Tạp chí Khoa học và Công nghệ, Số 40, 2019

DISTRIBUTED FORMATION CONTROL AND OBSTACLE AVOIDANCE OF MULTI-ROBOT SYSTEM

HÀ TRỌNG NGHĨA, TRẦN THANH KẾT, NGUYỄN TẤN LUỸ

Faculty of Electronics Technology, Industrial University of Ho Chi Minh City

nguyentanluy@iuh.edu.vn

Abstract: This paper proposes a distributed control method for multi-mobile robots to avoid obstacles. Firstly, the Limit Cycle (LC) method is exploited to set the reference trajectory for robots to avoid obstacles. Secondly, the control rule that control a leading robot following the reference path is introduced. Thirdly, the algorithm that controls robots moving in a formation and avoiding obstacles based on the combination of the LC method and the reference trajectory tracking algorithm. Different from the distributed control algorithm in related documents, the algorithm in this paper ensures that the robot formation is not only maintained but also avoids obstacles when moving to the target. Finally, simulation and experimental results are conducted to verify the effectiveness of the proposed method.

Keywords: Obstacle Avoidance, Formation Control, Nonholonomic Mobile Robots, Control Architectures, Tracking Control.

1 INTRODUCTION

Robots are increasingly used in life to meet practical needs such as cargo robots, search and rescue robots, adventurous robots, ... Robots can replace human to work in environments with extreme conditions, toxic to human's health ... Recently, research is focused on the study of robot control according to swarm behavior of wild animals. The research directions are only control a single robot attached to the reference angle [3] [4], but these above studies ignore the reference position. There are also many research projects study on controlling a robot attached to the position, reference angle [1] [5]. Therefore, we can combine these algorithms above to create a more complete algorithm to control the robot smoothly.

In many practical applications, when operating a moving robot formation that encounters obstacles, we need a flexible algorithm that can find the trajectory of avoiding the obstacle in the most efficient way. As robots move in warehouses, when encountering obstacles on the road or another robot is on the way, the control algorithm must be flexible to find the path while being safe and most effective. Recently, although there have been many obstacle avoidance algorithms, such as using potential field methods [9], avoiding obstacle with ultrasonic sensor [10], using LC method to find out reference trajectory [3]. The LC method is best suited to the requirements of this paper and has the most flexible problem-solving capabilities. Some studies use the LC method to find out the reference trajectory then applying the backstepping kinematics to control the robot moving to the target [4] [5]. However, they are still not applied to a formation of many robots.

Compared to the existing literature, the new contributions of this paper are listed as follows: 1) exploiting a method to avoid obstacles for single robots to solve the problem of distributed formation control and obstacles avoidance of multi-robot system. 2) designing feedback control law to ensure that when robots avoid obstacles, the stable formation is remained.

The rest of this paper is organized as follows: Section 2 introduces the theoretical basis of limit cycle and reference trajectory tracking, proposed an algorithm to control the distributed formation and obstacles avoidance of multi-mobile robot system. Sections 3 and 4 are simulation and experimental results. Section 5 is the conclusion of the paper.

2 BASIS OF THE CONTROL LAW AND ALGORITHM

2.1 Tracking a reference trajectory

The mobile robot in Fig. 1 is a nonholonomic mechanical system. The system includes two control wheels mounted behind the vehicle and a multi-directional wheel placed in front of the vehicle. The movement of the mobile robot is achieved by two independent torque from DC motors for two wheels.

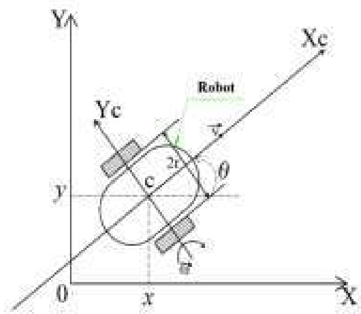

Figure 1. Mobile robot model of nonholonomic mechanical system

The position of the robot in Descartes coordinate system is represented by: $C(x, y)$ (m) is the central position of the robot, $\theta(rad)$ is the angle of the robot compared to the O_x axis, $d(m)$ is the distance from the center of the robot to robot's head position, $2r$ (m) is the distance between two wheels of the robot.

The kinematic equation of motion of leading robot [1]:

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{1}$$

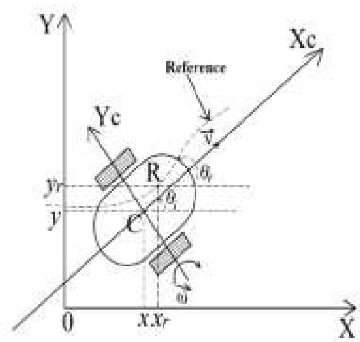

Figure 2. Errors between mobile robot and reference trajectory

In order to control the mobile robot following the reference trajectory, we must control the central position C of the robot following to the reference position R so that the deviation in position, angle and linear velocity reaches 0 when $t \to \infty$.

Figure 2 above describes the errors between mobile robot and reference trajectory. When the robot moves along the reference trajectory R, there will be position and angle errors. To determine the error parameters based on the paper [1], we have the following error equation:

$$e_p = T_e(q_r - q) \tag{2}$$

$$\Leftrightarrow \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \tag{3}$$

where e_1 and e_2 are the error in x and y directions between R and C, respectively. e_3 is the angle error between θ_r and θ.

From formula (3) to find out the law of kinematic control for mobile robots based on [1], the linear velocity and angular velocity are calculated as follows:

$$V_c = \begin{bmatrix} v_1 \\ \mathrm{w}_1 \end{bmatrix} = \begin{bmatrix} v_r \cos e_3 + k_1 e_1 \\ w_r + k_2 v_r e_2 + k_3 v_r \sin e_3 \end{bmatrix} \tag{4}$$

where v_r and w_r are reference linear velocity and angular velocity, respectively. v_r and w_r are linear velocity and angular velocity of the robot respectively and $K = [k_1 \quad k_2 \quad k_3]^T$ is a positive constant vector.

2.2 Formation control for three robots

In this section, configurations of a robot formation is introduced, then LC method is employed to design the control algorithm.

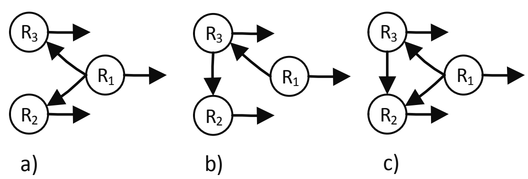

Figure 3. Control methods of the three-robot formation

We have three methods to control the robot formation (Fig. 3) in which R_1 is the leading robot, R_2 and R_3 in order are robot 2 and the robot 3: On Fig. 3 (a) is the first method $SB_{12}C \,\&\, SB_{13}C$. In this method, robot 1 is controlled to keep the desired distance with robot 2 by l_{12}^d and robot 1 is required to maintain the desired distance with robot 3 by l_{13}^d. The second method $SB_{12}C \,\&\, SB_{23}C$ (Fig. 3 (b)), this method keeps the desired distance from robot 1 to robot 2 by l_{12}^d and robot 2 to robot 3 by l_{23}^d by using control algorithm. The last method $SB_{12}C \,\&\, SB_{13}SB_{23}C$ on Fig. 3 (c) is the most stability one. In this method, Robot 3 is controlled to keep the desired distance from robot 1 and robot 2 by l_{13}^d, l_{23}^d ,respectively, robot 1 keeps the desired distance with robot 2 by l_{12}^d. Among the three control methods, the third method is more flexible than the other two. Therefore, this paper employs the third method to control the three-robot formation.

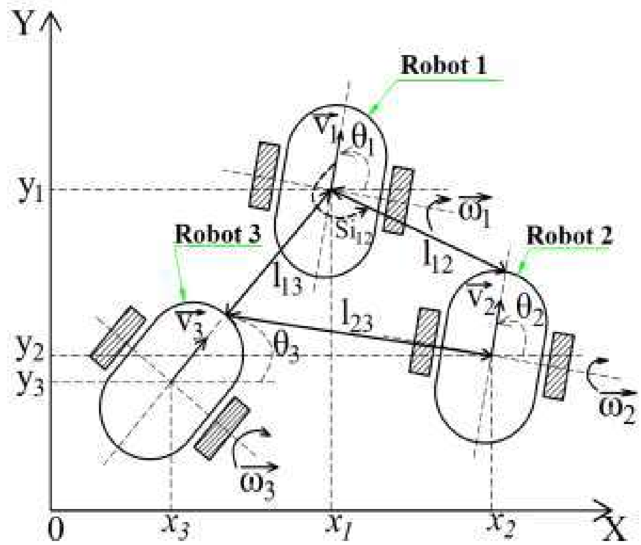

Figure 4. Three-robot formation

Consider a three-robot formation presented in Fig. 4, in which $l_{12}, l_{13}, l_{23}(m)$ respectively are the distances from the center of robot 1 to the center of robot 2, the center of robot 1 to the center of robot 3 and the center of robot 2 to the center of robot 3. Calculating the parameters l_{12}, l_{13}, l_{23} we have:

$$\begin{cases} l_{12}^x = x_1 - x_2 - d\cos\theta_1 \\ l_{12}^y = y_1 - y_2 - d\sin\theta_1 \\ l_{13}^x = x_1 - x_3 - d\cos\theta_1 \\ l_{13}^y = y_1 - y_3 - d\sin\theta_1 \\ l_{23}^x = x_2 - x_3 - d\cos\theta_2 \\ l_{23}^y = y_2 - y_3 - d\cos\theta_2 \end{cases} \Rightarrow \begin{cases} l_{12} = \sqrt{l_{12}^{x\,2} + l_{12}^{y\,2}} \\ l_{13} = \sqrt{l_{13}^{x\,2} + l_{13}^{y\,2}} \\ l_{23} = \sqrt{l_{23}^{x\,2} + l_{23}^{y\,2}} \end{cases} \tag{5}$$

where $\psi_{12}, \psi_{13}, \psi_{23}(rad)$ respectively are the directional angle between robot 1 and robot 2, robot 1 and robot 3, robot 2 and robot 3. Calculate the parameters $\psi_{12}, \psi_{13}, \psi_{23}$:

$$\begin{cases} \psi_{12} = \arctan\left(\dfrac{l_{12}^y}{l_{12}^x}\right) - \theta_1 + \pi \\ \psi_{13} = \arctan\left(\dfrac{l_{13}^y}{l_{13}^x}\right) - \theta_1 + \pi \\ \psi_{23} = \arctan\left(\dfrac{l_{23}^y}{l_{23}^x}\right) - \theta_2 + \pi \end{cases} \tag{6}$$

where $\theta_1, \theta_2, \theta_3(rad)$ are the angle of the robot 1, robot 2, robot 3 compared to the O_x axis respectively.

The kinematic equation is defined as [2]:

$$\begin{aligned} \dot{z} &= G_2(z, \theta_1, \theta_2, \theta_3)u_{23} + F_2(z)u_1 \\ \dot{\theta}_2 &= \omega_2, \dot{\theta}_3 = \omega_3 \end{aligned} \tag{7}$$

where $z = [l_{12} \quad \psi_{12} \quad l_{13} \quad \psi_{13}]^T$ is the system output, $u_{23} = [v_2 \quad \omega_2 \quad v_3 \quad \omega_3]^T$ is the input vector. Noting $\gamma_{ij} = \beta_{ij} + \psi_{ij}$ with $\beta_{ij} = \theta_i - \theta_j$, we obtain that:

$$G_2 = \begin{pmatrix} \cos\gamma_{12} & d\sin\gamma_{12} & 0 & 0 \\ \dfrac{-\sin\gamma_{12}}{l_{12}} & \dfrac{d\cos\gamma_{12}}{l_{12}} & 0 & 0 \\ 0 & 0 & \cos\gamma_{13} & d\sin\gamma_{13} \\ -\cos\psi_{23} & 0 & \cos\gamma_{23} & d\sin\gamma_{23} \end{pmatrix} \tag{8}$$

$$F_2 = \begin{pmatrix} -\cos\psi_{12} & 0 \\ \dfrac{\sin\psi_{12}}{l_{12}} & -1 \\ -\cos\psi_{13} & 0 \\ 0 & 0 \end{pmatrix} \tag{9}$$

$$p = \begin{pmatrix} k_1\left(l_{12}^d - l_{12}\right) \\ k_2\left(\psi_{12}^d - \psi_{12}\right) \\ k_3\left(l_{13}^d - l_{13}\right) \\ k_4\left(l_{23}^d - l_{23}\right) \end{pmatrix} \tag{10}$$

where $k = [k_1 \quad k_2 \quad k_3 \quad k_4]^T$ with k_1, k_2, k_3 and $k_4 > 0$.

The control input vector of robots 2 and robots 3 is defined as [2]:

$$u_{23} = [v_2 \quad \omega_2 \quad v_3 \quad \omega_3]^T = G_2^{-1}(p - F_2 u_1) \tag{11}$$

where $v_1, v_2, v_3\,(m/s)$ are linear velocities, $\omega_1, \omega_2, \omega_3\,(rad/s)$ are the angular velocity of robot 1, robot 2 and robot 3, respectively.

2.3 Limit-Cycle algorithm to avoid obstacles

Motivated by [3] , we set up the steps of LC method as follows:

Step 1: Draw a line *l* from the leading robot to the target in a global Descartes coordinate as follows:

$$ax + by + c = 0$$

Step 2: Treat variable obstacles as disturbing obstacle O_d's if the line *l* crosses them, else treat them as non-disturbing obstacles and $O_n^{'}s$.

Step 3: Move towards the target if there is no O_d.

Step 4: calculate the distance d from the center of the nearest disturbing obstacle, O_d to the line *l*, using:

$$d = \frac{aQ_x + bQ_y + c}{\sqrt{a^2 + b^2}} \tag{12}$$

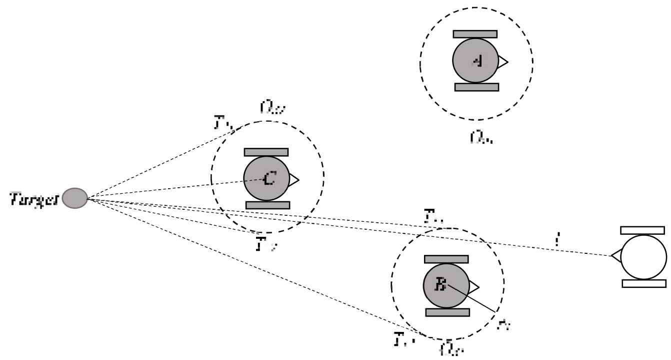

Figure 5. Navigation using the limit-cycle method

Then, the desired direction of the robot at each position is calculated by using the following equations:

$$\begin{cases} \dot{x}_r = \dfrac{d}{|d|} y + x\left(r^2{}_v - x^2 - y^2\right) \\ \dot{y}_r = -\dfrac{d}{|d|} y + x\left(r^2{}_v - x^2 - y^2\right) \end{cases} \tag{13}$$

$$x = R_x - G_x ; y = R_y - G_y \tag{14}$$

$$r_v = r_r + r_o + \delta \tag{15}$$

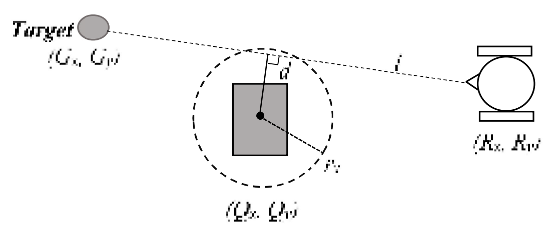

Figure 6. Describe parameters while robots avoid obstacle

where $\left(Q_x, Q_y\right), \left(G_x, G_y\right), \left(R_x, R_y\right)$ are the central positions of the obstacle, the target and the robots. r_v is calculated by the total radius of robot r_r , the radius of obstacle r_o and safety distance δ .

Remark 1: if d is positive, the robot avoids obstacles in a clockwise direction, otherwise, d is negative, and the robot avoids obstacles in the opposite direction to the clockwise.

Based on formula (13) and the laws of kinetic control [3], the desired angle and velocity is written as:

$$\begin{cases} \theta_r = \arctan\left(\dfrac{\dot{y}_r}{\dot{x}_r}\right) \\ v_r = \sqrt{\dot{x}_r^{\,2} + \dot{y}_r^{\,2}} \end{cases} \tag{16}$$

To calculate the angular velocity of Limit Cycle trajectory over time, the following formula is used:

$$\omega_r = \frac{d\theta_r}{dt} \tag{17}$$

3 SIMULATION RESULTS

In the section, we conduct a simulation to verify the stability of the system. If the simulation result are met the desired requirements, we will implement a real experiment.

We run the simulation with these following values: $r_v = 1\ (m)$, $l_{12}^d = l_{13}^d = l_{23}^d = 0.2\ (m)$ and $\Psi_{12}^d = 260^o$. If $d > 0$, the robot avoids obstacle in a clockwise direction as shown in Fig. 7.

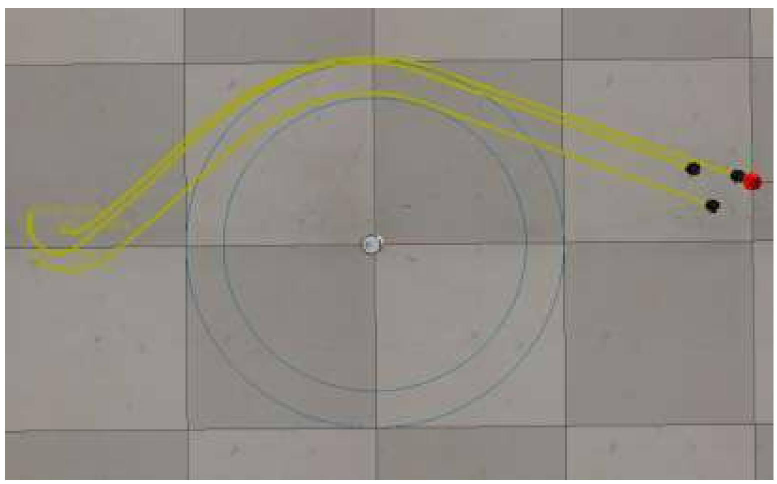

Figure 7. Obstacle avoidance trajectory of the three-robot formation (d > 0)

From the chart of $x - y$ coordinate on Fig. 8, it can be seen that the leading robot tracks the reference trajectory quite exactly and avoids obstacles with the radius $r_v = 1\ (m)$. It is also shown on Fig. 9 that, the errors e_1, e_2 gradually reduce to zero in a short time, so we can conclude that the leading robot tracks to the reference trajectory.

Fig. 10 shows that the algorithm works efficiency to keep the distance among 3 robots at the desired distance. Fig. 10 (a) compares measuring distance (red line) to the set distance (blue line) between robot 1 and robot 2 with the desired distance $l_{12}^d = 0.2\ (m)$. Fig 10 (b) compares measuring distance (red line) to the set distance (blue line) between robot 1 and robot 3 with the desired distance $l_{13}^d = 0.2\ (m)$. Fig 10 (c) compares measuring distance (red line) to the set distance (blue line) between robot 2 and robot 3 with the desired distance $l_{23}^d = 0.2\ (m)$. Fig 10 (d) compares measuring angle (red line) to the set angle (blue line) between robot 1 and robot 2 with the desired angle $\psi_{12}^d = 260^0$. The results on Fig. 10 show us the formation of three robots are kept at the desired distance and angle from the beginning of the l_{12} simulation to the end. So we can conclude the control theory is working well in the simulation environment.

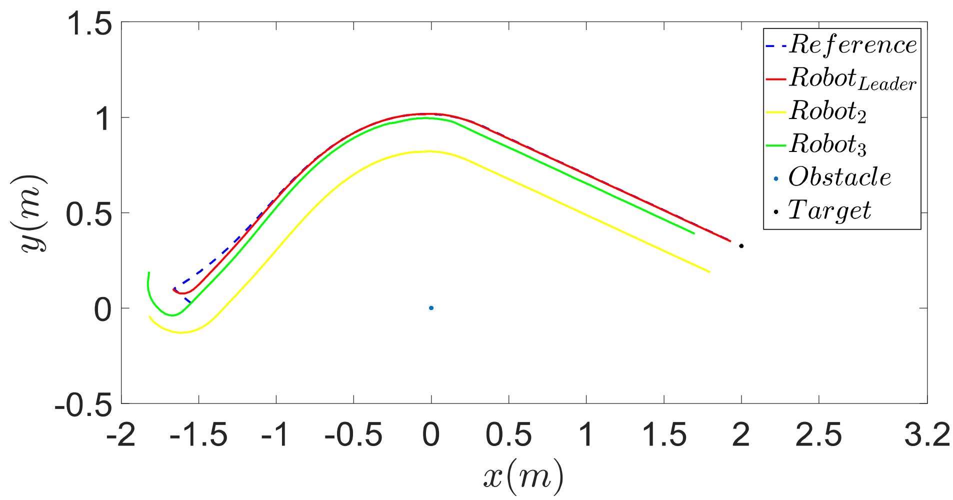

Figure 8. x-y trajectory of three-robot formation

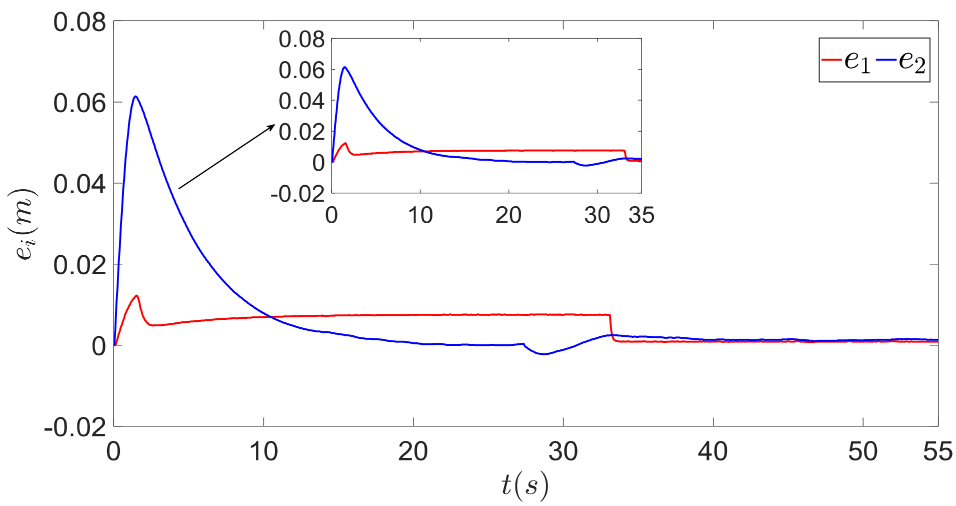

Figure 9. Position error of leading robot compared to the reference trajectory

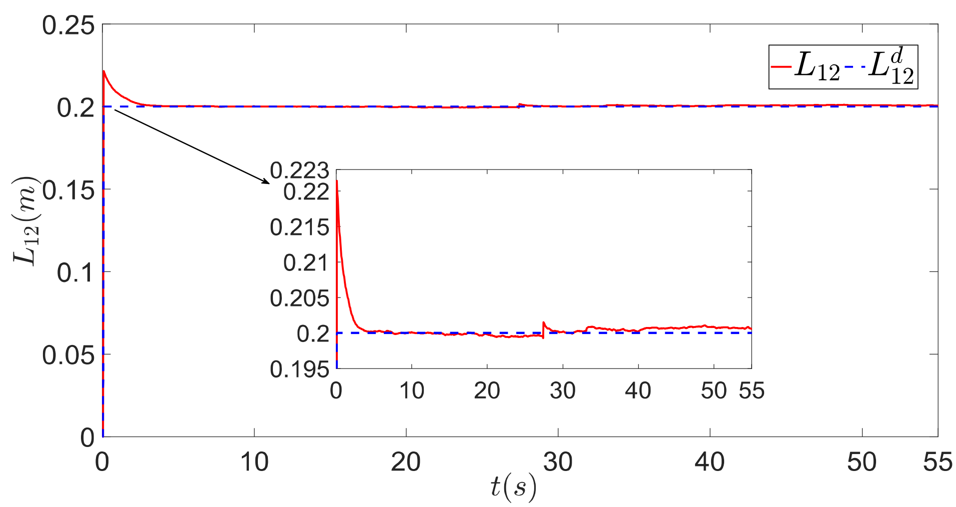

Figure 10.(a) Distance between the leading robot and the robot 2 compared to the desired distance

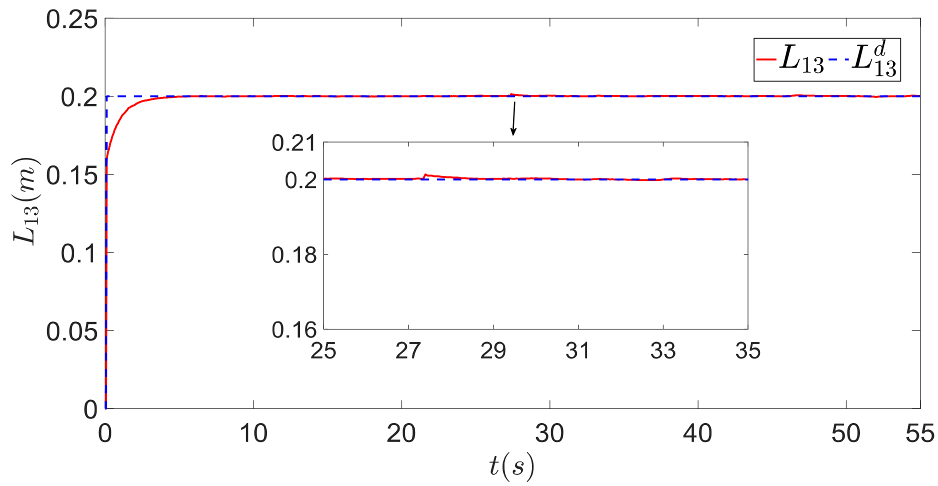

Figure 11.(b) Distance between the leading robot and the robot 3 compared to the desired distance

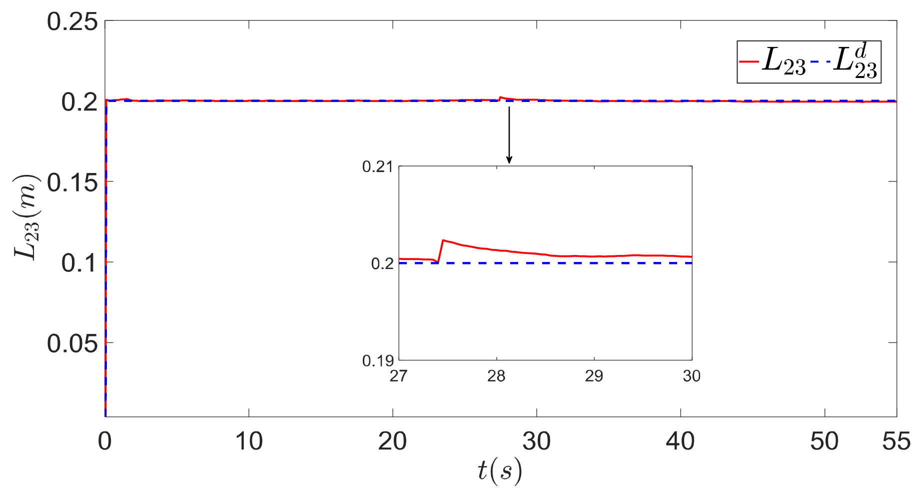

Figure 12.(c) Distance between the robot 2 and the robot 3 compared to the desired distance

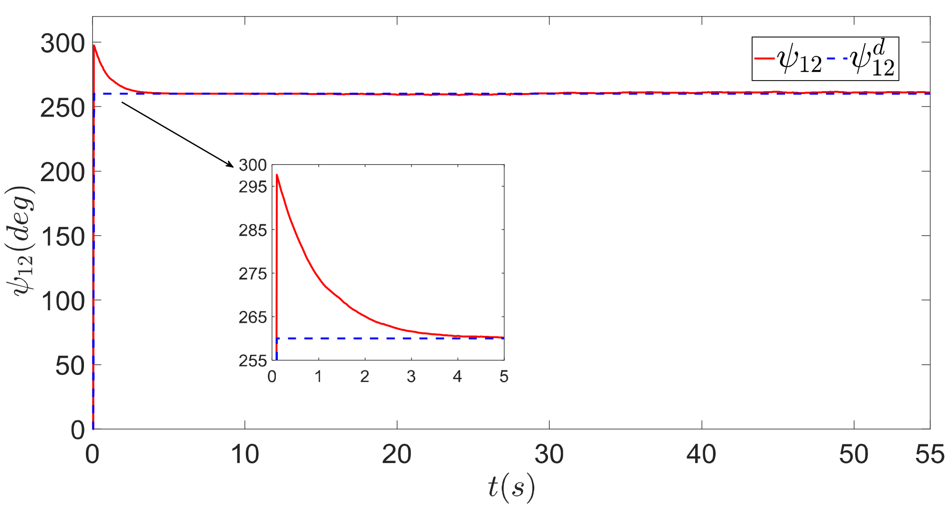

Figure 13.(d) Angle between the leading robot and the robot 2 compared to the desired distance

4 EXPERIMENTAL RESULT

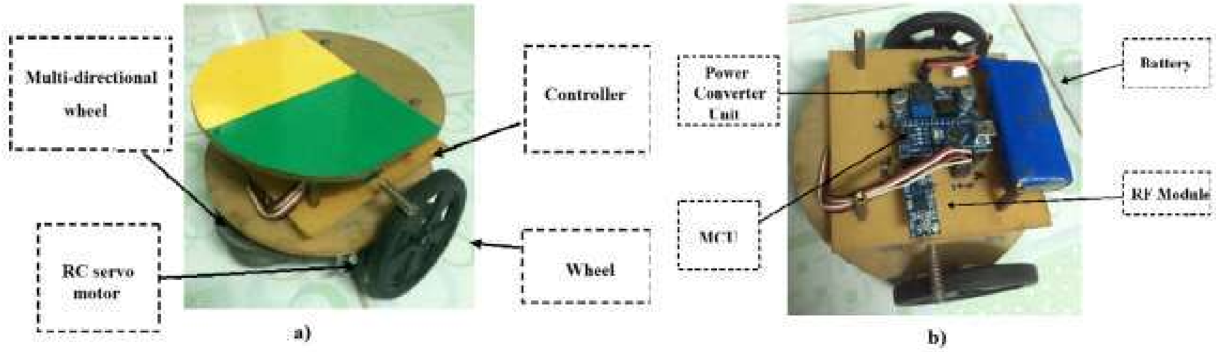

Figure 11. A real experimental model of nonholomonic mobile robot.

Now, the experimental is implemented. Firstly, we set up the hardware. The overall appearance of the robot is shown on Fig. 11(a). On the upper side of each robot is equipped with two different colors to distinguish it from other robot. Each pair of colors is different from others. The robot is controlled by two RC servo motors via two wheels. RC servo motor is include a driver and a feedback circuit to keep the desired speed of the motor. A multi-directional wheel is mounted in front of robot to balance the robot. Fig. 11(b) shows the circuit that receive the control signal and control the robot. The MCU gets data from RF module through UART communication and sends control value to RC servo motors. Then, a camera is connected to the computer, which is responsible for locating the position and direction of movement of the robot through image processing. The computer runs the control interface including image processing, control algorithms for the three-robot formation, and the computer is also connected to the RF transmitter module through COM port (RS232 standard). On each robot is also equipped with RF receiver module, which helps these robots receive control data from the PC, then the microcontroller on the robot outputs suitable control signal for the motor drive circuit to control two wheels of the robot. Through RF communication, the transmitter module receives data from the computer from COM port, then transmits data to the robots. Each frame data includes the left wheel and right wheel velocity (m/s) of each robot.

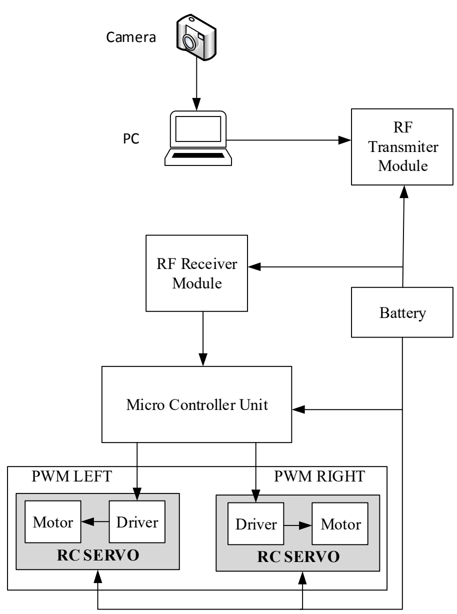

Figure 12. Schematic diagram of system

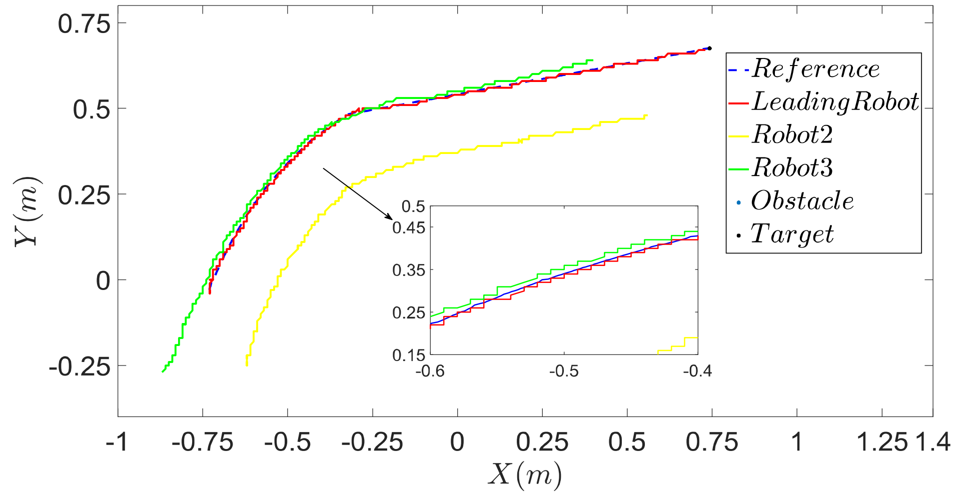

Figure 13. x - y trajectory of three-robot formation

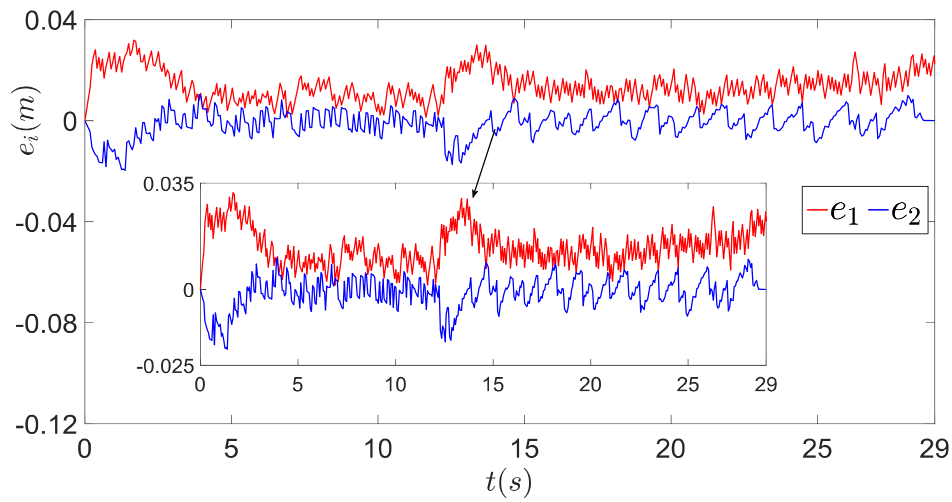

Figure 14. Position error of leading robot compared to the reference trajectory

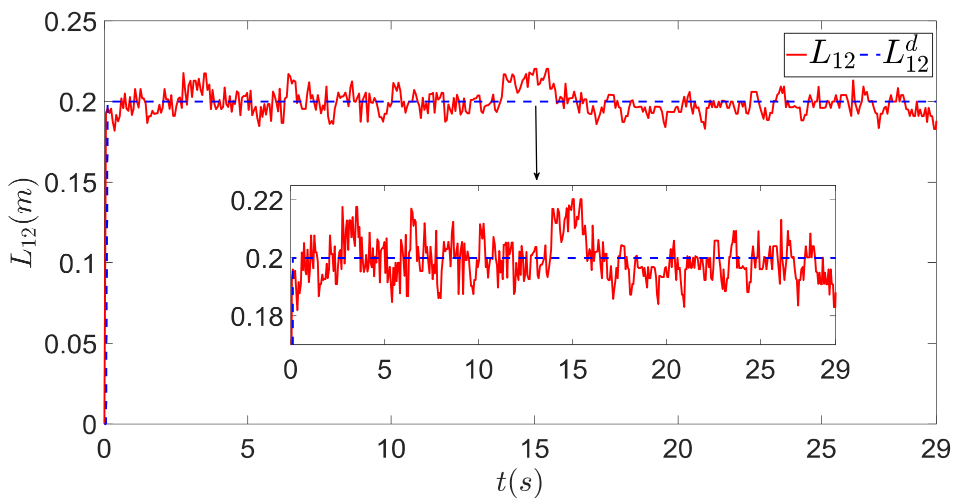

Figure 15.(a) Distance between the leading robot and robot 2 compared to the desired distance

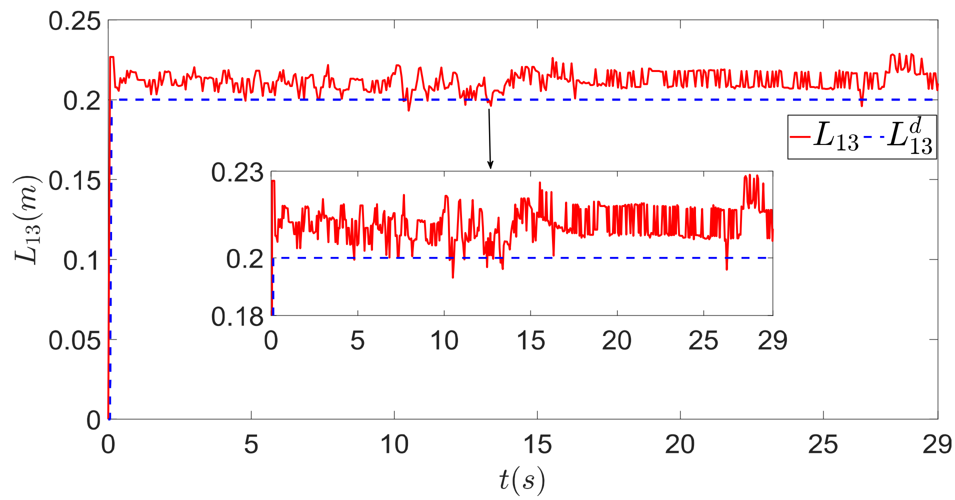

Figure 15.(b) Distance between the leading robot and robot 3 compared to the desired distance

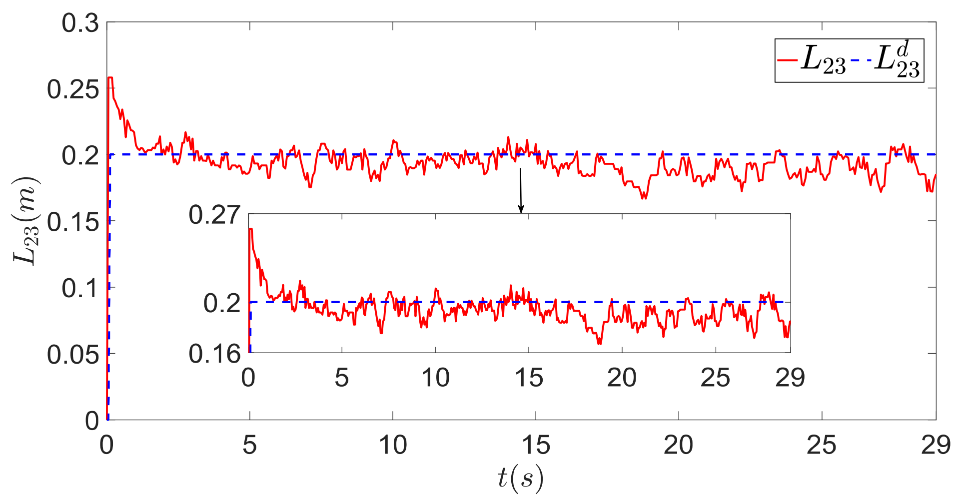

Figure 15.(c) Distance between the robot 2 and the robot 3 compared to the desired distance

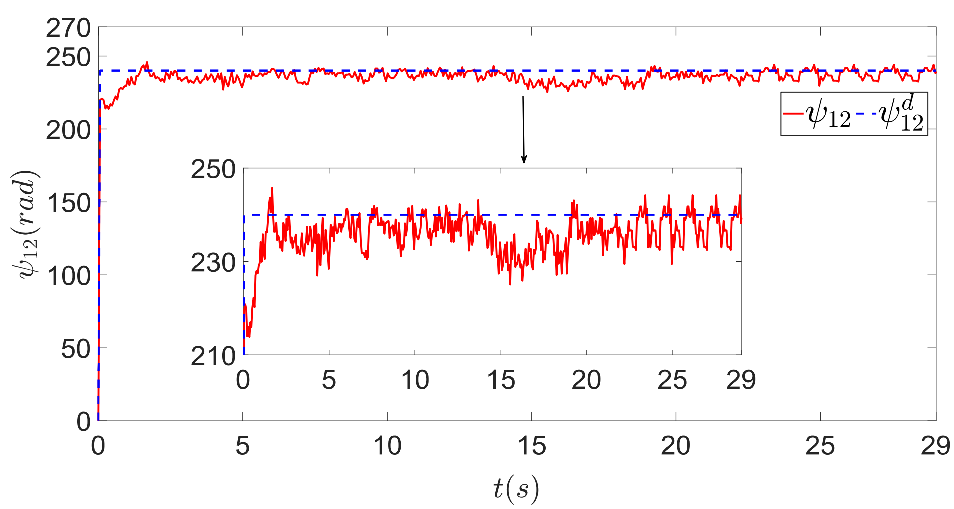

Figure 15. (d) Angle between the leading robot and the robot 2 compared to the desired angle

Figure 13 shows that the leading robot tracks the trajectory very well and the three-robot formation is guaranteed to maintain the desired distance and angle by the control algorithm while the formation moves to the target. Position and angular errors of leading robot (Fig. 14) gradually reduce to zero in a short time. Then we can conclude that the designed control algorithm performs well to keep leading robot tracking to the reference trajectory and maintain the formation. Fig. 15(a) shows distance between the leading robot and robot 2 compared to the desired distance. Fig. 15(b) shows distance between the leading robot and robot 3 compared to the desired distance. Fig. 15(c) shows distance between robot 2 and robot 3 compared to the desired distance. Fig. 15(d) shows angle between the leading robot and robot 2 compared to the desired angle. From the results, it can be seen that the followers are controlled to keep the distance and angle with leading robot as the desired value.

5 CONCLUSION

In this paper, backstepping kinematic control, limit-cycle and formation control of nonholomonic mobile robots are combined to design a control algorithm for a flexible formation to avoid obstacles and follow the leading robot. The simulation and experimental results show that the algorithm achieves the expected design purpose. The desired distance from each robot to the others in three-robot formation and the angle between leading robot and robot 2 are guaranteed. The future work will be focused on developing the algorithm for industrial robots.

6. ACKNOWLEDGEMENTS

This research is funded by Vietnam Ministry of Industry and Trade under grant number "DT KHCN-081/20".

7. REFERENCE

[1] R. Fierro and F. L. Lewis, "Control of a Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics," *Journal of Robotic Systems,* vol. 14, no. 3, pp. 149-227, March 1997.

[2] Aveek K. Das, Student Member, IEEE, Rafael Fierro, Member, IEEE, Vijay Kumar, Senior Member, IEEE, James P. Ostrowski, Member, IEEE, John Spletzer, and Camillo J. Taylor, Member, IEEE, "A Vision-Based Formation Control Framework," *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION,* Vols. 18, No. 5, October 2002.

[3] Dong-Han Kim, Jong-Hwan Kim, "A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer," *Robotics and Autonomous Systems,* vol. 42, no. 1, pp. 17-30, January 2003.

[4] Lounis Adouane, "Orbital Obstacle Avoidance Algorithm for Reliable and On-Line Mobile Robot Navigation," in *9th Conference on Autonomous Robot Systems and Competitions*, May 2009.

[5] Mehdi Mouad, Lounis Adouane, Djamel Khadraoui, Philippe Martinet, "Mobile Robot Navigation and Obstacles Avoidance based on Planning and Re-Planning Algorithm," *10th IFAC Symposium on Robot Control,* vol. 45, no. 22, pp. 622-628, 2012.

[6] Travis Dierks and S. Jagannathan, "Control of Nonholonomic Mobile Robot Formations: Backstepping Kinematics into Dynamics," *16th IEEE International Conference on Control Applications,* October 2007.

[7] Yue Li, Junyao Gao, Xuandong Su and Jingchao Zhao, "Cooperation control of multiple miniature robots in unknown obstacle environment," 24 October 2014.

[8] Phan Thành Long, Hồ Thanh Hoài, "Điều khiển đội hình Robot bầy đàn," Đại học Công Nghiệp Thành Phố Hồ Chí Minh, 2016.

[9] Yoram Koren, Johann Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," *IEEE International Conference on Robotics and Automation,* April 1991.

[10] Johann Borenstein, Yoram Koren, "Obstacle avoidance with ultrasonic sensors," *IEEE Journal on Robotics and Automation,* vol. 4, no. 2, April 1988.

[11] J. H. Kim, D. H. Kim, Y. J. Kim, and K. T. Seow, Soccer Robotics, 2004.

[12] Adrian Kaehler and Gary Bradski, Learning OpenCV 3, December 2016.

ĐIỀU KHIỂN ĐỘI HÌNH PHÂN TÁN VÀ TRÁNH VẬT CẢN CỦA HỆ THỐNG NHIỀU ROBOT

Tóm tắt: Bài báo này đề xuất phương pháp điều khiển phân tán bầy đàn cho nhiều robot di động tránh vật cản. Thứ nhất, thuật toán Limit Cycle được khai thác để lập quỹ đạo tham chiếu cho robot tránh vật cản. Thứ hai, luật điều khiển bám quĩ đạo tham chiếu cho robot dẫn đầu được được giới thiệu. Thứ ba, thuật toán điều khiển robot di chuyển theo đội hình và tránh vật cản dựa trên sự kết hợp của thuật toán Limit Cycle và điều khiển bám quĩ đạo tham chiếu được thiết kế. Khác với thuật toán điều khiển phân tán trong các tài liệu liên quan, thuật toán trong bài báo này bảo đảm rằng không chỉ đội hình robot được duy trì mà còn tránh được vật cản khi di chuyển đến mục tiêu. Cuối cùng, kết quả mô phỏng và thực nghiệm được tiến hành để kiểm chứng tính hiệu quả của phương pháp đề xuất.

Từ khóa: Tránh vật cản, Điều khiển đội hình, Robot phi holonom, Kiến trúc điều khiển, Điều khiển bám

Ngày nhận bài: 12/09/2019

Ngày chấp nhận đăng: 03/01/2020