $E(0) = 0$. It means that the output of the function (8) is smooth.

C. Trajectory Tracking Controller with Bounded Function

Considering the actual application, the formula (8) can be rewritten as,

$$\begin{aligned} &E_i(t) = \mathrm{sign}(e_i(t))\frac{B_i e_i(t)}{A_i + e_i(t)}(1 - \mathrm{e}^{-\mathrm{sign}(e_i(t))(A_i + e_i(t))t}), \\ &\quad i = x, y, \varphi \end{aligned} . \tag{11}$$

e_i are the tracking errors in robot own coordinate frame.

Based on the above analysis, if $E(t)$ were chosen to be the tracking errors $e(t)$, the jumping velocity caused by initial errors would be restrained.

Then we use E_i instead of the tracking error e_i and taking it into (6), we have the trajectory tracking control law described as formula (12).

$$\mathbf{U} = \begin{bmatrix} u \\ v \\ \omega \end{bmatrix} = \begin{bmatrix} k_1 E_x + \dot{X}_r\cos\theta + \dot{Y}_r\sin\theta \\ k_2 E_y - \dot{X}_r\sin\theta + \dot{Y}_r\cos\theta \\ k_3 E_\varphi + \omega_r \end{bmatrix}. \tag{12}$$

To analyze the control system stability, a Lyapunov candidate function has been constructed as follows.

$$V = \frac{1}{2}(B_x e_x^{\ 2} + B_y e_y^{\ 2} + B_\varphi e_\varphi^{\ 2} + k_1 E_x^{\ 2} + k_2 E_y^{\ 2} + k_3 E_\varphi^{\ 2}) \cdot \tag{13}$$

According to formula (11), when $e_i(t) \geq 0$, we can obtain its derivative,

$$\dot{E}_i = \begin{cases} -A_i E_i + (B_i - E_i)e_i \ , \ e_i \geq 0 \\ -A_i E_i + (B_i + E_i)e_i \ , \ e_i < 0 \end{cases}. \tag{14}$$

Then we can get the derivative of formula (13) when $e_i(t) \geq 0$.

$$\begin{aligned} \dot{V} &= B_x e_x \dot{e}_x + B_y e_y \dot{e}_y + B_\varphi e_\varphi \dot{e}_\varphi + k_1 E_x \dot{E}_x + k_2 E_y \dot{E}_y \\ &\quad + k_3 E_\varphi \dot{E}_\varphi \\ &= -B_x k_1 e_x E_x - B_y k_2 e_y E_y - B_\varphi k_3 e_\varphi E_\varphi \\ &\quad + k_1(-A_x - e_x) E_x^{\ 2} + k_1 B_x e_x E_x \\ &\quad + k_2(-A_y - e_y) E_y^{\ 2} + k_2 B_y e_y E_y \\ &\quad + k_3(-A_\varphi - e_\varphi) E_\varphi^{\ 2} + k_3 B_\varphi e_\varphi E_\varphi \\ &= k_1(-A_x - e_x) E_x^{\ 2} + k_1(B_x e_x - B_x e_x) E_x \\ &\quad + k_2(-A_y - e_y) E_y^{\ 2} + k_2(B_y e_y - B_y e_y) E_y \\ &\quad + k_3(-A_\varphi - e_\varphi) E_\varphi^{\ 2} + k_3(B_\varphi e_\varphi - B_\varphi e_\varphi) E_\varphi \\ &= -k_1(A_x + e_x) E_x^{\ 2} - k_2(A_y + e_y) E_y^{\ 2} \\ &\quad - k_3(A_\varphi + e_\varphi) E_\varphi^{\ 2} \end{aligned} . \tag{15}$$

In formula (14), $A_x > 0$, $A_y > 0$, $A_\varphi > 0$, $k_1 > 0$, $k_2 > 0$, $k_3 > 0$, then we have formula (16).

$$\dot{V} \leq 0. \tag{16}$$

Similarly, when $e(t) < 0$, we have $\dot{V} \leq 0$.

Then based on the Lyapunov theory, we can get that the system with the control law in formula (12) is asymptotically stabilized.

D. Design of Adaptive Law

The performance of the control system will be bad because that the outputs in formula (12) are limited. To improve the response speed and tracking accuracy of the control system, the non-negative constant A, which is not any more an invariable constant, is a time-varying variable. Then an adaptive law of A is designed as formula (17),

$$\dot{\hat{A}}_i = \alpha_i \hat{E}_i(\hat{E}_i - E_i), \ \ \alpha_i > 0. \tag{17}$$

Where $\hat{E}_i$ is the estimated value of E_i, and $\hat{A}_i$ is the estimated value of A_i.

$\tilde{E}_i = E_i - \hat{E}_i$ is the estimated error of E_i, considering the Lyapunov's function as formula (18),

$$V_2 = \frac{1}{2}\tilde{E}_i^{\ 2} + \frac{1}{2\alpha_i}(A_i - \hat{A}_i)^2 \cdot \tag{18}$$

When $e(t) > 0$, we have the derivative of formula (19).

$$\begin{aligned} \dot{V}_2 &= (E_i - \hat{E}_i)(\dot{E}_i - \dot{\hat{E}}_i) \\ &\quad - \frac{1}{\alpha_i}[(A_i - \hat{A}_i)\alpha_i \hat{E}_i(\hat{E}_i - E_i)] \\ &= (E_i - \hat{E}_i)[-A_i E_i + (B_i - E_i)e_i + \hat{A}_i \hat{E}_i \\ &\quad - (B_i - \hat{E}_i)e_i] - (A_i - \hat{A}_i)\hat{E}_i(\hat{E}_i - E_i) \\ &= (E_i - \hat{E}_i)[-A_i E_i + \hat{A}_i \hat{E}_i + (B_i - E_i - B_i + \hat{E}_i)e_i] \\ &\quad + (A_i - \hat{A}_i)\hat{E}_i(E_i - \hat{E}_i) \\ &= (E_i - \hat{E}_i)[-A_i(E_i - \hat{E}_i) - (E_i - \hat{E}_i)e_i] \\ &= -(A_i + e_i)(E_i - \hat{E}_i)^2 \end{aligned} . \tag{19}$$

Because that $e_i(t) > 0$, $A_i > 0$, then $\dot{V}_2 < 0$.

Similarly, when $e(t) < 0$, we have $\dot{V}_2 < 0$.

It means that the estimation errors in formula (18) can converge to zero in finite time and the control system is globally asymptotically stabilized.

Then we can obtain the adaptive trajectory tracking law as formula (20).

$$\mathbf{U} = \begin{bmatrix} u \\ v \\ \omega \end{bmatrix} = \begin{bmatrix} k_1 \hat{E}_x + \dot{X}_r \cos\theta + \dot{Y}_r \sin\theta \\ k_2 \hat{E}_y - \dot{X}_r \sin\theta + \dot{Y}_r \cos\theta \\ k_3 \hat{E}_\varphi + \omega_r \end{bmatrix}. \tag{20}$$

IV. Simulation and Analysis

We conducted numerical simulations at Matlab 2016Ra platform to assess the performance of the controller given in formula (20). The values of the control parameters correspond to a laboratory prototype built in our institution and they can be found as follows.

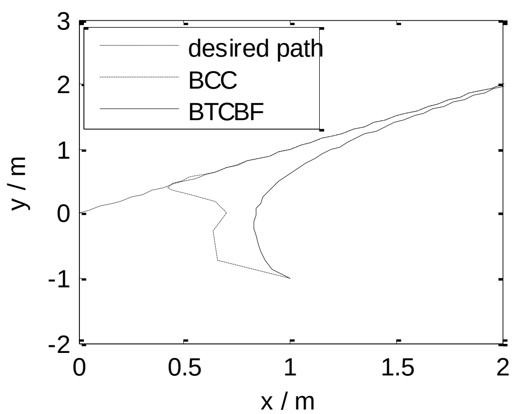
The speed constraints of our mobile robot are $|u| \leq 3m/s$, $|v| \leq 3m/s$ and $|\omega| \leq 3rad/s$, sampling time is 0.01s, parameters of BTCBF are $k_1 = 1$, $k_2 = 1$, $k_3 = 3$, $B_x = 3$, $B_y = 3$, $B_\varphi = 3$, $\alpha_x = 0.5$, $\alpha_y = 0.5$, $\alpha_\varphi = 0.5$, $\hat{A}_i(0) = 5$.

To confirm the algorithm's effectiveness, comparison experiments are done by using back-stepping tracking control (BCC, in lecture [30]), tracking control law (TCL) described in formula (12) and BTCBF in formula (20). Section A and Section B show the tracking results of a straight path and a circle path by BCC and BTCBF, and Section C gives the comparison results among three approaches.

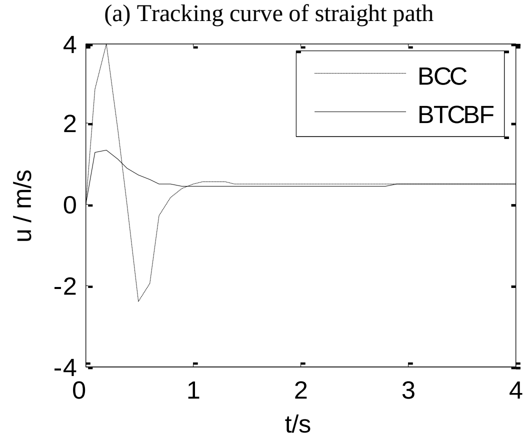
A. Results of Tracking a Straight Path

For the straight path, its equation is $Y_r = X_r$, angle is $\varphi_r = \pi/4$, the desired line velocities of robot are $u_r = 0.5\,m/s$, $v_r = 0.5\,m/s$, the desired angle velocity is $\omega_r = 0\,rad/s$, actual initial poses are $(1m, \ -1m, \ \pi/2rad)^{\mathrm{T}}$, and its reference poses are $(0,0,0)^{\mathrm{T}}$, then initial tracking errors are $(-1m, \ 1m, \ -\pi/2rad)^{\mathrm{T}}$, the simulation results are shown in Figure 4.

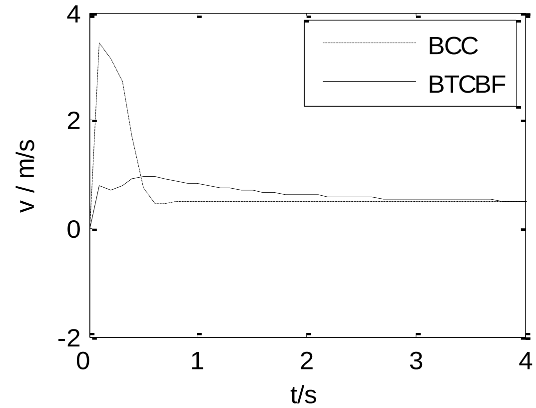
Figure 4 (a) shows that the robot with BTCBF can track the straight path and correct deviations quickly (about 3 s). From Figure 4 (b) to Figure 4 (d), at the time $t_0 = 0$, even if the initial errors are bigger, which are $(-1\,m,\ 1\,m,\ -\pi/2\,rad)^{\mathrm{T}}$, the outputs of the controller still remain at their zero neighborhoods. And the maximum of line velocities and angle velocity are $u = 0.68m/s$, $v = 0.69m/s$, $\omega = 1.6rad/s$, which are much smaller than their upper bounds. Figure 4 show that the robot with BCC can track the path more quickly than BTCBF. It is because that the BCC can produce the large outputs at the initial time, which have already exceeded the speed bounds, and the outputs of BCC are not smooth enough. It means that the motors of the robot must produce an infinite and abrupt driving torque at initial time, which is impossible in the actual system.



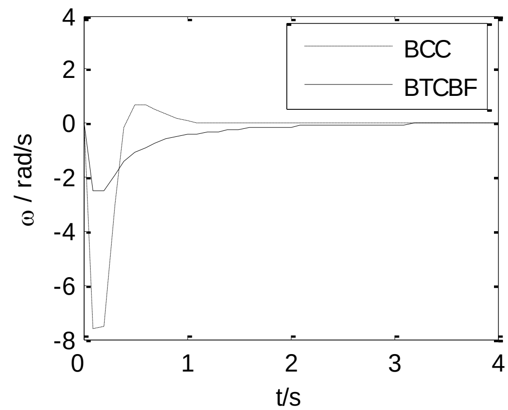
(a) Tracking curve of straight path



(b) Curve of line velocity u



(c) Curve of angle velocity v

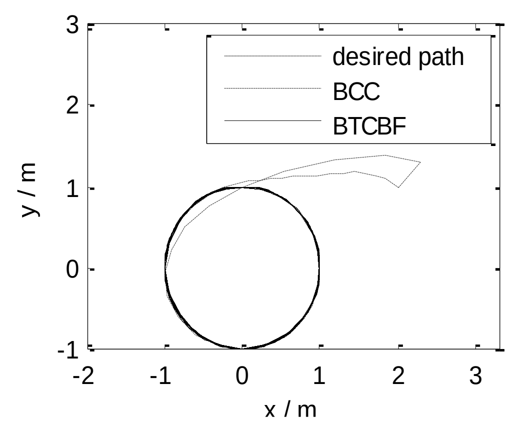


(d) Curve of angle velocity ω

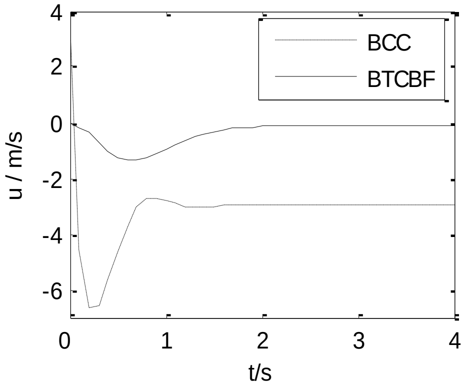
Fig. 4. Tracking results of straight path

B. Results of Tracking a Circle Path

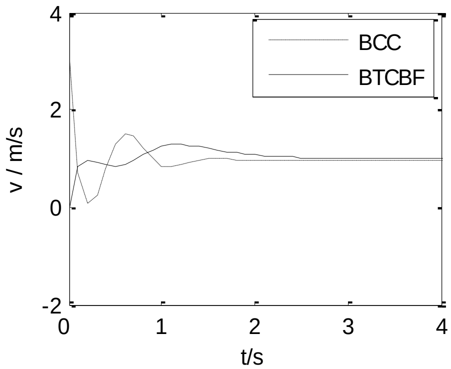
When the tracking trajectory is the circle path, its equation is $X_r = r\cos t$, $Y_r = r\sin t$, $\varphi_r = t$, where $r = 1m$, $0 \le t \le 15$, at time $t_0 = 0\ s$, actual initial poses are $(2.5m, 2m, \pi/2\,rad)^{\mathrm{T}}$, and its reference poses are $(1m, 0m, 0\,rad)^{\mathrm{T}}$, then initial tracking error is $(-1m, -2.5m, -\pi/2\ \ rad)^{\mathrm{T}}$. The simulation results are shown in Figure 5.



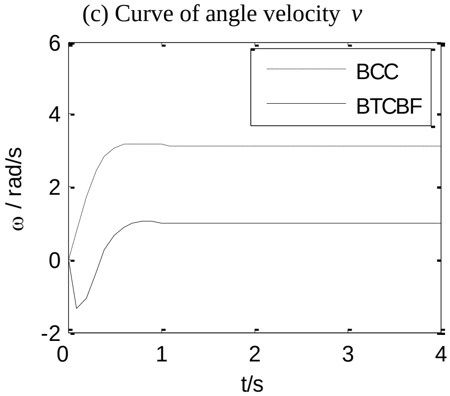
(a) Tracking curve of circle path



(b) Curve of line velocity u



(c) Curve of angle velocity v



(d) Curve of angle velocity ω

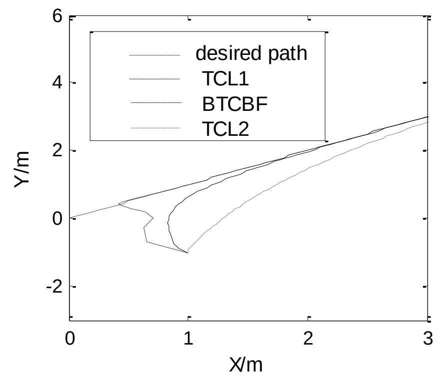
Fig. 5. Tracking results of circle path

Figure 5 (a) shows that the robot with BTCBF can also track the time varying path and correct deviations quickly. In the other subplots of Figure 5, the actual line velocities and angle velocity with BTCBF within their upper bounds are much smoother than that with BCC. Specially, in Figure 5 (b), the biggest value of the actual line velocity is $|u| = 6.6\ m/s$, it is obviously that actual line velocity u has already exceeded its maximum value.

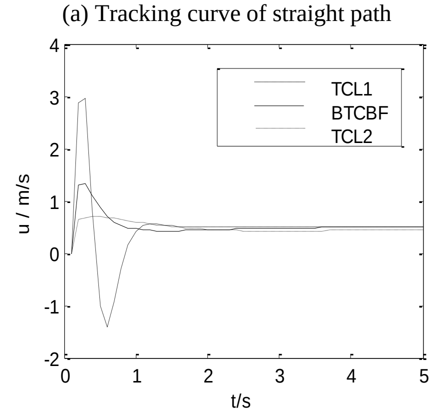
From Figure 4 and Figure 5, because of the large errors at initial moment, the velocities alter from zero to large values quickly, then, they reduce towards opposite direction. This action is done repeatedly until the velocities reach the proper values. In this process, the velocities produced by BCC, have a cyclical shock and maybe exceed their bounds at initial moment or when there are bigger errors. However, for the characters of bound function described in formula (8), the velocities produced by BTCBF are smooth and bounded no matter when the robot will track a desired path.

C. Comparison Results

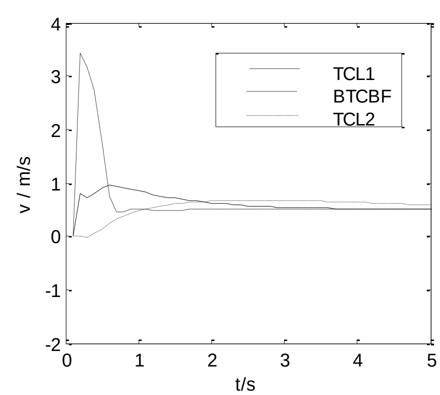
To confirm the algorithm's effectiveness, the experiments of tracking a straight path are done by using three approaches, BTCBF, TCL1 and TCL2. The parameters A_i and B_i in TCL1 are larger than that in TCL2. The parameters of TCL1 are $k_1 = 1$, $k_2 = 1$, $k_3 = 3$, $A_x = 10$, $A_y = 10$, $A_\varphi = 10$, $B_x = 10$, $B_y = 10$, $B_\varphi = 10$, the parameters of TCL2 are $k_1 = 1$, $k_2 = 1$, $k_3 = 3$, $A_x = 1$, $A_y = 1$, $A_\varphi = 1$, $B_x = 1$, $B_y = 1$, $B_\varphi = 1$. Comparison results can be found in Figure 6 and Table I.



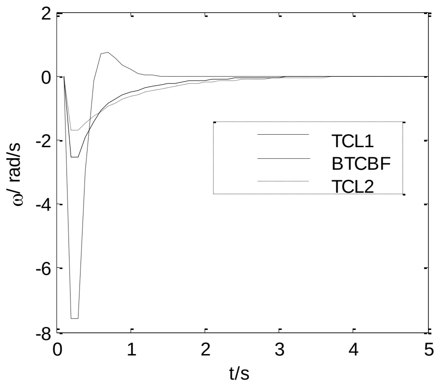
(a) Tracking curve of straight path



(b) Curve of line velocity u



(c) Curve of angle velocity v



(d) Curve of angle velocity ω

Fig. 6. Tracking results of straight path

Table. Ⅰ Comparison results of line velocity u

Algorithms	Stabilization time (s)	Max (m/s)	Mean (m/s)	Variance
TCL1	1.8	3.1958	0.4987	0.1440
BTCBF	3.3	1.3193	0.5011	0.0465
TCL2	6.1	0.6893	0.4978	0.0254

Figure 6 shows that the robot can track the desired path in time by using three different approaches. According to Figure 6 (b) and Table Ⅰ, the stabilization time of TCL1 is 1.8 second, which is shorter than the others. It means that the TCL1 has the fastest convergence speed. From Figure 6 the TCL1's velocity curves have larger oscillations. It is because that TCL1 has the larger parameters, with which TCL1 can product the larger outputs at the initial time. In Table Ⅰ the TCL1's maximum line velocity u is 3.1958 m/s, which has already exceeded the robot's real bound 3 m/s, however it is still smaller than its parameter $B_x = 10$ for the feature of formula (12).

On the other hand, by Figure 6 and Table Ⅰ, TCL2 has the slowest convergence speed and its velocity curves are smoother than others, because TCL2 has the smallest parameters A_i and B_i among three different approaches. Although the small parameters A_i and B_i can make the control outputs remain in their limits, but the response speed and tracking accuracy of the control system are reduced. By using BTCBF, the robot can track the desired path quickly with smooth velocity outputs and the control system has good performance. Because the parameters A_i can adjust via self-adaptive behaviors according to the errors situation and they are suitable for the control system.

V. Conclusions

The velocity jumping problem in trajectory tracking control for a four-wheel omni-directional mobile robot has been studied in this paper. Base on the kinemics model, a back-stepping controller is designed and improved by a designed bounded function, and the bounded function is applied to smooth the control output for conquering the velocity jumping problem. To further improve the response speed and tracking accuracy of the control system, a parameter adaptive law is proposed and used in the back-stepping controller. By using Lyapunov's stable theory, the controller scheme is demonstrated to be stable. For different tracking tasks, the effectiveness of the controller is demonstrated through simulation experiments. In the simulation the control system has proper outputs with smaller variances. All the simulation results indicate that the proposed control strategy is effective to solve the velocity jumping problem in path tracking.

References

[1] T. Kenjiro, T. Eri, F. Masahiro, N. Akito, K. Hirone, K. Masashi and T. Satoshi,"Planar omnidirectional crawler mobile mechanism development of actual mechanical prototype and basic experiments", *IEEE Robotics and Automation Letters*, vol. 3, no.1, pp. 395-402, 2018.

[2] Mohamed Oussama Ben Salem, Olfa Mosbahi,"Towards a Safe Development of Reconfigurable Robotic Systems",WSEAS Transactions on Systems and Control,vol. 14, pp. 359-374, 2019.

[3] S. Luís, N. Francisco, M.Ricardo Santos, S. João and E. João Sena, "Remote control system for a mobile platform with four Mecanum wheels", *International Journal of Mechatronics and Applied Mechanics*, vol. 2017, no.1, pp. 274-281, 2017.

[4] Lyubomira Miteva, Kamen Delchev, Kaloyan Yovchev, Evgeniy Krastev,"Design of Biped Robot with Anthropomorphic Gait",WSEAS Transactions on Systems and Control, vol. 14, pp. 257-263, 2019.

[5] G. T. Bae, S. W. Kim, D. Choi, C. Cho, W. S. Lee and S. C. Kang, "Omni-directional power-assist-modular(PAM) mobile robot for total nursing service system", in *Proc. 14th International Conference on Ubiquitous Robots and Ambient Intelligence*, Jeju, 2017, pp. 832-834.

[6] Y. B. Feng, C. J. Ding, X. Li and X. H. Zhao, "Integrating Mecanum wheeled omni-directional mobile robots in ROS", in *Proc. 2016 IEEE International Conference on Robotics and Biomimetics*, Qingdao, 2016, pp. 643-648.

[7] K. B. Zhang and L. Zhang, "Indoor omni-directional mobile robot that track independently", *Journal of Computers (Taiwan)*, vol. 29, no.2, pp. 118-135, 2018.

[8] S. T. Kao, Y. Wang and M. T. Ho, "Ball catching with omni-directional wheeled mobile robot and active stereo vision", in *Proc. 2017 IEEE International Symposium on Industrial Electronics*, Edinburgh, 2017, pp.1073-1080.

[9] J. Qian, B. Zi, D. M. Wang, Y. G. Ma and D. Zhang, "The design and development of an Omni-Directional mobile robot oriented to an intelligent manufacturing system", *Sensors (Switzerland),* vol. 17, no.9, pp. 118-125, 2017.

[10] T. R. Peng, J. Qian, B. Zi, J. K. Liu and X. W. Wang, "Mechanical design and control system of an omni-directional mobile robot for material conveying", in *Proc. 9th International Conference on Digital Enterprise Technology*, Nanjing, 2016, pp. 412-415.

[11] G. Runge, G. Borchert, P. Henke and A. Raatz, "Design and testing of a 2-DOF ball drive: an omnidirectional wheel for mobile robots", *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 81, no.2, pp. 195-213, 2016.

[12] Z. Q. Gao, Y. X. Yang, Y. P. Du and Y. Zhang, "Design and development of an omni-directional mobile robot for logistics of printing enterprise", *International Journal of Simulation: Systems, Science and Technology*, vol. 17, no.47, pp. 171-176, 2016.

[13] K. S. Yoon, M. S. Lee and Y. W. Sung, "A study on an omni-directional mobile robot for moving a double-parked car", *Transactions of the Korean Institute of Electrical Engineers*, vol. 67, no.3, pp. 440-447, 2018.

[14] Z. Guo, R. B. Yee, K. R. Mun and H. Y. Yu, "Experimental evaluation of a novel robotic hospital bed mover with omni-directional mobility", *Applied Ergonomics*, vol. 65, no.1, pp. 389-397, 2017.

[15] I. Doroftei, I. Conduraru and V. Horga, "Preliminary ideas on the odometry of an omni-directional mobile robot", *Advances in Intelligent Systems and Computing*, vol. 371, no.1, pp. 157-164, 2016.

[16] S. Nakayama, I. Matsumoto and M. Uchida, “Self-localization by omni-directional camera and luminous landmarks for autonomous mobile robot”, in *Proc. 10th IEEE Region Annual International Conference*, Penang, 2017, pp. 3490-3493.

[17] X. F. Liu, H. L. Chen, C. C. Wang, F. Hu and X. Q. Yang, “MPC control and path planning of omni-directional mobile robot with potential field method”, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10985 LNAI, no.1, pp. 170-181, 2018.

[18] C. Y. Chen, S. Y. Chiang and C. T. Wu, “Path planning and obstacle avoidance for omni-directional mobile robot based on Kinect depth sensor”, *International Journal of Embedded Systems*, vol. 8, no.4, pp.343-351, 2016.

[19] J. T. Zou and C. Y. Wang, “The application of ultrasonic positioning system for indoor navigation of omni-directional mobile robot”, *Sensors and Materials*, vol. 28, no.6, pp. 585-591, 2016.

[20] H. Chang, H. J. Zhang, X. Yang, W. Yang, C. Y. Chen and G. L. Yang, “An improved FastSLAM algorithm based on an omni-directional wheeled mobile robot”, in *Proc. 20th IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Banff, 2016, pp.884-888.

[21] Y. Yang, G. L. Yang, Y. Z. Tian, T. J. Zheng, L. Li and Z. H. Wang, “A robust and accurate SLAM algorithm for omni-directional mobile robots based on a novel 2.5D lidar device”, in *Proc. 13th IEEE Conference on Industrial Electronics and Applications*, Singapore, 2018, pp. 2123-2127.

[22] I. Galgamuwa, L. Samaranayake, P. Ekanayake and K. Liyanage, “Simplified controller for a three wheeled omni directional mobile robot”, *Control and Intelligent Systems*, vol.44, no.3, pp. 112-120, 2016.

[23] S. G. Ma, J. L. Zhao and C. Ren, “Passivity-based active disturbance rejection control of an omni-directional mobile robot”, *Control and Decision*, vol.33, no.6, pp. 1081-1086, 2018.

[24] C. C. Jiang and Y. N. Rui, “Research on the intelligent control technology of the uneven road omni-directional mobile robot”, *Scientia Sinica Technologica*, vol.47, no.5, pp. 463-471, 2017.

[25] T. F. Wu, H. C. Huang, P. S. Tsai, N. T. Hu and Z. Q. Yang, “The tracking control design of adaptive fuzzy CMAC for an omni-directional mobile robot”, *Journal of Computers (Taiwan)*, vol.28, no.1, pp. 247-260, 2017.

[26] D. Garcia, E. Gorrostieta, E. V. Soto, R. Cristian Rodriguez and D. Guillermo Diaz, “Learning from demonstration with Gaussian process approach for an omni-directional mobile robot”, *IEEE Latin America Transactions*, vol.16, no.4, pp. 1250-1255, 2018.

[27] T. T. Pham, D. Van Huong, C. N. Nguyen and T. Le Minh, “Online training the radial basis function neural network based on Quasi-Newton algorithm for omni-directional mobile robot control”, *Lecture Notes in Electrical Engineering*, vol.465, no.1, pp. 607-616, 2018.

[28] H. J. Kim and B. K. Kim, “Minimum-energy cornering trajectory planning with self-rotation for three-wheeled omni-directional mobile robots”, *International Journal of Control, Automation and Systems*, vol.15, no.4, pp. 1857-1866, 2017.

[29] J. B. Wang, J. P. Chen and Q .J. Xiao, “A minimum-energy trajectory tracking controller for four-wheeled omni-directional mobile robot”, in *Proc. 15th International Conference on Control, Automation, Robotics and Vision*, Singapore, 2018, pp. 48-53.

[30] Q. Z. Cui, X. Li, X. K. Wang and M. Zhang, “Backstepping control design on the dynamics of the omni-directional mobile robot”,*Applied Mechanics and Materials*, vol.203, no.1, pp. 51-56, 2012.

[31] H. C. Huang and C. C. Tsai, “Adaptive robust control of an omnidirectional mobile platform for autonomous service robots in polar coordinates”, *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol.51, no.4, pp. 439-460, 2008.

[32] H. H. Nguyen, C. K. Kim, V. C. Ho, T. L. Bui, H. K. Kim, C. H. Lee and S. B. Kim, “Tracking controller design for omni-directional automated guided vehicles using backstepping and model reference adaptive control”, *Lecture Notes in Electrical Engineering*, vol.46, no.5, pp.715-725, 2018.

Jianping Chen was born on Nov. 7, 1975**.** He received the Ph.D. degree in control theory and control engineering from Guangdong University of Technology, China. Currently, he works as a professor in Zhaoqing University, China. His major research interests include artificial intelligence and intelligent system, multi-robot cooperation, motion control of mobile robot, intelligent solar drying technology, and so on. He is invited as a reviewer by the editors of some journals, such as *Journal of Theoretical and Applied Information Technology, Journal of Communications, Journal of Zhaoqing University, International Journal of Information and Computer Security*, etc. He has published more than fifty papers in related journals or international conferences. In addition, he is a co-chair of International Conference on Control, Automation, Robotics and Vision in recent years.

Jianbin Wang was born on Apr. 11, 1982**.** He received the Ph.D degree in control theory and control engineering from Guangdong University of Technology, China. Currently, he is a researcher at Zhaoqing University, China. His major research interests include control theory and intelligent robot control. He has published many papers in related journals.