International Journal of Control, Automation, and Systems (2014) 12(5):1077-1089 ISSN:1598-6446 eISSN:2005-4092
DOI 10.1007/s12555-013-0021-0 http://www.springer.com/12555

Formation Control of Mobile Robots with Obstacle Avoidance based on GOACM using Onboard Sensors

Yanyan Dai and Suk Gyu Lee*

Abstract: This paper deals with the problem of formation control for nonholonomic mobile robots under a cluttered environment. When the obstacles are not detected, the follower robot calculates its waypoint to track, based on the leader robot's state. The proposed geometric obstacle avoidance control method (GOACM) guarantees that the robot avoids the static and dynamic obstacles using onboard sensors. Due to the difficulty for the robot to simultaneously get overall safe boundary of an obstacle in practice, a safe line, which is perpendicular to the obstacle surface, is used instead of the safe boundary. Since GOACM is executed to find a safe waypoint for the robot, GOACM can effectively cooperate with the formation control method. Moreover, the adaptive controllers guarantee that the trajectory and velocity tracking errors converge to zero with the consideration of the parametric uncertainties of both kinematic and dynamic models. Simulation and experiment results present that the robots effectively form and maintain formation avoiding the obstacles.

Keywords: GOACM, leader-follower formation control, limited on-board sensor, mobile robot.

1. INTRODUCTION

Formation control problem has been regarded as an important problem in the multi-robot system, in order to accomplish cooperative tasks [1]. The formation control approaches have been proposed as: behavior-based approach [2], virtual structure approach [3], and leader-follower approach [4]. Due to its easiness to implement, this paper focuses on leader-follower approach.

In static environment, artificial potential field algorithm [5,6] is one of the well-known methods for obstacle avoidance, because of its mathematical simplicity. The robot is attracted by the goal and repulsed away by the obstacles [5]. However, besides the stability problem, this method also suffers from a local minimal problem. Mixed-integer linear programming [7] is another powerful tool for planning a safe path, because of its modeling capability and the availability of good solver. A major disadvantage of MILP lies in its computational complexity, due to the NP-hard problem. Moreover, every obstacle avoidance trajectory has to be calculated in advance. In [8], one geometric obstacle avoidance method is proposed that the leader robot influences the obstacle avoidance behavior and guides the follower robot to a free area. If the leader robot cannot detect the obstacles but the follower robot has to avoid the obstacles, the obstacle avoidance motion will fail, because follower robot cannot navigate by itself.

In real-world, it is necessary to guarantee that multiple robots effectively avoid moving obstacles. To solve this problem, [9] proposes an approach for mobile robots, where the behavior of dynamic obstacle is known as a priori. A dynamic artificial potential field method is presented in [10]. The attractive potential is defined as a function of the relative position and velocity between the robot and the target. The repulsive potential is defined as the relative position and velocity between the robot and moving obstacles. The performance of this approach is not effective, due to the major differences in collision detection and the necessity to control the speed of the robot [11]. An improved artificial potential field method, by using an Anisotropic-Function, is discussed in [12]. However, the moving obstacle velocity is required to pre-set. [13] presents a dynamic obstacle avoidance method. The robot may not avoid the collision, if the velocity and the acceleration of the obstacle in the direction are higher than those of the robot.

In this paper, a simple but effective GOACM is proposed for the robot. GOACM is executed to help the robot find a new safe waypoint based on limited on-board sensor information, which includes the distance between the robot and the obstacle, the angle from x-axis to the measured range. With the consideration of the robot velocity constraints, a safe line, which is perpendicular to the obstacle surface, is used to calculate the waypoint for the robot instead of the safe boundary. In the dynamic environment, GOACM relies on the velocities of the robot and the obstacles. Since both GOACM and formation control method aim to find the waypoint for a robot, GOACM can effectively cooperate with the formation control method.

Manuscript received January 15, 2013; revised October 25, 2013; accepted April 8, 2014. Recommended by Associate Editor Yingmin Li under the direction of Editor Hyouk Ryeol Choi.

This research was supported by the 2012 Yeungnam University Research Grant.

Yanyan Dai and Suk Gyu Lee are with the Department of Electrical Engineering, Yeungnam University, Dae-hak Ro, Gyeongsan, Gyeongsangbukdo 712-749, Korea (e-mails: guangyanyan1129@hotmail.com, sglee@ynu.ac.kr).

* Corresponding author.

The trajectory tracking problem is widely solved using the kinematic model of a mobile robot [14], where 'perfect velocity' tracking is to generate the actual velocity controller. Based on [15], since it is difficult for the dynamics of the robot to produce the perfect velocity as the kinematic controller, the torque controller is used. In this paper, an adaptive tracking control algorithm is presented with the integration of an adaptive kinematic controller and a torque controller, based on [14,16].

2. PRELIMINARY FORMULATIONS

2.1. A nonholonomic mobile robot model

In Fig. 1, the robot R_i consists of a passive wheel and two actuated wheels to achieve the motion and orientation. The radius of both of actuated wheels is r_i. The distance between two actuated wheels is denoted by $2b_i$. The mass center of the mobile robot is located at O_i. M_i is located in the middle point between the right and left driving wheels. The distance between O_i and M_i is denoted by d_i. $\{o, x, y\}$ is an inertial Cartesian and $\{O_i, X, Y\}$ is the local frame fixed on the robot. The configuration of robot R_i can be $q_i = [x_i, y_i, \theta_i]^T$ in an inertial Cartesian frame. Define v_i and ω_i as the linear and angular velocities of the robot. The ordinary form of the robot R_i kinematic model is as

$$\dot{q}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i & 0 \\ \sin\theta_i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}. \tag{1}$$

Define $\nu_i = (\nu_{i1}, \nu_{i2})^T$ as the angular velocities of the right and left wheels of the robot R_i. The relationship of v_i, ω_i and ν_{i1}, ν_{i2} is as follows:

$$\nu_i = \begin{bmatrix} \nu_{i1} \\ \nu_{i2} \end{bmatrix} = \frac{1}{r_i} \begin{bmatrix} 1 & b_i \\ 1 & -b_i \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}. \tag{2}$$

The robot's trajectory is constrained to the horizontal plane. The wheels roll without slip. Define φ_{ir} and φ_{il} as the angles of the right and left driving wheels of robot R_i. The nonholonomic constraints can be as

$$\dot{y}_i \cos\theta_i - \dot{x}_i \sin\theta_i = 0, \tag{3}$$

$$\dot{x}_i \cos\theta_i + \dot{y}_i \sin\theta_i + b_i\dot{\theta}_i = r_i\dot{\varphi}_{ir}, \tag{4}$$

$$\dot{x}_i \cos\theta_i + \dot{y}_i \sin\theta_i - b_i\dot{\theta}_i = r_i\dot{\varphi}_{il}. \tag{5}$$

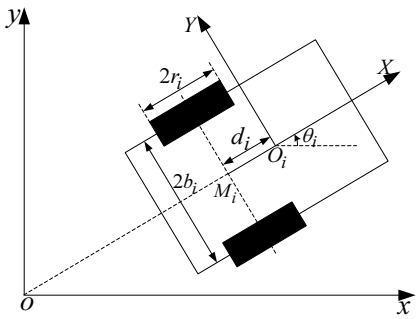


Fig. 1. Two-wheels nonholonomic mobile robot model.

The motion of the mobile robot R_i is given by

$$\dot{q}_i = S(q_i)v_i(t), \tag{6}$$

$$\bar{M}(q_i)\dot{v}_i + \bar{V}(q_i, \dot{q}_i)v_i + \bar{\tau}_{id} = \bar{B}\tau_i, \tag{7}$$

where

$$S(q_i) = \frac{r_i}{2}\begin{bmatrix} \cos\theta_i & \cos\theta_i \\ \sin\theta_i & \sin\theta_i \\ 1/b_i & -1/b_i \end{bmatrix},$$

$$\bar{M} = \left(\frac{r_i}{2b_i}\right)^2 \begin{bmatrix} (m_i b_i^2 + I_i) + I_{iw} & (m_i b_i^2 - I_i) \\ (m_i b_i^2 - I_i) & (m_i b_i^2 + I_i) + I_{iw} \end{bmatrix},$$

$$\bar{V} = \begin{bmatrix} 0 & \frac{r_i^2}{2b_i} m_{ic} d_i \dot{\theta}_i \\ -\frac{r_i^2}{2b_i} m_{ic} d_i \dot{\theta}_i & 0 \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$m_i = m_{ic} + 2m_{iw}, \quad I_i = m_{ic} d_i^2 + 2m_{iw} b_i^2 + I_{ic} + 2I_{iw},$$

where $\tau_i = (\tau_{i1}, \tau_{i2})^T$ is the torque applied on the right and left wheels of robot R_i. m_{ic} and m_{iw} are the mass of the body and wheel with a motor. I_{ic}, I_{iw}, and I_{im} are the moment of inertia of the body about the vertical axis through O_i, the wheel with a motor about the wheel axis, and the wheel with a motor about the wheel diameter, respectively. $\bar{\tau}_{id}$ is the bounded unknown disturbances of robot R_i.

Property 1: $\bar{M}(q_i)$ is symmetric and positive definite.

Property 2: $(\dot{\bar{M}} - 2\bar{V})$ is skew symmetric.

Assumption 1: The bounded disturbances $\bar{\tau}_{id}$ satisfies $\|\bar{\tau}_{id}\| \le c_{i0} + c_{i1}\|v_i\|$, with positive constants c_{i0} and c_{i1} [16].

2.2. The formation control model

Let R_{i-1} and R_i be the leader and the follower robot, respectively. l_{i0} is the desired distance between R_{i-1} and R_i. β_{i0} is the desired bearing angle from the orientation of the follower robot to the axis connecting R_{i-1} and R_i. The formation control model is as Fig. 2, where $\theta_i(t) = \theta_{i-1}(t)$. The waypoint $q_{iw} = (x_{iw}, y_{iw}, \theta_{iw})^T$ of follower robot R_i is denoted as

$$\begin{bmatrix} x_{iw}(t) \\ y_{iw}(t) \\ \theta_{iw}(t) \end{bmatrix} = \begin{bmatrix} x_{i-1}(t) - l_{i0}\cos(\beta_{i0} + \theta_i(t)) \\ y_{i-1}(t) - l_{i0}\sin(\beta_{i0} + \theta_i(t)) \\ \theta_{i-1}(t) \end{bmatrix}. \tag{8}$$

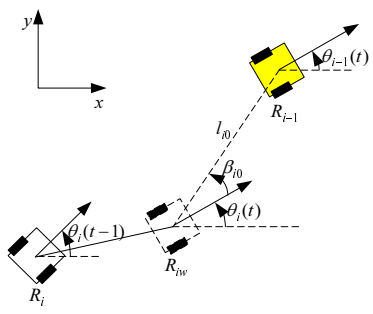


Fig. 2. The leader-follower formation control model.

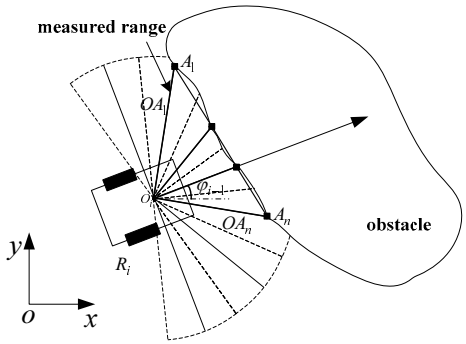


Fig. 3. The measured range finder model.

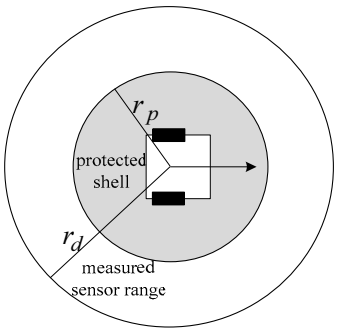


Fig. 4. The protected shell, and measured sensor range.

2.3. A measured range finder model

Due to good cost performance, the range finders are used to on-line cooperative obstacle detection and localization. The measured range data is noiseless without the presence of outlier. The measured range finder model is as Fig. 3. There are a set of intersections, based on the measured line and the surface of the obstacle. The line connecting the intersections is defined as the approximate surface of the obstacle for GOACM. The angle from each measured range to the orientation of the robot is assumed to be known, so the angle $\varphi_i (i = 1, 2, ..., n)$ from the measured range to x-axis can be calculated. A set of intersections $A_i (i = 1, 2, ..., n)$, on the surface of the obstacle, are numbered in a clockwise direction. The line connecting the mass of the robot and the intersection is denoted as OA_i, and the length of it is r_{OA_i}. GOACM is executed by the first intersection A_1 or the last intersection A_n.

3. OBSTACLE AVOIDANCE CONTROL

Assume two pseudo-shells surrounding each robot as shown in Fig. 4, where the biggest circle, and the solid circle filled in grey color describe the measured sensor range (radius = r_d), the protected shell (radius = r_p), respectively. Once the obstacles are inside the protected shell, the robot is to avoid obstacles.

3.1. Obstacle avoidance control

3.1.1 Static obstacle avoidance

In a static environment, when the measured distance is smaller than the radius of the protected shell, the robot considers to avoid an obstacle using GOACM as Fig. 5. Based on Section 2.3, GOACM is executed by the first intersection A_1 or the last intersection A_n. In order to avoid the static obstacle with less angular motion, the A_1 or A_n selection for GOACM relies on the smaller angle from current orientation of the robot to OA_1 or OA_n. In Fig. 5, the robot chooses point A_1. OA_2 is the nearest measured line to OA_1. Since it is difficult for the robot to simultaneously obtain overall boundary of the obstacle in practice, a safe line A_1S_1 is considered for the robot, which is perpendicular to A_1A_2 at A_1. The safe line is the minimal distance between the robot and the obstacle in the next time step. The goal of the obstacle avoidance is to guarantee that the obstacle is outside the protected

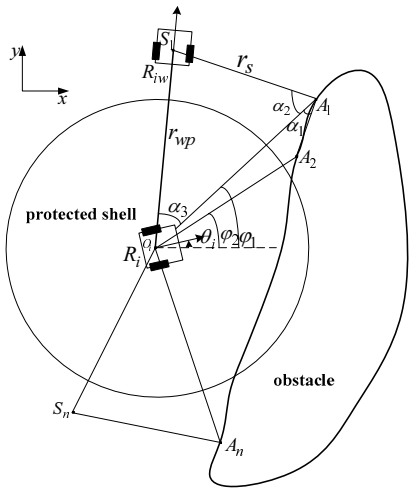


Fig. 5. Static obstacle avoidance for a robot using GOACM.

shell. Therefore, safe line's length r_s should be the same as or longer than the protected shell's radius. In the following discussion, define $r_s = r_p$.

Based on the safe line, the robot calculates its waypoint S_1 to avoid obstacle. φ_1 and φ_2 denote the angles from x-axis to OA_1 and OA_2, respectively. The distance of A_1A_2 is as

$$r_{A_1A_2} = \sqrt{r_{OA_1}^2 + r_{OA_2}^2 - 2r_{OA_1} r_{OA_2} \cos(\varphi_1 - \varphi_2)}. \quad (9)$$

The angle of α_1 are given by

$$\alpha_1 = \arccos\left(\frac{r_{OA_1}^2 + r_{A_1A_2}^2 - r_{OA_2}^2}{2 r_{OA_1} r_{A_1A_2}} \right). \quad (10)$$

In Fig. 6, if the angle α_1 is $\pi/2$, the safe waypoint S_1 will be on OA_1. If the direction of the robot is the same as direction of OA_1, and the length of OA_1 is r_s, S_1 will be located at O_i. The dead-lock problem is caused. To solve the problem, a small angle ε is used, and the angle α_2 is calculated in terms of α_1 and ε.

$$\alpha_2 = \begin{cases} \pi/2 - \alpha_1 & \alpha_1 \neq \pi/2 \\ \pi/2 - \alpha_1 + \varepsilon & \alpha_1 = \pi/2. \end{cases} \quad (11)$$

The distance connecting S_1 and O_i can be as

$$r_{wp} = \sqrt{r_{OA_1}^2 + {r_s}^2 - 2 r_{OA_1} \cdot r_s \cos\alpha_2}. \quad (12)$$

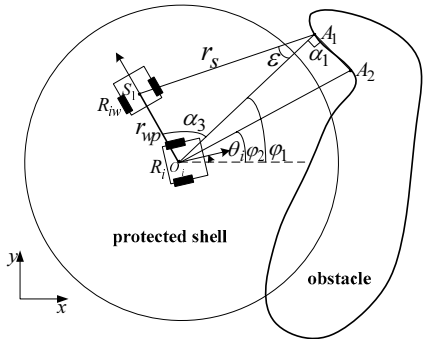


Fig. 6. Special case of GOACM: $\alpha_1 = \pi/2$.

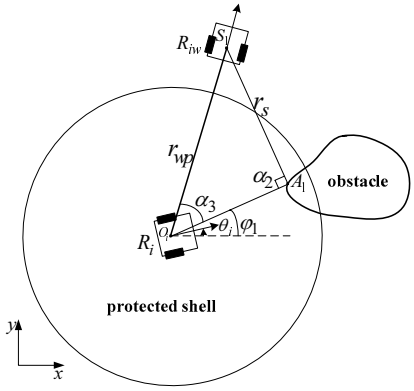


Fig. 7. Special case of GOACM: $\alpha_1 = 0$.

The difference of the orientation between OA_1 and OS_1 is defined as α_3 in

$$\alpha_3 = \arccos((r_{OA_1} - r_s \cos\alpha_2)/r_{wp}). \quad (13)$$

The waypoint q_{iw} to avoid an obstacle is as

$$\begin{bmatrix} x_{iw}(t) \\ y_{iw}(t) \\ \theta_{iw}(t) \end{bmatrix} = \begin{bmatrix} x_i(t-1) + r_{wp}\cos(\varphi_1 \pm \alpha_3) \\ y_i(t-1) + r_{wp}\sin(\varphi_1 \pm \alpha_3) \\ \varphi_1 \pm \alpha_3 \end{bmatrix}, \quad (14)$$

where r_{wp} is the distance to move for the next step, and $\varphi_1 \pm \alpha_3$ is the turning angle in the next step. Based on the orientation of the robot R_i, note that when the waypoint is on the left-side of the robot, select $\varphi_1 + \alpha_3$ in (14), otherwise, select $\varphi_1 - \alpha_3$. A big value of α_3 may cause a big value of $\varphi_1 \pm \alpha_3$. Based on [17], the robot effectively reaches its waypoint using the geometrical waypoint in cone method.

In Fig. 7, one measured line exists inside the measured sensor range, so $\alpha_1 = 0$. The safe line is assigned perpendicular to the measured line, so $\alpha_2 = \pi/2$. The safe waypoint is calculated in terms of (12)-(14).

3.1.2 Dynamic obstacle avoidance

The GOACM in a dynamic environment is shown in Fig. 8. For simplicity, the obstacle is assumed to be circle shape with known of the radius. Based on the measured range finder model and the radius of the obstacle, robot knows the obstacle location B. Define P_r and P_o as the position vectors of the robot and the obstacle, respectively. And P_{ro} is denoted as the relative position, $P_{ro} = P_o - P_r$. The distance between the robot and the obstacle is $r_{OB} = \|P_{ro}\|$. If the robot and the obstacle velocities are $\dot{P}_r$ and $\dot{P}_o$, the relative velocity of the robot to the obstacle is $\dot{P}_{or} = \dot{P}_r - \dot{P}_o$. Based on the smaller angle from the relative velocity of the robot and the obstacle to OA_1 or OA_n, the intersection point A_1 or A_n is chosen for the robot to apply GOACM. In Fig. 8, A_n is used for GOACM. The angle γ from x-axis to the line connecting O_i and B is

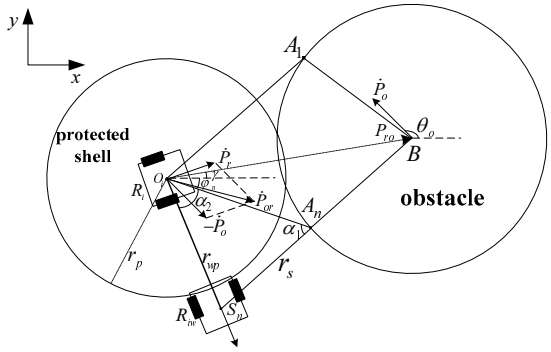


Fig. 8. Dynamic obstacle avoidance for the robot using GOACM.

$$\gamma = \text{a tan 2}((y_o - y_r),(x_o - x_r)). \quad (15)$$

φ_n denotes the angle from x-axis to OA_n. The distance of A_nB is calculated as

$$r_{A_nB} = \sqrt{r_{OB}^2 + r_{OA_n}^2 - 2r_{OB}r_{OA_n}\cos(\gamma - \varphi_n)}. \quad (16)$$

The direction of the safe line A_nS_n is defined the same as BA_n. α_1 can be calculated as

$$\alpha_1 = \pi - \arccos((r_{OA_n} - r_{OB}\cos(\gamma - \varphi_n))/r_{A_nB}). \quad (17)$$

The distance of OS_n can be calculated as

$$r_{wp} = \sqrt{{r_{OA_n}}^2 + {r_s}^2 - 2r_{OA_n}\cdot r_s\cos\alpha_1}. \quad (18)$$

The angle α_2 between OA_n and OS_n is as

$$\alpha_2 = \arccos((r_{OA_n} - r_s\cos\alpha_1)/r_{wp}). \quad (19)$$

The waypoint q_{iw} for the robot is as

$$\begin{bmatrix} x_{iw}(t) \\ y_{iw}(t) \\ \theta_{iw}(t) \end{bmatrix} = \begin{bmatrix} x_i(t-1) + r_{wp}\cos(\varphi_n \pm \alpha_2) \\ y_i(t-1) + r_{wp}\sin(\varphi_n \pm \alpha_2) \\ \varphi_n \pm \alpha_2 \end{bmatrix}, \quad (20)$$

where r_{wp} is the distance to move for the next step, and $\varphi_n \pm \alpha_2$ is the turning angle in the next step. Based on the orientation of the robot R_i, note that when the waypoint is on the left-side of the robot, select $\varphi_n + \alpha_2$ in (20), otherwise, select $\varphi_n - \alpha_2$.

3.1.3 Multiple obstacles avoidance

When the robot is required to avoid multiple obstacles at the same time, there are multiple intersections on the

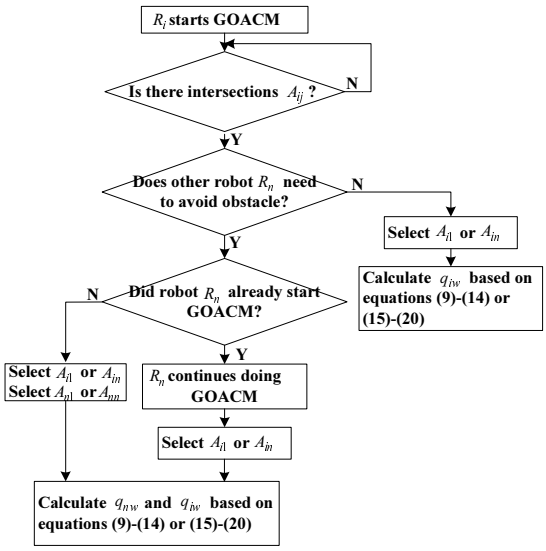


Fig. 9. Two robots avoid an obstacle with GOACM.

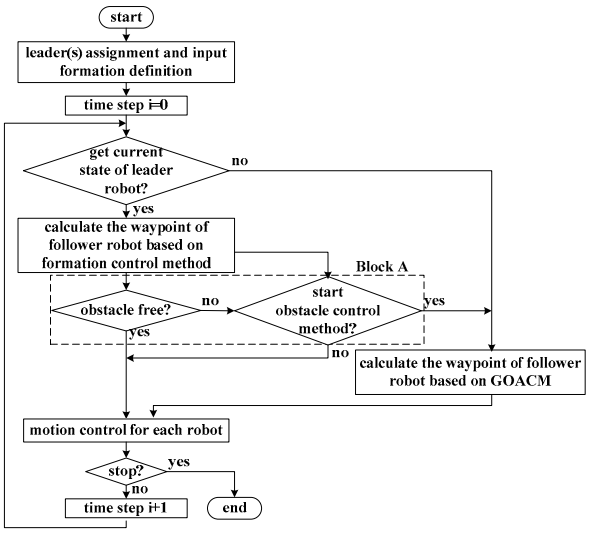


Fig. 10. The control flowchart combining the formation control method and GOACM.

surfaces of the obstacles. Define the intersection points as $A_1, A_2, ..., A_n (n \geq 2)$ in a clockwise direction. The robot chooses A_1 or A_n to apply GOACM, and the waypoint equation is as (14) or (20).

3.1.4 Multiple robots avoiding an obstacle

Another issue is that multiple robots avoid an obstacle. Multiple robots should choose proper intersections to apply GOACM, in order to avoid collision. Fig. 9 shows that two robots avoid an obstacle with GOACM. Define two robots as R_i and R_n, and the intersections as A_{ii} and A_{ni}, respectively. If R_i should start GOACM, there are three cases. Firstly, if robot R_n does not need to avoid an obstacle, robot R_i will select A_{i1} or A_{in}, and calculate q_{iw}. Secondly, if robot R_n already started GOACM, R_n will continue avoid an obstacle. Based on relative locations between R_i and R_n, R_i chooses proper intersection for GOACM. If R_i is on the left-side of R_n, robot R_i will choose A_{i1} to calculate q_{iw}; otherwise, robot R_i will select A_{in}. Thirdly, if robot R_n should also start obstacle avoidance, two robots will compare their locations. If R_i is on the left-side of R_n, A_{i1} and A_{nn} are selected for GOACM; otherwise, A_{in} and A_{n1} are selected.

3.2. Cooperation with the formation control method

Fig. 10 shows the robot control flowchart, which combines the formation control method and GOACM. According to block (A), the robot determines whether to start GOACM or not. If it is not necessary, the robot moves to the waypoint calculated by the formation control method. Otherwise, the robot calculates its safe waypoint based on GOACM and then moves to it.

Remark: The following two cases determine whether the robot starts GOACM or not as block (A) in Fig. 10.

Case 1: Define the angle ϕ_{wd}, which is from OA_i (used in GOACM) to the connection line between the robot and the waypoint (calculated by formation control method). If $|\phi_{wd}| > \pi/2$, the robot will not start GOACM.

Case 2: Regardless of the measured distance, if the robot cannot communicate with the leader robot, the follower robot has to start GOACM.

The minimum measured length between the robot and the obstacle is defined as r_m. The pseudo code of block (A) is as follows:

1. if ($r_m < r_d$)
2. { if ($r_m < r_p$)
3. {obstacle not free;
4. if ($|\phi_{wd}| < \pi/2$)
5. start GOACM;
6. else
7. do not start GOACM;
8. end}
9. else
10. obstacle free;
11. end}
12. else
13. obstacle free;
14. end

4. ROBUST ADAPTIVE CONTROLLERS

4.1. Adaptive control of the kinematic model

An adaptive tracking controller for the kinematic part is designed based on (6) and [14]. The waypoint model is

$$\dot{x}_{iw} = v_{iw} \cos\theta_{iw}, \tag{21}$$

$$\dot{y}_{iw} = v_{iw} \sin\theta_{iw}, \tag{22}$$

$$\dot{\theta}_{iw} = \omega_{iw}, \tag{23}$$

where v_{iw} and ω_{iw} are the waypoint input. Comparing the waypoint state q_{iw} with the current state q_{ic}, the tracking error posture can be described as

$$E_{iq} = \begin{bmatrix} x_{iE} \\ y_{iE} \\ \theta_{iE} \end{bmatrix} = \begin{bmatrix} \cos\theta_{ic} & \sin\theta_{ic} & 0 \\ -\sin\theta_{ic} & \cos\theta_{ic} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{iw} - x_{ic} \\ y_{iw} - y_{ic} \\ \theta_{iw} - \theta_{ic} \end{bmatrix}. \tag{24}$$

The derivative matrix $\dot{E}_{iq}$ can be derived as

$$\dot{E}_{iq}=\begin{bmatrix}\dot{x}_{iE}\\ \dot{y}_{iE}\\ \dot{\theta}_{iE}\end{bmatrix}=\begin{bmatrix}-1\\ 0\\ 0\end{bmatrix}v_{ic}+\begin{bmatrix}y_{iE}\\ -x_{iE}\\ -1\end{bmatrix}\omega_{ic}+\begin{bmatrix}v_{iw}\cos\theta_{iE}\\ v_{iw}\sin\theta_{iE}\\ \omega_{iw}\end{bmatrix}. \quad (25)$$

Based on [14], the linear and angular velocities v_{ic} and ω_{ic} are given as v_{if} and ω_{if}, respectively.

$$\begin{bmatrix}v_{if}\\ \omega_{if}\end{bmatrix}=\begin{bmatrix}v_{iw}\cos\theta_{iE}+K_{i1}x_{iE}\\ \omega_{iw}+K_{i2}v_{iw}y_{iE}+K_{i3}\sin\theta_{iE}\end{bmatrix}, \quad (26)$$

where K_{i1}, K_{i2}, and K_{i3} are positive constants. Using (2), (25) is

$$\dot{E}_{iq}=\begin{bmatrix}-\frac{r_i}{2}+\frac{r_i}{2b_i}y_{iE}\\ -\frac{r_i}{2b_i}x_{iE}\\ -\frac{r_i}{2b_i}\end{bmatrix}v_{i1}+\begin{bmatrix}-\frac{r_i}{2}-\frac{r_i}{2b_i}y_{iE}\\ \frac{r_i}{2b_i}x_{iE}\\ \frac{r_i}{2b_i}\end{bmatrix}v_{i2}+\begin{bmatrix}v_{iw}\cos\theta_{iE}\\ v_{iw}\sin\theta_{iE}\\ \omega_{iw}\end{bmatrix}. \quad (27)$$

If the parameters r_i and b_i are not known, the angular velocities of the left and right wheels cannot be obtained by (2). Based on [16], set $a_{i1}=1/r_i$ and $a_{i2}=b_i/r_i$, input velocities are chosen as

$$\begin{aligned}v_{ic}&=\begin{bmatrix}\hat{a}_{i1}&\hat{a}_{i2}\\ \hat{a}_{i1}&-\hat{a}_{i2}\end{bmatrix}\begin{bmatrix}v_{if}\\ \omega_{if}\end{bmatrix}\\ &=\begin{bmatrix}a_{i1}+\tilde{a}_{i1}&a_{i2}+\tilde{a}_{i2}\\ a_{i1}+\tilde{a}_{i1}&-a_{i2}-\tilde{a}_{i2}\end{bmatrix}\begin{bmatrix}v_{if}\\ \omega_{if}\end{bmatrix},\end{aligned} \quad (28)$$

where $\hat{a}_{i1}$ and $\hat{a}_{i2}$ are the estimate of a_{i1} and a_{i2}, and $\tilde{a}_{i1}=\hat{a}_{i1}-a_{i1}$, $\tilde{a}_{i2}=\hat{a}_{i2}-a_{i2}$. (27) can be as

$$\dot{E}_{iq}=\left(1+\frac{\tilde{a}_{i1}}{a_{i1}}\right)\begin{bmatrix}-v_{if}\\ 0\\ 0\end{bmatrix}+\left(1+\frac{\tilde{a}_{i2}}{a_{i2}}\right)\omega_{if}\begin{bmatrix}y_{iE}\\ -x_{iE}\\ -1\end{bmatrix}+\begin{bmatrix}v_{iw}\cos\theta_{iE}\\ v_{iw}\sin\theta_{iE}\\ \omega_{iw}\end{bmatrix}. \quad (29)$$

To design $\hat{a}_{i1}$ and $\hat{a}_{i2}$, based on [18], the Lyapunov function is chosen as (30), where $\dot{\tilde{a}}_{i1}=\dot{\hat{a}}_{i1}$, $\dot{\tilde{a}}_{i2}=\dot{\hat{a}}_{i2}$.

$$V_1=\frac{1}{2}\left(x_{iE}^2+y_{iE}^2\right)+\frac{\left(1-\cos\theta_{iE}\right)}{K_{i2}}+\frac{\tilde{a}_{i1}^2}{2\gamma_{i1}a_{i1}}+\frac{\tilde{a}_{i2}^2}{2\gamma_{i2}a_{i2}}, \quad (30)$$

γ_{i1} and γ_{i2} are positive constants. The differential of V_1 is

$$\begin{aligned}\dot{V}_1&=x_{iE}\dot{x}_{iE}+y_{iE}\dot{y}_{iE}+\frac{\dot{\theta}_{iE}\sin\theta_{iE}}{K_{i2}}+\frac{\tilde{a}_{i1}\dot{\tilde{a}}_{i1}}{\gamma_{i1}a_{i1}}+\frac{\tilde{a}_{i2}\dot{\tilde{a}}_{i2}}{\gamma_{i2}a_{i2}}\\ &=-K_{i1}x_{iE}^2-\frac{K_{i3}\left(\sin\theta_{iE}\right)^2}{K_{i2}}+\frac{\tilde{a}_{i1}}{\gamma_{i1}a_{i1}}\left(\dot{\hat{a}}_{i1}-\gamma_{i1}x_{iE}v_{if}\right)\\ &\quad+\frac{\tilde{a}_{i2}}{\gamma_{i2}a_{i2}}\left(\dot{\hat{a}}_{i2}-\gamma_{i2}\frac{\omega_{if}\sin\theta_{iE}}{K_{i2}}\right).\end{aligned} \quad (31)$$

The parameters in (31) are chosen as

$$\dot{\hat{a}}_{i1}=\gamma_{i1}x_{iE}v_{if},\quad \dot{\hat{a}}_{i2}=\gamma_{i2}\frac{\omega_{if}\sin\theta_{iE}}{K_{i2}}. \quad (32)$$

Equation (31) becomes

$$\dot{V}_1=-K_{i1}x_{iE}^2-\frac{K_{i3}\left(\sin\theta_{iE}\right)^2}{K_{i2}}\le 0. \quad (33)$$

As $t\to\infty$, E_{iq} is shown to be a stable equilibrium point. Using Barbalat's lemma, x_{iE} and θ_{iE} tend to zero as $t\to\infty$. Since $\tilde{a}_{i1}$ and $\tilde{a}_{i2}$ are bounded, $\dot{x}_{iE}$ and $\dot{\theta}_{iE}$ are bounded.

4.2. Adaptive control of the dynamic model

Define the velocity tracking errors of robot R_i as

$$E_{ic}=\nu_i-\nu_{ic}=\begin{bmatrix}\nu_{i1}-\nu_{i1c}\\ \nu_{i2}-\nu_{i2c}\end{bmatrix}. \quad (34)$$

Equation (7) can be rewritten as

$$\bar{M}(\dot{\nu}_{ic}+\dot{E}_{ic})+\bar{V}(\nu_{ic}+E_{ic})+\bar{\tau}_{id}=\bar{B}\tau_i. \quad (35)$$

The differential of E_{ic} multiply $\bar{M}$ can be described as

$$\begin{aligned}\bar{M}\dot{E}_{ic}&=-(\bar{M}\dot{\nu}_{ic}+\bar{V}\nu_{ic})-\bar{V}E_{ic}-\bar{\tau}_{id}+\bar{B}\tau_i\\ &=-Y_{ic}P_i-\bar{V}E_{ic}-\bar{\tau}_{id}+\bar{B}\tau_i,\end{aligned} \quad (36)$$

where

$$\bar{M}\dot{\nu}_{ic}+\bar{V}\nu_{ic}=Y_{ic}P_i,\ Y_{ic}=\begin{bmatrix}\dot{\nu}_{i1c}&\dot{\nu}_{i2c}&\dot{\theta}_i\nu_{i2c}\\ \dot{\nu}_{i2c}&\dot{\nu}_{i1c}&-\dot{\theta}_i\nu_{i1c}\end{bmatrix},\ \text{and}$$

$$P_i=\left[\frac{r_i^2}{4b_i^2}(m_ib_i^2+I_i)+I_{iw}\quad\times\frac{r_i^2}{4b_i^2}(m_ib_i^2-I_i)\quad\frac{r_i^2}{2b_i}m_{ic}d_i\right]^T.$$

Based on [16], the torque controller is as

$$\tau_i=\bar{B}^{-1}(-K_{id}E_{ic}+Y_{ic}\hat{P}_i-u_{is}), \quad (37)$$

where K_{id} is a diagonal positive-definite design matrix. $\hat{P}_i$ is the estimate of P_i, and $\tilde{P}_i=\hat{P}_i-P_i$. $u_{is}=A\,\mathrm{sgn}(E_{ic})$, where $\mathrm{sgn}(E_{ic})$ is a sign function, and A is a positive-definite controller gain. Define $A=\hat{c}_{i0}+\hat{c}_{i1}\|\nu_i\|$, $\hat{c}_{i0}$ and $\hat{c}_{i1}$ are the estimate of c_{i0} and c_{i1} in Assumption 1, and $\tilde{c}_{i0}=\hat{c}_{i0}-c_{i0}$, $\tilde{c}_{i1}=\hat{c}_{i1}-c_{i1}$. To design $\hat{c}_{i0}$, $\hat{c}_{i1}$ and $\hat{P}_i$, the Lyapunov function is chosen as (38).

$$V_2=V_1+\tfrac{1}{2}E_{ic}^T\bar{M}E_{ic}+\tfrac{1}{2}\tilde{P}_i^T\Gamma^{-1}\tilde{P}_i+\frac{\tilde{c}_{i0}^2}{2\gamma_{i3}}+\frac{\tilde{c}_{i1}^2}{2\gamma_{i4}}, \quad (38)$$

where γ_{i3} and γ_{i4} are positive constants. $\dot{V}_2$ is as

$$\begin{aligned}\dot{V}_2 &= \dot{V}_1 + E_{ic}^T \bar{M}\dot{E}_{ic} + \frac{E_{ic}^T \dot{\bar{M}} E_{ic}}{2} + \tilde{P}_i^T \Gamma^{-1} \dot{\tilde{P}}_i + \frac{\tilde{c}_{i0}\dot{\tilde{c}}_{i0}}{\gamma_{i3}} + \frac{\tilde{c}_{i1}\dot{\tilde{c}}_{i1}}{\gamma_{i4}} \\ &= \dot{V}_1 - E_{ic}^T K_{id} E_{ic} + \tilde{P}_i^T \Gamma^{-1}(\dot{\tilde{P}} + \Gamma Y_{ic}^T E_{ic}) \\ &\quad - E_{ic}^T \bar{\tau}_{id} - A E_{ic}^T \operatorname{sgn}(E_{ic}) + \frac{\tilde{c}_{i0}\dot{\tilde{c}}_{i0}}{\gamma_{i3}} + \frac{\tilde{c}_{i1}\dot{\tilde{c}}_{i1}}{\gamma_{i4}}. \end{aligned} \quad (39)$$

The parameters in (39) are selected as

$$\dot{\tilde{P}}_i = -\Gamma Y_{ic}^T E_{ic}, \quad \dot{\tilde{c}}_{i0} = \gamma_{i3}\|E_{ic}\|, \quad \dot{\tilde{c}}_{i1} = \gamma_{i4}\|v_i\|\|E_{ic}\|. \quad (40)$$

Therefore, (39) can be rewritten as

$$\begin{aligned}\dot{V}_2 &= \dot{V}_1 - E_{ic}^T K_{id} E_{ic} - E_{ic}^T \bar{\tau}_{id} - A E_{ic}^T \operatorname{sgn}(E_{ic}) \\ &\quad + \frac{\tilde{c}_{i0}\dot{\tilde{c}}_{i0}}{\alpha_{i0}} + \frac{\tilde{c}_{i1}\dot{\tilde{c}}_{i1}}{\alpha_{i1}} \\ &\le \dot{V}_1 - E_{ic}^T K_{id} E_{ic} - \left(A - \|\bar{\tau}_{id}\|\right)\|E_{ic}\| + \frac{\tilde{c}_{i0}\dot{\tilde{c}}_{i0}}{\alpha_{i0}} + \frac{\tilde{c}_{i1}\dot{\tilde{c}}_{i1}}{\alpha_{i1}} \\ &\le \dot{V}_1 - E_{ic}^T K_{id} E_{ic} \\ &\quad - \tilde{c}_{i0}\left(\|E_{ic}\| - \frac{\dot{\tilde{c}}_{i0}}{\alpha_{i0}}\right) - \tilde{c}_{i1}\left(\|v_i\|\|E_{ic}\| - \frac{\dot{\tilde{c}}_{i1}}{\alpha_{i1}}\right) \\ &= \dot{V}_1 - E_{ic}^T K_{id} E_{ic} \le 0. \end{aligned} \quad (41)$$

V_2 is non-increasing with time, so it is bounded. Using Barbalat's lemma, x_{iE}, θ_{iE} and E_{ic} tend to zero as $t \to \infty$. θ_{iE} and $\ddot{\theta}_{iE}$ are bounded. This implies $\dot{\theta}_{iE}$ is uniformly continuous. From Barbalat's lemma, $\dot{\theta}_{iE} \to 0$ as $t \to \infty$. Based on (25), (26) and (34), variable $\dot{\theta}_{iE}$ can be described by the equation $K_{iy} v_{iw} y_{iE} \to 0$. v_{iw} is a bounded function of time. These imply that y_{iE} has to converge to zero. $E_{iq} = 0$ is an asymptotically stable point as $t \to \infty$. The error $E = (E_{iq}^T \quad E_{ic}^T)^T = 0$ is an asymptotically stable equilibrium point.

5. SIMULATIONS

The physical parameters of the robot are defined as $b_i = 0.125$ m, $d_i = 0.15$ m, $r_i = 0.08$ m, $m_{ic} = 4$ kg, $m_{iw} = 1$ kg, $I_{ic} = 2.325$ kgm^2, $I_{iw} = 0.005$ kgm^2 and $I_{im} = 0.0025$ kgm^2. The parameters for the adaptive controllers are as $K_{i1} = 0.1$, $K_{i3} = 0.28$, $K_{id} = diag\{0.5, 0.5\}$, $\gamma_{i1} = \gamma_{i2} = 0.001$, $\gamma_{i3} = \gamma_{i4} = 0.01$ and $\Gamma = diag\{0.01, 0.01, 0.01\}$. In the simulations, the time step is 0.2s. Input noise is $|\tau_{id}| \le 3$. The velocity constraints of follower robot 1 are $v_{\max} = 1.1$ m/s, $\omega_{\max} = \pi/2$ rad/s. The velocity constraints of follower robot 2 are $v_{\max} = 0.7$ m/s, $\omega_{\max} = \pi/2$ rad/s.

5.1. Static obstacle avoidance

Fig. 11 depicts simulation results that three robots form a formation in sinusoid trajectories while avoiding static obstacles. The predefined leader robot's obstacle-free trajectory is sinusoidal. If there are no obstacles, the two follower robots and the leader robot should form and maintain formation with the desired distance $l_{10} = l_{20} = 1.2$ m, and the desired orientation $\beta_{10} = -\pi/3$ and $\beta_{20} = \pi/3$. $r_s = r_p = 0.35$ m is selected for follower robots 1 and 2. The initial states of the leader robot, follower robots 1 and 2 are as $(0,0,0)^T$, $(-1.5,1.2,0)^T$, $(-0.5,-2.2,\pi/6)^T$, respectively. The trajectory of the leader robot is as

$$x_0 = 0.38t, \quad y_0 = 1.5\sin(0.18t). \quad (42)$$

Follower robot 1 avoids a static rectangular obstacle and static multiple obstacles composing of two circles. Follower robot 2 avoids a rectangular obstacle and a circle obstacle. The result shows the follower robots can avoid any kind of obstacles using GOACM, in Fig. 11.

Figs. 12 and 13 show trajectory tracking errors of two follower robots while moving in the trajectories as Fig. 11. The state errors are small when the follower robots avoid the obstacles. Because the leader robot moves forward when the follower robots avoid the obstacles, there are oscillations after avoiding the obstacles, but the followers can quickly reform formation.

Figs. 14 and 15 show velocity tracking errors of two follower robots when tracking the trajectories in Fig. 11. Although oscillations occur, the velocity tracking errors approximately converge to zero.

For comparison with the result in Fig. 11, the artificial potential field method (APFM) is applied to avoid static obstacles for the follower robots. When avoiding the rectangular obstacle, the magnitude of the repulsive force

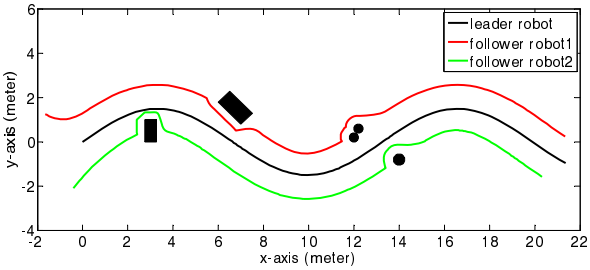


Fig. 11. The follower robots track the leader robot avoiding static obstacles using GOACM.

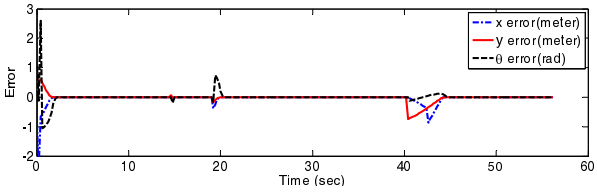


Fig. 12. Trajectory tracking errors of follower robot 1.

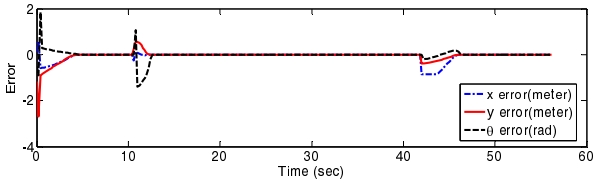


Fig. 13. Trajectory tracking errors of follower robot 2.

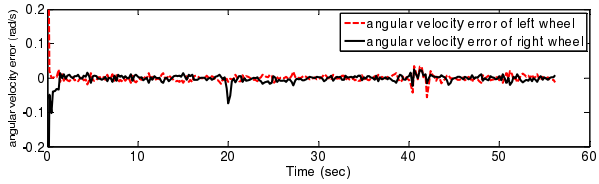


Fig. 14. Velocity tracking errors of follower robot 1.

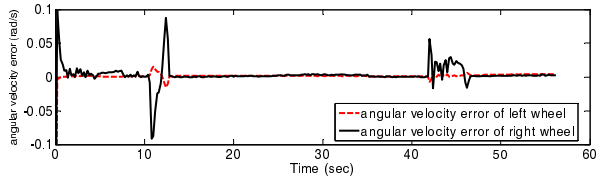


Fig. 15. Velocity tracking errors of follower robot 2.

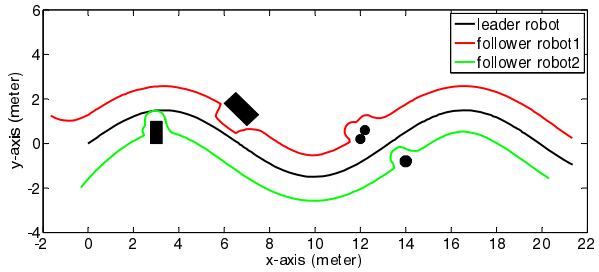


Fig. 16. The follower robots track the leader robot avoiding static obstacles using APFM.

Table 1. The average path length comparison between GOACM and APFM.

		path Length (meter)	
obstacle type		rectangular obstacle	multi- circular-obstacles
follower robot 1	GOACM	1.5758	1.0695
	APFM	3.9344	2.1900
obstacle type		rectangular obstacle	circular obstacle
follower robot 2	GOACM	2.3481	1.1942
	APFM	2.8993	2.6354

is calculated by taking the nearest point from the robot to the obstacle, while the direction is given from its center as [19]. The trajectories of the robots are described in Fig. 16, where each follower robot's path length is longer than in Fig. 11 when avoiding obstacles.

Table 1 shows the average path length of follower robots 1 and 2 avoiding the obstacles. The performance of GOACM is better than APFM. Using GOACM, the path length of follower robot 1 is reduced by 2.3586m and 1.1205m related to APFM, avoiding rectangular obstacle and multiple circular obstacles, respectively. For follower robot 2, compared with APFM, GOACM reduces the path length by 0.5512m avoiding the rectangular obstacle, and reduces the path length by 1.4412m, avoiding the circular obstacle.

Figs. 17-20 show the better performance with GOACM than with APFM. When avoiding the obstacles, relative distance and bearing angle errors between the leader robot and follower robot using GOACM are smaller than the errors using APFM. After avoiding the

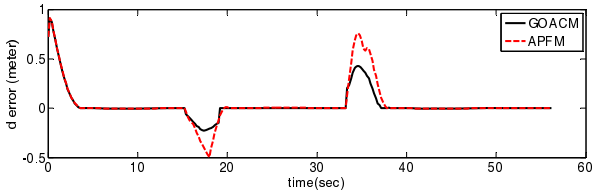


Fig. 17. Relative distance errors between the leader robot and follower robot 1.

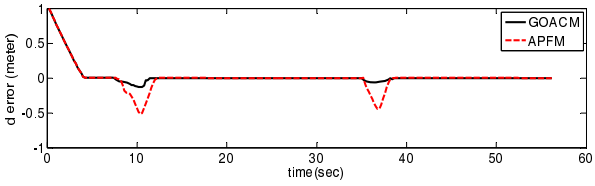


Fig. 18. Relative distance errors between the leader robot and follower robot 2.

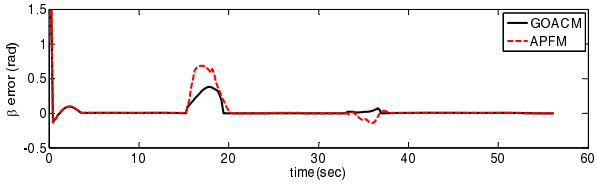


Fig. 19. Relative bearing angle errors between the leader robot and follower robot 1.

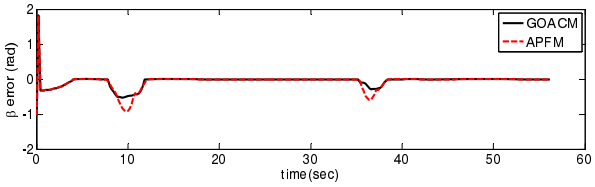


Fig. 20. Relative bearing angle errors between the leader robot and follower robot 2.

obstacles, GOACM contributes that the formation state errors more quickly converge to zero.

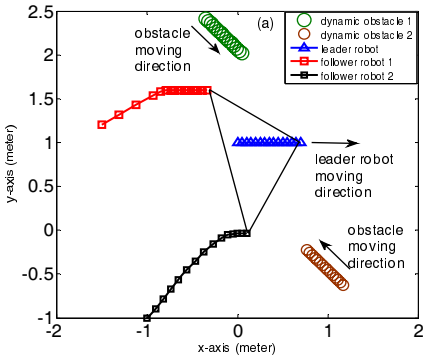
5.2. Dynamic obstacle avoidance

To make the simulations realistic, the uncertainties are added to the simulated measured position of the dynamic obstacles. If there are no obstacles, two follower robots and the leader robot form and maintain the formation with the desired distance $l_{10} = l_{20} = 1.2$ m, the bearing angle $\beta_{10} = -\pi/6$ and $\beta_{20} = \pi/3$. $r_s = r_p = 0.7$ m is selected for follower robots 1 and 2. The initial states of the leader robot, follower robots 1 and 2 are $(0,1,0)^T$, $(-1.5,1.2,0)^T$, $(-1,-1,\pi/6)^T$, respectively. The trajectories of dynamic obstacles 1 and 2 are defined as

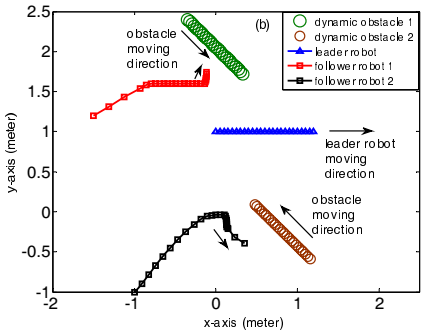
$$x_{o1} = 0.1\sqrt{2}t - 0.35, \quad y_{o1} = -0.1\sqrt{2}t + 2.42, \tag{43}$$

$$x_{o2} = -0.1\sqrt{2}t + 1.15, \quad y_{o2} = 0.1\sqrt{2}t - 0.58. \tag{44}$$

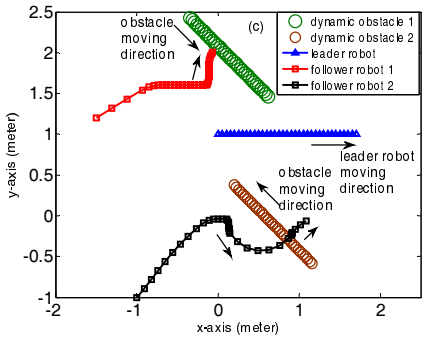
Fig. 21 shows the trajectories of three robots and two obstacles. At $t = 3$s, the dynamic obstacles do not enter



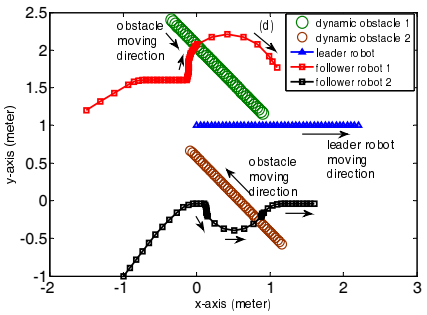
(a) t =3s.



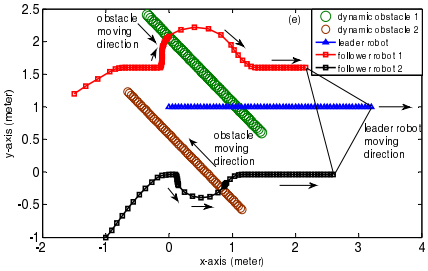
(b) t =5s.



(c) t =7s.



(d) t = 9s.



(e) t =13s.

Fig. 21. Trajectories of the three robots and the two dynamic obstacles.

the protected shells of the follower robots, so both of the follower robots form a triangle formation with the leader robot. At $t = 5$s, since both dynamic obstacles enter the protected shells of the follower robots, follower robots 1 and 2 avoid obstacles using GOACM. At $t = 7$s, after completion of avoiding dynamic obstacle 2, follower robot 2 attempts to track the leader robot, while follower robot 1 still avoids the dynamic obstacle 1. At $t = 9$s, follower robot 1 tracks the leader robot, and follower robot 2 reforms the formation with the leader robot as the desired distance and bearing angle. At $t = 13$s, both of the follower robots reform and maintain the formation with the leader robot.

Figs. 22 and 23 show state errors of two follower robots moving in a dynamic environment. The results show that the state errors can effectively converge to zero. After avoiding the dynamic obstacles, the followers can quickly come back to the formation.

The velocity tracking errors of the two follower robots are shown in Figs. 24 and 25, when moving as the trajectories in Fig. 21(e). In Fig. 24, the velocity errors are close to zero. In Fig. 25, although oscillations occur, the velocity tracking errors finally converge to zero.

To compare with the result in Fig. 21(e), two follower robots avoid the same dynamic obstacles using APFM in Fig. 26. For both follower robots, the obstacle avoidance path lengths are longer than them in Fig. 21(e).

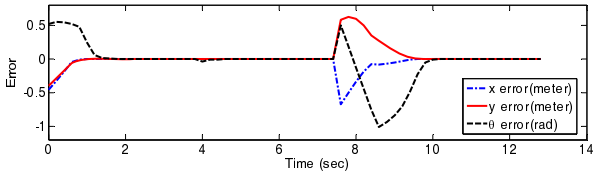


Fig. 22. Trajectory tracking errors of follower robot 1.

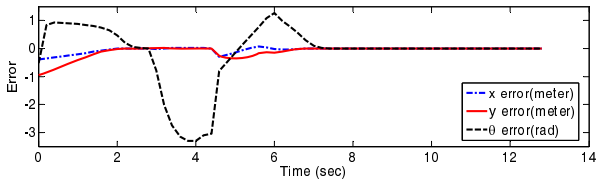


Fig. 23. Trajectory tracking errors of follower robot 2.

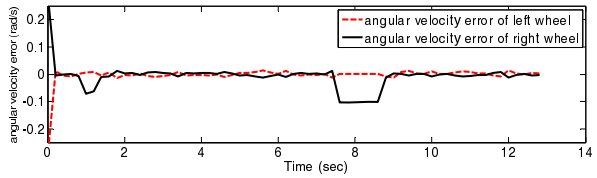


Fig. 24. Velocity tracking errors of follower robot 1.

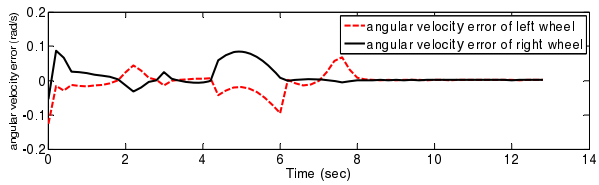


Fig. 25. Velocity tracking errors of follower robot 2.

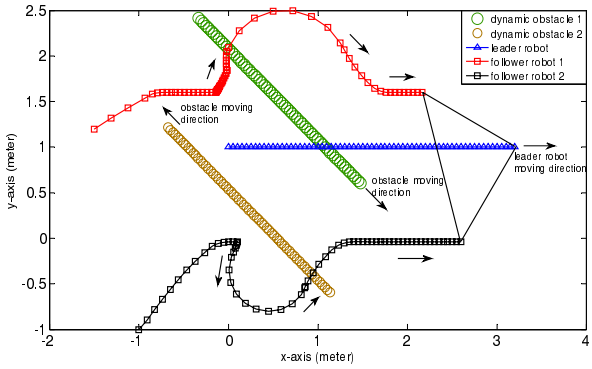


Fig. 26. Trajectories of the three robots and the two dynamic obstacles compared with Fig. 17(e).

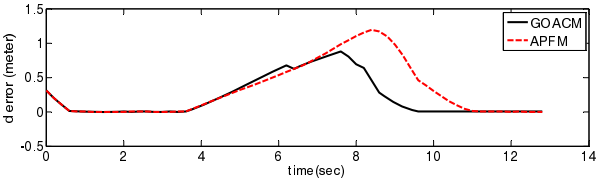


Fig. 27. Relative distance errors between the leader robot and follower robot 1.

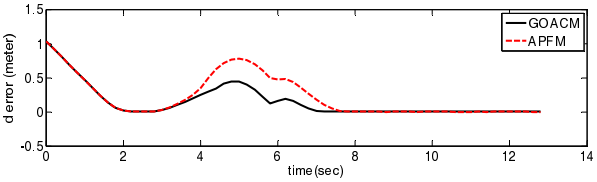


Fig. 28. Relative distance errors between the leader robot and follower robot 2.

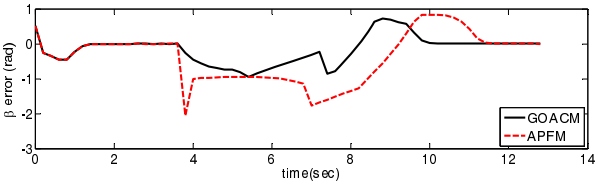


Fig. 29. Relative bearing angle errors between the leader robot and follower robot 1.

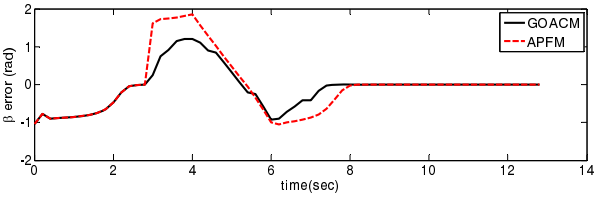


Fig. 30. Relative bearing angle errors between the leader robot and follower robot 2.

Using GOACM and APFM, Figs. 27-30 show the comparison results of the formation state errors, including relative distance errors and relative bearing angle errors between the leader robot and follower robots 1 and 2, respectively. The formation state errors using GOACM are smaller than the errors using APFM.

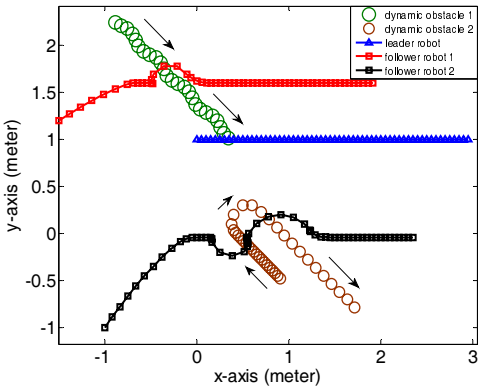


Fig. 31. Trajectories of three robots and two dynamic obstacles.

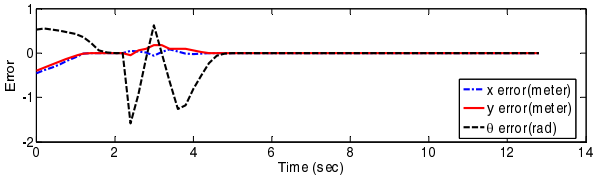


Fig. 32. Trajectory tracking errors of follower robot 1.

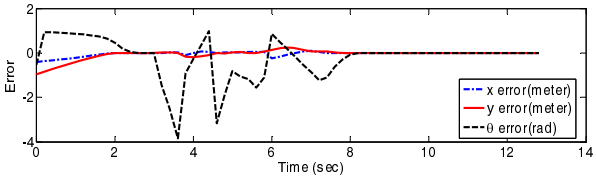


Fig. 33. Trajectory tracking errors of follower robot 2.

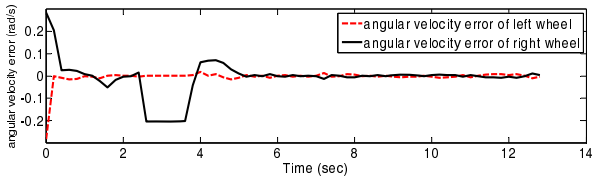


Fig. 34. Velocity tracking errors of follower robot 1.

Fig. 31 shows two follower robots avoid two dynamic obstacles while forming formation with the leader robot. Dynamic obstacle 1 moves in a sinusoid trajectory. During the process, dynamic obstacle 2 turns 180 degree and accelerates. In Fig. 31, follower robot 1 effectively avoids dynamic obstacle 1. Since the dynamic obstacle 2 changes its direction, follower robot 2 applies GOACM by choosing different intersections. Follower robot 2 can also effectively avoid dynamic obstacle 2. After avoiding the obstacles, two follower robots form a formation with leader robot with the desired distance $l_{10} = l_{20} = 1.2$ m, and the desired bearing angle $\beta_{10} = -\pi/6$ and $\beta_{20} = \pi/3$.

Figs. 32 and 33 present trajectory tracking errors of two follower robots avoiding two dynamic obstacles. The state errors can correctly converge to zero.

In Figs. 34 and 35, the velocity tracking errors can converge to zero.

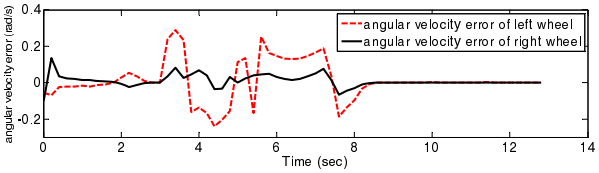


Fig. 35. Velocity tracking errors of follower robot 2.

5.3. Multiple robots avoiding an obstacle

Fig. 36 shows that two follower robots simultaneously avoid the same obstacle. Follower robot 1 is located on the left-side of follower robot 2, so follower robots 1 and 2 choose A_{11} and A_{2n} to avoid the obstacle, respectively. After avoiding the obstacle, three robots can effectively and efficiently reform and maintain formation.

Figs. 37 and 38 show trajectory tracking errors of two follower robots while moving in the trajectories as Fig. 36. Based on the results, the trajectory tracking errors can finally converge to zero.

The velocity tracking errors of two follower robots are shown in Figs. 39 and 40. The velocity tracking errors approximately converge to zero.

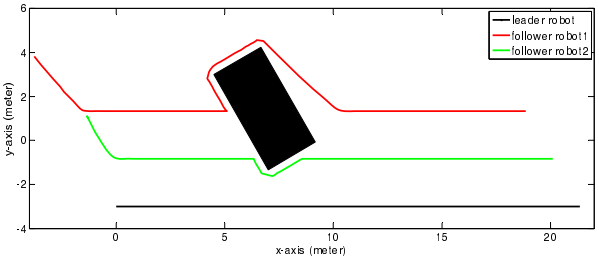


Fig. 36. Two follower robots simultaneously avoid one obstacle.

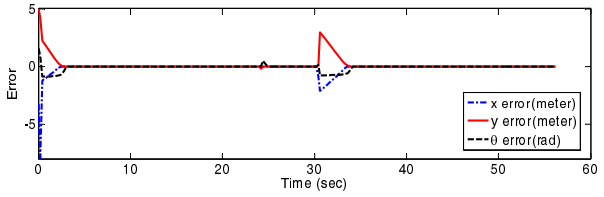


Fig. 37. Trajectory tracking errors of follower robot 1.

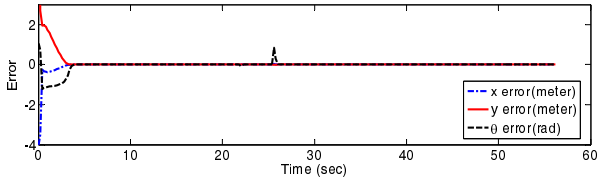


Fig. 38. Trajectory tracking errors of follower robot 2.

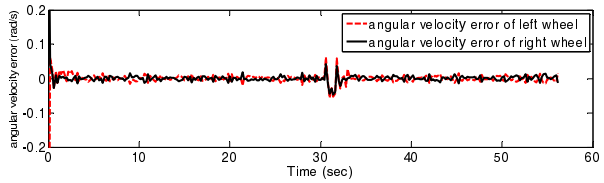


Fig. 39. Velocity tracking errors of follower robot 1.

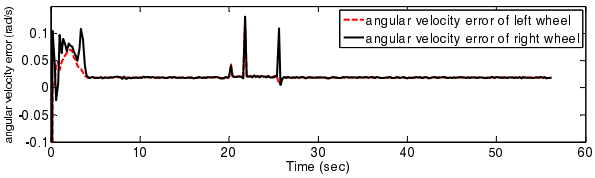


Fig. 40. Velocity tracking errors of follower robot 2.

6. EXPERIMENTS

In order to validate the proposed methods, experiments were conducted using three robots. The experiments include three robots avoiding static obstacles, and two robots avoiding a dynamic obstacle.

6.1. Experimental setup

The test field is a 1.85m by 2.25 m wooden bed for three nonholonomic mobile robots as shown in Fig. 41(a). A CCD camera (640×480) is located approximately 2.45 m above the floor to measure the positions and attitudes of each robot. For each robot as Fig. 41(b), the diameter of each wheel is 4.3 cm, and the distance between the left and right wheels is 6.9 cm. The weight of the robot is 0.4 kg. The robot is equipped five onboard modules: communication module, sensor module, MCU module, DC motor module, and power module. The robots use bluetooth to communicate with a computer. In sensor module, compass and encoder are used to detect robot angles and positions. The distances between robot and obstacle are measured by ultrasonic sensors. Onboard processing is carried out by an ATmega 2560 AVR for communication and sensor control, and a DSP for motor control. The gear ratio of DC motor is 8:1. The experiment procedure for each robot is as:

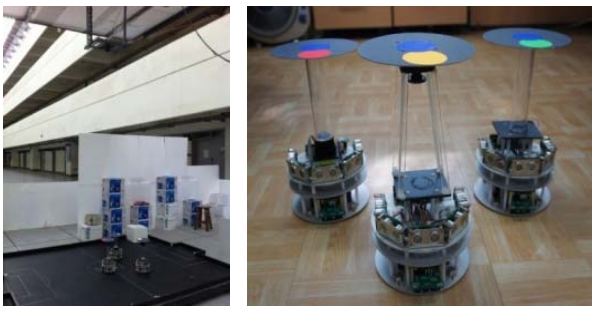
1) The robot detects the distances between itself and an obstacle using ultrasonic sensors, and transmits the measured distances to server PC.

2) In server PC, if the minimal measured distance is smaller than the radius of the robot protected shell, the waypoint is calculated based on GOACM. If no obstacle, the waypoint is given by the leader robot's information.

3) Server PC calculates the torques and sends motion commands to the robot by Bluetooth.

4) DC motors are actuated on the robot.

5) The robot transmits encoder values and angle values to server PC.



(a) Test bed. (b) Three robots.

Fig. 41. Experiment system.

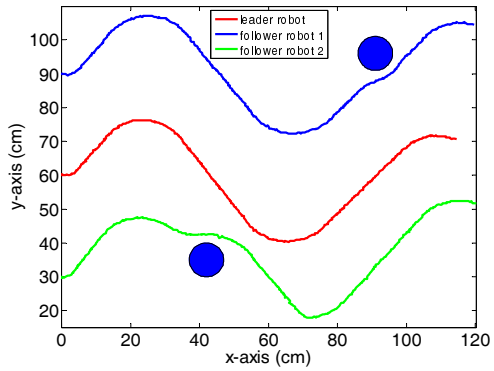


Fig. 42. The sinusoid trajectories of three robots avoiding two circular static obstacles.

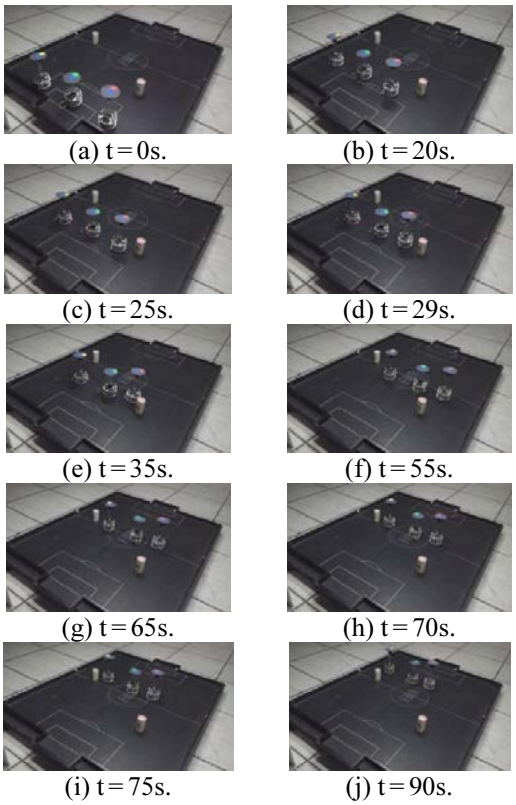


Fig. 43. Snapshots of three robots avoiding two static obstacles.

6) Localizations are carried out on server PC based on the encoder values and angle values.

7) The adaptive controllers are utilized to minimize the trajectory and velocity tracking errors.

6.2. Experimental result

In the experiment, the parameters for the algorithm are as $K_{i1} = 0.1$, $K_{i2} = 0.2$, $K_{i3} = 0.28$, $K_{id} = diag\{0.5, 0.5\}$, $\gamma_{i1} = \gamma_{i2} = 0.001$, $\gamma_{i3} = \gamma_{i4} = 0.01$, $\Gamma = diag\{0.01, 0.01, 0.01\}$. The radius of the protected shell of each robot is 8 cm.

6.2.1 Static obstacle avoidance

The initial states of the leader robot, follower robots 1 and 2 are $(0\text{ cm}, 60\text{ cm}, 0\,\text{deg})^T$, $(0\text{ cm}, 90\text{ cm}, 0\,\text{deg})^T$, and $(0\text{ cm}, 30\text{ cm}, 0\,\text{deg})^T$, respectively. The desired distances for follower robots 1 and 2 with the leader robot are $l_{10} = 30$ cm, and $l_{20} = 30$ cm, respectively. The desired angles of the follower robots 1 and 2 with the leader robot are $\beta_{10} = -\pi/2$, and $\beta_{20} = \pi/2$, respectively. Fig. 42 shows the sinusoid trajectories of three robots forming and maintaining a line formation. Snapshots of video are presented in Fig. 43. Based on GOACM, follower robot 1 avoids a circular obstacle at around 70 s, and follower robot 2 avoids a circular obstacle at around 25 s. After avoiding the obstacles, both follower robots reform the formation with the leader robot.

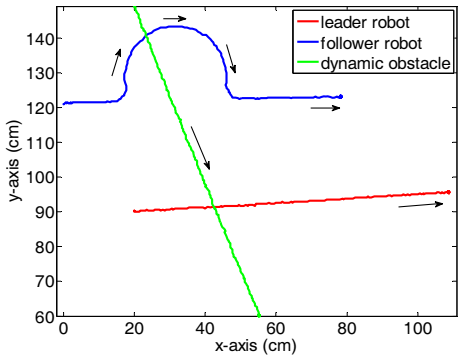


Fig. 44. The trajectories of two robots avoiding a dynamic obstacle.

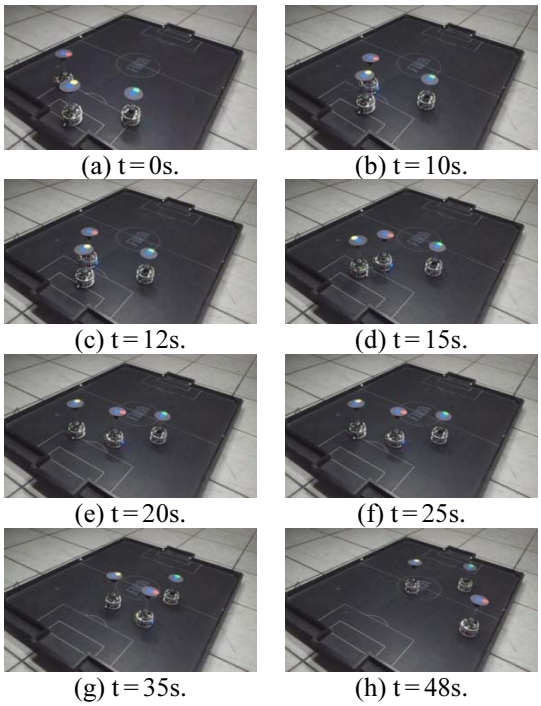


Fig. 45. Snapshots of two robots avoiding a dynamic obstacle.

6.2.2 Dynamic obstacle avoidance

In Fig. 44, a robot acts as a dynamic obstacle, and a follower robot avoids the dynamic obstacle while maintaining formation with a leader robot. The initial states of the leader robot and the follower robot

are $(20\text{ cm}, 90\text{ cm}, 0\,\text{deg})^T$ and $(0\text{ cm}, 120\text{ cm}, 0\,\text{deg})^T$, respectively. The desired distance for the follower robot with the leader robot is defined as $l_{10} = 36$ cm. The desired angle of the follower robot with the leader robot is $\beta_{10} = -56\,\text{deg}$. Snapshots of video are presented in Fig. 45. The follower robot avoids the dynamic obstacle at around 70s.

7. CONCLUSIONS

This paper presents a group of nonholonomic mobile robots form and maintain formation under a cluttered environment. In order to avoid the static and dynamic obstacles, GOACM is proposed for the robot using limited on-board sensor information, and considering the velocity constraints of each robot. In case of dynamic environment, the trajectories of the dynamic obstacles are not pre-known. Since both the leader-follower formation control method and GOACM focus on finding waypoints for the robot, GOACM can effectively cooperate with the formation control method. The adaptive controllers are used to minimize the robot's trajectory and velocity tracking errors. The simulation results show the better performance based on GOACM than the performance based on APFM. The simulation and experimental results demonstrate the effectiveness of GOACM to avoid the static and dynamic obstacles.

REFERENCES

[1] M. Defoort, T. Floquet, A. Kökösy, and W. Perruquetti, "Sliding-mode formation control for cooperative autonomous mobile robots," *IEEE Trans. on Ind. Electron*., vol. 55, no. 11, pp. 3944-3953, 2008.

[2] T. Balch and R. C. Arkin, "Behavior-based formation control for multi-robot teams," *IEEE Trans. on Robot. and Autom*., vol. 14, no. 6, pp. 926-939, 1998.

[3] X. Chen and Y. M. Li, "Stability on adaptive NN formation control with variant formation patterns and interaction topologies," *Int. J. of Adv. Robot. Syst.*, vol. 5, no. 1, pp. 69-82, 2008.

[4] T. Gustavi and X. M. Hu, "Observer-based leader-following formation control using onboard sensor information," *IEEE Trans. on Robot*., vol. 24, no. 6, pp. 1457-1462, 2008.

[5] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. of Robot. Res*., vol. 5, no. 1, pp. 90-98, 1986.

[6] D. W. Kim, T. A. Lasky, and S. A. Velinsky, "Autonomous Multi-mobile robot system: simulation and implementation using fuzzy logic," *Int. J. of Control, Auto., and Systems*, vol. 11, no. 3, pp. 545-554, 2013.

[7] M. G. Earl and R. D. Andrea, "Iterative MILP methods for vehicle-control problems," *IEEE Trans. On Robot*., vol. 21, no. 6, pp. 1158-1167, 2005.

[8] S. C. Liu, D. L. Tan, and G. J. Liu, "Formation control of mobile robots with active obstacle avoidance," *Acta Autom. Sinica*, vol. 33, no. 5, pp. 529-535, 2007.

[9] K. Fujimura and H. Samet, "A hierarchical strategy for path planning among moving obstacles," *IEEE Trans. on Robot. and Autom*., vol. 5, no. 1, pp. 61-69, 1989.

[10] S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Auton. Robots*, vol. 13, no. 3, pp. 207-222, 2002.

[11] F. Belkhouche, "Reactive path planning in a dynamic environment," *IEEE Trans. on Robot*., vol. 25, no. 4, pp. 902-911, 2009.

[12] C. Leng, C. Cao, and Y. Huang, "A motion planning method for omni-directional mobile robot based on anisotropic characteristics," *Int. J. of Adv. Robot. Syst.*, vol. 5, no. 4, pp. 327-340, 2008.

[13] R. L. Williams and J. H. Wu, "Dynamic obstacle avoidance for an omnidirectional mobile robot," *J. of Robot*., vol. 2010, Article ID 901365, 2010.

[14] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for a nonholonomic mobile robot," *Proc. of IEEE/RSJ Int. Workshop Intell. Robots and Syst*., pp. 1236-1241, 1991.

[15] T. Das and I. N. Kar, "Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots," *IEEE Trans. on Control Syst. Tech.*, vol. 14, no. 3, pp. 501-510, 2006.

[16] J. B. Wu, G. H. Xu, and Z. P. Yin, "Robust adaptive control for a nonholonomic mobile robot with unknown parameters," *J. Control Theory Appl.*, vol. 7, no. 2, pp. 212-218, 2009.

[17] Y. Y. Dai and S. G. Lee, "The leader-follower formation control of nonholonomic mobile robots," *Int. J. of Control, Auto., and Systems*, vol. 10, no. 2, pp. 350-361, 2012.

[18] M. Krstic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*, Wiley, New York, 1995.

[19] M. Mohan, D. Busquets, R. L. de. Màntaras, and C. Sierra, "Integrating a potential field based pilot into a multiagent navigation architecture for autonomous robots," *Proc. of Int. Conf. on Info. in Control, Auto. and Robot*., pp. 287-290, 2004.

Yanyan Dai received her B.S. degree from Tianjin University of Science and Technology, China, in 2009, and her M.S. degree in Electrical Engineering from Yeungnam University, Korea, in 2011, where she is currently working toward a Ph.D. degree in Robotics and Control. Her research interests include the formation control of multiple mobile robots and SLAM.

Suk Gyu Lee received his B.S. and M.S. degrees in Electrical Engineering from Seoul National University, in 1979 and 1981, respectively, and he received his Ph.D. degree in Electrical Engineering from UCLA in 1990. His research interests include robotics, SLAM, nonlinear control and adaptive control.